



# ARCHIVO DEL LABORATORIO DE DOCUMENTACIÓN GEOMÉTRICA DEL PATRIMONIO

LABORATORY FOR THE GEOMETRIC DOCUMENTATION OF  
HERITAGE'S ARCHIVE


Sección de proyectos fin de carrera / Undergraduate master  
projects section

# 2

Información general / General information		
AUTOR:	Ion MARÍN MARTÍN	:AUTHOR
TUTOR:	José Miguel BLANCO ARBE	:TUTOR
TITULACION:	Ingeniería Informática. Facultad de Informática de San Sebastián (UPV-EHU)	:DEGREE
TITULO:	Desarrollo de una solución para el acceso integrado a fuentes de información sobre Patrimonio histórico. Análisis de las alternativas e implementación de una solución basada en una aplicación de búsqueda orientada en bases de datos heterogéneas	:TITLE
FECHA:	febrero 2009 / February 2009	:DATE
NUMERO:	LDGP_pfc_002	:NUMBER
IDIOMA:	español / Spanish	:LANGUAGE

Resumen	
TITULO:	Desarrollo de una solución para el acceso integrado a fuentes de información sobre Patrimonio histórico. Análisis de las alternativas e implementación de una solución basada en una aplicación de búsqueda orientada en bases de datos heterogéneas
RESUMEN:	Se desarrolla una base de datos a través de Internet que permite recuperar recursos relacionados con el patrimonio histórico.
DESCRIPTORES NATURALES:	patrimonio, bases de datos, web
DESCRIPTORES CONTROLADOS:	(Procedentes del Tesouro UNESCO [ <a href="http://databases.unesco.org/thessp/">http://databases.unesco.org/thessp/</a> ]) Patrimonio Cultural, Base de datos, Fotogrametría

Abstract	
TITLE:	Development of a solution for the integrated access to sources of information about historic Heritage. Analysis of alternatives for the implementation of a solution based on a searching application orientated to heterogeneous data bases
ABSTRACT:	This project carried out a data base running in a web environment which searches information about Heritage.
NATURAL KEYWORDS:	, web
CONTROLLED KEYWORDS:	(From the UNESCO's thesaurus [ <a href="http://databases.unesco.org/thesaurus/">http://databases.unesco.org/thesaurus/</a> ]) Cultural Heritage, Databases, Internet

Derechos / Rights		
DERECHOS:	Está permitido citar y extraer el texto, siempre que la fuente sea claramente identificada (respecto a la consideración de "no comercial" ver el apartado "otros derechos"). / <b>Permission is granted to quote and take excerpts from this text, provided that the source of such material is fully acknowledged (for the "non commercial" label see below in "others rights").</b>	:RIGHTS
		
OTROS:		:OTHERS

Renuncia de responsabilidad / Disclaimer		
DESCARGO:	El uso de la información contenida en este documento se hará bajo la completa responsabilidad del usuario. / <b>The use of the information contained in this document will be under the exclusive responsibility of the user.</b>	:DISCLAIMER

Estructura / Framework		
ID PERMANENTE:	<a href="http://hdl.handle.net/10810/7097">http://hdl.handle.net/10810/7097</a>	:PERMANENT ID
ESTRUCTURA:	<ul style="list-style-type: none"> <li>• <b>ldgp_pfc002_lonMM.pdf</b>: este documento / <b>this document.</b></li> </ul>	:FRAMEWORK

**Cita completa recomendada / Recommended full citation**

CITA:	MARÍN MARTÍN, Ion. <i>Desarrollo de una solución para el acceso integrado a fuentes de información sobre Patrimonio histórico. Análisis de las alternativas e implementación de una solución basada en una aplicación de búsqueda orientada en bases de datos heterogéneas.</i> Proyecto fin de carrera de Ingeniería Informática. Facultad de Informática de San Sebastián (Universidad del País Vasco-Euskal Herriko Unibertsitatea UPV/EHU). 2009.	:CITATION
-------	---	-----------

## **0. ABSTRACT**

Actualmente existen diversas organizaciones e instituciones en Álava y alrededores que desarrollan una actividad importante en lo que al patrimonio histórico se refiere. Estas organizaciones e instituciones disponen de numerosa información sin clasificar lo que provoca que dicha información no pueda ser consultada por los aficionados del patrimonio histórico.

Como principio de una solución al problema, el Laboratorio de Documentación Geométrica del Patrimonio, se propuso encontrar la manera de conseguir reunir toda la información existente, y presentársela al usuario a través de la red. Esta idea obtuvo como resultado el Proyecto de Fin de Carrera titulado “Desarrollo de una solución para acceso integrado a fuentes de información sobre patrimonio histórico. Análisis de las alternativas e implementación de una solución basada en una aplicación de búsqueda orientada en bases de datos heterogéneas” y descrito a lo largo de esta memoria.

Dicho proyecto consta de una parte de investigación, en la que el alumno, Ion Marín, analiza las posibles soluciones al problema, explicando para cada una de ellas su estructura, sus ventajas, inconvenientes y limitaciones, y una fase de desarrollo.

La fase de desarrollo está fundamentada en la fase de investigación, ya que una vez que se desarrolla la fase de investigación, el alumno considera que lo que más se adapta tanto a los requerimientos del cliente, como a las circunstancias actuales es desarrollar una solución basada en una aplicación de búsqueda orientada en bases de datos heterogéneas.

A lo largo de la memoria se describe de manera exhaustiva cómo se ha desarrollado la aplicación de búsqueda orientada, las aplicaciones complementarias que han sido necesarias para obtener un producto satisfactorio, así como una fase de pruebas y unas conclusiones finales. Unido a esto se explica con claridad la gestión del proyecto y las relaciones establecidas con el cliente a lo largo del proyecto.

Por último destacar que el proyecto realizado está disponible en la Web a través de la página Web del Laboratorio de Documentación Geométrica del Patrimonio de la Universidad del País Vasco.

## **PALABRAS CLAVE**

Bases de datos heterogéneas, bases de datos, aplicación de búsqueda orientada, sistema de acceso integrado a fuentes de información.

## ÍNDICE DE CONTENIDOS

<b>1. INTRODUCCIÓN .....</b>	<b>1</b>
1.2. PROPUESTA DE PROYECTO.....	3
<b>2. ANTECEDENTES Y CONTEXTO.....</b>	<b>5</b>
<b>3. INTERÉS Y VIABILIDAD DEL PROYECTO .....</b>	<b>7</b>
3.1 INTERÉS EN EL PROYECTO.....	7
3.2 VIABILIDAD DEL PROYECTO.....	8
3.3 PROYECTOS SIMILARES .....	9
<b>4. DOCUMENTO DE OBJETIVOS DEL PROYECTO .....</b>	<b>13</b>
4.1 OBJETIVOS .....	13
4.2 MOTIVACIÓN.....	14
4.3 DESCRIPCIÓN .....	15
4.4 RECURSOS NECESARIOS PARA EL DESARROLLO DEL PROYECTO.....	19
4.4.1 RECURSOS HARDWARE .....	19
4.4.2 RECURSOS SOFTWARE.....	20
4.4.3 EDITOR DE DESARROLLO.....	20
4.4.4 SOPORTE DEL SISTEMA DE INFORMACIÓN .....	20
4.5 PLANIFICACIÓN .....	22
4.6. DIRECCIÓN DE RIESGOS DEL PROYECTO .....	39
4.6.1 IDENTIFICACIÓN DE RIESGOS .....	39
4.6.2 CUANTIFICACIÓN DE LOS RIESGOS .....	40
4.6.3 DESARROLLO DE RESPUESTAS A RIESGOS .....	41
4.6.4 CONTROL DE RESPUESTAS A RIESGOS.....	44
4.7 ENTREGABLES DEL PROYECTO.....	45

<b>5. ANÁLISIS .....</b>	<b>47</b>
5.1 ANÁLISIS DE REQUERIMIENTOS .....	47
5.2 ANÁLISIS DE LAS HERRAMIENTAS .....	48
5.2.1 COMPARATIVA DE DESARROLLADORES WEB.....	48
5.2.1.1 ELECCIÓN DE DESARROLLADOR DE APLICACIONES WEB.....	48
5.2.1.1.1 DESCRIPCIÓN DE LAS DIFERENTES DESARROLLADORES WEB .....	48
5.2.1.1.2 VENTAJAS E INCONVENIENTES DE CADA UNA DE LAS PLATAFORMAS.....	50
5.2.1.1.3 COMPARATIVA DE LOS LENGUAJES DE PROGRAMACIÓN DE CADA UNA DE LAS PLATAFORMAS .....	53
5.2.1.1.4 CONCLUSIÓN.....	55
5.2.2 XAMPP.....	57
5.2.2.1 APACHE.....	57
5.2.2.2 MYSQL.....	58
5.2.2.3 PHP .....	59
5.3 ANÁLISIS DE LOS DATOS .....	62
5.4 ANÁLISIS DE LAS POSIBLES SOLUCIONES .....	64
5.4.1 BASE DE DATOS CENTRALIZADA.....	65
5.4.1.1 DESCRIPCIÓN .....	65
5.4.1.2 DISEÑO DE BASE DE DATOS CENTRALIZADA .....	65
5.4.1.3 BASE DE DATOS CENTRALIZADA APLICADA A NUESTRO PROYECTO .....	68
5.4.2 BASE DE DATOS DISTRIBUIDA.....	71
5.4.2.1 DESCRIPCIÓN .....	71
5.4.2.2 DISEÑO DE BASES DE DATOS DISTRIBUIDAS .....	77

5.4.2.3 BASE DE DATOS DISTRIBUIDA APLICADA A NUESTRO PROYECTO .....	85
5.4.3 BASES DE DATOS DISTRIBUIDAS Y HETEROGÉNEAS .....	89
5.4.3.1 DESCRIPCIÓN .....	89
5.4.3.2 DISEÑO DE LA ARQUITECTURA DEL SISTEMA DE ACCESO INTEGRADO A BASES DE DATOS DOCUMENTALES BASADA EN ÁRBOLES DE CONCEPTOS. ....	89
5.4.3.3 DISEÑO DE LA ARQUITECTURA DEL SISTEMA DE ACCESO INTEGRADO A BASES DE DATOS DOCUMENTALES BASADA EN ÁRBOLES DE CONCEPTOS APLICADO A NUESTRO PROYECTO. ....	97
5.5 ELECCIÓN DE LA PROPUESTA MÁS ADECUADA .....	101
5.6 CASOS DE USO.....	105
5.6.1 CASOS DE USO DEL ADMINISTRADOR .....	106
5.6.2 CASOS DE USO DEL USUARIO.....	109
5.6.3 CASOS DE USO DEL USUARIOWEB .....	112
<b>6. DISEÑO DE LA SOLUCIÓN.....</b>	<b>115</b>
6.1. INTRODUCCIÓN .....	115
6.2 APLICACIÓN QUE DA SOPORTE A LA BASE DE DATOS QUE CLASIFICA DATOS SOBRE LA INFORMACIÓN EXISTENTE EN LA PÁGINAS WEB.....	117
6.2.1 BASES DE DATOS DE LA APLICACIÓN .....	117
6.2.1.1 BASE DE DATOS CENTRALIZADA “SITIOS WEB”.....	117
6.2.1.1.1 MODELO ENTIDAD-RELACIÓN BASE DE DATOS “SITIOS WEB”.....	118
6.2.1.1.2 MODELO RELACIONAL BASE DE DATOS “SITIOS WEB”.....	119
6.2.1.1.3 RELACIONES DE LA BASE DE DATOS .....	121
6.2.1.2 BASE DE DATOS CENTRALIZADA “ADMINISTRADORES” .....	122
6.2.1.2.1 MODELO ENTIDAD-RELACIÓN BASE DE DATOS “ADMINISTRADORES”.....	123



6.2.1.2.2 MODELO RELACIONAL BASE DE DATOS“ ADMINISTRADORES” .....	123
6.2.1.2.3 RELACIONES DE LA BASE DE DATOS .....	124
6.2.2 INTERFACES DE LA APLICACIÓN.....	125
6.2.3 PERFILES DE LA APLICACIÓN.....	126
6.2.4 ARQUITECTURA DE QUE DA SOPORTE A LA BASE DE DATOS QUE CLASIFICA DATOS SOBRE LA INFORMACIÓN EXISTENTE EN LAS PÁGINAS WEB.....	127
6.2.4.1 CAPA DE PRESENTACIÓN .....	128
6.2.4.2 LÓGICA DE NEGOCIO.....	129
6.2.4.3 CAPA DE ACCESO A DATOS.....	129
6.2.5 FUNCIONALIDADES DEL SISTEMA .....	129
6.2.5.1 ACCEDER AL SISTEMA.....	130
6.2.5.2 NUEVO SITIO WEB.....	132
6.2.5.3 MODIFICAR SITIO WEB.....	134
6.2.5.4 ELIMINAR SITIO WEB .....	136
6.2.5.5 NUEVA SUB-PÁGINA WEB .....	138
6.2.5.6 MODIFICAR SUB-PÁGINA WEB .....	140
6.2.5.7 ELIMINAR SUB-PÁGINA WEB .....	143
6.2.5.8 INTRODUCIR NUEVO PATRIMONIO .....	146
6.2.5.9 MODIFICAR PATRIMONIO.....	149
6.2.5.10 ELIMINAR PATRIMONIO.....	152
6.2.5.11 INTRODUCIR NUEVO ADMINISTRADOR.....	155
6.2.5.12 MODIFICAR ADMINISTRADOR.....	157
6.2.5.13 ELIMINAR ADMINISTRADOR.....	159
6.2.5.14 VER MOVIMIENTOS.....	161

6.2.5.15 VER ADMINISTRADORES .....	162
6.2.5.16 MODIFICAR CONTRASEÑA .....	163
6.3 APLICACIÓN DE BÚSQUEDA ORIENTADA .....	165
6.3.1 INTERFAZ DE APLICACIÓN DE BÚSQUEDA ORIENTADA .....	165
6.3.2 ARQUITECTURA DE LA APLICACIÓN DE BÚSQUEDA ORIENTADA .....	170
6.3.2.1 INTERFAZ DE USUARIO .....	172
6.3.2.1.1 COMPONENTES DE LA “INTERFAZ DE USUARIO” .....	173
6.3.2.2 MEDIADOR .....	174
6.3.2.2.1 COMPONENTES DEL MEDIADOR .....	174
6.3.2.3 SISTEMAS ENVOLTORIOS (WRAPPERS) .....	175
6.3.2.3.1 COMPONENTES DE LOS SISTEMAS ENVOLTORIO (WRAPPERS) .....	175
6.3.2.4 BASES DE DATOS .....	176
6.3.2.4.1 COMPONENTES DE LAS “BASES DE DATOS” .....	176
6.3.3 ÁRBOL DE CONCEPTOS Y ÁRBOLES DE CORRESPONDENCIAS .....	177
6.3.3.1 DESCRIPCIÓN DE LOS ÁRBOLES .....	177
6.3.3.1.1 RELACIONES DE DESCRIPCIÓN .....	178
6.3.3.1.2 RELACIONES DE GENERALIZACIÓN /ESPECIALIZACIÓN .....	178
6.3.3.2 ÁRBOLES DE CONCEPTOS .....	182
6.3.3.2.1 CONCEPTOS BÁSICOS .....	182
6.3.3.2.2 UTILIDADES DE LOS ÁRBOLES DE CONCEPTOS .....	182
6.3.3.2.3 ÁRBOL DE CONCEPTOS DE NUESTRO SISTEMA .....	184
6.3.3.3 ÁRBOLES DE CORRESPONDENCIAS .....	185
6.3.3.3.1 CONCEPTOS BÁSICOS .....	185

6.3.3.3.2	ÁRBOLES DE CORRESPONDENCIAS EXISTENTES EN NUESTRO SISTEMA .....	185
6.3.3.3.2.1	BASE DE DATOS “SITIOS WEB” .....	186
6.3.3.3.2.2	BASE DE DATOS DEL LABORATORIO DE DOCUMENTACIÓN GEOMÉTRICA DEL PATRIMONIO DE LA UPV/EHU .....	186
6.3.4	Lenguajes de comunicación entre capas .....	188
<b>7.</b>	<b>IMPLEMENTACIÓN DE LA APLICACIÓN .....</b>	<b>189</b>
7.1	INTRODUCCIÓN .....	189
7.2.	APLICACIÓN QUE DA SOPORTE A LA BASE DE DATOS QUE CLASIFICA DATOS SOBRE LA INFORMACIÓN EXISTENTE EN LA PÁGINAS WEB.....	189
7.2.1	INTRODUCCIÓN.....	189
7.2.2	DISTRIBUCIÓN FÍSICA.....	190
7.2.3	INTERFACES DE LA APLICACIÓN .....	190
7.2.4	CONTROLES .....	194
7.2.4.1	CONTROLES DE ACCESO .....	194
7.2.4.2	CONTROLES EN LAS INSERCIONES.....	194
7.2.4.3	CONTROLES EN LAS MODIFICACIONES.....	195
7.2.4.4	CONTROLES EN LAS ELIMINACIONES.....	196
7.2.5	CONTRASEÑA .....	196
7.2.6	SESIONES.....	196
7.2.7	PERFILES.....	197
7.3.	APLICACIÓN DE BÚSQUEDA ORIENTADA .....	198
7.3.1	EXPLICACIÓN DE LA APLICACIÓN DE BÚSQUEDA ORIENTADA .....	198
7.3.1.1	GENERADOR DE INTERFAZ DE CONSULTA.....	199
7.3.1.2	Lenguaje consulta, constructor y distribuidor de consultas.....	204

7.3.1.2.1 LENGUAJE CONSULTA.....	205
7.3.1.2.2 MÓDULO CONSTRUCTOR Y DISTRIBUIDOR DE CONSULTAS. ....	206
7.3.1.3 TRADUCCIÓN DE CONSULTAS.....	208
7.3.1.3.1 ALGORITMO DE TRADUCCIÓN .....	208
7.3.1.4 PRESENTACIÓN DE RESULTADOS.....	211
7.3.2 EJEMPLO DE PROCESO DE CONSULTA .....	213
7.3.2.1 LENGUAJE CONSULTA, CONSTRUCTOR Y DISTRIBUIDOR DE CONSULTAS .....	213
7.3.2.2 CONSTRUCTOR Y DISTRIBUIDOR DE CONSULTAS .....	216
7.3.2.3 TRADUCCIÓN DE CONSULTAS.....	220
7.3.2.4 PRESENTACIÓN DE RESULTADOS.....	223
<b>8. IMPLANTACIÓN DE LA SOLUCIÓN EN EL ORDENADOR DE LA UNIVERSIDAD DEL PAÍS VASCO .....</b>	<b>229</b>
<b>9. FASE DE PRUEBAS .....</b>	<b>233</b>
9.1 PRUEBAS DE LA APLICACIÓN QUE DA SOPORTE A LA BASE DE DATOS QUE ALMACENA INFORMACIÓN SOBRE LAS PÁGINAS WEB. ....	233
9.1.1 PRUEBA 1 .....	233
9.1.2 PRUEBA 2.....	234
9.1.3 PRUEBA 3.....	234
9.1.4 PRUEBA 4.....	235
9.1.5 PRUEBA 5.....	236
9.1.6 PRUEBA 6.....	236
9.1.7 PRUEBA 7.....	237
9.1.8 PRUEBA 8.....	238
9.1.9 PRUEBA 9.....	238
9.1.10 PRUEBA 10 .....	239

9.1.11 PRUEBA 11 .....	239
9.1.12 PRUEBA 12 .....	240
9.1.13 PRUEBA 13 .....	240
9.1.14 PRUEBA 14 .....	241
9.1.15 PRUEBA 15 .....	242
9.2 PRUEBAS DE LA APLICACIÓN DE BÚSQUEDA ORIENTADA .....	242
9.2.1 PRUEBA 1 .....	242
9.2.2 PRUEBA 2 .....	243
9.2.3 PRUEBA 3 .....	243
9.2.4 PRUEBA 4 .....	244
9.2.5 PRUEBA 5 .....	244
9.2.6 PRUEBA 6 .....	245
9.2.7 PRUEBA 7 .....	245
9.3 CONCLUSIONES .....	246
9.3.1 CONCLUSIONES DE LA APLICACIÓN QUE DA SOPORTE A LA BASE DE DATOS QUE ALMACENA INFORMACIÓN SOBRE LAS PÁGINAS WEB .....	246
9.3.2 CONCLUSIONES DE LA APLICACIÓN DE BÚSQUEDA ORIENTADA .....	246
<b>10. SEGUIMIENTO DEL PROYECTO .....</b>	<b>249</b>
10.1 CAMBIOS EN EL PROYECTO .....	249
10.2 SEGUIMIENTO DE LA GESTIÓN DEL PROYECTO .....	250
10.2.1 FASE “PREVIO” .....	250
10.2.2 FASE “INICIACIÓN” .....	251
10.2.3 FASE “DESARROLLO” .....	253
10.2.3.1 SUB-FASE “ESTUDIO PREVIO” .....	253
10.2.3.2 SUB-FASE “ANÁLISIS” .....	254

10.2.3.3 SUB-FASE DISEÑO .....	256
10.2.3.4 SUB-FASE IMPLEMENTACIÓN .....	262
10.2.3.5 SUB-FASE IMPLANTACIÓN .....	266
10.2.3.6 SUB-FASE PRUEBAS .....	268
10.2.3.7 TOTAL FASE DESARROLLO .....	269
10.2.4 TOTAL PROYECTO .....	269
10.3 CONCLUSIONES.....	272
<b>11. CONCORDANCIA ENTRE LOS RESULTADOS OBTENIDOS Y LOS OBJETIVOS PLANTEADOS .....</b>	<b>275</b>
<b>12. RELACIONES CON EL CLIENTE .....</b>	<b>277</b>
12.1 FASE DEL PRE-PROYECTO .....	277
12.2 FASE DE ACUERDO.....	278
12.3 FASE DE SEGUIMIENTO .....	279
12.4 FASE DE ENTREGA, PRUEBAS Y ACEPTACIÓN DE RESULTADOS.....	281
12.4.1 PRUEBAS.....	282
12.4.2 CONCLUSIONES POR PARTE DEL CLIENTE .....	292
<b>13. CONCLUSIONES.....</b>	<b>295</b>
13.1 LINEAS FUTURAS.....	295
13.2 CONCLUSIONES SOBRE EL PROYECTO.....	295
13.3 CONCLUSIONES PERSONALES .....	298
<b>14. BIBLIOGRAFÍA.....</b>	<b>301</b>

## INDICE DE FIGURAS

Figura 1. Ejemplo del buscador de www.muchoviaje.com. ....	9
Figura 2. Ejemplo del buscador de www.atrapalo.com. ....	10
Figura 3. Ejemplo del buscador de www.edreams.es. ....	10
Figura 4. Ejemplo de BD Centralizada. ....	16
Figura 5. Ejemplo de BD Distribuida ....	17
Figura 6. Diagrama de Gantt. Fases globales. ....	22
Figura 7. Diagrama de Gantt. Fase “Previo”. ....	23
Figura 8. Diagrama de Gantt. Fase “Iniciación”. ....	23
Figura 9. Diagrama de Gantt. Fase “Desarrollo”. ....	25
Figura 10. Ejemplo de arquitectura en capas. ....	66
Figura 11. Posible solución basada en una base de datos centralizada. ....	68
Figura 12. Ejemplo de una base de datos distribuida. ....	72
Figura 13. Arquitectura de tres capas con varias bases de datos en la capa de datos. ....	85
Figura 14. Sistema de Acceso Integrado a Bases de Datos Documentales ....	90
Figura 15. Árbol de Conceptos de un Sistema de Acceso Integrado (esquemático). ....	94
Figura 16. Parte de un Árbol de Conceptos de un Sistema de Acceso Integrado (XML). ....	95
Figura 17. Traducción de lenguaje de consulta (XML) a lenguaje de la base de datos (SQL). ..	96
Figura 18. Fragmento de un Árbol de Correspondencias (en XML). ....	97
Figura 19. Arquitectura del sistema de acceso integrado a base de datos documentales aplicado a nuestro problema. ....	98
Figura 20. Casos de uso del Administrador. ....	106
Figura 21. Casos de uso del Usuario. ....	109
Figura 22. Casos de uso del UsuarioWeb. ....	112
Figura 23. Modelo entidad-relación de la base de datos centralizada. ....	118

Figura 24. Modelo Entidad-Relación de la base de datos “Movimientos” .....	123
Figura 25. Arquitectura de la aplicación que trabaja contra las bases de datos “sitios Web” y “administradores” .....	128
Figura 26. Diagrama de Flujo de la funcionalidad “Acceder al sistema” .....	130
Figura 27. Diagrama de Flujo de la funcionalidad “Nuevo sitio Web” .....	132
Figura 28. Diagrama de Flujo de la funcionalidad “Modificar sitio Web” .....	134
Figura 29. Diagrama de Flujo de la funcionalidad “Eliminar sitio Web” .....	136
Figura 30. Diagrama de Flujo de la funcionalidad “Nueva subpágina Web” .....	138
Figura 31. Diagrama de Flujo de la funcionalidad “Modificar subpágina Web” .....	141
Figura 32. Diagrama de Flujo de la funcionalidad “Eliminar subpágina Web” .....	144
Figura 33. Diagrama de Flujo de la funcionalidad “Introducir nuevo patrimonio” .....	147
Figura 34. Diagrama de Flujo de la funcionalidad “Modificar patrimonio” .....	150
Figura 35. Diagrama de Flujo de la funcionalidad “Eliminar patrimonio” .....	153
Figura 36. Diagrama de Flujo de la funcionalidad “Introducir nuevo administrador” .....	155
Figura 37. Diagrama de Flujo de la funcionalidad “Modificar Administrador” .....	157
Figura 38. Diagrama de Flujo de la funcionalidad “Eliminar administrador” .....	159
Figura 39. Diagrama de Flujo de la funcionalidad “Ver Movimientos” .....	161
Figura 40. Diagrama de Flujo de la funcionalidad “Ver Movimientos” .....	162
Figura 41. Diagrama de Flujo de la funcionalidad “Modificar subpágina Web” .....	163
Figura 42. Calculadora de Windows .....	167
Figura 43. Frase de LNA para Atributos de tipo cadenas de caracteres .....	168
Figura 44. Frase de LNA para Atributos de tipo cadenas de caracteres .....	169
Figura 45. Ejemplo de Aproximación Navegacional .....	170
Figura 46. Arquitectura de la aplicación .....	171
Figura 47. Un concepto y sus atributos .....	178



Figura 48. Ejemplo de una relación de Generalización / Especialización.....	179
Figura 49. Parte general del concepto Patrimonio. ....	180
Figura 50. Parte general del concepto Edificios. ....	181
Figura 51. Árbol de Conceptos del Sistema. ....	184
Figura 52. Diagrama Entidad-Relación de la Base de Datos Proyectos.....	186
Figura 53. Interfaz de Autenticación .....	191
Figura 54. Interfaz tipo formulario común.....	191
Figura 55. Lista desplegable.....	192
Figura 56. Interfaz de borrado .....	192
Figura 57. Ejemplo de Interfaz de información al usuario.....	193
Figura 58. Interfaz de Movimientos en el sistema.....	193
Figura 59. Arquitectura de la aplicación. ....	198
Figura 60. Capas “Interfaz de Usuario” y “Mediador” de la arquitectura general. ....	199
Figura 61. Esqueleto de Frase de Especialización general.....	201
Figura 62. Esqueleto de Frase de Especialización para el concepto patrimonio .....	201
Figura 63. Esqueleto de Frase de Descripción global .....	202
Figura 64. Esqueleto de Frase de Descripción para el concepto Arquitectura civil.....	202
Figura 65. Diagrama de Flujo del Algoritmo del Generador de la Interfaz de Consulta .....	204
Figura 66. Diagrama de Flujo Algoritmo del módulo Constructor y Distribuidor de consulta. ....	207
Figura 67. Sistema envoltorio.....	209
Figura 68. Diagrama de flujo del algoritmo de traducción de consultas.....	210
Figura 69. Página asociada a la relación de Generalización/ Especialización del concepto Patrimonio. ....	213
Figura 70. Página asociada a la relación de Generalización/ Especialización del concepto Edificios. ....	214

Figura 71. Página de descripción del concepto “Arquitectura Religiosa”.....	214
Figura 72. Página de descripción del atributo “Nombre”.....	215
Figura 73. Página de descripción del atributo “Población”.....	216
Figura 74. Página de descripción del atributo “Provincia”.....	216
Figura 75. Primer paso de la construcción de la consulta. ....	217
Figura 76. Segundo paso de la construcción de la consulta. ....	217
Figura 77. Tercer paso de la construcción de la consulta.....	217
Figura 78. Cuarto paso de la construcción de la consulta. ....	218
Figura 79. Quinto paso de la construcción de la consulta.....	219
Figura 80. Consulta alternativa.....	220
Figura 81. Paso 1. Traducción del concepto “Arquitectura Religiosa”.....	221
Figura 82. Paso 2. Traducción del atributo “Nombre”.....	222
Figura 83. Paso 3 y 4. Traducción de los atributos “Población” y “Provincia”.....	222
Figura 84. Consulta para Base de Datos LDGP.....	223
Figura 85. Consulta para Base de Datos Páginas Web.....	223
Figura 86. Presentación al usuario de las Base de Datos implicadas en la consulta. ....	225
Figura 87. Presentación al usuario de los datos la Base de Datos “Páginas Web” .....	226
Figura 88. Descripción de una de las páginas Web de la solución. ....	227
Figura 89. Planificación inicial de la fase “Previo” .....	250
Figura 90. Resultado real de la fase “Previo”.....	251
Figura 91. Planificación inicial de la fase “Iniciación” .....	252
Figura 92. Resultado real de la fase “Iniciación” .....	252
Figura 93. Planificación inicial de la sub-fase “Estudio previo” .....	253
Figura 94. Resultado real de la sub-fase “Estudio previo” .....	253

Figura 95. Planificación inicial de la sub-fase “Análisis” .....	255
Figura 96. Resultado real de la sub-fase “Análisis” .....	255
Figura 97. Planificación inicial de la sub-fase “Diseño” .....	257
Figura 98. Resultado real de la sub-fase “Diseño” .....	257
Figura 99. Planificación inicial de la sub-fase “Implementación” .....	262
Figura 100. Resultado real de la sub-fase “Implementación” .....	263
Figura 101. Planificación inicial de la sub-fase “Implantación” .....	267
Figura 102. Resultado real de la sub-fase “Implantación” .....	267
Figura 103. Planificación inicial de la sub-fase “Pruebas” .....	268
Figura 104. Resultado real de la sub-fase “Pruebas” .....	268
Figura 105. Diagrama de Gantt correspondiente a la planificación inicial. ....	270
Figura 106. Diagrama de Gantt correspondiente al resultado real.....	271
Figura 107. Comparación (en horas) entre la planificación inicial (1ª barra) y el resultado real (2ª barra) .....	272
Figura 108. Comparación por fases (en horas) entre la planificación inicial y el resultado real .....	273
Figura 109. Comparación por tareas de la fase desarrollo (en horas) entre la planificación inicial y el resultado real .....	274

## ÍNDICE DE TABLAS

Tabla 1. Cuantificación de los riesgos. ....	41
Tabla 2. Fragmentación horizontal. Paso 1. ....	81
Tabla 3. Fragmentación horizontal. Paso 2 (Nombre-sucursal=Guadarrama). ....	82
Tabla 4. Fragmentación horizontal. Paso 2 (Nombre-sucursal=Cercedilla). ....	82
Tabla 5. Fragmentación vertical. Paso 1. ....	83
Tabla 6. Fragmentación vertical. Paso 2. Nombre-sucursal, Nombre-cliente, Id-tupla ....	83
Tabla 7. Fragmentación vertical. Paso 2. Número-cuenta, Saldo, Id-tupla ....	83
Tabla 8. Clasificación de datos del patrimonio histórico. ....	86
Tabla 9. Fragmentación horizontal de los datos. Diputación Foral de Álava. ....	87
Tabla 10. Fragmentación horizontal de los datos. Archivo Municipal. ....	87
Tabla 11. Fragmentación horizontal de los datos. UPV/EHU. ....	87
Tabla 12. Tabla Proyecto. ....	187
Tabla 13. Responsables de las pruebas. ....	233
Tabla 14. Fase “Previo”. Comparativa planificación inicial y resultado real. ....	251
Tabla 15. Fase “Iniciación”. Comparativa planificación inicial y resultado real. ....	252
Tabla 16. Sub-Fase “Estudio previo”. Comparativa planificación inicial y resultado real. ....	254
Tabla 17. Sub-Fase “Análisis”. Comparativa planificación inicial y resultado real. ....	256
Tabla 18. Sub-Fase (parte 1) “Diseño”. Comparativa planificación inicial y resultado real. ....	258
Tabla 19. Sub-Fase (parte 2) “Diseño”. Comparativa planificación inicial y resultado real. ....	260
Tabla 20. Sub-Fase “Diseño”. Comparativa planificación inicial y resultado real. ....	261
Tabla 21. Sub-Fase (parte 1) “Implementación”. Comparativa planificación inicial y resultado real. ....	263
Tabla 22. Sub-Fase (parte 2) “Implementación”. Comparativa planificación inicial y resultado real. ....	265

Tabla 23. Sub-Fase “Implementación”. Comparativa planificación inicial y resultado real. ....	266
Tabla 24. Sub-Fase “Implantación”. Comparativa planificación inicial y resultado real .....	267
Tabla 25. Sub-Fase “Pruebas”. Comparativa planificación inicial y resultado real.....	268
Tabla 26. Fase “Desarrollo” Comparativa planificación inicial y resultado real.....	269
Tabla 27. Fase “Desarrollo” Comparativa planificación inicial y resultado real.....	269

# 1. INTRODUCCIÓN

Este documento describe el desarrollo del Proyecto de Fin de Carrera (PFC) denominado “Desarrollo de una solución para el acceso integrado a fuentes de información sobre Patrimonio histórico. Análisis de las alternativas e implantación de una solución basada en una aplicación de búsqueda orientada en bases de datos heterogéneas”. El objetivo de este proyecto es conseguir una unificación de toda la información sobre el patrimonio histórico existente en diversas organizaciones, y elaborar una aplicación que sirva como puente para acceder a dicha información. Para conseguir esto, el proyecto consta de una parte de investigación, en la cual se analizan las diferentes soluciones posibles, y una parte de desarrollo, en la cual se desarrollan dos aplicaciones, una para dar soporte a una base de datos, también creada en el Proyecto de Fin de Carrera, que almacena información sobre los datos existentes en las páginas Web de aquellas organizaciones que trabajan con el patrimonio histórico, y otra que funcione como una aplicación de búsqueda orientada en bases de datos heterogéneas, en donde dichas bases de datos sean en un principio, la base de datos comentada anteriormente y la base de datos del Laboratorio de Documentación Geométrica del Patrimonio.

Para conseguir desarrollar esta solución el proyecto consta de diferentes fases que son descritas, algunas de manera exhaustiva, a lo largo de esta memoria. Esto es debido a que además del contenido de investigación y desarrollo, el proyecto conlleva una parte extensa dedicada a la gestión del mismo.

En la primera fase aparecerá un documento que nos explica cuál era la situación antes del comienzo del proyecto. La segunda fase consta de una primera parte en la cual se estudia la viabilidad del proyecto, y una segunda parte donde se explica el Documento de Objetivos del Proyecto, gracias al cual se acerca al lector a diferentes aspectos de la gestión inicial del proyecto.

La tercera fase es la denominada fase de desarrollo, en la cual se lleva a cabo un estudio previo de los datos, herramientas, y un análisis exhaustivo sobre las diferentes alternativas existentes para desarrollar una solución al proyecto. Además incluye la parte del diseño de la aplicación. Se trata de un diseño completo, en donde se explican cada una de las funcionalidades. Se puede decir que se trata del talón de Aquiles en el cual se apoya la implementación realizada. En cuanto a la implementación, en la memoria se lleva a cabo una explicación detallada acerca de cómo han sido implementadas las dos aplicaciones comentadas en el primer párrafo. Para finalizar esta fase de desarrollo, en la memoria se hace referencia al servidor de la universidad en donde están alojadas las aplicaciones, y además añade una fase de pruebas realizadas previamente a la instalación de la aplicación en el servidor de la Universidad del País Vasco.

Para terminar la memoria se ha descrito el seguimiento del proyecto, indicando para cada fase la duración inicial estimada y la duración real de misma, describiendo

los problemas que han acontecido a lo largo del proyecto, y las medidas que se han tomado al respecto. Además se incluye un apartado en el que se detallan las relaciones mantenidas con el cliente durante el desarrollo del Proyecto de Fin de Carrera, en el cual se explica, cómo surgió el proyecto, cómo se ha desarrollado, y el grado de satisfacción del cliente en cuanto al resultado se refiere.

Para completar la memoria del PFC se ha incluido un amplio apartado de anexos. En los anexos se detallan las organizaciones existentes en Álava y alrededores cuya actividad principal está relacionada con el patrimonio. Se explica de manera breve como se han clasificados los datos que han dado lugar al Árbol de Conceptos, en el cual se basa la aplicación de búsqueda orientada. Aparecen explicados, además, conceptos clave de las aplicaciones desarrolladas.

Como complemento a los anexos, aparecen los manuales de mantenimiento, y los manuales de uso de cada una de las aplicaciones. Por último también se describen todas las actas de reunión correspondientes a cada una de las reuniones que han sido mantenidas a lo largo del proyecto tanto con el cliente, José Manuel Valle y Álvaro Rodríguez, como con el tutor del proyecto, José Miguel Blanco.

## **1.2. PROPUESTA DE PROYECTO**

Nombre del alumno: Ion Marín Martín

### Introducción

El proyecto a realizar está previsto para cubrir una serie de necesidades expuestas por el Laboratorio de Documentación Geométrica del Patrimonio de la UPV/EHU.

Este proyecto de fin de carrera consta del estudio y desarrollo de una base de datos que almacene meta información sobre el patrimonio digital. A continuación, con el objetivo de profundizar en lo que va a ser el desarrollo del proyecto, se detallarán las características fundamentales de dicho proyecto.

### Descripción del proyecto

Actualmente, existen diversas instituciones (Diputación, Ayuntamiento, investigadores, empresas, organizaciones sin ánimo de lucro) o el propio Laboratorio de Documentación Geométrica del Patrimonio de la UPV/EHU, que poseen información digital sobre el patrimonio. Esta información se compone de fotos, planos, documentos de texto acerca del patrimonio, modelos 3d, etc.

Una vez conocidos estos datos, el objetivo de este proyecto es el de conseguir la accesibilidad a los metadatos de dicha información. La idea del proyecto es poder ofrecer al usuario una serie de información acerca de lo que él ha solicitado. Por ejemplo, si el usuario desea conseguir fotos de la catedral de Santa María de Vitoria-

Gasteiz, podrá conocer el lugar donde se encuentran esas fotos, el formato en el que están, el año en el que se tomaron... Es decir, lo que se pretende reunir son metadatos, la información en sí no estaría disponible.

Para poder desarrollar esta idea, es necesario estudiar de manera detallada cada una de las diferentes propuestas que nos puedan ir saliendo en el desarrollo del proyecto. De momento, y sin tener apenas conocimientos en bases de datos distribuidas, contamos con estas alternativas:

- Desarrollar un modelo de base centralizado de datos que permita la interacción entre la meta información disponible en las diversas instituciones u organizaciones, con lo que se pretende una búsqueda centralizada de la información que permitirá consultar la disponibilidad y ubicación de la información que necesiten los diferentes usuarios.
- Desarrollar un modelo de base de datos, y que cada “socio” lo coloque en su servidor con la meta información que dicho socio tenga disponible. Cuando se realice una consulta de información, el sistema hará una búsqueda en el modelo de base de datos de cada socio.
- Desarrollar un wiki, en el que cada colaborador deje disponible la meta información que considere importante.
- A lo largo del proyecto pueden aparecer más alternativas que serán tomadas en cuenta para el desarrollo de dicho proyecto.

Con el objetivo de escoger aquella alternativa que más se adecúe a nuestras necesidades, cada una de las alternativas será estudiada y probada en un pequeño experimento, siempre y cuando se crea conveniente. Esta parte, unida a la búsqueda de diversas alternativas para el desarrollo del proyecto, es la que denominamos fase de investigación del proyecto.

Además de todo esto deberemos estudiar para cada propuesta aspectos de seguridad física, seguridad en bases de datos, temas de concurrencia...

Para poder conseguir la información almacenada necesitaremos un buscador dentro de un sitio Web, para que cada usuario pueda consultar información que le pueda interesar. Para poder desarrollar este buscador necesitamos primero conocer la meta información que se va a almacenar.

Una vez conocida la información disponible y las primeras propuestas para reunir la meta información debemos realizar un estudio sobre las herramientas a utilizar. Lo que se pretende es utilizar herramientas que no queden obsoletas a corto plazo, y que puedan ser fácilmente actualizables. Para conseguir esto utilizaremos estándares que más se adecúen a nuestras necesidades.





## **2. ANTECEDENTES Y CONTEXTO**

Existen, en la actualidad, diferentes organizaciones cuya actividad principal es el patrimonio histórico. Cada una de estas organizaciones tiene en su poder una serie de datos que son, en ocasiones, utilizados por la propia organización para presentar información a diversos usuarios. Sin embargo existen diversas organizaciones cuyos datos no están expuestos al público, lo que provoca que mucha información acerca del patrimonio histórico quede completamente en desuso.

Hoy en día, si un usuario necesita información sobre un elemento particular del patrimonio histórico, tiene dos opciones, bien buscar a través de la Web información sobre dicho elemento, con lo que eso conlleva (información no relacionada con el tema, información inadecuada, necesidad de filtrar algunos contenidos, selección de aquello más importante, búsqueda entre páginas...), o bien acercarse a las oficinas de cada organización y consultar allí la información, con el esfuerzo que supone acercarse hasta cada organización... En definitiva no existe, actualmente, un sistema que permita al usuario realizar una búsqueda sobre un determinado elemento patrimonial, otorgándole información de manera clara, concisa y breve.

Lo que pretende este proyecto es conseguir reunir toda esa información, y facilitar al usuario la búsqueda de datos a través de una aplicación de búsqueda orientada. Para ello deberá contar con el apoyo de aquellas organizaciones que poseen datos acerca del patrimonio histórico. El objetivo de este proyecto es conseguir proporcionar al usuario un sistema de búsqueda orientada que le permita realizar consultas de cualquier tipo sobre el patrimonio, y cuya respuesta sea concisa, clara, y un espacio limitado de tiempo. Para esto el sistema de búsqueda orientada pretende contar en un futuro con aquellas bases de datos relacionadas con las organizaciones cuya actividad principal es el patrimonio histórico.

Por lo tanto, lo que se quiere conseguir es un producto innovador que sea una referencia para todos los usuarios en cuanto al patrimonio histórico se refiere, y que sea el punto de partida para conseguir la unificación de toda la información de patrimonio histórico de las organizaciones de Álava. El mayor beneficiario de este producto será el propio usuario.



### **3. INTERÉS Y VIABILIDAD DEL PROYECTO**

Con el fin de decidir si es positivo desarrollar o no un determinado proyecto, es imprescindible analizar diferentes factores. En primer lugar considero necesario que el producto que se va a elaborar cumpla uno de los siguientes requisitos.

- Crear un producto innovador que facilite la resolución de problemas o incluso los evite.
- Crear un producto ya existente y venderlo a un precio más competitivo que el original.
- Crear un producto ya existente pero que aporte funcionalidades de valor añadido.
- Crear un producto para cubrir las necesidades expuestas por el cliente.

En mi caso, aunque considero que podría cumplir tres de los cuatro puntos, (ya que me atrevería a decir que el producto a realizar puede considerarse innovador en su sector, aunque luego veremos que en otros ámbitos o sectores ya se utilizan aplicaciones muy similares. Por otro lado si ya existiera alguno como el que se pretende desarrollar, seguramente el precio de nuestro proyecto sería más competitivo, ya que su coste va a ser cero), diría que lo que realmente cumple es el punto cuatro.

#### **3.1 INTERÉS EN EL PROYECTO**

Para que un proyecto se lleve a cabo es necesario, en primer lugar, que alguien se interese por él, que el proyecto pueda tener un futuro uso, ya que elaborar algo para que quede en el olvido, lo considero una pérdida de tiempo y sobre todo de recursos, tanto físicos como mentales. En general, todo pedido solicitado por algún cliente tiene, si éste es desarrollado correctamente, un futuro uso. En mi caso es así, un cliente solicitó un servicio, que intentaré llevar a cabo de la manera más brillante posible en el espacio de tiempo concedido. El interés de mi cliente me hace ver la existencia de alguien interesado en que el proyecto que tengo entre manos salga adelante de la forma más correcta posible.

Los motivos por lo que el cliente se fijó en mí para el desarrollo de este proyecto son fáciles de explicar. No existe, a priori, ninguna aplicación en la universidad que se asemeje a la aplicación que se va a desarrollar de la cual ayudarse. Contratar una empresa externa para que realice el proyecto tiene un coste, que desconozco, pero que el cliente considera innecesario. La opción más apropiada es que algún profesional informático de la Universidad de la UPV/EHU realizara el proyecto, lo cual supondría coste cero, pero ocuparía al profesional sus horas de trabajo, por lo que se perdería un profesional. Con todo esto, para nuestro cliente, lo más adecuado es contar con un alumno de la Universidad de la UPV/EHU, futuro Ingeniero Informático al que únicamente le queda por elaborar su Proyecto de Fin de Carrera para terminar la carrera.

El alumno se encargaría de realizar el proyecto con el cual elaboraría su PFC. El coste sería cero y además las dos partes buscan el mismo objetivo, la mejor elaboración posible del proyecto. El cliente para conseguir que la aplicación funcione de la mejor forma posible, y el alumno para alcanzar la mejor nota posible en su Proyecto Fin de Carrera.

### 3.2 VIABILIDAD DEL PROYECTO

Antes de comenzar a realizar un proyecto, es importante analizar si el proyecto es viable, es decir, si a priori, éste puede llevarse a cabo sin problemas. Para analizar la viabilidad de un proyecto es necesario analizar la viabilidad en diferentes aspectos.

**Viabilidad económica.** Es una parte muy importante del proyecto, consta en realizar una estimación de costes, fijar un presupuesto, y analizar si con la aportación económica del cliente sale rentable llevarlo a cabo.

En este proyecto, el único coste es el de la electricidad, y la amortización del ordenador que utilizo, ya que el coste de la mano de obra es de 0 €

**Viabilidad estratégica / comercial.** Esta parte se asemeja a la explicada en el interés del proyecto. Es imprescindible un análisis del mercado para conocer si el producto a realizar ya es utilizado por otra empresa que se nos ha adelantado, o si se adapta a las necesidades de los usuarios de hoy en día.

En mi caso, como he comentado anteriormente, no tiene prácticamente importancia, ya el cliente me pide que elabore una aplicación que preste un determinado servicio, yo desarrollo la aplicación y cumplo con mi deber.

**Viabilidad técnica.** Para elaborar cada proyecto es necesaria una serie de herramientas determinadas. Es importante analizar su disponibilidad, y su perfecto funcionamiento antes de comenzar a desarrollar el proyecto ya que una herramienta en mal estado puede implicar la compra de una nueva y por consiguiente un gasto más.

Para llevar a cabo este proyecto disponemos de todas las herramientas necesarias, desde servidores de prueba hasta programas necesarios, pasando por ordenadores portátiles...

**Viabilidad organizacional / operativa.** Se considera tan importante como evidente. Es necesario que el equipo humano que va a realizar el proyecto tenga unos mínimos conocimiento del proyecto que va a desarrollar. Además es necesario tener certeza de que el número de trabajadores no sea tan pequeño como para no poder asegurar con certeza el acabado del proyecto en los plazos establecidos.

Quizás este sea el aspecto más difícil de asegurar en mi proyecto. Anteriormente a este proyecto nunca había trabajado con bases de datos a un nivel tan profundo, por lo que considero que aunque mis conocimientos sobre bases de datos no son escasos y habiéndolos incrementado durante esta última semana, aún no sean lo suficientemente buenos como para llevar a cabo el proyecto. Sin embargo, considero que el proyecto es

largo y hay cabida para la mejora en lo que a mis conocimientos se refiere. Por otro lado, aunque requiera esfuerzo, considero que el proyecto puede ser completado de forma satisfactoria con una única persona trabajando en él, yo, en el plazo de tiempo indicado.

### 3.3 PROYECTOS SIMILARES

Tras la reunión con el cliente, en la cual se explicó de manera más concisa los resultados que deseaba el cliente del proyecto a desarrollar, comenzamos a buscar proyecto de similar índole.

Precisamente sobre patrimonio digital no encontramos ninguna aplicación similar, sin embargo, relacionados con otros temas encontramos aplicaciones que nos pueden servir como referencia en el desarrollo de la aplicación.

Para que nos podamos hacer una idea de cuáles y cómo son estas aplicaciones, a continuación se presentan una serie de ellas muy parecidas entre sí.

- [www.muchoviaje.com](http://www.muchoviaje.com). Se trata de una aplicación consistente en lo siguiente, una página Web que contiene un buscador, en el cual se indica una ciudad origen una ciudad destino, la fecha de vuelo y el número y tipo de pasajeros. Acorde con los criterios fijados por el usuario, se mostrarán todos los vuelos disponibles en todas las compañías de vuelo. En términos de aplicación, desconozco si utilizan una base de datos centralizada con toda la información o una base de datos distribuida, donde cada compañía tiene un fragmento de la base de datos con sus vuelos.



Figura 1. Ejemplo del buscador de [www.muchoviaje.com](http://www.muchoviaje.com).

- [www.atrapalo.com](http://www.atrapalo.com) Se trata de una aplicación prácticamente igual a la primera. Consiste en una página Web que contiene un buscador en el que se fijan unos

criterios muy similares a los de muchoviaje.com (Ciudad de salida, destino, hora...). Además de vuelos, se pueden buscar hoteles. La aplicación, según la búsqueda, muestra bien todos los vuelos disponibles en todas las compañías de vuelo, o bien todos los hoteles disponibles en las compañías de hoteles.

Figura 2. Ejemplo del buscador de [www.atrapalo.com](http://www.atrapalo.com).

- De la misma índole que los dos ejemplos anteriores existen algunas aplicaciones más como por ejemplo <http://www.edreams.es/>.

Figura 3. Ejemplo del buscador de [www.edreams.es](http://www.edreams.es).

Como se puede observar, existen proyectos similares al que voy a desarrollar, no de la misma índole, pero si comunes desde el punto de vista informático. Esto no hace más que incrementar mis posibilidades de éxito, ya que si alguien lo ha hecho, ¿Por qué no voy a poder hacerlo yo? Si bien, probablemente este tipo de aplicaciones haya sido llevado a cabo por un grupo de profesionales de la informática, y para mi proyecto únicamente cuento con una persona, considero que con esfuerzo y dedicación se puede conseguir una aplicación más que satisfactoria.





## 4. DOCUMENTO DE OBJETIVOS DEL PROYECTO

### 4.1 OBJETIVOS

El **objetivo principal** del proyecto a realizar es conseguir un acceso uniformado a la meta-información relacionada con el patrimonio digital existente en las diferentes instituciones u organizaciones como pueden ser Diputación, Ayuntamiento o el propio Laboratorio de Documentación Geométrica del Patrimonio de la UPV/EHU, de tal forma que cualquier usuario pueda conocer la información disponible en cada institución y acceder a ella en el supuesto de que sea permitido por dicha institución, sea cual sea su ubicación física. Se trata de conseguir reunir toda la información disponible sobre el patrimonio de Álava, con el fin de facilitar su accesibilidad a cualquier usuario interesado en la materia. Todo esto además debe realizarse en un plazo de tiempo determinado ya que debe de estar terminado para comienzos del año 2009.

Durante el desarrollo del proyecto existen otros objetivos de menor alcance, tanto en tiempo como en volumen de trabajo, a los que denominamos sub-objetivos u objetivos secundarios. Alguno de los **sub-objetivos** que considero más importantes para el desarrollo del proyecto son los siguientes:

Durante la fase de análisis, existe una sub-fase que cuyo objetivo es la elección de aquella propuesta que nos permita conseguir la solución final más adecuada. Para ello se deberá realizar un profundo análisis de las propuestas reunidas anteriormente. Esta fase es de vital importancia ya que en ella se deberán de analizar correctamente cada uno de los pros y contras de cada propuesta, para en tareas posteriores poder escoger la solución más acertada para este proyecto.

Otro de los objetivos importantes del proyecto recae en la fase de diseño del mismo. El objetivo de esta fase, no es otro, que conseguir un diseño robusto, seguro y fiable, para que su consiguiente implementación sea sencilla. Para esto se deberá haber realizado un profundo estudio de los datos disponibles en cada institución u organización.

Para que el objetivo principal del proyecto se consiga, es muy importante ir consiguiendo todos los sub-objetivos, tanto los comentados anteriormente, como muchos otros que por resultar tan obvios he considerado prescindir de ellos. Para poder evaluar el desarrollo de estos sub-objetivos existe una fase que se prolonga durante todo el proyecto y se denomina "*seguimiento del proyecto*". Gracias a su cumplimiento, podremos analizar si los sub-objetivos se van realizando correctamente, y tomar las medidas necesarias para conseguir que los sub-objetivos posteriores sean culminados de manera brillante.

## 4.2 MOTIVACIÓN

Durante mi formación académica en la Universidad del País Vasco (UPV/EHU) no he tenido la oportunidad de realizar un proyecto de investigación extenso. Aunque sí que es cierto que en algunas asignaturas, sobre todo aquellas relacionadas con la programación, he llevado a cabo trabajos en los que he tenido que dedicar horas a la investigación, ya sea de las herramientas a utilizar, del lenguaje de programación o de la base de datos a utilizar, nunca me había permitido hacerlo de una forma tan extensa como, a priori, me lo podía permitir un Proyecto de Fin de Carrera.

Mediante un profesor, con quien había trabajado en el Proyecto de Fin de Carrera de la Ingeniería Técnica en Informática de Gestión, me llegó una propuesta de realización de un proyecto de investigación, en el que trabajaría con el Laboratorio de Documentación Geométrica del Patrimonio de la UPV/EHU. Acordé reunirme un día con ellos para que me explicaran en qué iba a consistir el proyecto, cuáles eran los objetivos del mismo, y donde empezaba y terminaba mi participación.

Desde el principio me convenció la propuesta, ya que podía realizar un proyecto de investigación que, en principio y si no existen problemas en su desarrollo, va a ser utilizado una vez finalizado. Además la parte del proyecto relacionada con la investigación está relacionada con las bases de datos, ámbito en el que únicamente conozco aquello relacionado con aplicaciones de software, es decir conexiones a bases de datos, operaciones en bases de datos, desconexiones de bases de datos... por lo que me ayudaría a aprender mucho más en un ámbito de la informática que considero muy importante en los tiempos que corren. Además el proyecto me permitiría trabajar bien desde mi casa o bien en la Universidad del País Vasco (UPV/EHU) de Vitoria-Gasteiz.

El proyecto, además del apartado de investigación, contiene una parte práctica que consiste en diseñar una base de datos que contenga toda la meta-información del patrimonio digital. Por lo que además de un parte de investigación el proyecto tiene una parte práctica en donde se podrán aplicar los conocimientos adquiridos durante la investigación.

El único problema que me surgió fue debido a mi desconocimiento total del alcance que tiene un proyecto de fin de carrera para una Ingeniería en Informática. No estaba seguro de que el proyecto que tengo previsto realizar fuera suficiente para abarcar un proyecto de fin de carrera, sin embargo, mi tutor del proyecto, José Miguel Blanco, vista la propuesta que le envíe, cree que es un proyecto interesante.

En resumen, considero que es un proyecto muy interesante para mí, creo que me va a aportar mucho, no sólo en conocimientos de bases de datos, sino también en el trato con el cliente. Voy a conocer cómo se trabaja según las exigencias de un cliente, lo cual me va a venir muy bien para mi futura salida laboral. Además el trabajo de investigación que voy a tener que realizar, va a ayudarme a conocerme más a mi mismo en lo que a mis métodos de trabajo se refiere.

### 4.3 DESCRIPCIÓN

Actualmente, existen diversas instituciones (Diputación, Ayuntamiento, investigadores, empresas, organizaciones) o el propio Laboratorio de Documentación Geométrica del Patrimonio de la UPV/EHU, que poseen información digital sobre el patrimonio. Esta información se compone de fotos, planos, documentos de texto, modelos 3d, etc.

Para poder acceder a esta información, hoy en día, es necesario acercarse a cada una de estas instituciones u organizaciones, con el esfuerzo y tiempo que conlleva. Además, una vez allí, desconoces el tipo de información sobre el patrimonio que está disponible en dicha organización hasta que no buscas o preguntas a algún responsable. Todo esto supone que cantidad de individuos aficionados o atraídos por el patrimonio de su provincia desconozcan información sobre el mismo que les resultaría, sin lugar a duda, interesante. Además de muchos otros, que en momentos puntuales, por simple curiosidad, o quizás porque vayan a realizar una visita turística a algún patrimonio de la provincia, no acceden a la información sobre el mismo por desconocer su ubicación física o incluso su existencia.

Para intentar facilitar la búsqueda de información sobre elementos patrimoniales, el Laboratorio de Documentación Geométrica del Patrimonio de la UPV/EHU ha solicitado que lleve a cabo una investigación sobre la manera más adecuada de reunir la información sobre el patrimonio existente en las diferentes instituciones u organizaciones.

Para ello, en primer lugar es necesaria una reunión con el cliente para que nos explique los datos que se quieren reunir, y sobre todo la manera de acceder a ellos. En función de las peticiones del cliente y del estudio de investigación elaborado por la persona que realiza el proyecto, en este caso yo, se intentará dar una solución lo más adecuada posible, siempre teniendo en cuenta los recursos disponibles.

En la reunión mantenida con el cliente se obtuvieron las primeras ideas claras sobre el proyecto. En primer lugar, el cliente indicó que la idea era elaborar un buscador en el que fijando una serie de criterios, que todavía están por definir, ya que para definirlos es imprescindible un estudio de los datos y de los meta-datos que posee cada una de las instituciones u organizaciones, se realizará una búsqueda y se indicase al usuario toda la información disponible acorde con los criterios fijados. Es decir, por ejemplo, un usuario necesita conocer la información existente sobre las catedrales de Vitoria-Gasteiz, entonces fijando los criterios relacionados con estos datos, la aplicación le mostraría toda la información existente (fotos, modelos 3D, artículos, memorias, planos...), según los criterios fijados, en todas y cada una de las instituciones u organizaciones.

Una vez conocidas las intenciones del cliente, “únicamente” queda investigar alternativas, realizar estudios sobre ellas, comparativas... y por supuesto implementar aquella que consideremos más apropiada para las exigencias del cliente. Además de la investigación que se vaya a llevar a cabo, es necesario mantener nuevas reuniones con

el cliente para conocer sus preferencias acerca de las propuestas, ya que el cliente, a fin y al cabo, es para quién se realiza este proyecto.

En un principio existen tres alternativas posibles, de las cuales las dos primeras son las que más opciones tienen de ser la elegida según las indicaciones dadas por el cliente. Las alternativas posibles son las siguientes:

- Desarrollar un modelo de base centralizado de datos que permita la interacción entre la meta información disponible en las diversas instituciones u organizaciones, con lo que se pretende una búsqueda centralizada de la información que permitirá consultar la disponibilidad y ubicación de la información que necesiten los diferentes usuarios. Esto es lo que se conoce como una Base de Datos Centralizada.

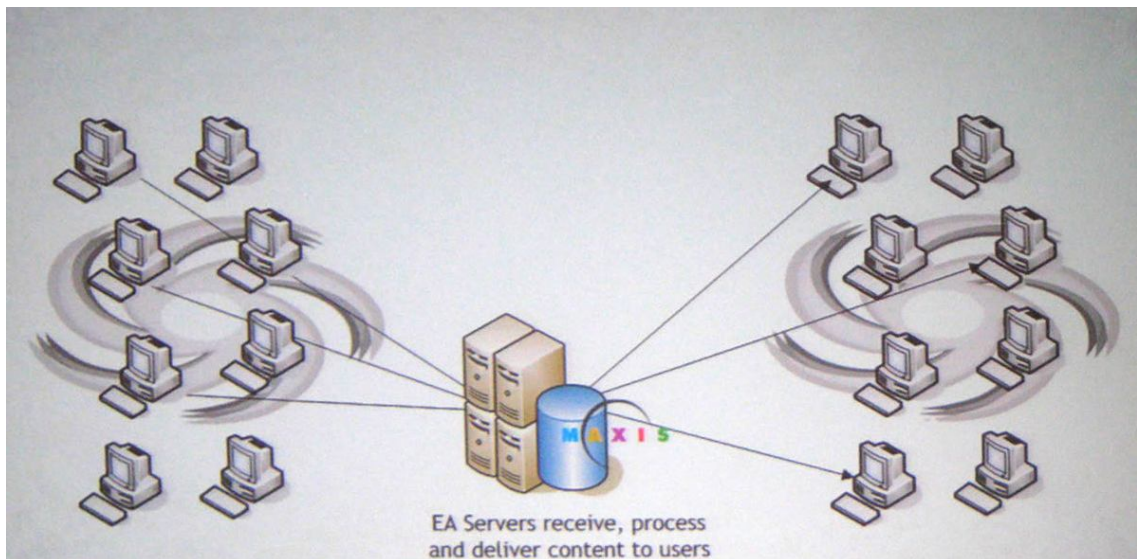


Figura 4. Ejemplo de BD Centralizada.

- Desarrollar un modelo de base de datos, y que cada “socio” lo coloque en su servidor con la meta información que dicho socio tenga disponible. Cuando se realice una consulta de información, el sistema hará una búsqueda en el modelo de base de datos de cada socio. Esto se conoce como una Base de Datos Distribuida.

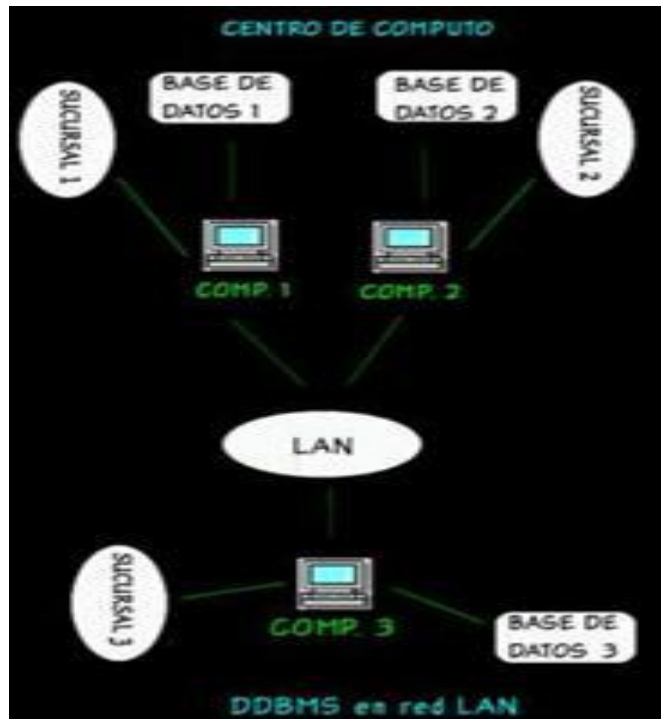


Figura 5. Ejemplo de BD Distribuida

- Desarrollar un wiki, en el que cada colaborador deje disponible la meta información que considere importante.
- A lo largo del proyecto pueden aparecer más alternativas que serán tomadas en cuenta para el desarrollo de dicho proyecto.

La tarea que debemos de realizar no es otra que realizar un profundo análisis sobre estas propuestas incluida alguna otra que pueda ir surgiendo a medida que avanza el desarrollo del proyecto.

Una vez elaborado dicho análisis, basándonos en comparativas, costes, prioridades del cliente, e incluso pequeños ejemplos de funcionamiento, escogeremos la opción que consideremos más adecuada. Esta opción escogida será la propuesta que llevaremos a cabo.

Para llevar a cabo la propuesta escogida, necesitaremos realizar una fuerte fase de diseño que nos permita conseguir una base de datos robusta.

Una vez terminada la aplicación, ésta será puesta en marcha lo antes posible con el fin de que los usuarios puedan disfrutar de este nuevo servicio que se les va a facilitar.

Para la elaboración del proyecto se utilizarán herramientas de software libre que además son estándares según el W3C.

En definitiva, se pretende realizar una aplicación fundamentada en una base de datos que facilite a cualquier usuario la búsqueda de información sobre el patrimonio.

## 4.4 RECURSOS NECESARIOS PARA EL DESARROLLO DEL PROYECTO

Para el desarrollo del proyecto vamos a necesitar una serie de recursos tanto software como hardware. Con el fin de conocer de antemano las herramientas a utilizar se disponen a continuación tanto los elementos hardware como los elementos software necesarios.

### 4.4.1 RECURSOS HARDWARE

- **Ordenador portátil** Asus, Intel Core 2 Duo T7300 con 2G de memoria RAM y 160 GB de disco duro. Este será el ordenador principal en el cual se desarrolle el proyecto. En él se implementará la parte experimental del proyecto. También almacenara todos aquellos documentos relacionados con la gestión del mismo, así como los documentos que forman la memoria final.
- **Servidor** en el cual trabaja el Laboratorio de Documentación Geométrica del Patrimonio de la UPV/EHU. El cliente nos da vía libre para utilizar este servidor, tanto como para almacenar pruebas parciales como el resultado final. El uso de este servidor nos facilitará la realización de pruebas.
- **Dispositivo USB** Verbatim de 4GB de almacenamiento. Dicho dispositivo nos permitirá almacenar las copias de seguridad que vayamos realizando, así como transportar información de un equipo a otro.
- **Ordenadores portátiles, ordenadores de sobremesa e incluso más servidores**, que el cliente pone a mi disposición en caso de que sea necesario. En cuanto a los ordenadores, es muy positivo saber que en el caso de avería, pérdida, o robo del ordenador portátil con el que trabajo, es posible conseguir otro con el que continuar trabajando en un corto periodo de tiempo. En cuanto a la posibilidad de contar con más de un servidor, nos facilita la realización de pruebas entre servidores antes de implantar la aplicación en los servidores de cada una de las instituciones u organizaciones.
- **Servidores de las diferentes instituciones y organizaciones.** Una vez terminado el proyecto en sí, es necesario, su implantación allí donde se vaya a utilizar, es decir, en los servidores de las diferentes instituciones y organizaciones. Para esto, es necesario que sus servidores funcionen correctamente, algo que sin lugar a dudas, no depende de nuestro proyecto.
- Se dispone de conexión a internet segura y con unas características óptimas, tanto en el lugar habitual de trabajo, mi domicilio, como en la Universidad del País Vasco (UPV/EHU), lugar ocasional de trabajo.



#### **4.4.2 RECURSOS SOFTWARE**

- Sistema operativo Windows Vista.
- MySQL 6.0.3 Alpha. Para el desarrollo del proyecto utilizaremos MySQL como sistema de gestión de bases de datos.
- Servidor Apache.
- PHP 5. Indispensable para el desarrollo del proyecto.
- Navegadores Web Internet Explorer, Mozilla Firefox, Opera para realizar todo tipo de pruebas que nos aseguren el correcto funcionamiento de nuestro buscador.
- Microsoft Office Visio 2003 para elaborar gráficos, esquemas y dibujos que se incluirán en la documentación.
- Microsoft Office Project 2003 para la elaboración de planificaciones y diagramas de Gantt.
- Microsoft Office 2007 para la elaboración documentos, entregables...

#### **4.4.3 EDITOR DE DESARROLLO**

El editor de desarrollo escogido para llevar a cabo el proyecto es EditPlus.

Fuente: Wikipedia (<http://en.wikipedia.org/wiki/EditPlus>) (Consultado el 2-IX-2008)

*“EditPlus es editor de texto para el sistema operativo Microsoft Windows, desarrollado por Kim Sangil de ES-Computing. El editor contiene herramientas para los programadores, incluidas las de sintaxis, las conversiones de tipo de archivo, las expresiones regulares para búsqueda y reemplazo, la grabación de pulsaciones de teclas, el corrector ortográfico, pleno apoyo a Unicode edición, auto-realización, plegado de código, y mucho más.”*

El principal motivo por el que he escogido este editor es debido a que puedo utilizarlo gratuitamente.

#### **4.4.4 SOPORTE DEL SISTEMA DE INFORMACIÓN**

A día de hoy es difícil saber cuál va a ser el soporte del sistema de información. En un principio el proyecto se va a desarrollar en un servidor local, existen plataformas que permiten el uso de servidores locales (como es el caso de la herramienta XAMPP), para posteriormente instalarlo en los servidores de la Universidad del País Vasco. La

idea de trabajar con un servidor local, es permitir al encargado de desarrollar el proyecto trabajar con su ordenador portátil desde donde desee.

Por lo tanto el soporte del sistema de información es aquel que proporcione la Universidad del País Vasco, el cuál ahora desconocemos.

## 4.5 PLANIFICACIÓN

Para realizar la planificación y gestión del proyecto se ha optado por la herramienta ofimática Microsoft Office Project 2003. Esta herramienta nos permitirá definir la planificación de una manera más visual, y mantener una disposición muy definida de cada una de las fases. La disposición de las tareas se sitúa en la parte izquierda. A cada una de estas les acompaña su duración, en términos diarios, su fecha de comienzo, su fecha de finalización y, en el caso de que tuviera, la tarea de la cual depende su comienzo. En la parte derecha se describe gráficamente cada una de las tareas, su duración y las dependencias entre ellas.

Con el fin de facilitar el seguimiento al lector les aclaramos la nomenclatura que sigue la herramienta utilizada. Las barras negras son fases del proyecto o tareas que engloban otras tareas. Existen dependencias entre fases y entre fases y tareas. Estas dependencias se muestran como flechas de color negro. Por otra parte, las barras azules son exclusivamente tareas y las flechas azules responden a dependencias entre tareas.

Atendiendo a la planificación se ha establecido una jornada laboral de cinco días semanales con diferentes horas diarias de dedicación al proyecto durante los meses en los que se extiende el proyecto. Las horas de trabajo no serán siempre las mismas debido a que durante el mes de julio, mes en el que comenzó el proyecto las horas eran sensiblemente menores a las que serán durante el resto del año. Además durante el primer mes, dedicado a reuniones con el cliente y constantes correcciones sobre la propuesta del proyecto y la planificación del mismo supondrá un trabajo menor a diario que tendrá como consecuencia que la tarea dure más días de lo previsto. Durante el resto del proyecto se dedicarán, siempre y cuando se pueda proseguir con alguna tarea, unas 7 u 8 horas diarias.

En circunstancias normales se trabaja de lunes a viernes dejando así los dos días el fin de semana para cubrir posibles desviaciones ocurridas durante los días de labor.

En principio los días festivos a nivel nacional se consideran festivos como tal. Las vacaciones se producen durante el mes de agosto, y los días comprendidos entre el 14 y el 21 de septiembre.

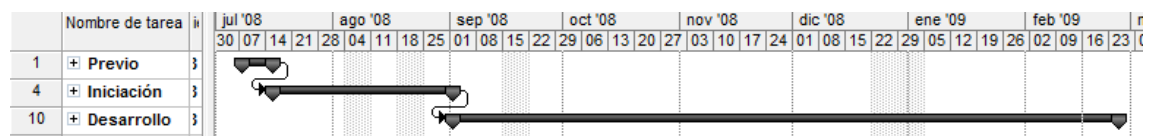


Figura 6. Diagrama de Gantt. Fases globales.

El proyecto, como se puede observar en el Diagrama de Gantt de la figura superior, está dividido en tres fases globales más una de seguimiento del proyecto (que no se incluye), que a su vez se subdividen en diferentes fases. A continuación se comentarán las diferentes fases globales con sus correspondientes sub-fases.

## ✓ Previo

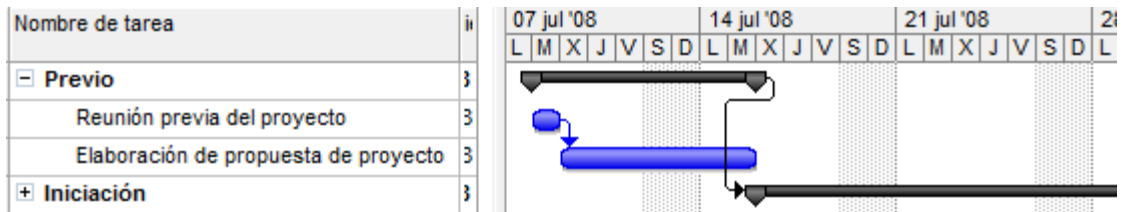


Figura 7. Diagrama de Gantt. Fase “Previo”.

### ○ Reunión previa del proyecto

La reunión se da lugar en la Universidad de la UPV/EHU de Vitoria-Gasteiz, a la cual asiste por una parte el cliente, una representación del Laboratorio de Documentación Geométrica del Patrimonio de la UPV/EHU y por otra el encargado de llevar a cabo el proyecto, Ion Marín Martín.

En esta reunión se cuestionan puntos como el enfoque, el alcance o los objetivos del proyecto así como el punto de partida y la duración estimada aproximada del mismo.

La dedicación estimada asciende a ~ 1 hora.

### ○ Elaboración de propuesta de proyecto

Esta tarea consiste en plasmar lo acontecido en la reunión en un documento que indique en qué va a consistir el proyecto, cómo se va a desarrollar, cual es su finalidad...

El entregable resultante será un documento que contenga lo anterior.

La dedicación estimada asciende a ~ 7 horas.

## ✓ Iniciación

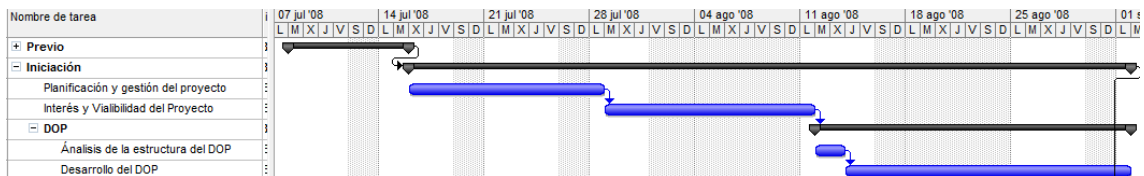


Figura 8. Diagrama de Gantt. Fase “Iniciación”.

### ○ Planificación y gestión del proyecto

Esta tarea consiste en la planificación del desarrollo del proyecto. En ella se establece la duración de cada una de las tareas así como la fecha

inicial del proyecto, obteniendo como resultado la fecha fin del mismo. Además de la duración se establecen las dependencias entre las tareas.

El entregable será un documento de Microsoft Office Project 2003 (tendrá una extensión .mpp) que recoja la propia planificación del proyecto.

La dedicación estimada asciende a ~ 10 horas.

#### ○ Interés y viabilidad del proyecto

La tarea consiste en una valoración del proyecto que se va a realizar visto desde puntos de vista diferentes. Por un lado explica los motivos por los que el proyecto se va a llevar a cabo, por otro lado explica las razones que llevan al cliente a estar interesado en el proyecto. Desde otro punto de vista el desarrollador del proyecto demuestra la viabilidad del proyecto. Para terminar se proponen una serie de proyectos similares que nos pueda servir de referencia o de ayuda para el desarrollo del proyecto.

El entregable de la tarea será un documento, que formará parte de la memoria del proyecto y que tendrá el mismo nombre que la tarea.

La dedicación estimada asciende a ~ 4 horas.

#### ○ D.O.P. (Documento de Objetivos del Proyecto)

##### ▪ Análisis de la estructura del D.O.P.

Debido a la falta de experiencia en el desarrollo de proyectos y al desconocimiento de lo que es un Documento de Objetivos del Proyecto, conviene que, antes de empezar a desarrollar el D.O.P., analice ejemplos de otros que hayan sido realizados por alguno de mis compañeros, con el fin de conocer su estructura, los pasos a seguir...

Para ello se estudiarán Documentos de Objetivos del Proyecto de otros proyectos de fin de carrera que están disponibles en la biblioteca de la facultad de informática de San Sebastián, FISS, o prestados por mi tutor del proyecto.

La dedicación estimada asciende a ~ 5 horas.

##### ▪ Desarrollo del D.O.P.

Esta tarea tendrá como objetivo recoger todos aquellos documentos relacionados con los objetivos del proyecto.

El entregable relacionado será una serie de documentos que formarán parte de la memoria como Documentos de Objetivos del Proyecto.

La dedicación estimada asciende a ~ 30 horas.

## ○ Desarrollo

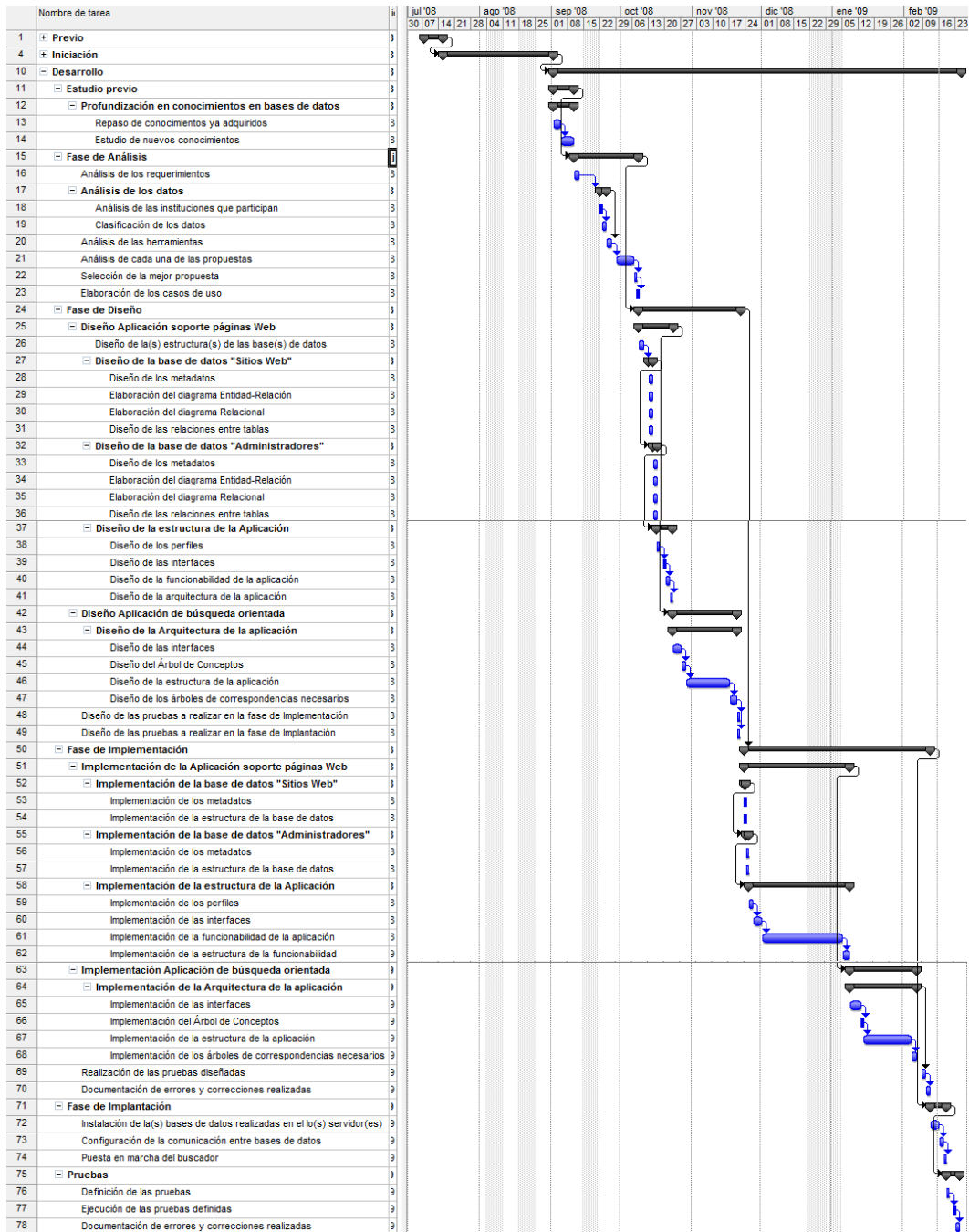


Figura 9. Diagrama de Gantt. Fase "Desarrollo".

## ○ Estudio previo

- Profundización en conocimientos en bases de datos

- ❖ Repaso de conocimientos ya adquiridos

Esta tarea consiste en el repaso de los conceptos más importantes de aquellas asignaturas de bases de datos que tenga relación con lo que se va a desarrollar en el proyecto.

La dedicación estimada asciende a ~ 20 horas.

- Estudio de nuevos conocimientos

Esta tarea consiste en el estudio de otras asignaturas que no han sido cursadas por el alumno durante la carrera, como puede ser Administración de Bases de Datos, que pueden resultar muy útiles para el desarrollo del proyecto.

La dedicación estimada asciende a ~ 25 horas.

## ○ Fase Análisis

- Análisis de requerimientos

Esta tarea consiste en recoger en un documento aquello que el cliente quiere obtener del proyecto que se desarrolla. Por esto es importante elaborar dicho documento bajo la supervisión del cliente.

El entregable de esta tarea será un documento que recoja lo anterior.

La dedicación estimada asciende a ~ 5 horas.

- Análisis de los datos

- ❖ Análisis de las instituciones que participan

Con esta tarea se pretende listar todas aquellas instituciones que proporcionen datos al proyecto.

El entregable será un documento que recoja una descripción sobre dichas instituciones junto con el material que proporcionan.

La dedicación estimada asciende a ~ 5 horas.

#### ❖ Clasificación de los datos

Para cumplimentar esta tarea se requiere elaborar un documento que incluya todas las aportaciones en forma de datos que cada institución aporta al proyecto. Para esto es necesario un estudio de las diferentes fuentes de datos.

Además esta tarea incluye un análisis de los datos que se han obtenido de las diferentes fuentes. Esta parte también quedará plasmada en un documento.

La dedicación estimada asciende a ~ 12 horas.

#### ▪ Análisis de las herramientas

Una vez realizado el estudio de los datos, se llevará a cabo un estudio de las herramientas. En él se realizarán comparativas sobre diversas herramientas, para escoger aquella que más se adecúe a las características del proyecto.

El entregable de esta tarea será un documento que recoja esta información.

La dedicación estimada asciende a ~ 15 horas.

#### ▪ Análisis de cada una de las propuestas

Esta tarea consiste en el análisis de cada una de las propuestas existentes para solucionar el presente proyecto. El número de propuestas se desconoce pero sí se conoce que tras este análisis deben quedar un máximo de dos propuestas, preferiblemente una.

Para esto debemos analizar la viabilidad de cada propuesta, realizar comparativas entre ellas para poder analizar cuál es la más viable de todas. Debemos tener en cuenta factores como los recursos disponibles, el tiempo estimado, el coste económico adicional si lo hubiera,...

El entregable de esta tarea será un documento en el que se analicen las propuestas.

La dedicación estimada asciende a ~ 50 horas.

#### ▪ Selección de la mejor propuesta

Una vez realizado un análisis completo de cada una de las propuestas en esta tarea deberemos elegir un máximo de dos propuestas, preferiblemente una, y eliminar el resto. Esta tarea



consiste meramente en elegir, estudiando el análisis realizado de cada tarea cuál es la mejor para nuestro proyecto.

El entregable será un documento en el que se expliquen los motivos de la elección de algunas y el rechazo de otras propuestas.

La dedicación estimada asciende a ~ 5 horas.

- Elaboración de los casos de uso

Esta tarea consiste en la elaboración de los casos de usos que se corresponden con las dos aplicaciones que se pretenden desarrollar a lo largo del proyecto de fin de carrera.

El entregable será un documento en el que se recojan dichos casos de uso.

La dedicación estimada asciende a ~ 6 horas.

- Fase de Diseño

- Diseño Aplicación soporte páginas Web

- ❖ Diseño de la(s) estructura(s) de las base(s) de datos

Esta tarea consiste en diseñar cuáles van a ser las bases de datos para la aplicación, de qué tipo deben de ser...

La dedicación estimada asciende a ~ 8 horas

- ❖ Diseño de la base de datos "Sitios Web"

- Diseño de los metadatos

En esta tarea se comienza el diseño de las bases de datos necesarias para llevar a cabo el proyecto. En esta tarea en concreto se diseñarán los metadatos de todas las bases de datos necesarias.

El entregable resultante será un documento que recoja el diseño de los metadatos de la bases de datos.

La dedicación estimada asciende a ~ 3 horas

- Elaboración del diagrama Entidad-Relación

Esta tarea consiste en la elaboración del diagrama Entidad-Relación correspondiente a la bases de datos “Sitios Web”

El entregable resultante será un documento que recoja dicho diseño en el que se incluye un esquema gráfico y una explicación del mismo.

La dedicación estimada asciende a ~ 5 horas

- Elaboración del diagrama Relacional

Una vez elaborado el diagrama Entidad-Relación es necesario realizar su paso a diagrama relacional. Esto es precisamente lo que se lleva a cabo en esta tarea.

El entregable resultante será un documento que recoja dicho diseño en el que se incluye un esquema gráfico y una explicación del mismo.

La dedicación estimada asciende a ~ 3 horas

- Diseño de las relaciones entre tablas

Una vez que se han diseñado los diagramas de Entidad-Relación y Relacional queda por definir la relaciones que van a tener las tablas entre ellas. Esto es precisamente lo que se lleva a cabo en esta tarea.

Esta tarea quedará reflejada dentro del entregable correspondiente al diseño de bases de datos.

La dedicación estimada asciende a ~ 3 horas

#### ❖ Diseño de la base de datos "Administradores"

- Diseño de los metadatos

En esta tarea se comienza el diseño de las bases de datos necesarias para llevar a cabo el proyecto. En esta tarea en concreto se diseñarán los metadatos de todas las bases de datos necesarias.

El entregable resultante será un documento que recoja el diseño de los metadatos de la bases de datos.

La dedicación estimada asciende a ~ 3 horas

- Elaboración del diagrama Entidad-Relación

Esta tarea consiste en la elaboración del diagrama Entidad-Relación correspondiente a la bases de datos “Sitios Web”

El entregable resultante será un documento que recoja dicho diseño en el que se incluye un esquema gráfico y una explicación del mismo.

La dedicación estimada asciende a ~ 5 horas

- Elaboración del diagrama Relacional

Una vez elaborado el diagrama Entidad-Relación es necesario realizar su paso a diagrama relacional. Esto es precisamente lo que se lleva a cabo en esta tarea.

El entregable resultante será un documento que recoja dicho diseño en el que se incluye un esquema gráfico y una explicación del mismo.

La dedicación estimada asciende a ~ 3 horas

- Diseño de las relaciones entre tablas

Una vez que se han diseñado los diagramas de Entidad-Relación y Relacional queda por definir la relaciones que van a tener las tablas entre ellas. Esto es precisamente lo que se lleva a cabo en esta tarea.

Esta tarea quedará reflejada dentro del entregable correspondiente al diseño de bases de datos.

La dedicación estimada asciende a ~ 3 horas

- ❖ Diseño de la estructura de la aplicación

- Diseño de los perfiles

Lo que en esta tarea se lleva a cabo es la distinción entre los diferentes grupos de usuarios que pueden acceder a la aplicación, ya que cada uno de los grupos tendrá unas posibilidades de acceso determinadas.

El entregable resultante será un documento que recoja los perfiles en los que se va a basar la aplicación.

La dedicación estimada asciende a ~ 7 horas

- Diseño de las interfaces

Esta tarea consiste en la elaboración de un documento que recoja el tipo de interfaces que van a ser utilizadas en el desarrollo del proyecto.

El entregable resultante será un documento que recoja dicha información.

La dedicación estimada asciende a ~ 7 horas

- Diseño de la funcionabilidad de la aplicación

En esta tarea se pretende llevar a cabo es realizar un estudio sobre las funcionalidades que va a necesitar el usuario para mantener actualizada la bases de datos “sitios Web”.

No requiere entregable.

La dedicación estimada asciende a ~ 14 horas

- Diseño de la estructura de la funcionabilidad

Esta tarea recoge la estructura que debe seguir la aplicación para conseguir desarrollar las funcionalidades que se desarrollan en el punto anterior.

Esta tarea quedará reflejada dentro del entregable correspondiente a la arquitectura de la aplicación.

La dedicación estimada asciende a ~ 7 horas

- Diseño Aplicación de búsqueda orientada

- ❖ Diseño de la Arquitectura de la aplicación

- Diseño de las interfaces

Lo que se pretende con esta tarea es explicar al lector el tipo de interfaces que se pretenden desarrollar a lo largo del proyecto.

El entregable resultante será un documento que recoja el diseño de las interfaces de la aplicación de búsqueda orientada o aplicación general del sistema.

La dedicación estimada asciende a ~ 15 horas

- Diseño del Árbol de Conceptos

Una vez que se ha llevado a cabo el estudio de los datos y su posterior análisis y clasificación, el siguiente paso, el que recoge esta tarea, es elaborar un Árbol de Conceptos que sea consecuente con dicha clasificación de los datos.

El entregable resultante recogerá dicho Árbol de Conceptos, y una descripción del mismo.

La dedicación estimada asciende a ~ 15 horas

- Diseño de la estructura de la aplicación

En esta tarea se pretende llevar a cabo el diseño de la estructura de la aplicación. Se trata de una tarea compleja y de larga duración, ya que en ella se basa toda la aplicación, y su fracaso puede conllevar con casi total seguridad el fracaso de la implementación de la misma.

El entregable resultante será un documento que explique cómo es el diseño de la aplicación.

La dedicación estimada asciende a ~ 80 horas

- Diseño de los Árboles de Correspondencias necesarios

Además para cada base de datos que se instale en nuestro sistema es necesario atribuirle un Árbol de Correspondencias. Esto es lo que se pretende desarrollar en esta tarea.

El entregable resultante recogerá dichos Árboles de Correspondencias, y una descripción de los mismos.

La dedicación estimada asciende a ~ 22 horas

- Diseño de las pruebas a realizar en la fase de implementación

Esta tarea consiste en recoger en un documento con todas y cada una de las pruebas que se van a llevar a cabo durante la fase de implementación, para posteriormente llevarlas a cabo en dicha fase.

El entregable de esta tarea será un documento que recoja todas las pruebas que se van a llevar a cabo durante la fase de implementación. Este documento formará parte de la memoria.

La dedicación estimada asciende a ~ 6 horas.

- Diseño de las pruebas a realizar en la fase de implantación

Esta tarea consiste en recoger en un documento con todas y cada una de las pruebas que se van a llevar a cabo durante la fase de implantación, para posteriormente llevarlas a cabo en dicha fase.

El entregable de esta tarea será un documento que recoja todas las pruebas que se van a llevar a cabo durante la fase de implantación. Este documento formará parte de la memoria.

La dedicación estimada asciende a ~ 6 horas.

- Fase de Implementación

- Implementación de Aplicación soporte páginas Web

- ❖ Implementación de la base de datos "Sitios Web"

- Implementación de los metadatos

- En esta tarea se lleva a cabo la implementación de los metadatos de una de las bases de datos necesarias para llevar a cabo el proyecto.

- El entregable resultante de esta tarea se podrá comprobar en la base de datos "Sitios Web"

- La dedicación estimada asciende a ~ 1 horas

- Implementación de la estructura de la base de datos

- Esta tarea consiste en implementar la base de datos "Sitios Web"

- El entregable resultante de esta tarea se podrá comprobar en la base de datos "Sitios Web"

La dedicación estimada asciende a ~ 7 horas

❖ Implementación de la base de datos "Administradores"

○ Implementación de los metadatos

En esta tarea se lleva a cabo la implementación de los metadatos de una de las bases de datos necesarias para llevar a cabo el proyecto.

El entregable resultante de esta tarea se podrá comprobar en la base de datos "Administradores"

La dedicación estimada asciende a ~ 1 horas

○ Implementación de la estructura de la base de datos

Esta tarea consiste en implementar la base de datos "Administradores"

El entregable resultante de esta tarea se podrá comprobar en la base de datos "Administradores"

La dedicación estimada asciende a ~ 7 horas

❖ Implementación de la estructura de la aplicación

○ Implementación de los perfiles

Lo que en esta tarea se trata es de llevar a cabo aquello que se diseñó en el apartado *Diseño de los perfiles*. Gracias a esto se podrán tratar diferentes grupos de usuarios.

El entregable resultante se podrá comprobar en la aplicación, así como en un documento relacionado con los perfiles de la aplicación, dentro del apartado "Implementación de la aplicación" de la memoria.

La dedicación estimada asciende a ~ 15 horas

○ Implementación de las interfaces

Lo que en esta tarea se trata es de llevar a cabo aquello que se diseñó en el apartado *Diseño de las interfaces*.

El entregable resultante se podrá comprobar en la aplicación. Además se podrá encontrar información sobre la implementación de las

interfaces dentro del apartado “Implementación de la aplicación” de la memoria.

La dedicación estimada asciende a ~ 14 horas

- Implementación de la funcionalidad de la aplicación

Lo que en esta tarea se trata es de llevar a cabo aquello que se diseñó en el apartado *Diseño de la funcionalidad de la aplicación*.

El entregable resultante se podrá comprobar, tanto en la aplicación, como en la memoria, ya que en la misma aparecerá para cada funcionalidad una explicación de la misma, acompañada de un diagrama de flujo y un diagrama de flujo de eventos. Todo esto dentro del apartado “Implementación de la aplicación” de la memoria.

La dedicación estimada asciende a ~ 100 horas

- Implementación de la estructura de la funcionalidad

Lo que en esta tarea se trata es de llevar a cabo aquello que se diseñó en el apartado *Diseño de la estructura de la funcionalidad de la aplicación*.

El entregable resultante se podrá comprobar en la aplicación.

La dedicación estimada asciende a ~ 22 horas

- Implementación Aplicación de búsqueda orientada

- ❖ Implementación de la Arquitectura de la aplicación

- Implementación de las interfaces

Lo que se pretende con esta tarea es desarrollar aquellas interfaces reflejadas en el apartado *Diseño de las interfaces*.

El entregable resultante se podrá comprobar en la aplicación que se desarrolle. Además se podrá encontrar en la memoria en el apartado “Implementación de la aplicación” de la memoria, relacionado con las interfaces de la aplicación de búsqueda orientada.



La dedicación estimada asciende a ~ 21 horas

○ Implementación del Árbol de Conceptos

Lo que se pretende con esta tarea es implementar el Árbol de Conceptos reflejado en el apartado *Diseño del Árbol de Conceptos*.

El entregable resultante se podrá comprobar en la aplicación que se desarrolle. Además el árbol de Conceptos formará parte de la memoria en uno de sus anexos.

La dedicación estimada asciende a ~ 8 horas

○ Implementación de la estructura de la aplicación

En esta tarea se implementa lo que se diseñó en el apartado *Diseño de la estructura de la aplicación*.

El entregable resultante se podrá comprobar en la aplicación que se desarrolle. A su vez, en el apartado “Implementación de la aplicación” de la memoria, nos encontraremos una explicación detallada de la misma.

La dedicación estimada asciende a ~ 110 horas

○ Implementación de los Árboles de Correspondencias necesarios

Esta tarea consiste en implementar los Árboles de Correspondencias necesarios para desarrollar la aplicación.

El entregable resultante se podrá comprobar en la aplicación que se desarrolle. Además los árboles de Correspondencias de cada una de las bases de datos del sistema formarán parte de la memoria en uno de sus anexos.

La dedicación estimada asciende a ~ 12 horas

▪ Realización de las pruebas diseñadas

Esta tarea consiste en la realización de pruebas a lo largo de toda la fase de implementación. Esta tarea nos servirá para descubrir puntos flacos de nuestra aplicación o posibles puntos de mejora.

La dedicación estimada asciende a ~ 10 minutos por prueba.

- Documentación de errores y correcciones realizadas

Tras la realización de cada prueba es necesario hacer una valoración de la misma, es decir, documentar como a resultado la prueba, documentar sus errores si es preciso, y realizar las correcciones realizadas.

El entregable de esta tarea será un documento que reúna lo anteriormente escrito, es decir, que documente el resultado de las pruebas los errores que se han producido y las correcciones llevadas a cabo para subsanar dichos errores.

La dedicación estimada asciende a ~ 1/2 horas por prueba.

#### ○ Fase de implantación

- Instalación de la(s) bases de datos realizadas en el/lo(s) servidor(es)

Una vez finalizada la parte experimental del proyecto, falta por realizar su implantación allí donde se necesita su funcionamiento. Esto es exactamente lo que se va a realizar en esta tarea. Se instalarán las bases de datos necesarias para que el proyecto realizado funcione correctamente.

El entregable de esta tarea será un documento que explique el funcionamiento real de la aplicación realizada.

La dedicación estimada asciende a ~ 12 horas.

- Configuración de la comunicación entre bases de datos

Además de la instalación de las bases de datos en los equipos necesarios, tal y como se ha comentado en fases anteriores, es necesario crear una comunicación entre bases de datos. Para ello deberemos realizar una serie de ajustes en cada uno de los equipos.

Esta tarea tendrá como entregable un documento que complete el entregable de la fase anterior.

La dedicación estimada asciende a ~ 12 horas.

- Puesta en marcha del buscador

Esta tarea consiste en dejar accesible el buscador en una página Web, para que a través de él cualquier usuario pueda conseguir la información que desee sobre el patrimonio digital.

La dedicación estimada asciende a ~ 7 horas.

## ○ Pruebas

### ▪ Definición de las pruebas

En esta tarea se definirán todas las pruebas que se van a realizar para comprobar el correcto funcionamiento del buscador on-line realizado. Estas pruebas serán realizadas por individuos ajenos al proyecto y con diferentes conocimientos sobre la informática con el fin de valorar la aplicación y con la opción de sugerir ciertas mejoras.

El entregable de esta tarea será un documento en el que se indicará el tipo de prueba que se va a realizar, así como la persona que la va a llevar a cabo.

La dedicación estimada asciende a ~ 7 horas.

### ▪ Ejecución de las pruebas definidas

Una vez definidas las pruebas, en esta tarea se llevará a cabo su ejecución. Se ejecutará cada una de las pruebas, dando paso en la siguiente tarea a analizar sus resultados, tanto satisfactorios como insatisfactorios.

El entregable de esta tarea será un documento que indique los resultados de las pruebas.

La dedicación estimada asciende a ~ 8 horas.

### ▪ Documentación de errores y correcciones realizadas

Una vez llevadas a cabo cada una de las pruebas, es necesario analizar los resultados obtenidos. Se deben corregir los errores, y comprobar que las nuevas soluciones aportadas son válidas.

El entregable de esta tarea será un documento que indique el tipo de errores cometidos y las consiguientes correcciones que hay que realizar.

La dedicación estimada asciende a ~ 20 minutos por prueba.

## 4.6. DIRECCIÓN DE RIESGOS DEL PROYECTO

La dirección de riesgos del proyecto incluye los procesos relacionados con la identificación, análisis y respuesta a los riesgos del proyecto. Incluye maximizar los efectos positivos de los distintos eventos y minimizar sus efectos negativos. A continuación se irá haciendo referencia a los procesos anteriormente citados.

### 4.6.1 IDENTIFICACIÓN DE RIESGOS

La identificación de los riesgos consiste en determinar qué tipo de riesgos es más probable que afecten al proyecto y documentar las características de cada uno.

- ✓ **Padecimiento de algún tipo de enfermedad por la persona que desarrolla el proyecto.**

Puede darse el caso que el encargado de llevar a cabo el proyecto caiga enfermo, ya sea por un periodo corto como por un periodo largo. Obviamente en función del tipo las consecuencias serán unas u otras, ya que en este caso es imposible que se produzca una sustitución de personal.

- ✓ **Existencia de algún problema técnico grave en el ordenador en el cual se desarrolla el proyecto o bien la pérdida o el robo del mismo, que implique la pérdida de los datos.**

Existe la posibilidad de que suceda algún imprevisto relacionado con el ordenador, ya sea un problema técnico o la pérdida o robo del mismo en el cual se encuentren los datos del proyecto. Esto puede conllevar, por una parte a la pérdida de los datos del proyecto, y por otra parte a la pérdida de los programas que se utilizan en el mismo. El problema puede ser de diferente magnitud en función de si se realizaron las copias de seguridad oportunas, y si es así, de la fecha en que se hicieron.

- ✓ **Avería, robo o pérdida de la(s) máquina(s) servidor(as)**

Puede producirse un robo, una pérdida o un problema técnico importante en el servidor(es) en el/los cual/es se va ubicar físicamente el/los modelo(s) de la(s) base(s) de dato(s). Las consecuencias de este problema pueden ser muy diferentes en función de la avería (si es una avería), en función de la existencia o no de los datos en otra ubicación física, y de la probabilidad de reemplazo del servidor averiado, perdido o robado.

- ✓ **Unión de los dos riesgos anteriores**

En un caso extremo puede producirse la avería, el robo, o la pérdida simultánea del servidor(es) y del ordenador portátil en el cual se desarrolla el

proyecto. Este caso, puede traer consecuencias devastadoras si por ejemplo las copias de seguridad se encuentran únicamente en dichos ordenadores.

✓ **Desviaciones como consecuencia de una mala gestión**

En un comienzo se realiza una planificación en función de lo que el encargado del proyecto prevé que va a durar cada fase del mismo, sin embargo, como es lógico, lo real muchas veces no tiene nada que ver con lo planificado. Además, esa discordancia suele ser provocada porque en la realidad, el tiempo dedicado siempre es mayor que el tiempo planificado.

✓ **Errores relacionados con el análisis, el diseño o la implementación**

Se suele decir que “*el que tiene boca se equivoca*”, por lo tanto es obvio que durante el desarrollo del proyecto se cometerán equivocaciones y fallos que pueden implicar un retraso en la planificación del proyecto.

✓ **Malos entendidos con algún individuo relacionado con el proyecto**

En este proyecto intervienen principalmente tres partes, el cliente, el tutor del proyecto, José Miguel Blanco, y la persona encargada del proyecto, Ion Marín. Es bastante común, que pueda existir algún problema de comunicación entre cualquiera de las tres partes que puede desencadenar en malos entendidos que pueden implicar un retraso en el desarrollo del proyecto.

✓ **Cambios del producto en función de peticiones del cliente**

Aunque en principio se mantuvo una reunión con el cliente en base a fijar el alcance del proyecto, es probable que a medida que éste se va desarrollando se produzcan modificaciones al respecto, ya que en un momento dado, el cliente pueda pedir un diseño determinado sobre alguna parte del proyecto, o puede que quiera que se añada una nueva funcionalidad, que se cambie alguna cosa...

#### **4.6.2 CUANTIFICACIÓN DE LOS RIESGOS**

La cuantificación de riesgos implica evaluar los riesgos y las interacciones entre riesgos para determinar el rango de los posibles resultados del proyecto. Para esto, se va a elaborar una lista, ordenada en función de la probabilidad que se considera puede producirse cada riesgo (de más a menos), en donde aparecen cada uno de los riesgos clasificados en probabilidad baja, media y alta según la probabilidad que sucedan.

Riesgo	Probabilidad alta	Probabilidad media	Probabilidad baja
Cambios del producto en función de peticiones del cliente	X		
Desviaciones como consecuencia de una mala gestión	X		
Errores relacionados con el análisis, el diseño o la implementación	X		
Malos entendidos con algún individuo relacionado con el proyecto		X	
Padecimiento de algún tipo de enfermedad por la persona que desarrolla el proyecto.		X	
Existencia de algún problema técnico grave en el ordenador en el cual se desarrolla el proyecto o bien la pérdida o el robo del mismo, que implique la pérdida de los datos.			X
Avería, robo o pérdida de la(s) máquina(s) servidor(as)			X
Unión de los dos riesgos anteriores			X

Tabla 1. Cuantificación de los riesgos.

#### 4.6.3 DESARROLLO DE RESPUESTAS A RIESGOS

El desarrollo de respuestas a riesgos implica la definición de pasos para aprovechar las oportunidades y responder a las amenazas. Las respuestas a las amenazas se pueden agrupar dentro de una de estas tres categorías:

**Evitar:** Eliminando una amenaza específica, normalmente eliminando la causa. Es prácticamente imposible eliminar todo el riesgo, pero ciertos sucesos con riesgo pueden frecuentemente eliminarse.

**Mitigar:** Reduciendo, en la medida que sea posible, el daño que pueda producir un determinado riesgo, bien reduciendo la probabilidad que ocurra (por ejemplo, comprando un antivirus para que no tengamos problemas con nuestro equipo), o bien

reduciendo el reduciendo el impacto que produce, ya sea económico como en términos tiempo.

**Aceptar:** Aceptando el riesgo y sus efectos si este se produce. Existen casos en los que realizar una medida preventiva puede suponer un coste muy alto tanto económico como de tiempo en relación de la probabilidad que existe de que un determinado riesgo se produzca. Es decir, para prevenir un riesgo del cual existe una remota probabilidad que suceda, podemos invertir mucho dinero, o tiempo.

A continuación analizaremos la respuesta a llevar a cabo en el caso de que se produjera cada uno de los riesgos citados anteriormente.

✓ **Padecimiento de algún tipo de enfermedad por la persona que desarrolla el proyecto.**

**Aceptar.** Ante el riesgo de enfermedad sólo podemos intentar cuidarnos día a día lo máximo que podamos, tal y como hacemos en nuestro día a día. Obviamente, si se produjera alguna enfermedad había que aceptarla.

✓ **Existencia de algún problema técnico grave en el ordenador en el cual se desarrolla el proyecto o bien la pérdida o el robo del mismo, que implique la pérdida de los datos.**

**Mitigar.**

- **De manera previa al suceso.** Con el fin de evitar la pérdida de los datos nos debemos de asegurar de realizar copias de seguridad de forma diaria, o incluso en periodos de tiempo más breve. Para evitar un problema técnico, bastará con realizar un uso correcto del equipo y para evitar una infección del mismo se aconseja el uso de un antivirus. Ante la pérdida o robo del equipo lo único que se puede hacer es tener cuidado.
- **De manera posterior al suceso.** La ausencia de nuestro ordenador, ya sea por motivos técnicos o por pérdida o robo del mismo, puede causar diferentes problemas, desde la pérdida de datos y programas, hasta la imposibilidad de trabajar con él desde casa. Para recuperar los datos y programas, es suficiente con recuperar las copias de seguridad e instalar los programas. Para poder trabajar desde casa, únicamente podemos pedir tanto al cliente como a algún conocido el préstamo de un ordenador.

✓ **Avería, robo o pérdida de la(s) máquina(s) servidor(as)**

**Mitigar.**

- **De manera previa al suceso.** Con el fin de evitar la pérdida de los datos nos debemos de asegurar de realizar copias de seguridad cada vez que se produzca algún cambio (ya que al servidor se irá subiendo la aplicación a medida que se va completando). Para evitar un problema técnico, bastará

con realizar un uso correcto del servidor. Ante la pérdida o robo del servidor lo único que se puede hacer es tener cuidado.

- **De manera posterior al suceso.** La ausencia del servidor, ya sea por motivos técnicos o por pérdida o robo del mismo, puede causar diferentes problemas, desde la pérdida de datos y programas, hasta la imposibilidad de trabajar, realizar cualquier tipo de pruebas.... Para recuperar los datos y programas, es suficiente con recuperar las copias de seguridad e instalar los programas. Para poder trabajar con el servidor, únicamente podemos pedir tanto al cliente, que es el encargado de tener el servidor a punto que pida otro prestado.

✓ **Unión de los dos riesgos anteriores**

**Mitigar.**

- **De manera previa al suceso.** Incluye la descripción de “*De manera previa al suceso*” de los dos riesgos anteriores.
- **De manera posterior al suceso.** Además de los problemas que nos surgen debido a la ausencia tanto del servidor como de nuestro equipo, comentadas en “*De manera posterior al suceso*” de los dos riesgos anteriores, se nos presenta una posible pérdida de los datos, ya que las copias de seguridad del equipo pueden estar almacenadas en el servidor y viceversa. Para resolver esto contaremos con una tercera ubicación física, un dispositivo externo de almacenamiento de datos, en donde guardaremos las distintas copias de seguridad.

✓ **Desviaciones como consecuencia de una mala gestión**

**Mitigar.**

- **De manera previa al suceso.** Realizar la fase de gestión del proyecto de una manera consecuente, es decir, otorgándole el tiempo y el valor que se merece y se necesita. Así mismo, otorgar un periodo de margen ya sea en cada una de las fases o al final del proyecto.
- **De manera posterior al suceso.** Si sucede algún incumplimiento de fechas intentar retomar el ritmo apretando un poco en fases posteriores.

✓ **Errores relacionados con el análisis, el diseño o la implementación**

**Aceptar.** En este aspecto es casi imposible poder prever el riesgo, ya que cuando se realiza un proyecto se intenta siempre hacerlo de la mejor manera posible, y aún así uno siempre hay cosas que están mal, y hay que volverlas a repetir.

✓ **Malos entendidos con algún individuo relacionado con el proyecto**



**Mitigar/Aceptar.** Para que si los malos entendidos se producen no produzcan consecuencias graves, es importante mantener un contacto permanente entre las tres partes. Preferiblemente mantener un contacto en persona cada cierto tiempo, ya que en muchos casos no es posible mantenerlo en menor espacio de tiempo, y mantener contacto cuanto menos semanal vía e-mail.

✓ **Cambios del producto en función de peticiones del cliente**

**Mitigar.** Para que los cambios se puedan adaptar cuanto antes, y no cuando sea demasiado tarde (cuando nos obligue a deshacer cosas), procurar estar en un continuo contacto con el cliente, mostrándole lo que se va completando y preguntándole si tiene alguna sugerencia para lo que posteriormente se va a realizar.

#### **4.6.4 CONTROL DE RESPUESTAS A RIESGOS**

Para controlar que se cumple el plan de respuestas a riesgos no se llevará ninguna a cabo media compleja, únicamente se realizará un seguimiento en las fechas marcadas como hitos del proyecto, con el fin de asegurarse de que se cumplen las medidas preventivas, y de si se ha llevado a cabo el plan de respuesta a riesgos en el caso de que algún riesgo se haya producido.

## 4.7 ENTREGABLES DEL PROYECTO

Después de realizar una descripción sobre el proyecto, de describir su alcance, de explicar los motivos por los que se produce el proyecto, de listar los recursos necesarios para cumplimentar el proyecto, de describir la planificación del mismo, y de realizar un profundo análisis de riesgos, queda por describir la esencia del proyecto, ¿qué es lo que se va a obtener de él?

Obviamente, antes de empezar un proyecto es necesario definir lo que se va a obtener del mismo, para por una parte poder realizar una comparativa sobre lo que se pretendía obtener y lo que se ha obtenido finalmente, y por otra parte, como en este caso, para satisfacer las necesidades del cliente. Lo que se obtendrá al finalizar el proyecto es lo siguiente.

1. Una solución fundamentada del problema que se nos presenta. Es decir, una aplicación que consiga reunir la meta-información de cada una de las instituciones u organizaciones. El entregable constaría del diseño y de la implantación de la estructura de la base de datos, y de las aplicaciones que se desarrollen.
2. Unido a la solución del punto anterior, un buscador situado en la Web, en el que fijando unos criterios de búsqueda, otorgue la información disponible, según los criterios fijados, en todas y cada una de las instituciones u organizaciones. El entregable reuniría el código del buscador, más las conexiones con las diferentes bases de datos.
3. Manual de usuario de la/s aplicación/es.
4. Manual de mantenimiento de la/s aplicación/es.
5. Memoria del proyecto, donde se incluye un estudio sobre la investigación realizada acerca de cuál se ha considerado la opción más correcta para reunir los meta-datos y los propios datos.



## 5. ANÁLISIS

### 5.1 ANÁLISIS DE REQUERIMIENTOS

En todos los proyectos existen una serie de requerimientos o necesidades que delimitan la funcionalidad mínima que el producto a desarrollar debe ofrecer. Un proyecto que no cumple los requerimientos que se le impusieron no es un proyecto válido si no que más bien se podría considerar un fracaso.

Principalmente, los requerimientos de un producto afectan a su funcionalidad, es decir, exigen que el producto que se va a desarrollar ofrezca una serie de características que beneficien a quien vaya a adquirir el producto, el cliente en nuestro caso. Estos requerimientos se conocen como requisitos funcionales.

Debido a que el proyecto que esta memoria describe se basa principalmente en la elaboración de un producto en función de las exigencias de un cliente determinado, se ha considerado que sea el propio cliente quien elabore una pequeña lista con los requerimientos que debe abordar este proyecto. Una vez descritos los mismos, éstos serán aprobados por el desarrollador del proyecto, Ion Marín.

A continuación se listan los requerimientos exigidos por el cliente para obtener un producto satisfactorio.

- ✓ Al tratarse de un desarrollo experimental, se pide obtener una solución y realizar una valoración crítica de la misma. La solución no será óptima debido a que se desconocen varios factores transcendentales y no se cuenta con la colaboración del conjunto de instituciones, pero será un primer paso fundamental sobre el cual se podrá empezar a recabar esta colaboración y diseñar soluciones más adecuadas.
- ✓ La solución adoptada debe funcionar como un buscador web convencional de tal manera que el usuario pueda realizar las búsquedas de la manera más intuitiva posible.
- ✓ Las búsquedas podrán realizarse directamente por palabras clave desde el nivel inicial o ir desplegando jerarquías para ir limitando el ámbito de búsqueda, por ejemplo [Patrimonio] -> [Patrimonio edificado] -> [Arquitectura religiosa].
- ✓ Las búsquedas deben realizarse tanto sobre información web previamente seleccionada como sobre bases de datos de instituciones relacionadas con el patrimonio alavés.
- ✓ Debido al carácter dinámico de la información que se va a catalogar y ofrecer, debe proporcionarse la manera de ir actualizando los enlaces a los contenidos y este proceso debe realizarse de la manera más simple y ágil posible.

## 5.2 ANÁLISIS DE LAS HERRAMIENTAS

### 5.2.1 COMPARATIVA DE DESARROLLADORES WEB

#### 5.2.1.1 ELECCIÓN DE DESARROLLADOR DE APLICACIONES WEB

Existen diversas tecnologías para el desarrollo de aplicaciones Web, sin embargo, hoy en día, se podrían destacar tres como las más punteras, PHP, JSP (Java) y .net (ASP.Net). Para el desarrollo del proyecto que se va realizar se propuso escoger entre una de estas tres. Para llevar a cabo esta elección, primero se realizará una breve descripción de cada una de ellas, posteriormente se analizarán sus ventajas e inconvenientes, el tercer paso será la elaboración de una comparativa entre ellas, y por último la conclusión donde se explicarán los motivos de la elección realizada.

##### 5.2.1.1.1 DESCRIPCIÓN DE LAS DIFERENTES DESARROLLADORES WEB

- **JavaServer Pages (JSP) (JAVA)**

Fuente: [http://es.wikipedia.org/wiki/Java\\_Server\\_Pages](http://es.wikipedia.org/wiki/Java_Server_Pages) (“Consultado el 15-X-2009”)

*“Es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.*

*Esta tecnología es un desarrollo de la compañía Sun Microsystems. La Especificación JSP 1.2 fue la primera que se liberó y en la actualidad está disponible la Especificación JSP 2.1.*

*Las JSP's permiten la utilización de código Java mediante scripts. Además es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Librerías de Etiquetas (TagLibs o Tag Libraries) externas e incluso personalizadas.”*

- **PHP**

Fuente <http://es.wikipedia.org/wiki/Php> (“Consultado el 15-X-2009”)

*“PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.*

*PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdof en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.*

*PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser embebido dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, aunque el número de sitios en PHP ha declinado desde agosto de 2005. Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web. La más reciente versión principal del PHP fue la versión 5.2.6 de 1 de mayo de 2008.”*

- **.Net (Asp.Net)**

Fuente: <http://es.wikipedia.org/wiki/.NET> (“Consultado el 15-X-2009”)

*“.NET es un proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. Basado en ella, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado.*

*.NET podría considerarse una respuesta de Microsoft al creciente mercado de los negocios en entornos Web, como competencia a la plataforma Java de Sun Microsystems y a los diversos framework de desarrollo web basados en PHP. Su propuesta es ofrecer una manera rápida y económica, a la vez que segura y robusta, de desarrollar aplicaciones –o como la misma plataforma las denomina, soluciones– permitiendo una integración más rápida y ágil entre empresas y un acceso más simple y universal a todo tipo de información desde cualquier tipo de dispositivo.*

*La plataforma .NET de Microsoft es un componente de software que puede ser añadido al sistema operativo Windows. Provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones, y administra la ejecución de los programas escritos específicamente con la plataforma. Esta solución es el producto principal en la*

*oferta de Microsoft, y pretende ser utilizada por la mayoría de las aplicaciones creadas para la plataforma Windows.*

*.NET Framework se incluye en Windows Server 2008 y Windows Vista. De igual manera, la versión actual de dicho componente puede ser instalada en Windows XP, y en la familia de sistemas operativos Windows Server 2003. Una versión "reducida" de .NET Framework está disponible para la plataforma Windows Mobile, incluyendo teléfonos inteligentes.*

#### **5.2.1.1.2 VENTAJAS E INCONVENIENTES DE CADA UNA DE LAS PLATAFORMAS**

*Una vez descritas las tres tecnologías, se van a listar las ventajas e inconvenientes de cada una de ellas. No se trata de escoger una necesidad y realizar una comparación sobre qué herramienta la cubre mejor, sino de listar las cualidades de cada una de ellas, así como sus debilidades, con el fin de poder realizar una comparación entre lo que cada una de ellas puede aportar a nuestra aplicación.”*

- **Java Server Pages (JSP) (JAVA)**

##### **Ventajas**

- Soporta sin problemas aplicaciones robustas.
- Es un lenguaje multiplataforma.
- Son fáciles de mantener, siempre y cuando estén bien desarrolladas (modificar, ampliar...)
- No aumentan significativamente el tiempo de respuesta en el caso de que la aplicación tenga que satisfacer miles o millones de peticiones concurrentes.
- Es más completo que PHP. Hay cosas que se pueden hacer con JAVA que son muy difíciles de hacer en PHP.
- Permite una distribución correcta del código (división del código en capas...)

##### **Inconvenientes**

- Son pocas las empresas que ofrecen hosting en Java, y además es mucho más caro que en PHP.
- Las aplicaciones en Java ocupan más memoria

- **PHP**

**Ventajas**

- Es totalmente gratuito
- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL
- Su portabilidad. PHP está disponible para una gran cantidad de sistemas operativos diferentes, Linux, freeBSD, Solaris, IRIX, MS windows.
- Es un lenguaje que está tan extendido que es sencillo encontrar información sobre código, imágenes...
- PHP está pensado la programación en web, y es esa condición la que hace de PHP un lenguaje muy potente y muy flexible a la hora de crear sitios dinámicos rápidos, fiables y relativamente fáciles de programar.

**Inconvenientes**

- No posee una abstracción de base de datos estándar, sino bibliotecas especializadas para cada motor (a veces más de una para el mismo motor).
- Por su diseño dinámico no puede ser compilado y es muy difícil de optimizar.
- Por sus características favorece la creación de código desordenado y complejo de mantener.
- No posee adecuado manejo de internacionalización, Unicode, etc.

- **.Net (Asp.Net)**

**Ventajas**

- Documentación. .NET ofrece mucha documentación de ayuda (herramientas, debuggers, editores) incluida en la IDE y de soporte. Esto simplifica el desarrollo y la implementación.



- Rendimiento: Todos los códigos que se ejecutan en el ambiente .NET son compilados, lo cual proporciona un gran rendimiento a diferencia de versiones interpretadas.
- Movilidad. Las aplicaciones pueden ser desplegadas en una amplia variedad de dispositivos.
- Escalabilidad y flexibilidad. Escalabilidad es la capacidad de un sistema para soportar más carga de trabajo, usualmente debida al aumento de usuarios que lo utilizan. .NET ofrece métodos de escalabilidad como la carga balanceada que permite a un clúster de servidores (varios servidores) colaborar y dar un servicio de forma simultánea. En cuanto a la flexibilidad: el modo de programación que se emplea permite agregar nuevos módulos sin modificar la aplicación en su totalidad.
- Estándar abierto: La plataforma .NET está basada en estándares (HTML, XML, SOAP, WSDL, UDDI).

### **Inconvenientes**

- Licencias: Es un código cerrado, no hay licencias libres. La infraestructura para desarrollar en .NET representa un alto costo para las empresas.
- NET no es multiplataforma. La plataforma .NET sólo está disponible para la familia Windows.

### 5.2.1.1.3 COMPARATIVA DE LOS LENGUAJES DE PROGRAMACIÓN DE CADA UNA DE LAS PLATAFORMAS

#### Sintaxis

- **Java:** Es el que mejor sintaxis tiene.
- ✓ **PHP:** No es demasiado buena.
- ✓ **VB.Net:** Quizás no esté a la altura de Java, pero es bastante completa.

#### Aprendizaje

- **Java:** Es el más difícil de los tres.
- ✓ **PHP:** Es el más sencillo
- ✓ **VB.Net:** Es más sencillo de aprender que java, aunque es más complicado que PHP.

#### Velocidad de desarrollo

- **Java:** Es el más lento.
- ✓ **PHP:** Es rápido si se usa algún framework.
- ✓ **VB.Net:** Es el más rápido, debido a la cantidad de componentes que tiene a su servicio.

#### IDE (Integrated Development Environments):

- ✓ **Java:** J2EE tiene varias herramientas comerciales, pero Eclipse probablemente sea la mejor.
- ✓ **PHP:** Para PHP no existe una que destaque sobre las demás, aunque también se puede utilizar Eclipse.
- **VB.Net:** Tiene Visual Studio que es una gran aplicación pero de coste elevado.

#### Soporte orientado a objetos:

- ✓ **Java:** Es su especialidad.

- **PHP:** No está a la altura de los otros dos, aunque ha mejorado mucho en las últimas versiones.
- ✓ **VB.Net:** Trabaja muy bien con objetos.

### **Rendimiento**

- **Java:** Más pesado que PHP.
- ✓ **PHP:** Suele ganar en velocidad y recursos.
- **VB.Net:** Más pesado que PHP.

### **Servidor Web:**

- ✓ **Java** Tiene versiones comerciales y open source.
- ✓ **PHP:** Tiene versiones comerciales y open source.
- **VB.Net:** Sólo funciona con IIS.

### **Librerías y frameworks:**

- ✓ **Java:** Tiene muchas, la mayoría gratuitas y open source.
- ✓ **PHP:** Tiene muchas, la mayoría gratuitas y open source.
- **VB.Net:** Tiene muchas, pero de pago.

### **Soporte y comunidad:**

- ✓ **Java:** Tiene numerosos foros de consulta pero menos que PHP.
- ✓ **PHP:** Al estar tan extendido cuenta con numerosos foros de consulta.
- ✓ **VB.Net:** Tiene mucho soporte, además la mayoría de los foros, grupos de usuarios y comunidades de desarrolladores están manejados por Microsoft.

### **Coste**

- ✓ **Java:** Puede desarrollarse con herramientas gratuitas y de pago.
- ✓ **PHP:** Es totalmente gratuito.

- **VB.Net:** Es de pago.

#### 5.2.1.1.4 CONCLUSIÓN

Para poder tomar una decisión acertada sobre cuál es el desarrollador de aplicaciones Web más apropiado para el proyecto, primero se debe analizar las necesidades y las dimensiones del mismo, además de los recursos económicos disponibles. En cuanto a los recursos económicos, las herramientas que se deben utilizar para el desarrollo de este proyecto han de ser de coste cero. Esta restricción provoca que la decisión sobre cuál es el desarrollador de aplicaciones que más se ajusta a las características del proyecto, se transforme en la decisión de cuál es la plataforma que más se adapta a las características del mismo. Esto es, no sólo se debe comparar PHP, Visual Basic .net, y Java, sino que la comparación debe llevarse a cabo también, es aspectos como el entorno de desarrollo, el servidor Web, o el propio Sistema Operativo.

Por lo tanto debemos comparar el framework completo formado por Windows, IIS, .Net y Visual Basic.net, con los framework completos formados, por un lado, por Linux/Windows, Apache, PHP (Conocida como la plataforma XAMPP) más un entorno de desarrollo gratuito (existen varios) , y por otro lado, Linux/Windows, Apache Tomcat, Java más un entorno de desarrollo gratuito.

Como se puede observar el Sistema Operativo no es lo más influyente en la decisión a tomar, ya que existen entornos de desarrollo para Windows, Sistema Operativo en el que se desarrollará el proyecto, para cualquiera de las plataformas (.net, XAMPP, Apache Tomcat + Java). Sin embargo, el coste del entorno de desarrollo no es el mismo para todas las plataformas. .NET utiliza Visual Studio, que es de pago, mientras que tanto XAMPP como Java + Apache Tomcat, tienen a sus disposición numerosos entornos de desarrollo gratuitos. Además el servidor Web utilizado por la plataforma .net es IIS, que es de Windows, y de pago, mientras que los servidores Web Apache, son libres. Debido a esto, y a pesar de que VB.Net está considerado como un lenguaje de programación Web completo y de manejo sencillo, de debe descartar la elección de .NET como plataforma de desarrollo de aplicaciones Web, ya que su entorno de desarrollo es de pago.

Rechazada la posibilidad de utilizar la plataforma de desarrollo de software .NET, el análisis se centra en las otras dos posibilidades. Se puede decir, tal y como se puede observar a lo largo del análisis de las herramientas realizado, que se trata de dos desarrolladores Web muy diferentes en cuanto a programación se refiere, sin embargo muy similares a los servicios que ofrecen. Tanto los servidores Web, como los entornos de desarrollo son bastante similares (incluso pueden ser los mismos), por lo que la decisión va basarse en el lenguaje de programación que más se adapte a las características del proyecto a desarrollar. Centrándonos en los lenguajes de programación en sí, es obvio, que Java es más potente que PHP, sin embargo, en parte debido a esto, su velocidad de desarrollo, sin embargo, es menor que la de PHP. A favor

de Java está su seguridad, y su gran soporte para aplicaciones de grandes dimensiones. Por el contrario PHP tiene ventaja en aspectos como el tiempo estimado de aprendizaje del lenguaje, aunque esto puede ser relevante o no, dependiendo de si el encargado de desarrollar el proyecto ha trabajado o no con alguno de los dos lenguajes previamente, y su rendimiento en aplicaciones no muy extensas.

Si se analizan las necesidades del proyecto que se está desarrollando nos damos cuenta que es más necesario conseguir una aplicación sencilla y rápida, que una aplicación robusta. La aplicación que se está llevando a cabo no requiere de programación extensa en cuanto a dimensiones, por lo que se considera que con PHP será más que suficiente.

Además, para el desarrollo del proyecto se necesita un lenguaje de programación capaz de leer y escribir datos e documentos XML, y PHP nos permite realizar todo esto perfectamente.

En definitiva, se utilizará PHP, embebido en la plataforma XAMPP, como desarrollador de aplicaciones Web para llevar a cabo el proyecto.

## 5.2.2 XAMPP

Fuente: <http://es.wikipedia.org/wiki/XAMPP> (“Consultado el 15-X-2009”)

*“XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris, y MacOS X.*

### **Características y requisitos**

*XAMPP solamente requiere descargar y ejecutar un archivo zip, tar, o exe, con unas pequeñas configuraciones en alguno de sus componentes que el servidor Web necesitará. XAMPP se actualiza regularmente para incorporar las últimas versiones de Apache/MySQL/PHP y Perl. También incluye otros módulos como OpenSSL y phpMyAdmin. Para instalar XAMPP se requiere solamente una pequeña fracción del tiempo necesario para descargar y configurar los programas por separado.*

### **Aplicaciones**

*Oficialmente, los diseñadores de XAMPP sólo pretendían su uso como una herramienta de desarrollo, para permitir a los diseñadores de sitios Webs y programadores testear su trabajo en sus propios ordenadores sin ningún acceso a Internet. En la práctica sin embargo, XAMPP es utilizado actualmente para servidor de sitios Web, y con algunas modificaciones es generalmente lo suficientemente seguro para serlo. Con el paquete se incluye una herramienta especial para proteger fácilmente las partes más importantes.*

*A continuación se va a describir cada una de las herramientas que componen la plataforma XAMPP.*

#### **5.2.2.1 APACHE**

*El servidor **HTTP Apache** es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1<sup>[1]</sup> y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA Http 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que Behelendorf eligió ese nombre porque quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de*

*internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server (un servidor "parcheado").*

*El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.*

*Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.*

*Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años. (Estadísticas históricas y de uso diario proporcionadas por Netcraft).*

*La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache.*

### **Ventajas**

- *Modular*
- *Open source*
- *Multi-plataforma*
- *Extensible*
- *Popular (fácil conseguir ayuda/suporte)*
- *Gratuito*

### **5.2.2.2 MYSQL**

*MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems— desarrolla MySQL como software libre en un esquema de licenciamiento dual.*

*Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero las empresas que quieran incorporarlo en productos privativos pueden*

*comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.*

*Al contrario que proyectos como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, MySQL es propiedad y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.*

*Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson, y Michael Widenius.*

### **5.2.2.3 PHP**

*PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.*

*PHP es un acrónimo recursivo que significa **PHP Hypertext Pre-processor** (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdof en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.*

*PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser embebido dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, aunque el número de sitios en PHP ha declinado desde agosto de 2005. Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web. La más reciente versión principal del PHP fue la versión 5.2.6 de 1 de mayo de 2008.*

#### **Visión general**

*El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta.*



*También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.*

*Aunque todo en su diseño está orientado a facilitar la creación de página web, es posible crear aplicaciones con una interfaz gráfica para el usuario, utilizando la extensión PHP-Qt o PHP-GTK. También puede ser usado desde la línea de órdenes, de la misma manera como Perl o Python pueden hacerlo, a esta versión de PHP se la llama PHP CLI (Command Line Interface).*

*Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Mediante extensiones es también posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos.*

*Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.*

*PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como UNIX (y de ese tipo, como Linux o Mac OS X) y Windows, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.*

*PHP es una alternativa a las tecnologías de Microsoft ASP y ASP.NET (que utiliza C# VB.NET como lenguajes), a ColdFusion de la compañía Adobe (antes Macromedia), a JSP/Java de Sun Microsystems, y a CGI/Perl. Aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, existe además un IDE (entorno de desarrollo integrado) comercial llamado Zend Studio. Recientemente, CodeGear (la división de lenguajes de programación de Borland) ha sacado al mercado un entorno integrado de desarrollo para PHP, denominado **Delphi for PHP**. Existe un módulo para Eclipse uno de los IDE más populares.*

### **Características**

- **Ventajas**
  - *Es un lenguaje multiplataforma.*
  - *Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL*
  - *Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).*

- *Posee una amplia documentación en su página oficial ([2]), entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.*
- *Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.*
- *Permite las técnicas de Programación Orientada a Objetos.*
- *Biblioteca nativa de funciones sumamente amplia e incluida.*
- *No requiere definición de tipos de variables.*
- *Tiene manejo de excepciones (desde php5).*

- ***Desventajas***

- *No posee una abstracción de base de datos estándar, sino bibliotecas especializadas para cada motor (a veces más de una para el mismo motor).*
- *No posee adecuado manejo de internacionalización, Unicode, etc.*
- *Por su diseño dinámico no puede ser compilado y es muy difícil de optimizar.*
- *Por sus características favorece la creación de código desordenado y complejo de mantener.”*

### 5.3 ANÁLISIS DE LOS DATOS

Durante algo más de cinco semanas hemos estado inmersos en conversaciones con las diferentes instituciones de Álava y las distintas organizaciones dedicadas a diversas actividades relacionadas con el patrimonio de dicha provincia. Muchas de estas organizaciones consideran el proyecto como una iniciativa muy positiva que puede resultar de gran utilidad para conseguir reunir de una forma estructurada los datos del patrimonio alavés. Tanto es así, que varias de las organizaciones han mostrado su interés en colaborar, de forma material (aportando datos), con el proyecto. Sin embargo, a la hora de materializar esta colaboración, existen diversas dificultades que han dado como consecuencia que, por el momento, se descarte su colaboración.

Dos de las dificultades más importantes son, por un lado, la escasa disponibilidad de datos informatizados que tiene cada organización, y por otro lado, el retraso, en términos de tiempo, que supone para la persona encargada de llevar a cabo el proyecto, esperar a que cada organización, en el supuesto que se produjera, permitiera analizar y utilizar sus bases de datos.

Debido a estas dos dificultades principalmente, decidimos utilizar los datos ya existentes en las diferentes páginas Web de cada institución u organización. Con estos datos la idea es realizar la misma aplicación de búsqueda orientada, sólo que en vez de buscar en las bases de datos de las organizaciones, lo hiciera en sus sitios Web. Además, para completar la aplicación, la búsqueda también se haría en una base de datos, esta base de datos sería la del Laboratorio de Documentación Geométrica del Patrimonio de la Universidad del País Vasco (UPV/EHU). De esta forma, si en un futuro se quiere integrar una base de datos al sistema, se podrá hacer siempre y cuando se consiga aislar la base de datos de la aplicación de búsqueda, pero esto ya lo comentaremos más adelante.

Primeramente, nos centramos en la búsqueda orientada que se realiza en las diferentes páginas Web. Para poder llevar a cabo una búsqueda de este tipo, es imprescindible clasificar el contenido de los diferentes sitios Web. Por esto, nuestra primera tarea fue elaborar una lista, enumerando, de las diferentes organizaciones que realizan alguna actividad relacionada con el patrimonio, aquellas que disponían de página Web (Ver Anexo A.1. Descripción de las entidades contactadas). También se contó con sitios Web de instituciones como la Diputación Foral de Álava o el Ayuntamiento de Vitoria-Gasteiz. Una vez realizado este paso, había que analizar cada página Web, comprobar que apartados eran interesantes, cuáles no, había que realizar una clasificación dentro de cada sitio Web de los datos que se encontraban en el mismo... (Ver Anexo A.2. Clasificación de los datos)

Terminado el análisis y clasificación de los datos de cada sitio Web, el siguiente paso es realizar una clasificación global de los datos de todos los sitios Web. Este paso es uno de los más complicados, ya que nos damos cuenta de que, por un lado, existen

páginas Web con un contenido muy superficial, y por otro lado, existen otras con un contenido muy detallado. Por ejemplo, existen páginas que te citan iglesias, o simplemente realizan una descripción global sobre las mismas, contrario a esto, nos encontramos con páginas Web que describen de manera detallada cada una de las iglesias que se encuentran en los diferentes municipios de nuestra provincia. Otro de los problemas que nos encontramos es los diferentes tipos de datos que pueden existir sobre un patrimonio determinado. Por ejemplo, en algunos sitios Web aparecen solamente imágenes, por el contrario, en otros la información que se muestra es únicamente descriptiva. Además de esto, nos encontramos con páginas Web que utilizan su propio buscador que trabaja contra su base de datos, base de datos que obviamente nosotros nos podemos utilizar directamente, por lo que habrá que clasificar esa información de manera que en función de lo que el usuario quiera buscar, el sistema le muestre dicho buscador como una posible solución. Estos son sólo algunos de los problemas que nos hemos ido encontrando a lo largo de la clasificación de datos.

A pesar de los problemas anteriores, finalmente hemos conseguido realizar una clasificación, a nuestro juicio, sencilla y bastante intuitiva del contenido de las páginas Web.

Como se ha comentado anteriormente, además del uso de las páginas Web como fuentes de datos, el proyecto también va a contar con otra fuente de datos diferente. Esta fuente de datos va a ser la base de datos del Laboratorio de Documentación Geométrica del Patrimonio de la Universidad del País Vasco (UPV/EHU). Obviamente los datos existentes en la base de datos también deben ser clasificados. Siguiendo con la clasificación realizada anteriormente para los datos de los sitios Web de las organizaciones, nos hemos dado cuenta que los datos existentes en la base de datos son clasificables perfectamente dentro de esta clasificación, por lo que, en principio, consideramos la clasificación como válida para los datos con los que contamos.

Finalmente, se consiguió realizar una clasificación de los datos de manera que en mayor o menor medida todos los datos disponibles en las diversas fuentes de datos estuvieran representados. La consecuencia de esta clasificación de los datos ha sido el diseño del árbol de conceptos (Ver Anexo B.1. Árbol de Conceptos).

En el supuesto que en un futuro se unieran nuevas organizaciones al sistema que aportaran datos que no pudieran ser recogidos por nuestra clasificación, bastaría con modificar el Árbol de Conceptos de la aplicación para que los nuevos datos pudieran ser clasificados.

## 5.4 ANÁLISIS DE LAS POSIBLES SOLUCIONES

El proyecto en el que estamos inmersos en estos momentos nació para cubrir una serie de necesidades expuestas por el Laboratorio de Documentación Geométrica del Patrimonio de la UPV/EHU. La necesidad principal se basa en lo siguiente. Actualmente, existen diversas instituciones (Diputación, Ayuntamiento, investigadores, empresas, organizaciones sin ánimo de lucro) o el propio Laboratorio de Documentación Geométrica del Patrimonio de la UPV/EHU, que poseen información digital sobre el patrimonio. Esta información se compone de fotos, planos, documentos de texto acerca del patrimonio, modelos 3d, etc. El objetivo principal de este Proyecto de Fin de Carrera consiste en “reunir” toda esa información, de manera que cuando un usuario necesite cualquier información sobre el patrimonio, no tenga que ir buscando la información página por página sino que directamente, y mediante una sola aplicación, consiga la información que busca.

Cuando se empezó a tomar contacto entre las dos partes, cliente y desarrollador, hacia mediados de julio, el cliente tenía claro lo que quería conseguir, sin embargo desconocía completamente la manera de llevarlo a cabo, desconocía las instituciones que estaban dispuestas a colaborar, los datos de los que se dispondría, desconocía también si se iba a tener la posibilidad de manejar sus fuentes de datos... En definitiva, se sabía lo que se quería conseguir pero se desconocía, el cómo se iba a llevar a cabo, y los medios con los que se contaban.

El cliente se comprometió a establecer reuniones con las diferentes instituciones con el fin de informarles del proyecto que se iba a realizar y conocer si éstas estaban dispuestas a colaborar aportando, entre otras cosas, sus datos, el acceso a los mismos... Como ocurre siempre en estos casos, las conclusiones derivadas de las reuniones se posponían, ya que por un lado era difícil encontrar un día para reunirse, y por otro lado, los responsables de algunas instituciones, que generalmente eran las personas con las que el cliente mantenía las reuniones, desconocía como tenían estructurados los datos, y aunque mostraban interés en el proyecto que se iba a llevar a cabo, desconocían completamente en qué podían colaborar ellos.

Desde las primeras reuniones entre el cliente y el desarrollador del proyecto se barajaron dos opciones, *Base de Datos Centralizada* y *Bases de Datos Distribuidas*, a la que más adelante se sumó una tercera, *Arquitectura para Federación de Bases de Datos Documentales basada en Ontologías*. Mientras el cliente conseguía reunir datos de las diferentes instituciones, el encargado de llevar a cabo el proyecto comenzó a estudiar las diferentes opciones. Esta parte del proyecto se consideraba muy importante, ya que el Proyecto de Fin de Carrera se consideraba un proyecto que estaba formado por una parte de investigación y por una parte experimental. La parte de investigación consistía en, dado el problema inicial, analizar diferentes soluciones al problema, y argumentar cuál es la mejor para nuestro caso. Obviamente, aún seleccionando una posible solución para nuestro problema, es bastante probable que para nuestro caso concreto debamos de utilizar una arquitectura nueva.

Para realiza el análisis de estas tres posibles soluciones, primero se realizará una descripción teórica de cada una de ellas, con el fin de introducir al lector en la arquitectura a tratar. Seguidamente se explicará el diseño de dicha arquitectura.

Por último, se detallará cómo sería cada solución aplicada a nuestro caso particular, aportando ventajas e inconvenientes.

#### **5.4.1 BASE DE DATOS CENTRALIZADA**

##### **5.4.1.1 DESCRIPCIÓN**

Fuente: <http://www.monografias.com/trabajos35/comparativa-bases-datos/comparativa-bases-datos.shtml> (“Consultado el 17-X-2009”)

*“Un sistema de base de datos centralizado es aquel que se ejecuta en un único sistema computacional sin tener, para tal efecto, que interactuar con otros computadores. El rango de estos sistemas comprende desde los sistemas de bases de datos mono usuario ejecutándose en computadores personales hasta los sistemas de bases de datos ejecutándose en sistemas de alto rendimiento.*

*Normalmente los sistemas de base de datos mono usuarios no suelen proporcionar muchas de las facilidades que ofrecen los sistemas multiusuario, en particular no tienen control de concurrencia y tienen precarios o inexistentes sistemas de recuperación.*

*Dado que las máquinas en las cuales se utilizan los sistemas mono usuarios son comúnmente computadores de propósito general, la arquitectura de estas máquinas es siempre parecida (de 1 a 2 procesadores que comparten la memoria principal) por tanto los sistemas de base de datos que se ejecutan sobre estas máquinas no intentan dividir una consulta simple entre los distintos procesadores, sino que ejecutan cada consulta en un único procesador posibilitando así la concurrencia de varias consultas. Este tipo de sistemas dan la sensación de una mayor productividad (puesto que pueden ejecutar un mayor número de transacciones por segundo) a pesar de que cada transacción individualmente no se ejecute más rápido. Por el contrario las máquinas paralelas tienen un gran número de procesadores y los sistemas de base de datos que ahí se ejecutan siempre tenderán a paralelizar las tareas simples (consultas) que solicitan los usuarios.”*

##### **5.4.1.2 DISEÑO DE BASE DE DATOS CENTRALIZADA**

Unas de las primeras opciones que se tuvieron en cuenta desde el principio fue la de realizar una informatización conjunta de todos los datos disponibles en las diferentes organizaciones. La Arquitectura de este modelo sería una arquitectura de tres capas, en la cual una capa estaría dedicada al almacenamiento de los datos.

Fuente: [http://es.wikipedia.org/wiki/Arquitectura\\_de\\_tres\\_niveles](http://es.wikipedia.org/wiki/Arquitectura_de_tres_niveles) (“Consultado el 17-X-2009”)

*“La programación por capas es un estilo de programación en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario.*

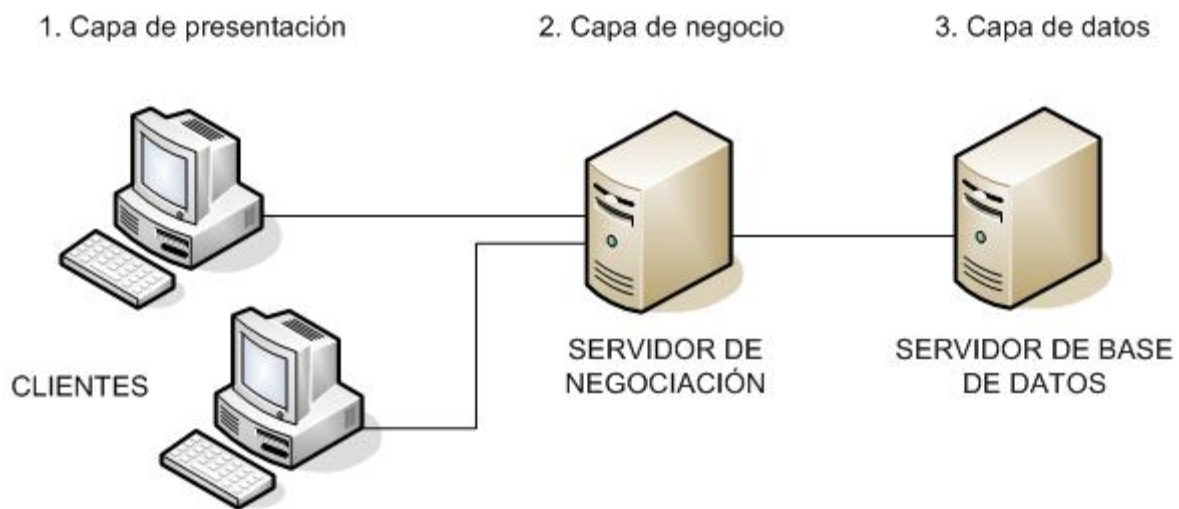


Figura 10. Ejemplo de arquitectura en capas

*La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. Un buen ejemplo de este método de programación sería el modelo de interconexión de sistemas abiertos.*

*Además, permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API que existe entre niveles.*

*En el diseño de sistemas informáticos actual se suele usar las arquitecturas multinivel o Programación por capas. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten).*

*El diseño más en boga actualmente es el diseño en tres niveles (o en tres capas).*

#### ▪ **Capas y niveles**

- ✓ **Capa de presentación:** es la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de

proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario.

- ✓ **Capa de negocio:** es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.
- ✓ **Capa de datos:** es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Todas estas capas pueden residir en un único ordenador, si bien lo más usual es que haya una multitud de ordenadores en donde reside la capa de presentación (son los clientes de la arquitectura cliente/servidor). Las capas de negocio y de datos pueden residir en el mismo ordenador, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más ordenadores. Así, si el tamaño o complejidad de la base de datos aumenta, se puede separar en varios ordenadores los cuales recibirán las peticiones del ordenador en que reside la capa de negocio.

Si, por el contrario, fuese la complejidad en la capa de negocio lo que obligase a la separación, esta capa de negocio podría residir en uno o más ordenadores que realizarían solicitudes a una única base de datos. En sistemas muy complejos se llega a tener una serie de ordenadores sobre los cuales corre la capa de datos, y otra serie de ordenadores sobre los cuales corre la base de datos.

En una arquitectura de tres niveles, los términos "capas" y "niveles" no significan lo mismo ni son similares.

El término "capa" hace referencia a la forma como una solución es segmentada desde el punto de vista lógico:

*Presentación/ Lógica de Negocio/ Datos.*

En cambio, el término "nivel" corresponde a la forma en que las capas lógicas se encuentran distribuidas de forma física. Por ejemplo:



- Una solución de tres capas (presentación, lógica, datos) que residen en un solo ordenador (Presentación + lógica + datos). Se dice que la arquitectura de la solución es de tres capas y un nivel.
- Una solución de tres capas (presentación, lógica, datos) que residen en dos ordenadores (presentación + lógica, lógica + datos). Se dice que la arquitectura de la solución es de tres capas y dos niveles.
- Una solución de tres capas (presentación, lógica, datos) que residen en tres ordenadores (presentación, lógica, datos). La arquitectura que la define es: solución de tres capas y tres niveles.”

### 5.4.1.3 BASE DE DATOS CENTRALIZADA APLICADA A NUESTRO PROYECTO

Una vez que se ha introducido al lector en el tema de las bases de datos centralizadas, vamos a conocer cómo se aplicaría el diseño de la misma a nuestro proyecto. Para ello, se llevará a cabo una posible arquitectura de la solución. Además se analizará dicha arquitectura argumentando ventajas e inconvenientes.

Una solución basada en una base de datos centralizada mantendría, tal y como aparece en la arquitectura explicada en el punto anterior, una estructura de tres capas (Capa de presentación, capa de negocio y capa de datos). A continuación aparece un esquema global de la arquitectura de la posible solución. Más adelante se explicará cada una de las capas, tanto su contenido como su funcionamiento.

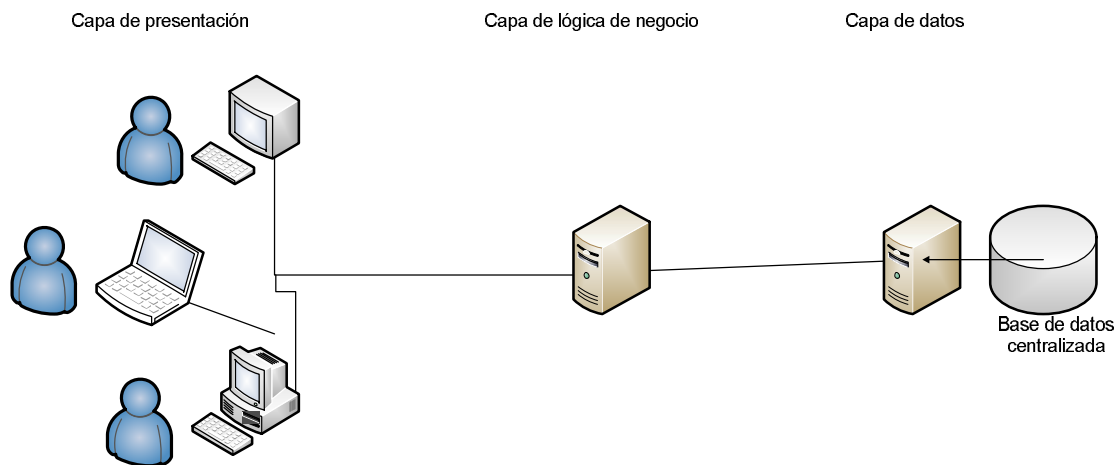


Figura 11. Posible solución basada en una base de datos centralizada.

En la figura superior se puede apreciar bien las tres capas en las que se divide la arquitectura. Antes de comenzar con la explicación de cada una de ellas se va a explicar el funcionamiento general de la aplicación, para poder posteriormente aplicarla en esta arquitectura.

La aplicación consta de un buscador, en el cual el usuario, indicando una serie de restricciones a determinadas características, requiere información al sistema sobre el patrimonio de Álava. Cuando el usuario se conecta al sistema, lo primero que le aparece es el buscador de patrimonio. No se trata de un buscador normal, si no de un programa de búsqueda orientada, es decir, es el sistema quien ayuda al usuario a encontrar lo que busca. Una vez que el usuario ha descrito las restricciones sobre los atributos que deseaba, se envía dicha información a la capa de lógica de negocio. En esta capa se procesan los datos recibidos y mediante la comunicación con la capa de datos, capa en donde se encuentran los datos y encargada de acceder a los mismos, se procesa la solución que es posteriormente ofrecida al usuario a través de la capa de presentación.

Para conocer cómo se estructuraría la aplicación en la arquitectura que se ha presentado, se llevará a cabo la descripción de cada una de las capas.

- ✚ **Capa de presentación.** La capa de presentación es la capa que ve el usuario cuando se conecta al sistema. En la aplicación a desarrollar, lo que se encontraría el usuario al conectarse al sistema, sería un buscador. En dicho buscador, el sistema le permitiría al usuario escribir una serie de restricciones a unas propiedades determinadas del patrimonio. Una vez que el usuario haya terminado de presentar las restricciones que ha creído oportunas, dicho usuario pulsaría el botón consultar. Cuando el usuario pulsa dicho botón, el sistema recoge las restricciones realizadas por el usuario, y envía dicha información a la capa de lógica de negocio.
- ✚ **Capa de la lógica de negocio.** En esta capa residen los programas y las reglas que la aplicación debe seguir para que funcione correctamente. Es la capa que se encarga del tratamiento de los datos. En función de los datos que recibe procesa unos u otros métodos para conseguir ofrecer al usuario aquello que ha requerido, y que ha sido enviado a la capa de la lógica de negocio a través de la capa de presentación. En nuestra aplicación, ésta sería la capa en donde residiría el peso de la misma. Aquí se encontrarían todos los métodos elaborados por el desarrollador del proyecto. Esta capa se comunica con la capa de datos, sobre la cual realiza consultas sobre datos que después utiliza para mostrar una solución al usuario a través de la Capa de presentación (con la que también se comunica).
- ✚ **Capa de datos.** Es la capa donde se encuentran los datos y se encarga de acceder a ellos. En nuestro proyecto, esta es la capa más importante, ya que requiere de una correcta clasificación de los datos. Se debe recordar que los datos disponibles están en organizaciones muy diferentes, y la información que contiene cada una de ellas es completamente distinta a la que contiene del resto. Para poder conseguir una correcta y eficiente clasificación de los datos se ha tenido que llevar a cabo un estudio minucioso de los datos que tiene disponibles, tanto en papel como en formato digital, cada una de las organizaciones dispuestas a colaborar en el proyecto. Esta capa es la encargada de comunicarse con la base de datos, y de devolver los datos que la capa de lógica de negocio ha

solicitado, para que estos puedan ser tratados, y devueltos en forma de solución al usuario.

### **Ventajas**

- ✓ Se trata de una arquitectura muy común, y con la que es posible que el encargado de llevar a cabo el proyecto haya trabajado previamente.
- ✓ Permite mantener la información en una misma base de datos, lo cual nos evitará de problemas como la redundancia.

### **Desventajas**

- ✓ Sería necesario conseguir el permiso de cada organización, para que nos facilitarán sus datos y poder utilizarlos en nuestro sistema, lo cual sin un coste económico, resulta bastante improbable.
- ✓ Un problema con la base de datos impediría el funcionamiento del sistema.
- ✓ Cada vez que una organización consiguiera nuevos datos sobre patrimonio existente o sobre nuevo patrimonio tendría que comunicárselo al encargado de administrar la base de datos para que este introduzca los nuevos datos en dicha base de datos.

## 5.4.2 BASE DE DATOS DISTRIBUIDA

### 5.4.2.1 DESCRIPCIÓN

- **Definición**

Fuente: <http://siul02.si.ehu.es/~jimena/ABD/fuentes/BDDistribuida.pdf>  
("Consultada el 18-X-2009")

*“BD que no está almacenada en una única posición física, sino que está repartida en distintos lugares geográficamente dispersos y conectados vía enlaces de comunicación, pudiendo los usuarios de cada nodo acceder a todos los datos.”*

- **Concepto general**

Fuente: <http://sacbeob.8m.com/tutoriales/bddistribuidas/index.htm> ("Consultada el 18-X-2009")

*“Una Base de Datos Distribuida es, una base de datos construida sobre una red computacional y no por el contrario en una máquina aislada. La información que constituye la base de datos esta almacenada en diferentes sitios en la red, y las aplicaciones que se ejecutan acceden a los datos en distintos sitios.*

*Una Base de Datos Distribuida es, por lo tanto, una colección de datos que pertenecen lógicamente a un solo sistema, pero se encuentra físicamente esparcido en varios "sitios" de la red. Un sistema de base de datos distribuidos se compone de un conjunto de sitios, conectados entre sí mediante algún tipo de red de comunicaciones, en el cual:*

- ❖ *Cada sitio es un sistema de base de datos en sí mismo, pero,*
- ❖ *Los sitios han convenido en trabajar juntos (si es necesario) con el fin de que un usuario de cualquier sitio pueda obtener acceso a los datos de cualquier punto de la red tal como si todos los datos estuvieran almacenados en el sitio propio del usuario.*

*En consecuencia, la llamada "base de datos distribuida" es, en realidad, una especie de objeto virtual, cuyas partes componentes se almacenan físicamente en varias bases de datos "reales" distintas ubicadas en diferentes sitios. De hecho, es la unión lógica de esas bases de datos. En otras palabras, cada sitio tiene sus propias bases de datos "reales" locales, sus propios usuarios locales, sus propios DBMS y programas para la administración de transacciones (incluyendo programas de bloqueo, bitácoras, recuperación, etc.), y su propio administrador local de comunicación de datos. Así pues, el sistema*

*de bases de datos distribuidas puede considerarse como una especie de sociedad entre los DBMS individuales locales de todos los sitios. Un nuevo componente de software en cada sitio (en el aspecto lógico, una extensión del DBMS local) realiza las funciones de sociedad necesarias; y es la combinación de este nuevo componente y el DBMS ya existente lo que constituye el llamado "sistema de administración de bases de datos distribuidas" (DDBMS, distributed database management system)."*

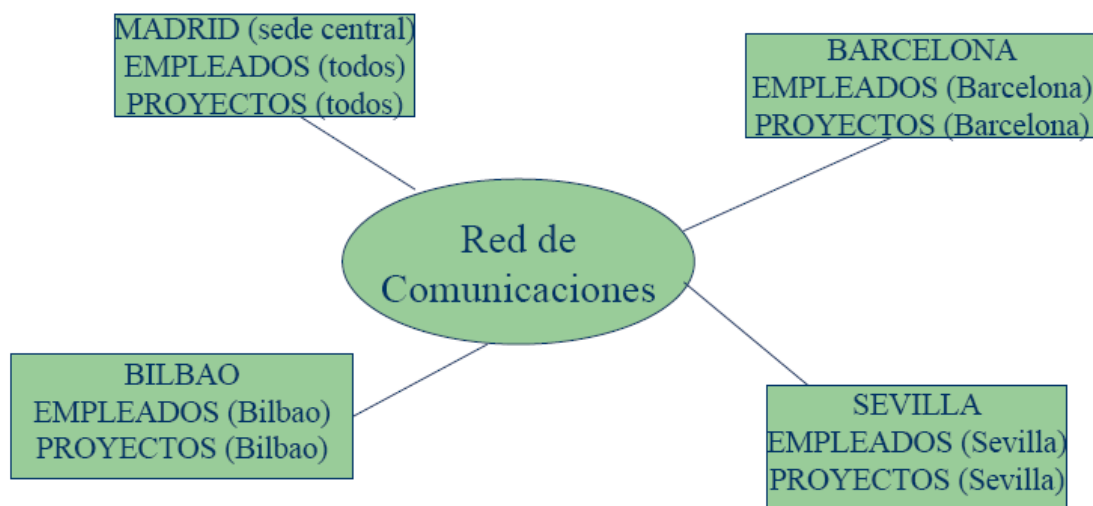


Figura 12. Ejemplo de una base de datos distribuida.

- **Principio fundamental de los sistemas de bases de datos distribuidos**

Fuente: <http://sacbeob.8m.com/tutoriales/bddistribuidas/cap1.html> (“Consultada el 18-X-2009”)

*“Desde el punto de vista del usuario, un sistema distribuido deberá ser idéntico un sistema no distribuido. En otras palabras, los usuarios de un sistema distribuido deberán comportarse exactamente como si el sistema no estuviera distribuido. Todos los problemas de los sistemas distribuidos son (o deberían ser) internos o a nivel de realización, no externos o a nivel del usuario. Llamaremos al principio fundamental recién identificado la "regla cero" de los sistemas distribuidos. La regla cero conduce a varios objetivos o reglas secundarios - doce en realidad- las siguientes:*

- ✓ ***Autonomía local.***

*Los sitios de un sistema distribuido deben ser autónomos. La autonomía local significa que todas las operaciones en un sitio dado se controlan en ese sitio; ningún sitio X deberá depender de algún otro sitio Y para su buen funcionamiento (pues de otra manera el sitio X podría ser*

*incapaz de trabajar, aunque no tenga en sí problema alguno, si cae el sitio Y, situación a todas luces indeseable). La autonomía local implica también un propietario y una administración locales de los datos, con responsabilidad local: todos los datos pertenecen " en realidad" a una base de datos local, aunque sean accesibles desde algún sitio remoto. Por tanto, las cuestiones de seguridad, integridad y representación en almacenamiento de los datos locales permanecen bajo el control de la instalación local.*

✓ **No dependencia de un sitio central.**

*La autonomía local implica que todos los sitios deben tratarse igual; no debe haber dependencia de un sitio central "maestro" para obtener un servicio central, como por ejemplo un procesamiento centralizado de las consultas o una administración centralizada de las transacciones, de modo que todo el sistema dependa de ese sitio central. Este segundo objetivo es por tanto un corolario del primero (si se logra el primero, se logrará pro fuerza el segundo). Pero la "no dependencia de un sitio central" es deseable por sí misma, aun si no se logra la autonomía local completa. Por ello vale la pena expresarlo como un objetivo separado.*

*La dependencia de un sitio central sería indeseable al menos por las siguientes razones: en primer lugar, ese sitio central podría ser un cuello de botella; en segundo lugar, el sistema sería vulnerable; si el sitio central sufriera un desperfecto, todo el sistema dejaría de funcionar.*

✓ **Operación continua.**

*En un sistema distribuido, lo mismo que en uno no distribuido, idealmente nunca debería haber necesidad de apagar a propósito el sistema. Es decir, el sistema nunca debería necesitar apagarse para que se pueda realizar alguna función, como añadirse un nuevo sitio o instalar una versión mejorada del DBMS en un sitio ya existente.*

✓ **Independencia con respecto a la localización.**

*La idea básica de la independencia con respecto a la localización (también conocida como transparencia de localización) es simple : no debe ser necesario que los usuarios sepan dónde están almacenados físicamente los datos, sino que más bien deben poder comportarse - al menos desde un punto de vista lógico - como si todos los datos estuvieran almacenados en su propio sitio local. La independencia con respecto a la localización es deseable porque simplifica los programas de los usuarios y sus actividades en la terminal. En particular, hace posible la migración de datos de un sitio a otro sin anular la validez de ninguno de esos*

*programas o actividades. Esta posibilidad de migración es deseable pues permite modificar la distribución de los datos dentro de la red en respuesta a cambios en los requerimientos de desempeño.*

✓ ***Independencia con respecto a la fragmentación.***

*Un sistema maneja fragmentación de los datos si es posible dividir una relación en partes o "fragmentos" para propósitos de almacenamiento físico. La fragmentación es deseable por razones de desempeño: los datos pueden almacenarse en la localidad donde se utilizan con mayor frecuencia, de manera que la mayor parte de las operaciones sean sólo locales y se reduzca al tráfico en la red. Por ejemplo, la relación empleados EMP podría fragmentarse de manera que los registros de los empleados de Nueva York se almacenen en el sitio de Nueva York, en tanto que los registros de los empleados de Londres se almacenan en el sitio de Londres.*

*Existen en esencia dos clases de fragmentación, horizontal y vertical, correspondientes a las operaciones relacionales de restricción y proyección; respectivamente. En términos más generales, un fragmento puede ser cualquier subrelación arbitraria que pueda derivarse de la relación original mediante operaciones de restricción y proyección (excepto que, en el caso de la proyección es obvio que las proyecciones deben conservar la clave primaria de la relación original). La reconstrucción de la relación original a partir de los fragmentos se hace mediante operaciones de reunión y unión apropiadas (reunión en el caso de fragmentación vertical, y la unión en casos de fragmentación horizontal).*

*Ahora llegamos a un punto principal: un sistema que maneja la fragmentación de los datos deberá ofrecer también una independencia con respecto a la fragmentación (llamada también transparencia de fragmentación). La independencia con respecto a la fragmentación (al igual que la independencia con respecto a la independencia con respecto a la localización) es deseable porque simplifica los programas de los usuarios y sus actividades en la terminal.*

✓ ***Independencia de réplica.***

*Un sistema maneja réplica de datos si una relación dada (ó en términos más generales, un fragmento dado en una relación) se puede representar en el nivel físico mediante varias copias almacenadas o réplicas, en muchos sitios distintos.*

*La réplica es deseable al menos por dos razones: en primer lugar, puede producir un mejor desempeño (las aplicaciones pueden*

*operar sobre copias locales en vez de tener que comunicarse con sitios remotos); en segundo lugar, también puede significar una mejor disponibilidad (un objeto estará disponible para su procesamiento en tanto esté disponible por lo menos una copia, al menos para propósitos de recuperación). La desventaja principal de las réplicas es desde luego que cuando se pone al día un cierto objeto copiado, deben ponerse al día todas las réplicas de ese objeto: el problema de la propagación de actualizaciones.*

*La réplica como la fragmentación, debe ser "transparente para el usuario". En otras palabras, un sistema que maneja la réplica de los datos deberá ofrecer también una independencia de réplica (conocida también como transparencia de réplica); es decir, los usuarios deberán poder comportarse como si sólo existiera una copia de los datos. La independencia de réplica es buena porque simplifica los programas de los usuarios y sus actividades en la terminal. En particular, permite la creación y eliminación dinámicas de las réplicas en cualquier momento en respuesta a cambios en los requerimientos, sin anular la validez de esos programas o actividades de los usuarios.*

✓ **Procesamiento distribuido de consultas.**

*En este aspecto debemos mencionar dos puntos amplios.*

*Primero consideremos la consulta "obtener los proveedores de partes rojas en Londres". Supongamos que el usuario está en la instalación de Nueva York y los datos están en el sitio de Londres. Supongamos también que son  $n$  los registros proveedor que satisfacen solicitud. Si sistema es relacional, consulta implicará en esencia dos mensajes: uno transmitir la solicitud Nueva York a Londres, y otro para devolver el conjunto resultante de  $n$  registros de Londres a Nueva York. Si, por otro lado, el sistema no es relacional, sino de un registro a la vez, la consulta implicará en esencia  $2n$  mensajes:  $n$  de Nueva York a Londres solicitando el siguiente registro, y  $n$  de Londres a Nueva York para devolver ese siguiente registro. Así, el ejemplo ilustra el punto de que un sistema relacional tendrá con toda probabilidad un mejor desempeño que uno no relacional (para cualquier consulta que solicite varios registros), quizá en varios órdenes de magnitud.*

*En segundo lugar, la optimización es todavía más importante en un sistema distribuido que en uno centralizado. Lo esencial es que, en una consulta como la anterior, donde están implicados varios sitios, habrá muchas maneras de trasladar los datos en la red para satisfacer la solicitud, y es crucial encontrar una estrategia suficiente. Por ejemplo, una solicitud de unión de una relación  $R_x$  almacenada en el sitio  $X$  y una*



*relación Ry almacenada en el sitio Y podría llevarse a cabo trasladando Rx a Y o trasladando Ry a X, o trasladando las dos a un tercer sitio Z.*

✓ ***Manejo distribuido de transacciones.***

*El manejo de transacciones tiene dos aspectos principales, el control de recuperación y el control de concurrencia, cada uno de los cuales requiere un tratamiento más amplio en el ambiente distribuido. Para explicar ese tratamiento más amplio es preciso introducir primero un término nuevo, "agente". En un sistema distribuido, una sola transacción puede implicar la ejecución de código en varios sitios (en particular puede implicar a actualizaciones en varios sitios). Por tanto, se dice que cada transacción está compuesta de varios agentes, donde un agente es el proceso ejecutado en nombre de una transacción dada en determinado sitio. Y el sistema necesita saber cuándo dos agentes son parte de la misma transacción; por ejemplo, es obvio que no puede permitirse un bloqueo mutuo entre dos agentes que sean parte de la misma transacción. La cuestión específica del control de recuperación, para asegurar, pues que una transacción dada sea atómica (todo o nada) en el ambiente distribuido, el sistema debe asegurarse de que todos los agentes correspondientes a esa transacción se comprometan al unísono o bien que retrocedan al unísono. Este efecto puede lograrse mediante el protocolo de compromiso en dos fases.*

*En cuanto al control de concurrencia, esta función en un ambiente distribuido estará basada con toda seguridad en el bloqueo, como sucede en los sistemas no distribuidos.*

✓ ***Independencia con respecto al equipo.***

*En realidad, no hay mucho que decir acerca de este tema, el título lo dice todo. Las instalaciones de cómputo en el mundo real por lo regular incluyen varias máquinas diferentes -máquinas IBM, DEC, HP, UNISYS, PC etc.- y existe una verdadera necesidad de poder integrar los datos en todos esos sistemas y presentar al usuario "una sola imagen del sistema". Por tanto conviene ejecutar el mismo DBMS en diferentes equipos, y además lograr que esos diferentes equipos participen como socios iguales en un sistema distribuido.*

✓ ***Independencia con respecto al sistema operativo.***

*Este objetivo es un corolario del anterior. Es obvia la conveniencia no sólo de poder ejecutar el mismo DBMS en diferentes equipos, sino también poder ejecutarlo en diferentes sistemas operativos*

*y lograr que una versión MVS y una UNIX y una PC/DOS participen todas en el mismo sistema distribuido.*

✓ ***Independencia con respecto a la red.***

*Si el sistema ha de poder manejar múltiples sitios diferentes, con equipo distinto y diferentes sistemas operativos, resulta obvia la conveniencia de poder manejar también varias redes de comunicación distintas.*

✓ ***Independencia con respecto al DBMS.***

*Bajo este título consideramos las implicaciones de relajar la suposición de homogeneidad estricta. Puede alegarse que esa suposición es quizá demasiado rígida. En realidad, no se requiere sino que los DBMS en los diferentes sitios manejen todos la misma interfaz; no necesitan ser por fuerza copias del mismo sistema.*

*Estas doce reglas no son todas independientes entre sí, ni son por fuerza exhaustivas, ni tienen todas la misma importancia (diferentes usuarios darán diferentes grados de importancia a diferentes reglas en diferentes ambientes). Sin embargo, sí son útiles como fundamento para entender la tecnología distribuida y como marco de referencia para caracterizar la funcionalidad de sistemas distribuidos específicos.*

*Un último punto introductorio: es importante distinguir los sistemas distribuidos de bases de datos verdaderos, generalizados, de los sistemas que tan solo ofrecen algún tipo de acceso remoto a los datos (llamados a veces sistemas de procesamiento distribuido o sistemas de red). En un " sistema de acceso remoto a los datos ", el usuario podría ser capaz de trabajar con datos de un sitio remoto, o aun con datos de varios sitios remotos al mismo tiempo, pero " se notan las costuras"; el usuario definitivamente está consciente (en mayor o menor grado) de que los datos son remotos, y debe comportarse de manera acorde. En cambio, en un sistema distribuido verdadero, las costuras son invisibles."*

#### **5.4.2.2 DISEÑO DE BASES DE DATOS DISTRIBUIDAS**

Fuente: Universidad de Castilla la Mancha- Arquitectura de Sistemas de Bases de Datos ("Consultada el 18-X-2009")

*"Consideremos una relación  $r$  que debe guardarse en la base de datos. Hay varios enfoques del almacenamiento de esa relación en la base de datos distribuida:*

- **Réplica.** El sistema conserva varias réplicas (copias) idénticas de la relación. Cada réplica se guarda en un emplazamiento diferente, lo que da lugar a la réplica de los datos. La alternativa a la réplica es guardar sólo una copia de la relación  $r$ .

- **Fragmentación.** La relación se divide en varios fragmentos. Cada fragmento se guarda en un emplazamiento diferente.

- **Réplica y fragmentación.** La relación se divide en varios fragmentos. El sistema conserva réplicas de cada fragmento.

Se va a realizar una descripción de cada uno de los enfoques:

#### ❖ **Réplica de los datos.**

Si se replica la relación  $r$ , se guardará una copia de la misma en dos o más emplazamientos. En el caso más extremo, se tendrá una réplica completa, en la que se guarda una copia en cada emplazamiento del sistema.

La réplica presenta un cierto número de ventajas e inconvenientes:

- **Disponibilidad (Ventaja).** Si falla uno de los emplazamientos que contienen la relación  $r$ , ésta aún se podrá encontrar en otro emplazamiento. Por tanto, el sistema puede seguir procesando las consultas que impliquen a  $r$ , a pesar del fallo en un emplazamiento.

- **Aumento del paralelismo (Ventaja).** En el caso de que la mayor parte de los accesos a la relación  $r$  sea de lectura de la misma, varios emplazamientos pueden procesar en paralelo las consultas que impliquen a  $r$ . Cuantas más réplicas de  $r$  haya, mayor será la posibilidad de que el dato buscado se halle en el emplazamiento en que se esté ejecutando la transacción. Por tanto, la réplica de los datos minimiza el tráfico de datos entre los emplazamientos.

- **Aumento de la sobrecarga en las actualizaciones (Inconveniente).** El sistema debe asegurarse de que todas las réplicas de la relación  $r$  sean consistentes; en caso contrario, puede dar lugar a resultados erróneos. Por tanto, siempre que se actualice  $r$ , la actualización debe extenderse a todos los emplazamientos que contengan réplicas. La consecuencia es un aumento en la sobrecarga. Por ejemplo, en un sistema bancario en el que la información sobre las cuentas se replica en varios emplazamientos, hay que asegurarse de que el saldo de una cuenta concreta coincida en todos los emplazamientos.

En general, la réplica mejora el rendimiento de las operaciones de lectura y aumenta la disponibilidad de los datos para las transacciones de sólo lectura. Sin embargo, las transacciones de actualización suponen una sobrecarga mayor. Por tanto, un buen parámetro para afrontar el grado de réplica consistiría en sopesar la cantidad de consultas de lectura que se efectuarán, así como el número de consultas de escritura

que se llevarán a cabo. En una red donde las consultas que se procesen sean mayoritariamente de lectura, se podría alcanzar un alto grado de réplica, no así en el caso contrario. El control de las actualizaciones concurrentes de los datos replicados por varias transacciones es más complicado que cuando se utiliza el enfoque centralizado del control de concurrencia. Se puede simplificar la administración de las réplicas de la relación  $r$  escogiendo una de ellas como copia principal de  $r$ . Por ejemplo, en un sistema bancario se puede asociar una cuanta con el emplazamiento en que se ha abierto. De manera parecida, en un sistema de reservas de unas líneas aéreas se puede asociar un vuelo con el punto en el que se origina.

#### ❖ **Fragmentación de los datos.**

Si la relación  $r$  está fragmentada, se dividirá en cierto número de fragmentos  $r_1, r_2, \dots, r_n$ . Estos fragmentos contendrán suficiente información como para permitir la reconstrucción de la relación original  $r$ .

Dado que una relación se corresponde esencialmente con una tabla y la cuestión consiste en dividirla en fragmentos menores, inmediatamente surgen dos alternativas lógicas para llevar a cabo el proceso: la división horizontal y la división vertical.

La división o fragmentación horizontal trabaja sobre las tuplas, dividiendo la relación en subrelaciones que contienen un subconjunto de las tuplas que alberga la primera. La fragmentación vertical, en cambio, se basa en los atributos de la relación para efectuar la división. Estos dos tipos de partición podrían considerarse los fundamentales y básicos. Sin embargo, existen otras alternativas.

Fundamentalmente, se habla de fragmentación mixta o híbrida cuando el proceso de partición hace uso de los dos tipos anteriores. La fragmentación mixta puede llevarse a cabo de tres formas diferentes: desarrollando primero la fragmentación vertical y, posteriormente, aplicando la partición horizontal sobre los fragmentos verticales (denominada partición VH), o aplicando primero una división horizontal para luego, sobre los fragmentos generados, desarrollar una fragmentación vertical (llamada partición HV), o bien, de forma directa considerando la semántica de las transacciones. Otro enfoque distinto y relativamente nuevo, consiste en aplicar sobre una relación, de forma simultánea y no secuencial, la fragmentación horizontal y la fragmentación vertical; en este caso, se generara una rejilla y los fragmentos formaran las celdas de esa rejilla, cada celda será exactamente un fragmento vertical y un fragmento horizontal (nótese que en este caso el grado de fragmentación alcanzado es máximo, y no por ello la descomposición resultará más eficiente).

Grado de fragmentación. Cuando se va a fragmentar una base de datos deberíamos sopesar qué grado de fragmentación va a alcanzar, ya que éste será un factor que influirá notablemente en el desarrollo de la ejecución de las consultas. El grado de fragmentación puede variar desde una ausencia de la división, considerando a las relaciones unidades de fragmentación; o bien, fragmentar a un grado en el cada tupla o atributo forme un fragmento. Ante estos dos casos extremos, evidentemente se ha de buscar un compromiso intermedio, el cual debería establecerse sobre las

*características de las aplicaciones que hacen uso de la base de datos. Dichas características se podrán formalizar en una serie de parámetros. De acuerdo con sus valores, se podrá establecer el grado de fragmentación del banco de datos.*

#### *Reglas de corrección de la fragmentación.*

*A continuación se enuncian las tres reglas que se han de cumplir durante el proceso de fragmentación, las cuales asegurarán la ausencia de cambios semánticos en la base de datos durante el proceso.*

- 1. **Plenitud.** Si una relación  $R$  se descompone en una serie de fragmentos  $R_1, R_2, \dots, R_n$ , cada elemento de datos que pueda encontrarse en  $R$  deberá poder encontrarse en uno o varios fragmentos  $R_i$ . Esta propiedad extremadamente importante asegura que los datos de la relación global se proyectan sobre los fragmentos sin pérdida alguna. Tenga en cuenta que en el caso horizontal el elemento de datos, normalmente, es una tupla, mientras que en el caso vertical es un atributo.*
- 2. **Reconstrucción.** Debe ser posible definir una operación relacional que permita reconstruir la relación  $R$  a partir de los fragmentos. Esta regla asegura que se preserven las dependencias funcionales.*
- 3. **Disyunción.** Si una relación  $R$  se descompone horizontalmente en una serie de fragmentos  $R_1, R_2, \dots, R_n$ , y un elemento de datos  $d_i$  se encuentra en algún fragmento  $R_j$ , entonces no se encuentra en otro fragmento  $R_k$  ( $k <> j$ ). Esta regla asegura que los fragmentos horizontales sean disjuntos. Si una relación  $R$  se descompone verticalmente, sus atributos primarios clave normalmente se repiten en todos sus fragmentos.*

*La fragmentación vertical es una excepción a esta regla, donde las claves primarias deben repetirse para permitir la reconstrucción. Esta regla asegura la mínima redundancia de los datos.*

✓ **Fragmentación horizontal**

Tomemos como ejemplo la siguiente relación cuenta:

Nombre-sucursal	Número-cuenta	Saldo
Guadarrama	C-305	100.000
Guadarrama	C-226	64.200
Cercedilla	C-177	41.000
Cercedilla	C-402	2.000.000
Guadarrama	C-155	12.400
Cercedilla	C-408	224.600
Cercedilla	C-639	150.000

Tabla 2. Fragmentación horizontal. Paso 1.

La relación  $r$  se divide en cierto número de subconjuntos  $r_1, r_2, \dots, r_n$ . Cada tupla de la relación  $r$  debe pertenecer al menos a uno de los fragmentos, de modo que se pueda reconstruir la relación original si fuera necesario.

Un fragmento puede definirse como una selección de la relación global  $r$ . Es decir, se utiliza un predicado  $P_i$  para construir un fragmento  $r_i$  de la manera siguiente:

$$r_i = \sigma_{P_i}(r)$$

Se puede obtener la reconstrucción de la relación  $r$  tomando la unión de todos los fragmentos, es decir:

$$r = r_1 \cup r_2 \cup \dots \cup r_n$$

La relación cuenta se puede dividir en  $n$  fragmentos diferentes, cada uno de los cuales consiste en tuplas de cuentas que pertenecen a una sucursal concreta. Si el sistema bancario sólo tiene dos sucursales (Guadarrama y Cercedilla), entonces sólo habrá dos fragmentos diferentes:

$$\text{cuenta1} = \sigma_{\text{nombre-sucursal}=\text{"Guadarrama"}}(\text{cuenta})$$

$$\text{cuenta2} = \sigma_{\text{nombre-sucursal}=\text{"Cercedilla"}}(\text{cuenta})$$

Los dos fragmentos se muestran a continuación:

*Cuenta1*

Nombre-sucursal	Número-cuenta	Saldo
Guadarrama	C-305	100.000
Guadarrama	C-226	64.200
Guadarrama	C-155	12.400

Tabla 3.Fragmentación horizontal. Paso 2 (Nombre-sucursal=Guadarrama).

*Cuenta2*

Nombre-sucursal	Número-cuenta	Saldo
Cercedilla	C-177	41.000
Cercedilla	C-402	2.000.000
Cercedilla	C-408	224.600
Cercedilla	C-639	150.000

Tabla 4.Fragmentación horizontal. Paso 2 (Nombre-sucursal=Cercedilla).

*En este ejemplo los fragmentos son disjuntos. Cambiando los predicados de selección utilizados para generar los fragmentos, se puede hacer que una tupla de  $r$  determinada aparezca en más de una de la  $r_i$ . Esta forma de réplica de los datos se discute con más profundidad más adelante.*

✓ **Fragmentación vertical**

*La fragmentación vertical de  $r(R)$  implica la definición de varios subconjuntos de atributos  $R_1, R_2, \dots, R_n$  del esquema  $R$  tales que*

$$R = R_1 \cup R_2 \cup \dots \cup R_n$$

*Cada fragmento  $r_i$  de  $r$  queda definido por*

$$r_i = \Pi_{R_i}(r)$$

*La fragmentación debe hacerse de modo que se pueda reconstruir la relación  $r$  a partir de los fragmentos tomando la reunión natural:*

$$r = r_1 \bowtie r_2 \bowtie r_3 \bowtie \dots \bowtie r_n$$

*Un modo de asegurar que la relación  $r$  pueda reconstruirse es incluir los atributos de la clave primaria de  $R$  en cada uno de los  $R_i$ . De manera más general, puede utilizarse cualquier superclave. A menudo resulta conveniente añadir un atributo especial, denominado *id-tupla*, al esquema  $R$ . El valor de *id-tupla* de una tupla es único, y se utiliza para distinguir esa tupla de las demás. El atributo *id-tupla* es, por tanto, un candidato a clave del esquema ampliado, y se incluye en cada uno de los  $R_i$ .*

La dirección física o lógica de una tupla puede utilizarse como id-tupla, dado que cada tupla tiene una dirección única.

Para ilustrar la fragmentación vertical se considerará relación depósito:

Nombre-sucursal	Número-cuenta	Nombre-cliente	Saldo	Id-tupla
Guadarrama	C-305	García	100.000	1
Guadarrama	C-226	Cordero	64.200	2
Cercedilla	C-177	Cordero	41.000	3
Cercedilla	C-402	Obeso	2.000.000	4
Guadarrama	C-155	Obeso	12.400	5
Cercedilla	C-408	Obeso	224.600	6
Cercedilla	C-639	Badorrey	150.000	7

Tabla 5.Fragmentación vertical. Paso 1.

A continuación se muestra una fragmentación vertical para la relación depósito:

*Esquema-depósito-1 = (nombre-sucursal, nombre-cliente, id-tupla)*

*Esquema-depósito-2 = (número-cuenta, saldo, id-tupla)*

*depósito1*

Nombre-sucursal	Nombre-cliente	Id-tupla
Guadarrama	García	1
Guadarrama	Cordero	2
Cercedilla	Cordero	3
Cercedilla	Obeso	4
Guadarrama	Obeso	5
Cercedilla	Obeso	6
Cercedilla	Badorrey	7

Tabla 6.Fragmentación vertical. Paso 2. Nombre-sucursal, Nombre-cliente, Id-tupla

*depósito2*

Número-cuenta	Saldo	Id-tupla
C-305	100.000	1
C-226	64.200	2
C-177	41.000	3
C-402	2.000.000	4
C-155	12.400	5
C-408	224.600	6
C-639	150.000	7

Tabla 7.Fragmentación vertical. Paso 2. Número-cuenta, Saldo, Id-tupla



Estas dos relaciones resultan de procesar:

$$\text{depósito1} = \sigma_{\text{Esquema-depósito-1}}(\text{depósito})$$

$$\text{depósito2} = \sigma_{\text{Esquema-depósito-2}}(\text{depósito})$$

Para reconstruir la relación depósito original a partir de los fragmentos hay que procesar:

$$\sigma_{\text{Esquema-depósito}}(\text{depósito1 depósito2})$$

La expresión depósito1 >< depósito2 es una forma especial de reunión natural. El atributo de reunión es id-tupla. Aunque el atributo id-tupla facilite la aplicación de la división vertical, no debe ser visible para los usuarios, dado que es un mecanismo interno de la aplicación y viola la independencia de los datos, que es una de las virtudes principales del modelo relacional.

#### ✓ **Fragmentación mixta**

La relación  $r$  se divide en una serie de relaciones fragmentarias  $r_1, r_2, \dots, r_n$ . Cada fragmento se obtiene como resultado de la aplicación del esquema de fragmentación horizontal o vertical a la relación  $r$ , o a un fragmento de  $r$  obtenido con anterioridad.

Supongamos la relación depósito vista anteriormente. Esta relación se divide en un principio en los fragmentos depósito1 y depósito2, tal y como se definieron anteriormente. Ahora se puede volver a dividir el fragmento depósito1 utilizando el esquema de fragmentación horizontal en los dos fragmentos siguientes:

$$\text{depósito1a} = \sigma_{\text{nombre-sucursal}=\text{"Guadarrama"}}(\text{depósito1})$$

$$\text{depósito1b} = \sigma_{\text{nombre-sucursal}=\text{"Cercedilla"}}(\text{depósito1})$$

Por tanto, la relación  $r$  queda dividida en tres fragmentos: depósito1a, depósito1b y depósito2. Cada uno de los fragmentos puede residir en un emplazamiento diferente.

#### ○ **Réplica y fragmentación de datos.**

Las técnicas descritas anteriormente para la réplica y fragmentación de datos pueden aplicarse de manera sucesiva a la misma relación. Es decir, se puede replicar un fragmento, las réplicas de los fragmentos se pueden volver a fragmentar, etc. Por ejemplo, considérese un sistema distribuido que consista en los emplazamientos  $E_1, E_2, \dots, E_{10}$ . Se puede fragmentar depósito en depósito1a, depósito1b y depósito2, y, por ejemplo, guardar una copia de depósito1a en los emplazamientos  $E_1, E_3$  y  $E_7$ ; una copia de depósito1b en los emplazamientos  $E_7$  y  $E_{10}$  y una copia de depósito2 en los emplazamientos  $E_2, E_8$  y  $E_9$ .

### 5.4.2.3 BASE DE DATOS DISTRIBUIDA APLICADA A NUESTRO PROYECTO

Antes de describir esta posible solución, es conveniente que repasemos en qué consiste el proyecto, y los problemas que debe solucionar el mismo. El objetivo principal de la aplicación a desarrollar consiste en reunir de forma lógica los datos situados en las diferentes organizaciones. Cada organización cuenta con sus datos, que están almacenados, bien, en una base de datos, bien, en una página Web estática, o bien, simplemente, en archivadores.

Teniendo en cuenta la teoría explicada anteriormente, y las características del proyecto, es obvio que se debe desarrollar un modelo de fragmentación horizontal de los datos, o en su defecto, un modelo que reúna una fragmentación horizontal de los datos y una réplica de los mismos.

La arquitectura general de esta solución puede ser perfectamente una arquitectura de tres niveles en donde la capa de datos está dividida en el acceso a varias bases de datos.

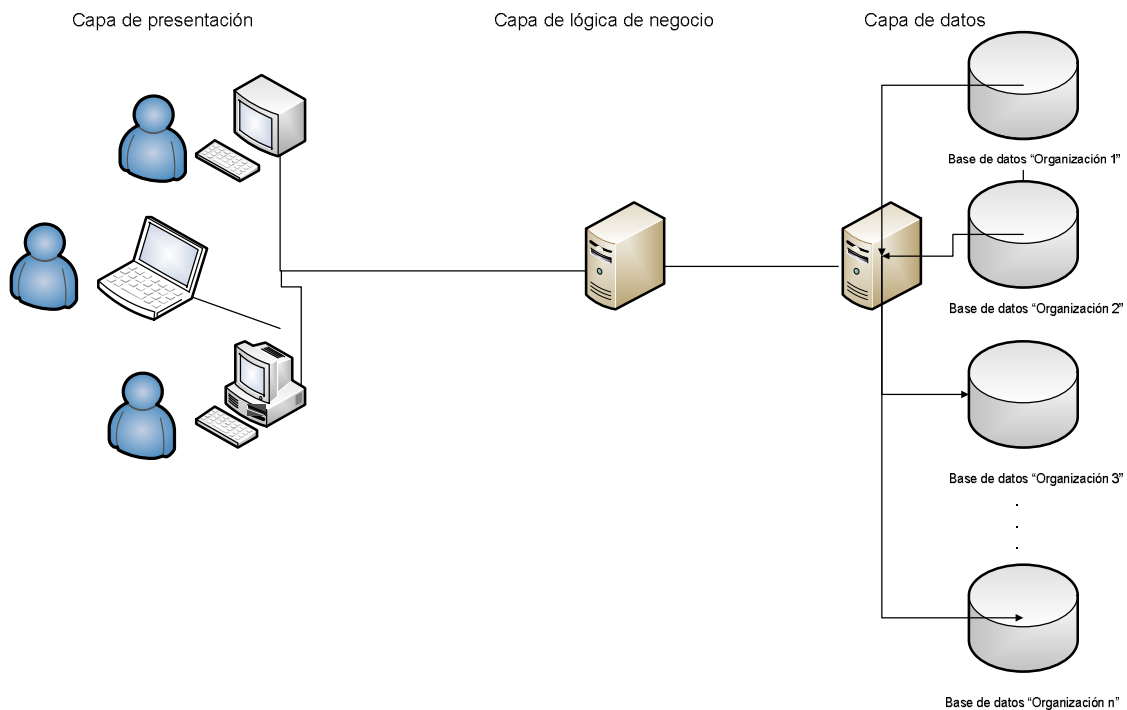



Figura 13.Arquitectura de tres capas con varias bases de datos en la capa de datos.

La arquitectura del sistema sería exactamente igual que en un sistema con una única base de datos centralizada, por lo que no se va a volver a explicar dicha arquitectura. Como se puede apreciar en la figura superior, la única diferencia entre las dos arquitecturas es la capa de datos. Para explicar este modelo se realizará una explicación que se basará únicamente en dicha capa de datos.

Únicamente para recordar las funciones que desempeña la capa de datos, realizaremos una pequeña descripción de la misma.

 **Capa de datos:** Lugar donde residen los datos. Es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

En el caso particular que estamos desarrollando, la capa de datos realizará las conexiones con cada base de datos, así como las consultas de datos pertinentes en cada momento.

Al igual que ocurría en el caso de que nuestro sistema albergara una base de datos centralizada, lo primero que debemos realizar es un minucioso análisis de los datos de cada una de las organizaciones, para poder realizar una óptima clasificación de los mismos. Se trata de una tarea difícil y de larga duración pero imprescindible. Una vez reunidos los datos y clasificados correctamente se lleva a cabo la fragmentación horizontal de los mismo. Para explicar dicha fragmentación, no ayudaremos de un ejemplo sobre lo que puede ser una posible solución de nuestro problema.

Tras elaborar una clasificación de los datos el resultado obtenido es una organización de los mismos como la siguiente.

Nombre	Tipo	Descripción	Organización
Iglesia de San Miguel	Iglesia	Iglesia situada en Vitoria-Gasteiz. Construida en el siglo XVI.	Diputación Foral de Álava
Catedral de Santa María	Catedral	Una de las catedrales de Vitoria-Gasteiz. Alberga un gran reclamo turístico.	Archivo Municipal.
Museo de Arqueología	Objetos	Posee objetos de diferentes épocas.	UPV/EHU
Santuario de la Encina	Excavaciones	Además de las excavaciones, los visitantes pueden contemplar un santuario del siglo XVII.	Diputación Foral de Álava
Puente Viejo	Elementos Menores	Situado un Armiñón (Álava)	Diputación Foral de Álava
Iglesia de San Pedro	Iglesia	Situada en Vitoria- Gasteiz.	UPV/EHU
Ermita de San Antonio	Ermita	Ermita del siglo XVI, con reformas durante los S.XVIII y XIX	Archivo Municipal

Tabla 8. Clasificación de datos del patrimonio histórico.

Antes de continuar se van a explicar los campos de la clasificación de la figura superior para entender mejor la explicación.

- **Nombre:** Indica el nombre del patrimonio.
- **Tipo:** Indica de qué tipo es el patrimonio (Excavación, iglesia, monasterio...)
- **Descripción:** Muestra una breve descripción del elemento patrimonial.
- **Organización:** Indica la organización que dispone de esos datos.

Si observamos los datos nos damos cuenta de que cada organización dispone de los suyos propios, y que no es necesario realizar una centralización de los mismos, si consiguiéramos acceder a los datos de cada una de las organizaciones de manera independiente. Obviamente el resultado de realizar una consulta en una base de datos centralizadas, y el de realizar n consultas en las diferentes bases de datos de cada organización debe ser el mismo.

En el ejemplo anterior, se puede dividir la clasificación de los datos en fragmentos según la organización a la que pertenezcan. Los fragmentos serían los siguientes:

Nombre	Tipo	Descripción	Organización
Iglesia de San Miguel	Iglesia	Iglesia situada en Vitoria-Gasteiz. Construida en el siglo XVI.	Diputación Foral de Álava
Santuario de la Encina	Excavaciones	Además de las excavaciones, los visitantes pueden contemplar un santuario del siglo XVII.	Diputación Foral de Álava
Puente Viejo	Elementos Menores	Situado un Armiñón (Álava)	Diputación Foral de Álava

Tabla 9. Fragmentación horizontal de los datos. Diputación Foral de Álava.

Nombre	Tipo	Descripción	Organización
Catedral de Santa María	Catedral	Una de las catedrales de Vitoria-Gasteiz. Alberga un gran reclamo turístico.	Archivo Municipal.
Ermita de San Antonio	Ermita	Ermita del siglo XVI, con reformas durante los S.XVIII y XIX	Archivo Municipal

Tabla 10. Fragmentación horizontal de los datos. Archivo Municipal.

Nombre	Tipo	Descripción	Organización
Museo de Arqueología	Objetos	Posee objetos de diferentes épocas.	UPV/EHU
Iglesia de San Pedro	Iglesia	Situada en Vitoria- Gasteiz.	UPV/EHU

Tabla 11. Fragmentación horizontal de los datos. UPV/EHU.

Cada uno de los fragmentos se encontraría situado en el servidor de la organización a la que corresponden los datos. Como se puede observar en las figuras superiores, la unión de todos los fragmentos es igual a la clasificación realizada en la Tabla 8.

$r = r_1 \cup r_2 \cup \dots \cup r_n$ , siendo  $r_1, r_2, \dots, r_n$  fragmentos en los que se divide la clasificación de los datos.

Cuando el usuario realiza una petición de datos sobre el patrimonio a través de la capa de presentación, la lógica de negocio procesa esa información solicitando a la capa de datos un conjunto de datos con los que poder construir una respuesta que presentar al usuario. La capa de datos es la encargada de elaborar esa petición de datos a cada una de las bases de datos del sistema, es decir, a cada base de datos correspondiente a cada organización. Para ello deberá abrir conexiones con cada una de ellas y posteriormente realizar la consulta apropiada para conseguir los datos necesarios y transmitirlos a la capa de la lógica de negocio.

### **Ventajas**

- ✓ Permite una autonomía local de los datos a cada una de las organizaciones.
- ✓ Consigue una transparencia total para el usuario.
- ✓ Mediante la replicación de información, las bases de datos distribuidas pueden presentar cierto grado de tolerancia a fallos haciendo que el funcionamiento del sistema no dependa de un solo lugar como en el caso de las bases de datos centralizadas.
- ✓ Posibilidad de compartir los datos al tiempo que se mantiene un cierto grado de control local.
- ✓ Una buena distribución de los datos, dará como resultado un mejor rendimiento.

### **Desventajas**

- ✓ Control y manejo de los datos. Dado que éstos residen en muchos nodos diferentes y se pueden consultar por nodos diversos de la red, la probabilidad de violaciones de seguridad es creciente si no se toman las precauciones debidas.
- ✓ Dado que los datos pueden estar replicados, el control de concurrencia y los mecanismos de recuperación son mucho más complejos que en un sistema centralizado.
- ✓ Al ser menos usuales que las bases de datos centralizadas su dificultad a la hora de implementarlas es mayor.

### **5.4.3 BASES DE DATOS DISTRIBUIDAS Y HETEROGÉNEAS**

#### **5.4.3.1 DESCRIPCIÓN**

Obviamente, tal y como su propio nombre indica las Bases de Datos Distribuidas y heterogéneas, son también Bases de datos distribuidas. Comparten características, ventajas e inconvenientes con las bases de datos distribuidas, sin embargo presentan un nuevo problema que no es otro que las bases de datos que integran el sistema son diferentes las unas a las otras.

A lo largo de estos últimos años se han desarrollado numerosas soluciones diferentes para afrontar el problema que supone integrar bases de datos diferentes dentro de un mismo sistema. De todas ellas, hubo una que, debido a su similitud en los contenidos con el proyecto que se va a desarrollar, nos resultó más interesante, y es la que a continuación se desarrolla. Se titula *Arquitectura del Sistema de Acceso Integrado a Bases de Datos Documentales basada en Árboles de Conceptos*.

#### **5.4.3.2 DISEÑO DE LA ARQUITECTURA DEL SISTEMA DE ACCESO INTEGRADO A BASES DE DATOS DOCUMENTALES BASADA EN ÁRBOLES DE CONCEPTOS.**

Durante el estudio de investigación llevado a cabo con el fin de encontrar la solución que se adecuara más al problema que se presenta en el Proyecto de Fin de Carrera una de las alternativas en la que se pensó fue una *Arquitectura del Sistema de Acceso Integrado a Bases de Datos Documentales basada en Árboles de Conceptos*.

De las posibles soluciones que se encontraron sobre integración de *bases de datos heterogéneas*, esta fue la que más llamó la atención al encargado de llevar a cabo el proyecto. Una de las principales razones por las que esta solución fue la que se decidió estudiar más profundamente fue porque las bases de datos, al igual que en el proyecto que se va a desarrollar, son documentales. Además, obviando su evidente grado de dificultad, la arquitectura parecía estar correctamente diseñada y bien argumentada en su conjunto, y lo más importante parecía que era factible su puesta a cabo.

Antes de explicar las ventajas e inconvenientes de esta arquitectura, y de entrar a valorar sobre la posibilidad real de llevarla a cabo, vamos a realizar una descripción detallada de la estructura de la arquitectura y de los elementos que la componen, para que el lector pueda comprender de manera más profunda las conclusiones obtenidas sobre la implantación de la misma.

En primer lugar se va a mostrar un esquema de la estructura, en el cual nos basaremos para explicar cada una de sus partes.

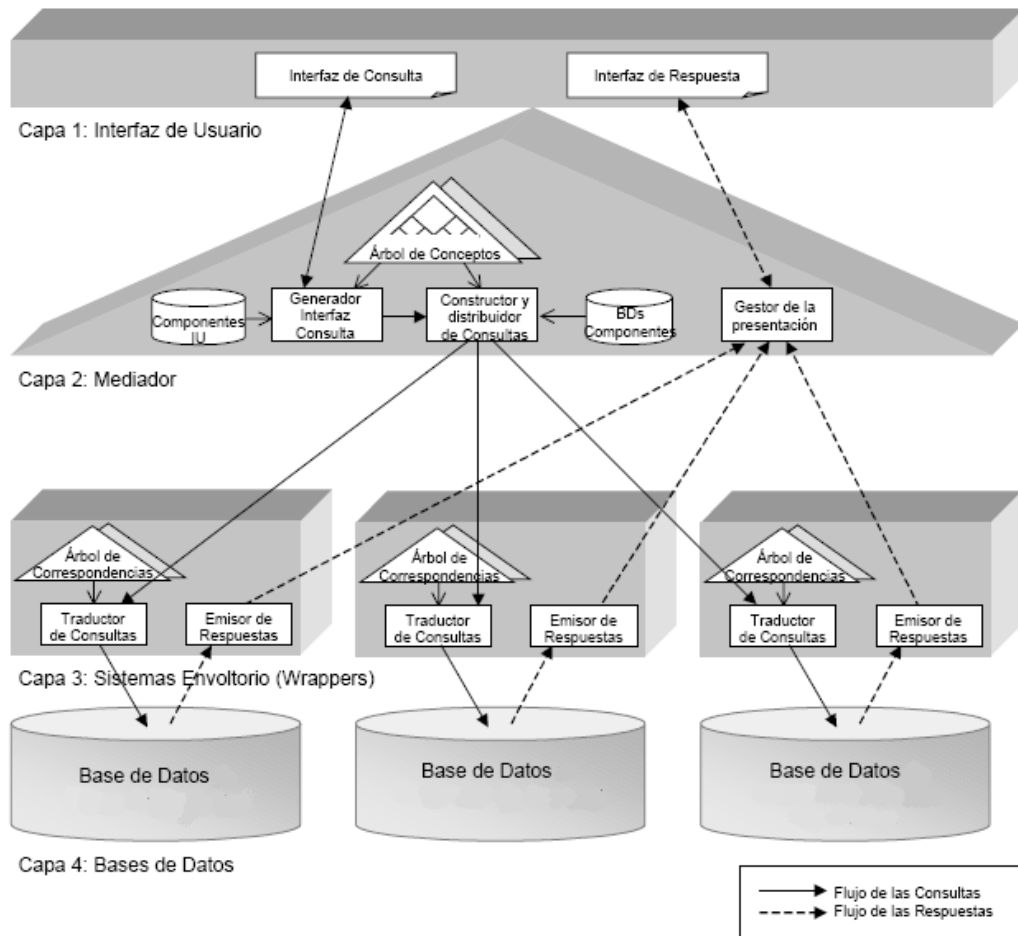


Figura 14. Sistema de Acceso Integrado a Bases de Datos Documentales

Como se puede observar la arquitectura se divide en cuatro capas. Para que la explicación sea más clara y concisa se comenzará realizando una explicación de cada una de las mismas.

o **Distribución por Capas**

Fuente: Tesis Doctoral. Arquitectura para Federación de Bases de Datos Documentales basada en Ontologías (“Consultada el 20-X-2009”)

✓ **Capa 1: Interfaz del Usuario**

*“Se trata de una capa conceptual, ya que no existen módulos de software propiamente dichos en la Interfaz de Usuario. Es el Mediador el que la genera cada vez que un usuario se conecta al sistema.*

*El diseño de la interfaz se basa en una utilización combinada de tres técnicas de diseño de Interfaces de Usuario (Lenguaje Natural Acotado, Metáforas*

*Cognitivas y Aproximación Navegacional). Conseguimos así que el Mediador sea capaz de generar una interfaz realmente fácil de usar, amigable y flexible de manera que el usuario pueda aprovechar las ventajas que ofrece el Sistema de Acceso Integrado que hemos diseñado y construido. Además, dado el caso y siempre en función de lo que se desee conseguir, la Interfaz de Usuario puede presentar una parte sencilla que permita hacer consultas muy simples a usuarios “comunes”, y otra parte que, sin incorporar demasiada dificultad, permita realizar consultas mucho más complejas a usuarios “expertos”.*

*Las principales características de la Interfaz de Usuario que es capaz de generar automáticamente un Sistema de Acceso Integrado son las siguientes:*

- *Proporciona los medios para expresar consultas, no sólo sobre los clásicos datos estructurados, sino que también facilita las consultas sobre el contenido de los documentos. Pensamos que no podía ser de otra manera ya que las bases de datos a las que el usuario va a acceder son documentales – almacenando, como ya hemos visto, tanto textos como datos estructurados sobre los documentos (además de tipos de datos multimedia, como imágenes).*
- *Proporciona los mecanismos necesarios para acceder a las bases de datos tanto a través de consultas globales a todo el conjunto de bases de datos, como consultas específicas a bases de datos de un dominio específico. Este es también un requisito imprescindible debido a que las bases de datos son heterogéneas, en cuanto a los textos y datos que almacenan.*
- *Es extremadamente intuitiva y fácil de usar. Requisito fundamental debido a que los usuarios del sistema van a ser usuarios web, en su mayoría, no informáticos, y, los usuarios web no podrán recibir entrenamiento alguno antes de acceder por primera vez al sistema.*
- *Como ya hemos dicho, la Interfaz de Usuario da la posibilidad de expresar consultas sencillas, a usuarios generales, y consultas más complejas y sofisticadas a usuarios expertos.*

## ✓ **Capa 2: Mediador**

*Esta capa engloba los módulos (y los Almacenes de Datos necesarios para soportar el funcionamiento de dichos módulos) encargados de hacer de nexo entre el usuario y las bases de datos componentes, haciendo transparente para el usuario las diferencias entre las mismas y su dispersión. Esta capa también incluye los Árboles de Conceptos, en los cuales aparecen todos los conceptos que existen en las bases de datos y que son relevantes para realizar consultas.*



A continuación se describen los elementos (módulos y almacenes) de esta capa:

- Generador de la Interfaz de Consulta: se encarga de generar de forma automática la interfaz Web de consulta, cada vez que un usuario se conecta al sistema. El Generador de la Interfaz de Consulta, para generar la interfaz Web de manera automática, interactúa con el Árbol de Conceptos, el cual explicaremos más adelante.
- Constructor y Distribuidor de Consultas: recibe las restricciones que expresa el usuario y construye el documento XML de la consulta (XML que sigue el DTD del Lenguaje de Consulta del sistema). Además, decide qué bases de datos están implicadas en la consulta y la dirige a dichas bases de datos;
- Gestor de la Presentación: construye la interfaz Web que presenta un resumen de las respuestas obtenidas de cada base de datos y gestiona la navegación a través de los datos y documentos, accediendo a la Interfaz de Respuesta implementada para cada base de datos.
- Componentes IU: es almacén en el que están las Frases en Lenguaje Natural Acotado y las Metáforas Cognitivas necesarias para generar la Interfaz de Consulta.
- Bases de Datos Componentes: almacena datos (nombre, descripción, parámetros de acceso, etc.) sobre las bases de datos integradas en el sistema.

### ✓ Capa 3: Sistemas Envoltorio (Wrappers)

Existe un Sistema Envoltorio por cada base de datos componente. Su función es ocultar al Mediador las diferencias entre las bases de datos. Este sistema gestiona todas las peticiones que el Mediador hace a las bases de datos. Es decir, traduce todas las peticiones que le envía el Mediador al lenguaje propio de la base de datos a la que está asociado, para que puedan ser ejecutadas.

Los Sistemas Envoltorio:

- Reciben las consultas del módulo Constructor y Distribuidor de Consultas en formato XML (Lenguaje de Consulta).
- Envían un resumen de las respuestas obtenidas por la consulta al Gestor de la Presentación.
- Cada Sistema Envoltorio incluye un Árbol de Correspondencias que contiene la información necesaria para llevar a cabo la traducción de las

*consultas (expresadas en el Lenguaje de Consulta) al lenguaje propio de la base de datos que tiene asociada.*

✓ **Capa 4: Bases de datos**

*Las bases de datos forman la cuarta capa de la arquitectura. Las tres bases de datos a las que actualmente se proporciona acceso integrado tienen las siguientes características:*

- *Heterogéneas, en cuanto a las máquinas que las albergan, al sistema operativo de dichas máquinas y a su estructura.*
- *Autónomas, es decir, totalmente independientes en funcionamiento. Esto implica, por un lado, que no es posible cambiar el esquema de ninguna de las bases de datos para adaptarlo al esquema global del Sistema de Acceso Integrado. Por otro lado, es posible que el esquema de las bases de datos cambie para satisfacer las necesidades de las aplicaciones propietarias. Las tres bases de datos tienen su propia Interfaz de Usuario vía Web. El hecho de que formen parte de nuestro Sistema de Acceso Integrado, no significa que éstas no puedan contestar consultas provenientes de su propia interfaz.*
- *Documentales, almacenan datos estructurados, textos y páginas digitalizadas (imágenes), por lo que una buena explotación de las bases de datos documentales debe permitir la aplicación de técnicas de recuperación de textos.*

✓ **Conceptos importantes**

- **Árboles de Conceptos**

*Los Árboles de Conceptos, tal y como se puede observar en la Figura 15, se sitúan en la capa del Mediador.*

*La principal función de los Árboles de Conceptos es almacenar los conceptos presentes en las bases de datos componentes, aunque únicamente aquellos que son interesantes para realizar consultas.*

*Sin embargo, existen otras funciones, además de para representar el esquema global del sistema, que pueden ser desempeñadas por los Árboles de Conceptos. Por ejemplo, una de las más importantes, es incluir en los Árboles de Conceptos información útil para generar la Interfaz de Consulta cada vez que un usuario se conecte, de este modo el sistema podrá interactuar de forma dinámica con el usuario, y adaptar su interfaz a las exigencias del usuario en cada caso. Además de esto también sirve para saber qué bases de datos están implicadas en cada consulta. De este modo se podrán enviar las consultas*

únicamente a esas bases de datos, en vez de a todas las que formen parte del Sistema de Acceso Integrado.

La forma más común de representar los Árboles de Conceptos es utilizar XML. La razón de esta elección es que este lenguaje permite definir formatos para intercambio de datos y metadatos, legibles por humanos y fácilmente parseables por máquinas.

A continuación se puede observar un Árbol de Conceptos de un Sistema de Acceso Integrado (de manera esquematizada), y una parte de dicho árbol de Conceptos en XML.

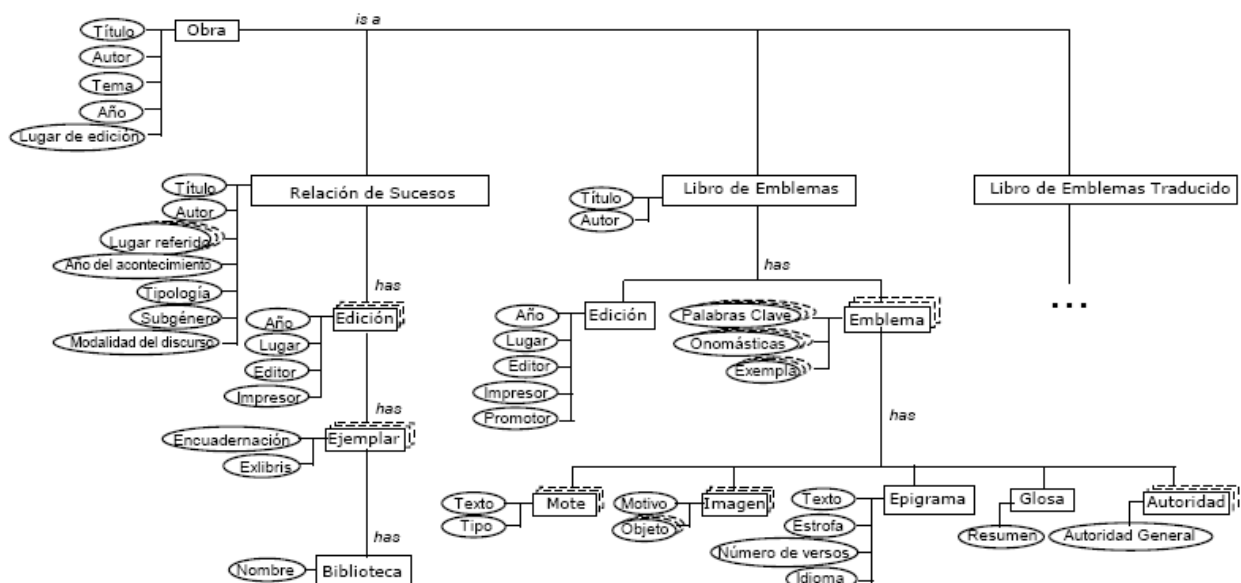


Figura 15. Árbol de Conceptos de un Sistema de Acceso Integrado (esquemático).

```

<arbol-conceptos>
<!-- Inicio Obra -->
  <concepto nombre="Obra">
    <atributo nombre="Titulo">
      <IU id="skel-" />
      <bd id="1" /> <bd id="2" /> <bd id="3" />
    </atributo>
    <atributo nombre="Autor">
      <IU id="ske5-" />
      <bd id="1" /> <bd id="2" /> <bd id="3" />
    </atributo>
    <atributo nombre="Tema">
      <IU id="ske36-" />
      <bd id="1" /> <bd id="2" /> <bd id="3" />
    </atributo>
    <atributo nombre="Año">
      <IU id="skel0-" />
      <bd id="1" /> <bd id="2" /> <bd id="3" />
    </atributo>
    <atributo nombre="Lugar de edición">
      <IU id="ske9-" />
      <bd id="1" /> <bd id="2" /> <bd id="3" />
    </atributo>
    <isa>
<!-- Inicio de Libro de Emblemas (isa de Obra) -->
      <IU id="metaphor" />
      <concepto nombre="Libro de Emblemas">
        <atributo nombre="Titulo">
          <IU id="skel-" />
          <bd id="1" /> <bd id="2" />
        </atributo>
        <atributo nombre="Autor">
          <IU id="ske5-" />
          <bd id="1" /> <bd id="2" />
        </atributo>
        <has>
<!-- Inicio de Edición (has de Libro de Emblemas) -->
          <concepto nombre="Edición">

```

Figura 16. Parte de un Árbol de Conceptos de un Sistema de Acceso Integrado (XML)

*Si observamos detenidamente la Figura XXXXXXXX se pueden apreciar las diferentes utilidades que tiene el Árbol de Conceptos. Por una parte se sirve para determinar el servicio que va a prestar la aplicación relacionada con la arquitectura. Otra utilidad, comentada anteriormente, es que indica en cada caso la interfaz que debe aparecer en función de la selección del usuario (etiqueta IU dentro de cada atributo o concepto). Por último nos indica las bases de datos en las que se encuentra el atributo o concepto seleccionado por el usuario (etiqueta bd dentro de cada concepto o atributo).*

#### ▪ Árboles de Correspondencias

*Los Árboles de Correspondencias tienen una estructura idéntica a la de los Árboles de Conceptos. Es una estructura en forma de árbol en la que los nodos son conceptos, con propiedades que llamamos atributos, y las ramas relaciones. Existen los mismos tipos de relaciones y la semántica de los*

conceptos y los atributos es la misma. Sin embargo, el propósito de los Árboles de Conceptos y los Árboles de Correspondencias son muy distintos. Los Árboles de Correspondencias servirán, como hemos dicho, para traducir las consultas que plantean los usuarios sobre los atributos del Árbol de Conceptos (y que el sistema almacena en XML) al lenguaje de la base de datos a la que está asociado dicho Árbol de Correspondencias.

Durante la traducción, para cada concepto que aparezca en el documento XML, se lee del Árbol de Correspondencias la Información de Correspondencia que tiene asociado dicho concepto y se añade a la sentencia SQL final. Asimismo, para cada atributo que exista en el documento XML de la consulta, se lee la Información de Correspondencia de dicho atributo y, antes de completar la sentencia SQL final con la nueva condición, se sustituye las etiquetas entre # por los valores concretos que haya establecido el usuario y que están recogidos en el documento XML.

CONSULTA EN XML	CONSULTA EN SQL
<pre> &lt;consulta&gt;   &lt;concepto nombre="Obra"&gt;     &lt;atributo nombre="Tema"&gt;       &lt;contiene limite="5"&gt;         &lt;valor cons="pecado"/&gt;         &lt;valor cons="clero"/&gt;         &lt;valor cons="Inquisición"/&gt;         &lt;valor cons="hoguera"/&gt;         &lt;valor cons="bruja"/&gt;       &lt;/atributo&gt;       &lt;atributo nombre="Año"&gt;         &lt;entre&gt;           &lt;valor cons="1500"/&gt;           &lt;valor cons="1650"&gt;         &lt;/atributo&gt;       &lt;/concepto&gt;     &lt;/consulta&gt; </pre>	<pre> select cod_obra, cod_emblem from obra ob, emblema em, edicion ed where ob.cod_obra = em.cod_obra and ed.cod_edic = ob.cod_edic and 5=   (select count(*)    from clave cl    where em.cod_obra = cl.cod_obra    and em.cod_emblem = cl.cod_emblem    and (cl.clave like "pecado"         or cl.clave like "clero"         or cl.clave like "Inquisición"         or cl.clave like "hoguera"         or cl.clave like "bruja")) and (ed.cod_edic &gt; 1500 and ed.cod_edic &lt; 1650) </pre>

Figura 17. Traducción de lenguaje de consulta (XML) a lenguaje de la base de datos (SQL).

En la Figura 18 se puede observar como gracias al Árbol de Correspondencias se puede obtener una consulta para una determinada base de datos a partir de una consulta generada en un lenguaje de consulta general. Esto se consigue debido a que el Árbol de Correspondencias asocia para cada concepto o atributo un código específico en lenguaje SQL, cuya suma de todas las partes resulta una consulta completa en lenguaje SQL que se procesará en la base de datos pertinente.

A continuación, se encuentra un pequeño fragmente de un Árbol de Correspondencias. Naturalmente, el lenguaje que utiliza es XML. Se puede observar que para atributo o concepto existe un código SQL asociado.”

```

<!-- Inicio concepto Obra -->
<concepto nombre="Obra">
  <correspondencia>
    <select> cod_obra, cod_emblem </select>
    <from> obras ob, emblema em </from>
    <where><fijo> ob.cod_obra = em.cod_obra </fijo></where>
  </correspondencia>
  <atributo nombre="Título">
    <correspondencia>
      <where>
        <fijo> and ob.titulo like '%#valor#%' </fijo>
        <opcional fin=""> and ob.titulo like '%#valor#%' </o
      </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Autor">
    <correspondencia>
      <where>
        <fijo> ob.cod_obra = em.cod_obra
          and ob.autor like '%#valor#%' </fijo>
      </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Tema">
    <correspondencia>
      <where>
        <fijo> and #limite# &lt;= (select count(*)
          from clave cl
          where em.cod_obra = cl.cod_obra
          and em.cod_emblem = cl.cod_emblem
          and clave in ('#valor#' </fijo>
        <parametro nombre="limite"/>

```

Figura 18. Fragmento de un Árbol de Correspondencias (en XML)

#### 5.4.3.3 DISEÑO DE LA ARQUITECTURA DEL SISTEMA DE ACCESO INTEGRADO A BASES DE DATOS DOCUMENTALES BASADA EN ÁRBOLES DE CONCEPTOS APLICADO A NUESTRO PROYECTO.

El principal inconveniente que se plantea en las otras dos arquitecturas expuestas como posible solución al problema que presenta este proyecto, es la dificultad que supone conseguir que las diferentes organizaciones permitan utilizar sus datos, para que estos sean clasificados y utilizados en una aplicación completamente independiente de la suya. En esta nueva arquitectura esto no es un problema puesto que no sucede. El buscador en el que se basa la aplicación del sistema, realizará una búsqueda por las diferentes bases de datos de las organizaciones, recapitulará toda la información existente en las bases de datos de las mismas, sobre aquello solicitado por el usuario, y mostrará una página en donde aparecen los resultados sobre la búsqueda que alberga cada página Web. El usuario sólo podrá acceder al contenido de la información a través de la interfaz de la página Web de la organización.

Para explicar de forma detallada como sería una arquitectura de este tipo aplicada a nuestra aplicación, nos vamos a basar en la Figura 19, en la que se muestra capa una de las capas de la arquitectura con sus correspondientes elementos.

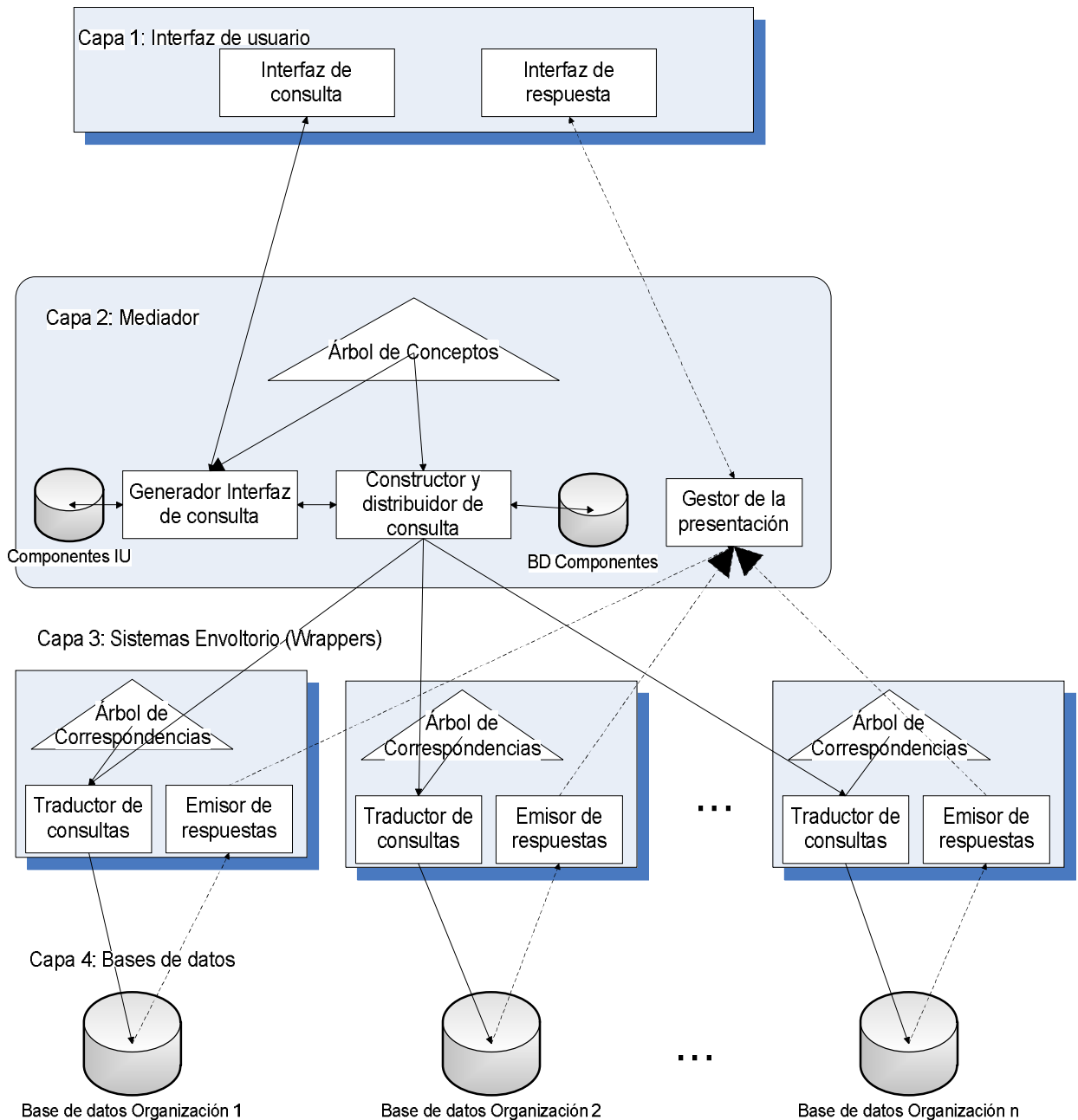


Figura 19. Arquitectura del sistema de acceso integrado a base de datos documentales aplicado a nuestro problema.

El funcionamiento de la aplicación sería el siguiente:

El usuario se conecta al sistema, y este de forma automática a través del *Generador de Interfaz de Consulta* le presenta una interfaz determinada. El usuario decide los atributos sobre los que quiere realizar la consulta y aplica, sobre los mismos, las restricciones que crea conveniente. Entre el usuario y el sistema existe una comunicación dinámica, ya que en función de las decisiones que va tomando el usuario, el sistema le muestra una interfaz u otra. Para ello el *Generador de Interfaz de Consulta*

se apoya en el *Árbol de Conceptos* que es quien le indica cual es el elemento de los almacenados en *Componentes IU* que tiene que mostrar al usuario. En nuestro caso, una consulta podría ser “*iglesias de nombre San Pedro en Álava*”.

Mientras el usuario va tomando decisiones acerca de los atributos sobre los que quiere realizar restricciones, *el Constructor y Distribuidos de consultas* va formando la consulta (en XML) que va a ser entregada a cada una de las bases de datos. Una vez que el usuario envía la petición, *el Constructor y Distribuidos de consultas* finaliza su elaboración de la consulta, e informándose en la bases de datos de BD componentes, envía la consulta a aquellas bases de datos que tengan información sobre los atributos sobre los que el usuario ha expresado condiciones. En nuestro ejemplo, *el Constructor y Distribuidos de consultas* elaborará la consulta y la enviará a aquellas bases de datos que tengan información sobre el atributo “*nombre*” y el atributo “*tipo*”.

Una vez enviada la consulta a las bases de datos pertinentes, los sistemas envoltorio de cada organización deben transformar la consulta del Lenguaje Consulta (en este caso XML) al lenguaje propio de la base de datos. Para conseguir esto cada sistema envoltorio cuenta con un *Traductor de Consultas* quien mediante su iteración con el *Árbol de Correspondencias* consigue transformar el Lenguaje Consulta en el lenguaje que utilice la base de datos de una organización determinada. En nuestro caso el Traductor de Consultas transformaría el Lenguaje Consulta (XML) al lenguaje de nuestra base de datos, por ejemplo SQL. El resultado sería algo así: “*Select \* from patrimonio where tipo='iglesia' and nombre='San Pedro' ”*

Una vez realizada la traducción, la consulta se ejecuta en la base de datos y el resultado se transfiere al *Emisor de Respuestas*, quién la envía al *Gestor de Presentación*, quien recibe los resultados de todas las consultas realizadas en cada base de datos, y las presenta al usuario mediante una *Interfaz de Respuesta*. En nuestro caso, el Gestor de Presentación daría formato a todas las respuestas recibidas del Emisor de Respuestas, mostrando al usuario a través de la Interfaz de Respuesta, todas las bases de datos en las que se ha encontrado información sobre “*iglesias de nombre San Pedro en Álava*” con su correspondiente contenido.

### **Ventajas**

- ✓ Permite una autonomía local de los datos a cada una de las organizaciones.
- ✓ Consigue una transparencia total para el usuario.
- ✓ Mediante la replicación de información, las bases de datos distribuidas pueden presentar cierto grado de tolerancia a fallos haciendo que el funcionamiento del sistema no dependa de un solo lugar como en el caso de las bases de datos centralizadas.
- ✓ Posibilidad de compartir los datos al tiempo que se mantiene un cierto de grado de control local.



- ✓ Una buena distribución de los datos, dará como resultado un mejor rendimiento.
- ✓ Permite a cada organización continuar trabajando con su base de datos, ya que para agregar su base de datos al sistema bastará con llevar a cabo un Sistema Envoltorio (Wrapper) para su base de datos, y ésta podrá ser conectada al sistema.
- ✓ El sistema es completamente escalable. Tanto si se quiere introducir información nueva sobre un elemento patrimonial ya existente, como si se quiere introducir información sobre un patrimonio nuevo, modificar datos o eliminarlos, cada organización lo puede hacer de manera completamente autónoma, sin alterar el funcionamiento del sistema.

### **Desventajas**

- ✓ Control y manejo de los datos. Dado que éstos residen en muchos nodos diferentes y se pueden consultar por nodos diversos de la red, la probabilidad de violaciones de seguridad es creciente si no se toman las precauciones debidas.
- ✓ Dado que los datos pueden estar replicados, el control de concurrencia y los mecanismos de recuperación son mucho más complejos que en un sistema centralizado.
- ✓ Al ser menos usuales que las bases de datos centralizadas su dificultad a la hora de implementarlas es mayor.
- ✓ Se trata de una arquitectura bastante atípica por lo que puede presentar problemas que desconozcamos totalmente.

## 5.5 ELECCIÓN DE LA PROPUESTA MÁS ADECUADA

Después de haber realizado un análisis de las diferentes soluciones ya existentes, a las que en este documento también se les hace referencia como diferentes propuestas, que se pueden llevar a cabo y un análisis profundo de los datos disponibles, estamos en condiciones de argumentar cuál es la propuesta que más se adapta a nuestras necesidades.

Tal y como se ha expuesto en el análisis de las diferentes propuestas, a priori y siempre y cuando se cumplan una serie de requisitos, todas las propuestas pueden llevarse a cabo. Sin embargo, tras realizar el análisis de los datos, existen propuestas que no pueden desarrollarse. En este documento se argumentará porqué alguna de las propuestas no es posible llevarla a cabo, y qué propuesta es la más indicada para las características de nuestro proyecto. En primer lugar se irán analizando las limitaciones que suponen las características de nuestro proyecto en cada una de las propuestas, para finalmente argumentar qué arquitectura se adaptaría mejor a los objetivos del proyecto.

En cuanto al diseño de una arquitectura de tres capas con una **base de datos centralizada**, se presentan una serie de inconvenientes. En primer lugar, es prácticamente imposible conseguir reunir los datos de diversas organizaciones para utilizarlas en nuestra base de datos. A pesar de que diversas organizaciones que trabajan en diferentes actividades relacionadas con el patrimonio hayan mostrado su interés en que este proyecto se desarrolle, se considera que es prácticamente imposible que todas ellas, o su mayoría, cedan sus datos para que una aplicación completamente ajena a su organización trabaje con ellos, sin que dichas organizaciones reciban beneficio económico alguno. En el caso de que esto fuera posible, y las organizaciones cedieran sus datos sin recibir nada a cambio, el tiempo que nos costaría recopilar toda esa información sobrepasa, y de manera exagerada, la duración estimada para el desarrollo de este proyecto.

Además de este principal inconveniente, existe otro que es también importante, la ausencia total de autonomía para las bases de datos de las organizaciones. Lo que se pretendía al comienzo del proyecto era conseguir, si era posible, elaborar una aplicación cuya arquitectura permitiera a cada organización mantener sus bases de datos ajenas a la aplicación que se va a desarrollar. Sin embargo, si, para el desarrollo de esta propuesta basada en una base de datos centralizada, es necesario realizar una clasificación de los datos y almacenar dicha clasificación en una única base de datos centralizada, no existirá autonomía alguna para las bases de datos de las organizaciones, ya que los datos se controlarán únicamente desde dicha base de datos centralizada.

Este inconveniente desemboca en otros inconvenientes, ya que si una organización consigue nueva información sobre el patrimonio, o desea modificar o eliminar algún dato incorrecto, no le bastará con introducir, modificar o borrar dicha información en su bases de datos sino que deberá ponerse en contacto con el administrador de la base de datos centralizada, para que éste introduzca, modifique o borre la información pertinente.

Debido a estos inconvenientes, es obvio, que esta propuesta no es la que mejor se adapta a las características del proyecto que se va a llevar a cabo. Esta solución sería posible, únicamente, si todas las organizaciones estuvieran de acuerdo en reunir toda la información en un solo sitio Web, y nombrar un administrador de la base de datos que se encargue de realizar todas las actualizaciones de los datos. Lo más costoso de esta propuesta sería el análisis de los datos y su posterior clasificación.

Respecto al diseño de una arquitectura de tres capas con **bases de datos distribuidas (y homogéneas)**, éste presenta ventajas respecto a la arquitectura anterior, sin embargo presenta problemas de incompatibilidad con las características de nuestro proyecto. Respecto a la arquitectura basada en una base de datos centralizada, la arquitectura basada en bases de datos distribuidas presenta mejoras como la autonomía en las distintas bases de datos de las organizaciones. En esta arquitectura cada organización tiene el control de su base de datos y es la aplicación quien, de forma transparente para el usuario, realiza la consulta en las diferentes bases de datos de las organizaciones, para presentar una respuesta al usuario. Cada organización puede realizar inserciones, modificaciones o borrados sin tener que informar de los cambios a las demás organizaciones.

Sin embargo, existen, aún, una serie de inconvenientes importantes. El principal inconveniente está íntimamente ligado al principal inconveniente de la arquitectura basada en una base de datos centralizada, no es otro que la diversidad de los datos. Cada organización dispone de sus propios datos, datos en formato digital, en papel, datos en páginas Web estáticas... Con los datos almacenados sucede algo similar, hay datos en tablas de Excel, datos en Access, datos en MySQL, datos en bases de datos Oracle... Por lo que para poder distribuir los datos en las bases de datos homogéneas, es imprescindible realizar una clasificación de los mismos, y distribuirlos en un mismo formato por las diferentes bases de datos. Por tanto estamos ante el mismo problema que en la arquitectura anterior, necesitamos que todas las organizaciones estén de acuerdo en actualizar el formato de sus datos, para posteriormente volver a ser clasificados, algo que en términos de tiempo, tal y como se ha comentado antes, sobrepasa la duración estimada del proyecto.

Por tanto podemos decir que, aunque con esta arquitectura las bases de datos ganan autonomía, no estamos ante la arquitectura que más se adapta a las características y restricciones de nuestro proyecto, por lo que no queda otro remedio que rechazar también esta propuesta.

La tercera alternativa es el Diseño de la Arquitectura del Sistema de Acceso Integrado a Bases de Datos Documentales basada en Árboles de Conceptos. Esta propuesta presenta una serie de ventajas respecto a las dos anteriores, sin embargo también presenta algunos inconvenientes.

Respecto a las ventajas que presenta esta tercera alternativa cabe destacar, la más importante de todas ellas, esta arquitectura permite a cada base de datos de cada organización mantener tanto su estructura actual, como su formato, como sus datos, además de su autonomía. Para lograr esto, tal y como hemos visto en la descripción del

análisis de cada una de las propuestas, la Arquitectura del Sistema de Acceso Integrado a Bases de Datos Documentales basada en Árboles de Conceptos contiene unos sistemas envoltorio (Wrappers), uno para cada base de datos, que se encarga de transformar el lenguaje de consulta que recibe de la aplicación en el lenguaje de cada base de datos, permitiendo así la autonomía de cada una de las bases de datos.

Con esta arquitectura se solucionan todos los problemas que presentaban las dos propuestas anteriores tanto la arquitectura de tres capas con una base de datos centralizada, como la arquitectura basada en bases de datos distribuidas.

Sin embargo, esta arquitectura presenta también una serie de inconvenientes. El inconveniente principal está relacionado con el análisis de los datos. Tal y como se puede apreciar en el documento titulado *Análisis de los datos*, los datos de las diferentes organizaciones, en su mayoría están únicamente disponibles en documentos estáticos en la Web, es decir, carecen de bases de datos, por lo que esta arquitectura no puede llevarse a cabo en su totalidad. Sin embargo, si existe una base de datos disponible, la del Laboratorio de Documentación Geométrica del Patrimonio de la UPV/EHU. Esta base de datos nos va a permitir implantar esta arquitectura, aunque únicamente, por el momento, exista una base de datos, ya que si en un futuro se desea agregar alguna más al sistema, únicamente se deberá construir el correspondiente sistema envoltorio (Wrapper) que permita traducir el lenguaje de consulta que envía la aplicación a la base de datos al lenguaje propio de la base de datos que se agrega al sistema.

Por otro lado, no se puede descartar la información que cada una de las organizaciones tiene expuesta en la Web, ya que esta nos puede servir para nuestra aplicación. Por lo tanto es probable que para introducir esta información tengamos que modificar, o mejor dicho, adaptar esta arquitectura a nuestras necesidades. Por tanto, podemos concluir que esta arquitectura puede ayudarnos a presentar una solución final a nuestro problema, sin embargo, es obvio, que debemos adaptar esta arquitectura a las necesidades de nuestra aplicación, y a las restricciones que nos vayan apareciendo, como en este caso han sido las conclusiones abstraídas del análisis de los datos.

En definitiva, y como conclusión, se puede argumentar que de las tres propuestas únicamente puede ayudarnos a la hora de realizar el diseño de la arquitectura de nuestra solución, la tercera alternativa, titulada, Arquitectura del Sistema de Acceso Integrado a Bases de Datos Documentales basada en Árboles de Conceptos, ya que es la única de las tres que va permitirnos elaborar una posible solución, sin tener la necesidad de contar directamente con los datos de las organizaciones. Además, otro de los factores clave es la escalabilidad de la aplicación, ya que va a permitir a nuevas organizaciones agregarse a la arquitectura, con el único requisito que supone el diseño de un sistema envoltorio para la base de datos de la organización que desea unirse al sistema. Otra ventaja que nos permite esta arquitectura, y que probablemente, sirva de “anzuelo” para incorporar nuevas organizaciones al sistema, es que, aunque el sistema de búsqueda orientada de la aplicación contenga interfaces de la propia aplicación, las soluciones de las consultas realizadas por el usuario, serán visualizadas en las interfaces de cada una

de las páginas Web de las organizaciones que tengan alguna respuesta relacionada con la consulta realizada por el usuario.

De las otras dos arquitecturas, quizás una base de datos centralizada pueda ayudarnos, por ejemplo para almacenar la información estática de las páginas Web, o para almacenar las Bases de datos de la arquitectura con sus respectivos atributos...

## 5.6 CASOS DE USO

Con el fin de obtener un análisis de los requerimientos funcionales de la aplicación en un formato universalmente reconocible se utilizará notación UML para la descripción de los casos de uso de las aplicaciones que van a ser desarrolladas en este proyecto. Concretamente una aplicación que sirva para dar soporte a la base de datos en la cual se van a almacenar la información existente en las páginas Web de las organizaciones cuya actividad principal está relacionada con el patrimonio, y una segunda aplicación de búsqueda orientada en bases de datos heterogéneas.

Entre las dos aplicaciones que tenemos, toman parte un total de tres actores, los cuales se describen a continuación.

1. Usuario de Internet.

Se considera usuario de Internet a cualquier persona que navegando por la red se conecta al sitio Web en el que está instalada la aplicación de búsqueda orientada.

2. Administrador

Es uno de los dos perfiles existentes en la aplicación de soporte a la base de datos que almacena información de las páginas Webs. El administrador tiene potestad absoluta para realizar cualquier funcionalidad de la aplicación

3. Usuario

Es el otro perfil existente en la aplicación de soporte a la base de datos que almacena información de las páginas Webs. El usuario tiene restricciones a la hora de realizar ciertas funcionalidades.

## 5.6.1 CASOS DE USO DEL ADMINISTRADOR

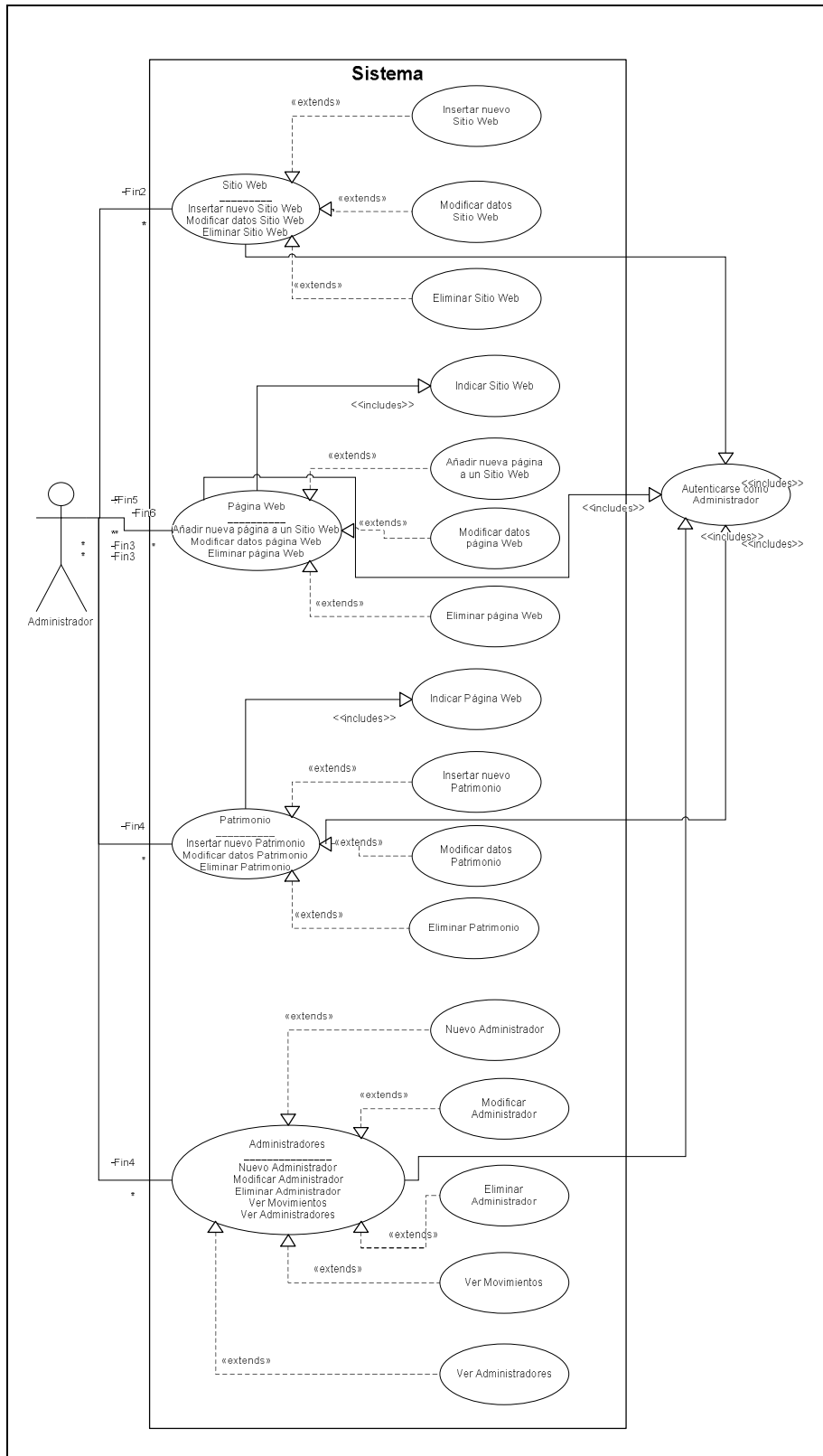


Figura 20. Casos de uso del Administrador.

- Insertar nuevo sitio Web

Este caso de uso permite a un individuo con el perfil de administrador introducir un nuevo sitio Web en la base de datos.

- Modificar datos sitio Web

El caso de uso “Modificar datos sitio Web” permite al administrador modificar los datos de un sitio Web determinado.

- Eliminar sitio Web

Como su propio nombre indica permite un administrador determinado eliminar del sistema el sitio Web que desee.

- Añadir una nueva página a un sitio Web

Esta funcionalidad ofrece al administrador la oportunidad de añadir una página a un sitio Web que ya forme parte del sistema.

- Modificar datos página Web

El caso de uso “Modificar datos página Web” permite al administrador modificar los datos de una página Web determinada, previa selección del sitio Web en el cual se encuentra dicha página.

- Eliminar página Web

Permite eliminar una página Web determinada, previa selección del sitio Web en el cual se encuentra dicha página.

- Insertar nuevo patrimonio

El caso de uso “Insertar nuevo patrimonio” otorga la posibilidad al administrador de introducir un nuevo elemento patrimonial, previa selección de la página y del sitio Web en donde se quiere introducir dicho elemento patrimonial.

- Modificar datos patrimonio

Esta funcionalidad otorga la posibilidad de modificar los datos de un elemento patrimonial, previa selección de la página, y el sitio Web en donde se encuentra el elemento a modificar.



- Eliminar patrimonio

Otorga la posibilidad de eliminar un elemento patrimonial del sistema, previa selección de la página, y el sitio Web en donde se encuentra el elemento a eliminar.

- Nuevo administrador

El caso de uso “Nuevo administrador” permite introducir un nuevo administrador (independientemente de su perfil) en el sistema.

- Modificar administrador

Permite al administrador la posibilidad de modificar los datos de un individuo que pertenece al sistema (sólo si este tiene perfil de usuario). Además permite al administrador modificar sus propios datos. Incluye modificar contraseña.

- Eliminar administrador

Otorga la posibilidad a un administrador de eliminar una persona que pertenece al sistema (sólo si este tiene perfil de usuario).

- Ver movimientos

Permite al administrador consultar los movimientos que se han realizado en el sistema. El sistema proporciona la fecha, la hora exacta, el movimiento realizado y la persona que lo ha llevado a cabo.

- Ver administradores

El caso de uso “Ver administradores” permite al administrador ver los administradores (independientemente de su perfil) que participan en el sistema.

- ✓ Para poder llevar a cabo todos y cada uno de estos casos de uso el administrador debe autenticarse como tal al comienzo de la aplicación.

## 5.6.2 CASOS DE USO DEL USUARIO

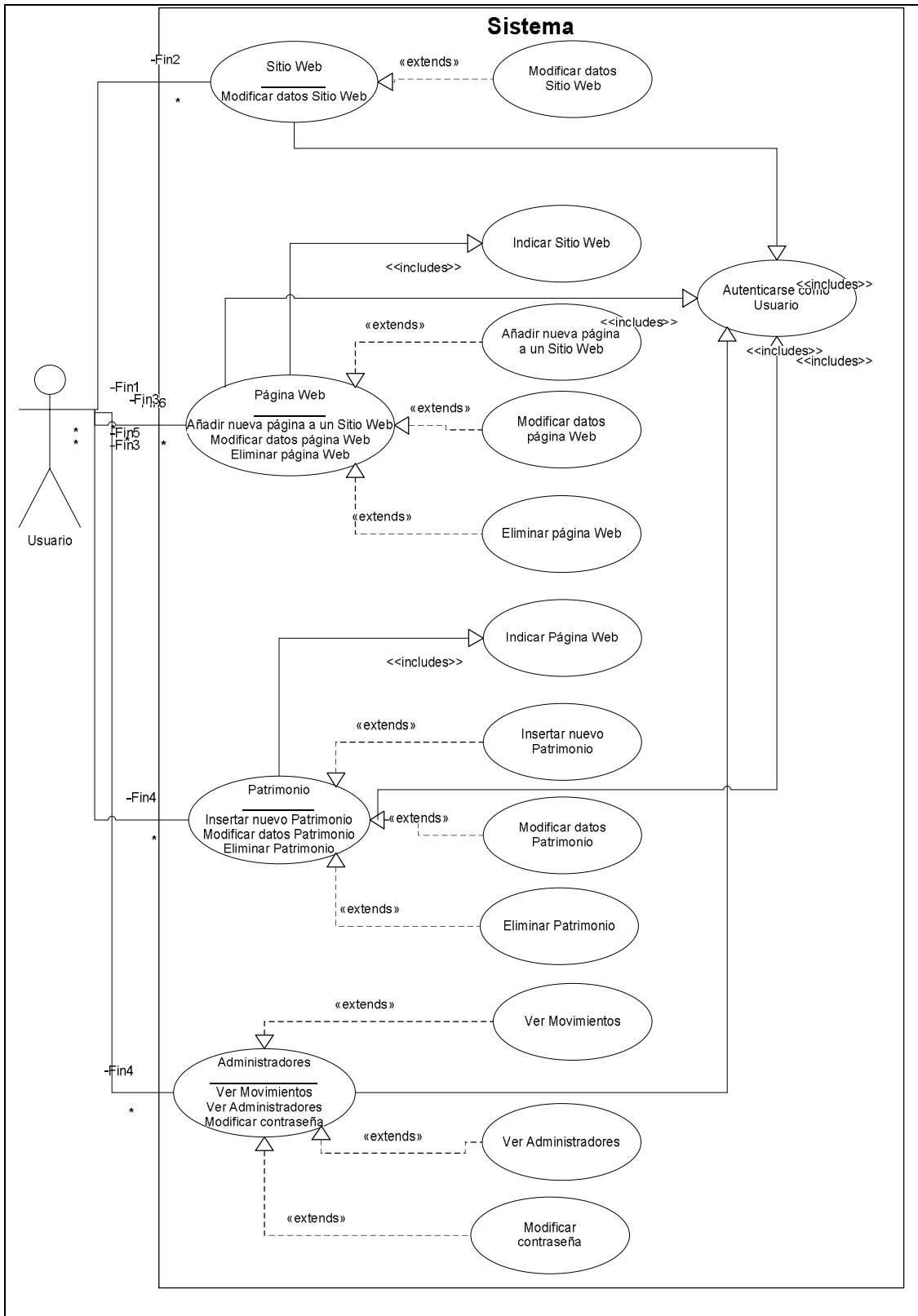


Figura 21. Casos de uso del Usuario.

- Modificar datos sitio Web

El caso de uso “Modificar datos sitio Web” permite al usuario modificar los datos de un sitio Web determinado.

- Añadir una nueva página a un sitio Web

Esta funcionalidad ofrece al usuario la oportunidad de añadir una página a un sitio Web que ya forme parte del sistema.

- Modificar datos página Web

El caso de uso “Modificar datos página Web” permite al usuario modificar los datos de una página Web determinada, previa selección del sitio Web en el cual se encuentra dicha página.

- Eliminar página Web

Permite eliminar una página Web determinada, previa selección del sitio Web en el cual se encuentra dicha página.

- Insertar nuevo patrimonio

El caso de uso “Insertar nuevo patrimonio” otorga la posibilidad al usuario de introducir un nuevo elemento patrimonial, previa selección de la página y del sitio Web en donde se quiere introducir dicho elemento patrimonial.

- Modificar datos patrimonio

Esta funcionalidad otorga la posibilidad de modificar los datos de un elemento patrimonial, previa selección de la página, y el sitio Web en donde se encuentra el elemento a modificar.

- Eliminar patrimonio

Otorga la posibilidad de eliminar un elemento patrimonial del sistema, previa selección de la página, y el sitio Web en donde se encuentra el elemento a eliminar.

- Modificar contraseña

Permite al usuario la posibilidad de modificar su contraseña.

- Ver movimientos

Permite al usuario consultar los movimientos que se han realizado en el sistema. El sistema proporciona la fecha, la hora exacta, el movimiento realizado y la persona que lo ha llevado a cabo.

- Ver administradores

El caso de uso “Ver administradores” permite al usuario ver los administradores (independientemente de su perfil) que participan en el sistema.

- ✓ Para poder llevar a cabo todos y cada uno de estos casos de uso el usuario debe autenticarse como tal al comienzo de la aplicación.

### 5.6.3 CASOS DE USO DEL USUARIOWEB

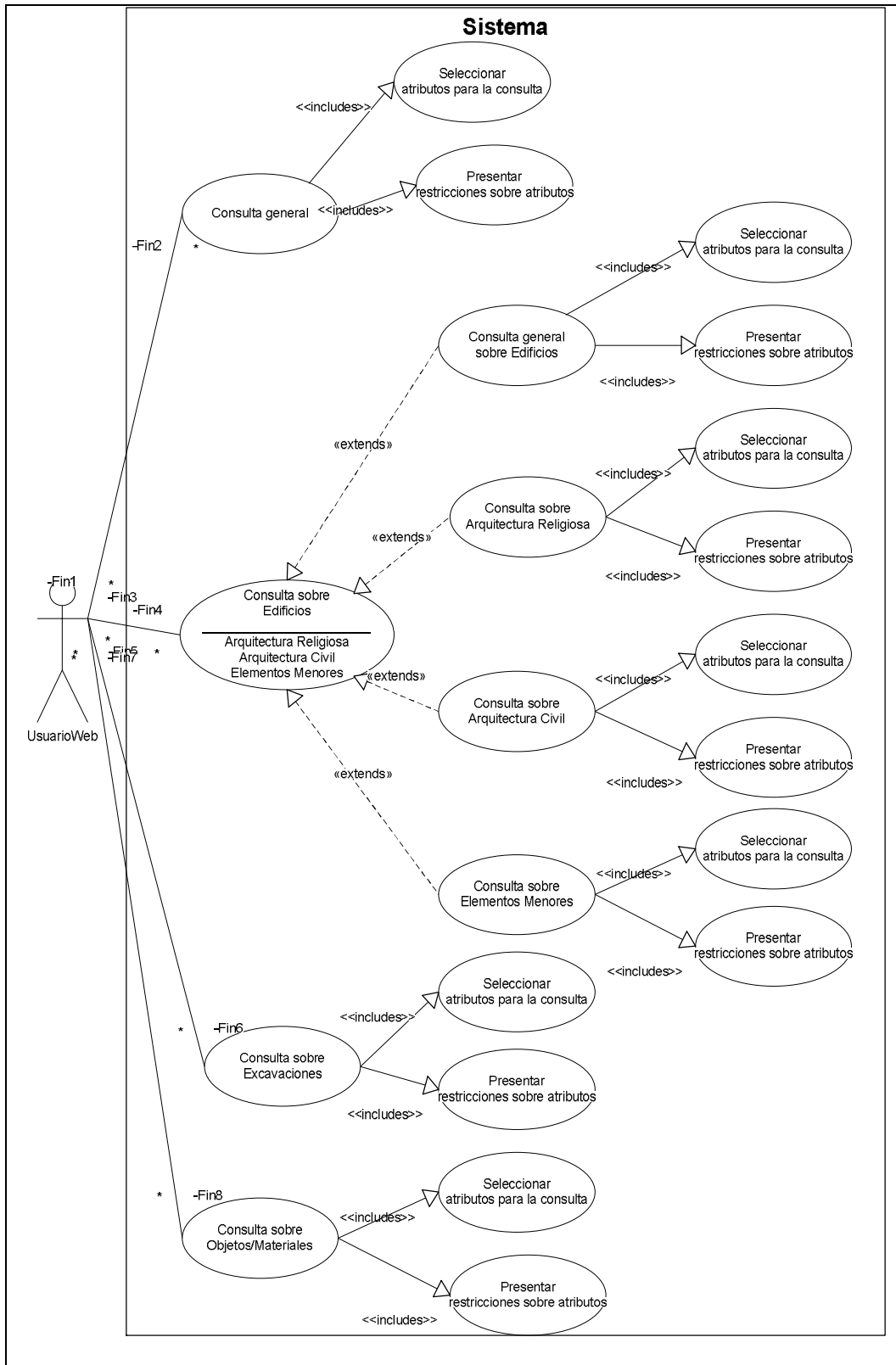


Figura 22. Casos de uso del UsuarioWeb.

- Consulta general

El caso de uso “Consulta general” permite al usuarioWeb llevar a cabo una consulta sobre el patrimonio en general.

- Consulta sobre Excavaciones

Esta funcionalidad ofrece al usuarioWeb la oportunidad de realizar una búsqueda sobre los elementos patrimoniales clasificados dentro de Excavaciones.

- Consulta general sobre Edificios

Permite al usuarioWeb realizar una consulta general sobre los elementos patrimoniales clasificados dentro del concepto general Edificios.

- Consulta sobre Arquitectura Religiosa

Esta funcionalidad ofrece al usuarioWeb la oportunidad de realizar una búsqueda sobre los elementos patrimoniales clasificados dentro de Arquitectura Religiosa.

- Consulta sobre Arquitectura Civil

El caso de usos “Consulta sobre Arquitectura Civil” permite al usuarioWeb desarrollar una búsqueda sobre elementos de Arquitectura Civil.

- Consulta sobre Elementos Menores

Esta funcionalidad ofrece al usuarioWeb la oportunidad de realizar una búsqueda sobre los elementos patrimoniales clasificados dentro de Elementos Menores.

- Consulta sobre Objetos

Permite al usuarioWeb realizar una consulta sobre los elementos patrimoniales clasificados dentro del concepto Objetos.



## **6. DISEÑO DE LA SOLUCIÓN**

### **6.1. INTRODUCCIÓN**

Tras haber realizado un amplio estudio de los datos disponibles, y haber analizado de forma detallada las diferentes posibles soluciones, que a priori, podrían ser útiles para la búsqueda de la solución al problema que se presenta en este proyecto, es el momento de elaborar el diseño de la arquitectura de nuestro sistema.

La solución que se pretende conseguir, es la implantación de un sistema de búsqueda orientada donde, una vez que el usuario introduzca las características de la búsqueda que desea realizar (atributos y restricciones), el sistema le presente en la solución, por un lado, cada una de las respuestas a esa consulta existentes en cada una de las páginas Web de las organizaciones, y por otro lado, la respuestas a la consulta que se encuentren en las diferentes bases de datos de nuestro sistema (una en principio).

Para conseguir que el sistema muestre las respuestas, a la consulta realizada por el usuario, con la información existente en las páginas Web de las organizaciones, primeramente hay que reunir esa información y clasificarla, para después poder informatizarla. Es decir, basándonos en la fase del análisis, en donde se hacía referencia tanto al análisis de los datos como al análisis de las diferentes soluciones ya existentes o diferentes propuestas para la solución de este proyecto, lo que se pretende llevar a cabo es desarrollar una base de datos centralizada, en donde resida la información, bien clasificada, del contenido de las diferentes páginas Web de las organizaciones. De esta forma, cuando el usuario envíe una consulta el sistema buscará en su base de datos los datos relacionados con la consulta, y podrá presentar su solución al usuario.

Sin embargo, además de la búsqueda de la información clasificada de las diferentes páginas Web de las organizaciones, se pretende que el sistema realice la búsqueda por las diferentes bases de datos que forman parte del sistema. Estas bases de datos pueden ser bases de datos ya construidas, o en construcción, pero completamente independientes y diferentes entre sí. Obviamente, el sistema centralizado que nos servía de solución para el caso anterior, no es válido para este problema, ya que aquí se está tratando con diferentes bases de datos independientes. Para solucionar este problema, volviendo de nuevo al análisis realizado sobre las diferentes soluciones existentes o diferentes propuestas de solución, nos encontramos con el Diseño de la Arquitectura del Sistema de Acceso Integrado a Bases de Datos Documentales basada en Árboles de Conceptos, en el que se integraban tres bases de datos documentales, completamente independientes entre sí, en un mismo sistema. Implantando, lo que sería un sistema similar al estudiado en esta parte del análisis, conseguiríamos presentar al usuario una solución a su consulta, formada por las diferentes soluciones conseguidas en cada una de las bases de datos que conformen nuestro sistema.

Como se ha podido observar se ha conseguido dotar de una solución a cada uno de los problemas, sin embargo, se presenta otro que consiste en elaborar una arquitectura que abarque las dos soluciones dadas. Para ello, se ha pensado en elaborar



una arquitectura similar a la Arquitectura del Sistema de Acceso Integrado a Bases de Datos Documentales basada en Árboles de Conceptos, pero tratando a la base de datos centralizada, en donde se encuentra clasificada la información de los sitios Web de las diferentes organizaciones, como una base de datos más, como si ésta fuera la base de datos de una organización que ha sido unida a nuestro sistema.

Sin embargo el proyecto no concluye con el desarrollo de esta arquitectura, ya que el mismo consta de una segunda parte. Esta segunda parte consiste en la elaboración de una aplicación para que el administrador de la base de datos pueda mantener actualizado el contenido de las páginas Web de la base de datos centralizada de nuestro sistema. Para ello la aplicación deberá consistir en una serie de formularios, bien organizados, donde el administrador de la bases de datos, pueda añadir, eliminar, o modificar datos de la fuente de datos centralizada de nuestro sistema.

Durante este apartado de diseño de la solución, se explicará el diseño de la base de datos centralizada que reúne la información de los sitios Web. Se analizará la organización de dicha base de datos, mostrando sus campos y atributos. Además se expondrán los diseños de las bases de datos que formen parte de nuestro sistema, explicando además sus funcionalidades. Se realizará una amplia descripción sobre el diseño de las interfaces, tanto de las interfaces correspondientes al sistema de búsqueda orientada, como las relacionadas con la aplicación que se va a desarrollar para el mantenimiento de la base de datos centralizada. Se expondrán las técnicas que se han llevado a cabo para desarrollar dichas interfaces. Por último se realizará una amplia descripción de la arquitectura que se va a desarrollar como solución al proyecto. En esta descripción, se analizarán cada una de las capas que forman esta arquitectura, así como los componentes que forman cada una de las capas.

Con el fin de realizar una descripción de todos los puntos descritos en el párrafo superior de manera ordenada, separaremos ambas aplicaciones como si fueran diferentes y explicaremos su diseño en cada una de ellas. La primera aplicación que será descrita es la Aplicación que da soporte a la base de datos que clasifica información sobre la información existente en las páginas Web. Una vez que terminemos con esta, se describirá la Aplicación de búsqueda orientada.

## **6.2 APLICACIÓN QUE DA SOPORTE A LA BASE DE DATOS QUE CLASIFICA DATOS SOBRE LA INFORMACIÓN EXISTENTE EN LA PÁGINAS WEB.**

### **6.2.1 BASES DE DATOS DE LA APLICACIÓN**

#### **6.2.1.1 BASE DE DATOS CENTRALIZADA “SITIOS WEB”**

Tal y como se ha descrito a lo largo de la introducción anterior, la arquitectura que se pretende desarrollar como solución al problema presentado en este proyecto, incluye una base de datos centralizada, en la que se clasifica toda la información existente en los sitios Web de las diferentes organizaciones que realizan una actividad relacionada con el patrimonio.

En este apartado se pretende presentar la estructura de dicha base de datos así como sus conceptos y atributos. Para ello se mostrarán los modelos entidad-relación y relacional por este orden. Además de esto se realizará una breve descripción de cada una de las tablas de la base de datos.

A continuación se presenta el modelo entidad-relación de la base de datos centralizada.

### 6.2.1.1.1 MODELO ENTIDAD-RELACIÓN BASE DE DATOS “SITIOS WEB”

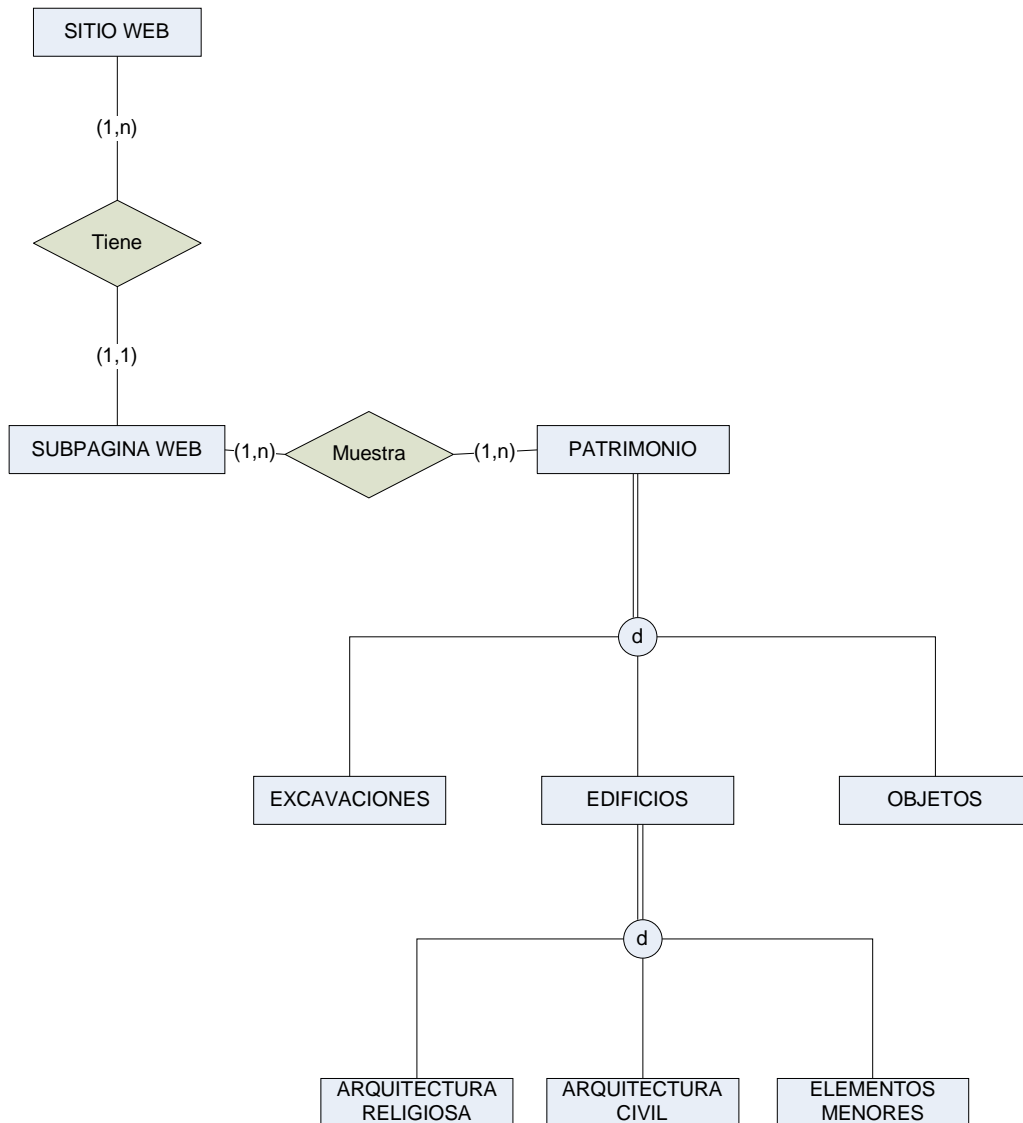


Figura 23. Modelo entidad-relación de la base de datos centralizada.

Tal y como se puede apreciar en la Figura 23, en la base de datos existen dos relaciones de Especialización/Generalización, en las cuales, en una de ellas la superclase (el concepto general) es “Patrimonio” y sus subclases (conceptos especializados) son “Excavaciones”, “Edificios” y “Objetos”. En la otra relación de este tipo existente en el modelo entidad-relación de la base de datos centralizada el concepto general o superclase es el concepto “Edificios” mientras que sus subclases o especializaciones son “Arquitectura Religiosa”, “Arquitectura Civil” y “Elementos Menores”. Cabe citar que existen diferentes tipos de relaciones de Especialización en función de las restricciones que se le apliquen a dicha relación. En el caso de este proyecto, se puede afirmar que existe una restricción de desunión, la cual especifica que las subclases de la especialización deben estar separadas. Esto significa que una

ocurrencia de la entidad puede ser miembro de cómo máximo una de las subclases de la especialización. Como se puede observar en la Figura 23 que representa el diagrama Entidad-Relación, ambas relaciones de especialización presentan un círculo, en donde se aprecia una “d”, que indica que en esa relación se da una restricción de desunión. La otra restricción que aparece en las dos relaciones de especialización de este diseño de base de datos, es la restricción de totalidad. Una restricción de especialización total especifica que cada ocurrencia de entidad de la superclase debe ser miembro de alguna subclase de la especialización. En el caso que aquí se abarca, se puede asegurar que tanto cualquier instancia de patrimonio como de edificio debe ser “excavaciones”, “edificios” u “objetos”, o “arquitectura religiosa”, “arquitectura civil” o “elementos menores” respectivamente

Con el fin de ampliar la explicación del diseño de la base de datos a continuación se presenta el modelo relacional de la base de datos.

#### **6.2.1.1.2 MODELO RELACIONAL BASE DE DATOS “SITIOS WEB”**

El paso de un modelo Entidad-Relación a modelo relacional se consigue llevando a cabo una serie de reglas, las cuales no vamos a comentar, ya que consideramos que no son relevantes para el desarrollo de este proyecto. Sin embargo, en el modelo Entidad-Relación de este proyecto aparecen dos relaciones de especialización, algo que no suele ser muy habitual, y que requiere de explicación. Para conseguir el paso del modelo Entidad-Relación al modelo Relacional cuando existe una relación de Especialización, se llevará a cabo lo siguiente. Se creará para cada subclase una relación con los atributos de la superclase más los atributos propios, donde la clave primaria será la de la superclase. Esta opción sólo es válida para las especializaciones con restricción de totalidad y desunión, como es nuestro caso, ya que si una ocurrencia de la superclase no pertenece a ninguna de las subclases se pierde, y si pertenece a más de una, sus datos aparecen de forma redundante en más de una relación. Para diferenciar las claves primarias y las claves externas, las primeras, sus atributos aparecerán en negrita y subrayados, mientras que las segundas se identificarán por tener su atributo subrayado y debajo de dicho atributo las siglas CE (Clave Externa) junto con la tabla a la cual hace referencia.

### Tabla PÁGINA WEB

<u>Dirección_Web</u>	Descripcion	Nombre
----------------------	-------------	--------

### Tabla SUBPAGINA WEB

<u>Direccion_Web</u>	Descripcion	Disponible_en_Web	Disponible_en_Organización	PaginaWeb
----------------------	-------------	-------------------	----------------------------	-----------

CE Pagina Web

### Tabla EXCAVACIONES

<u>Cod_patri</u>	Nombre	Periodo	Tipo_datos	Municipio	Poblacion	Provincia	Tipo_excavacion	<u>Direccion_Web</u>
------------------	--------	---------	------------	-----------	-----------	-----------	-----------------	----------------------

CE Subpagina  
Web

### Tabla OBJETOS

<u>Cod_patri</u>	Nombre	Periodo	Tipo_datos	Municipio	Poblacion	Provincia	Tipo_objeto	<u>Direccion_Web</u>
------------------	--------	---------	------------	-----------	-----------	-----------	-------------	----------------------

CE Subpagina  
Web

### Tabla ARQUITECTURA RELIGIOSA

<u>Cod_patri</u>	Nombre	Periodo	Tipo_datos	Municipio	Poblacion	Provincia	Tipo_arquitectura_religiosa	<u>Direccion_Web</u>
------------------	--------	---------	------------	-----------	-----------	-----------	-----------------------------	----------------------

CE Subpagina  
Web

### Tabla ARQUITECTURA CIVIL

<u>Cod_patri</u>	Nombre	Periodo	Tipo_datos	Municipio	Poblacion	Provincia	Tipo_arquitectura_civil	<u>Direccion_Web</u>
------------------	--------	---------	------------	-----------	-----------	-----------	-------------------------	----------------------

CE Subpagina  
Web

### Tabla ELEMENTOS MENORES

<u>Cod_patri</u>	Nombre	Periodo	Tipo_datos	Municipio	Poblacion	Provincia	Tipo_elementos_menores	<u>Direccion_Web</u>
------------------	--------	---------	------------	-----------	-----------	-----------	------------------------	----------------------

CE Subpagina  
Web

### 6.2.1.1.3 RELACIONES DE LA BASE DE DATOS

- **Tabla Página Web**

En esta tabla se almacenarán los datos descriptivos relevantes de las páginas Web de las diferentes organizaciones, como son la dirección Web de la página de cada organización, junto con el nombre de la organización y junto con una breve descripción del contenido de la página.

- **Tabla Subpágina Web**

La tabla Subpágina Web alojará en sus registros aquellos datos relacionados con la información descriptiva de las diferentes subpáginas existentes dentro de un sitio Web de una organización determinada. Se almacenará por cada subpágina, su dirección Web, la cual será su clave primaria, su descripción, que ayudará al usuario a conocer de antemano la información que se va a encontrar en dicha página, y su disponibilidad en la Web en el lugar físico en el que se encuentra la organización. La disponibilidad en Web recoge, como su propio nombre indica, aquello que el usuario puede encontrar en dicha subpágina, sobre determinada información, mientras que la información en organización recoge la información que recoge sobre un determinado tema la organización en su ubicación física.

#### **Relación de especialización**

Tal y como se ha explicado anteriormente la transformación a modelo Relacional ha dado como consecuencia una clasificación atípica de las tablas. Las tablas que aparecen a continuación son subclases de la superclase Patrimonio, por lo que todas ellas almacenarán los atributos que aquí se mencionan. Nombre del patrimonio, puede ser de la excavación, de un objeto u edificio (a partir de ahora, y durante este punto Patrimonio se refiere tanto a excavaciones como a objetos como a edificios), Periodo en el que se llevo a cabo ese patrimonio, Tipo\_datos que recoge esta información, Municipio, Población y Provincia en la cual se encuentra dicho Patrimonio. Además incluye un atributo, que realiza las funciones de clave externa, y se denomina Dirección\_Web.

- **Tabla Excavaciones**

Esta tabla almacena aquellos registros del Patrimonio que sean clasificados como “Excavaciones”. Además de los atributos mencionados anteriormente, almacena el tipo de excavación.

- **Tabla Objetos**

Esta tabla almacena aquellos registros del Patrimonio que sean clasificados como “Objetos”. Además de los atributos mencionados anteriormente, almacena el tipo de objeto.

## **Relación de especialización**

Como ya se conoce la base de datos que se pretende desarrollar cuenta con dos relaciones de especialización. Una de ellas se ha explicado anteriormente, y la restante es la que se explica a continuación. Antes de comenzar, mencionar que esta relación de especialización se encuentra inmersa en otra relación de especialización, la otra existente en el diseño de la base de datos. En este caso es el concepto general Edificios el que se divide en tres especializaciones como son “arquitectura religiosa”, “arquitectura civil” y “elementos menores”. Al estar inmersa en otra relación de especificación, cada tabla alberga los atributos de la relación anterior, mientras que de esta nueva relación no recoge ninguno, ya que carece de ellos.

- **Tabla Arquitectura Religiosa**

Esta tabla almacena aquellos registros del Patrimonio que sean clasificados como “Edificios” y dentro de “Edificios” como “Arquitectura Religiosa”. Además de los atributos mencionados anteriormente, almacena el tipo de Arquitectura Religiosa.

- **Tabla Arquitectura Civil**

Esta tabla almacena aquellos registros del Patrimonio que sean clasificados como “Edificios” y dentro de “Edificios” como “Arquitectura Civil”. Además de los atributos mencionados anteriormente, almacena el tipo de Arquitectura Civil.

- **Tabla Elementos Menores**

Esta tabla almacena aquellos registros del Patrimonio que sean clasificados como “Edificios” y dentro de “Edificios” como “Elementos Menores”. Además de los atributos mencionados anteriormente, almacena el tipo de Elementos Menores.

### **6.2.1.2 BASE DE DATOS CENTRALIZADA “ADMINISTRADORES”**

Además de construir una base de datos en la cual se almacene toda la información existente en la página Web de las diferentes organizaciones y relacionada con el patrimonio, se considera conveniente desarrollar otra base de datos, en la cual queden registrados todos y cada uno de los movimientos que se produzcan en la base de datos “sitios Web”. Esta no es, sino una forma de controlar en cada momento qué movimiento se produce, quién es la persona que realiza ese movimiento... con el fin de que en el caso que suceda algo extraño con la base de datos de la aplicación, se pueda obtener, consultando la base de datos “Administradores”, quién es el encargado de dicho suceso. Para el diseño de esta base de datos, al igual que con la base de datos “sitios Web” se ha desarrollado un modelo Entidad-Relación, el cual aparece en la Figura 24, y a partir de él se ha construido el modelo Relacional, que se puede observar en el apartado 6.2.1.2.2. Cabe citar que en el modelo Entidad-Relación, para diferenciar las claves primarias y las claves externas, en las primeras, sus atributos aparecerán en negrita y subrayados, mientras que en las segundas se identificarán por tener su atributo subrayado y debajo de dicho atributo las siglas CE (Clave Externa) junto con la tabla a la cual hace referencia.

#### 6.2.1.2.1 MODELO ENTIDAD-RELACIÓN BASE DE DATOS “ADMINISTRADORES”.

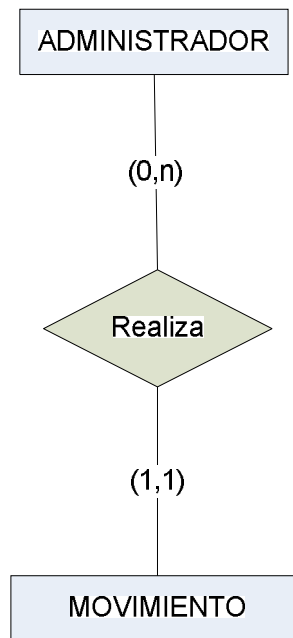


Figura 24. Modelo Entidad-Relación de la base de datos “Movimientos”

A diferencia de la base de datos “sitios Web”, la base de datos “Movimientos”, tal y como se observa en la Figura 24, únicamente dispone de dos tuplas y una relación entre ambas. Sin embargo estas dos tuplas nos permitirán almacenar abundante información acerca de los movimientos que se realizan en la base de datos “sitios Web”, ya sea a través de la aplicación o directamente sobre ella.

Con el fin de ampliar la explicación del diseño de la base de datos a continuación se presenta el modelo relacional de la base de datos.

#### 6.2.1.2.2 MODELO RELACIONAL BASE DE DATOS “ADMINISTRADORES”

Al ser una modelo Entidad-Relación simple, el paso de dicho modelo a modelo relacional se consigue de manera sencilla, únicamente llevando a cabo una serie de reglas, las cuales, tal y como se ha comentado anteriormente no vamos a explicar. A continuación se muestra el modelo Relacional de la base de datos “Administrador”.



### Tabla ADMINISTRADOR

<u>Id</u>	Nombre	Nick	Password	Teléfono	Correo
-----------	--------	------	----------	----------	--------

### Tabla MOVIMIENTOS

<u>Codigo_movimiento</u>	Id	Accion	Fecha
--------------------------	----	--------	-------

CE Administrador

### 6.2.1.2.3 RELACIONES DE LA BASE DE DATOS

- **Tabla Administrador**

En esta tabla se almacena toda la información relacionada con el administrador, es decir su nombre, su Nick. Si utiliza, su contraseña, su teléfono y su correo. Estas dos últimas se han incluido porque se consideran útiles para tener la posibilidad de ponerse en contacto con cualquier administrador de manera inmediata.

- **Tabla Movimientos**

La tabla Movimientos alojará en sus registros los movimientos que se producen en la base de datos “sitios Web”, concretamente almacena el código de movimiento, como simple clave principal, el id, atributo externo que hace referencia al administrador, la acción que se desarrolla en la base de datos, y la fecha en el que se produce ese movimiento o acción.

## 6.2.2 INTERFACES DE LA APLICACIÓN

Aunque, tal y como sea comentado en puntos anteriores, el sistema que almacena la clasificación de datos consta de dos bases de datos, una de ellas la propia base de datos que recoge la información, y la otra, aquella que almacena los movimientos que se producen en la primera, la aplicación que trabaje, en constante funcionamiento, contra ambas bases de datos será la misma, por lo que, obviamente, sus interfaces serán las mismas en muchos casos, y similares en otros tantos.

La aplicación sobre la que el administrador de la base de datos va a tener la posibilidad de mantener actualizada el contenido de la base de datos que reúne información sobre el patrimonio recogido en las diferentes páginas Web, constará de un número determinado de páginas Web (extensión .php3), aproximadamente unas 30. Dichas páginas Web estarán divididas en dos directorios principales, lo cuales serán “administradores” y “sitios web”. En el primer directorio se almacenarán las páginas cuyo contenido esté relacionado, con el alta y baja de administradores de la base de datos, o con modificaciones de los datos de cualquier administrador. Por el contrario, en el segundo directorio se encontrarán alojadas aquellas páginas que supongan la realización de alguna acción sobre el contenido de la base de datos “sitios Web”, ya sea añadir o eliminar información sobre un sitio Web o, bien, añadir o eliminar el propio sitio Web.

Además de esta clasificación principal, existirá un index.php3, que será la primera página que se presente al administrador de la página Web una vez de que se haya “logueado” en el sistema, es decir haya autenticado su identidad, y en la cual el administrador tendrá la opción de acceder a la página principal, bien de administradores, o bien, de sitios web, que se encuentra en su correspondiente directorio. Citar que dentro de cada directorio, es posible encontrar otros subdirectorios que contengan imágenes, etc.

Las interfaces que se van a desarrollar para permitir al administrador de las bases de datos interactuar con el sistema deben cumplir los siguientes requisitos:

- **Interfaces amigables.** El sistema debe disponer de interfaces agradables a la vista, y de sencilla navegación. Este requisito parece que carece de importancia, sin embargo es muy importante si se requiere que el sistema sea utilizado con asiduidad, ya que si el encargado de utilizarla considera que la aplicación es pesada o desagradable en cuanto a estética, es probable que en un futuro no muy lejano deje de utilizarla.
- **Utilidad y funcionalidad.** Obviamente es importante que el sistema sea capaz de realizar todo aquellos que el usuario necesita llevar a cabo.
- **Consistencia del portal entre los distintos navegadores.** No es el requisito más importante, sin embargo se considera necesario que al menos funcione en los dos navegadores más utilizados, Mozilla Firefox y Microsoft Explorer.

### **6.2.3 PERFILES DE LA APLICACIÓN**

A petición del cliente, y con el fin de posibilitar el acceso limitado a diversas funcionalidades del sistema, se pretende desarrollar dos perfiles diferentes: administrador, usuario.

En principio, y mientras el cliente no indique lo contrario, las funcionalidades para ambos perfiles serán similares, teniendo en cuenta los puntos que se indican a continuación.

- El administrador tendrá poder absoluto en la aplicación. Un administrador no tiene restricciones de ningún tipo.
- El usuario no tendrá potestad para crear nuevos individuos en el sistema, ya sean administradores o usuarios.
- El usuario tampoco podrá modificar los datos de ningún administrador ni usuario inclusive sus datos salvo su contraseña que sí podrá ser modificada.
- El sistema no le otorga la posibilidad al usuario de eliminar a ningún miembro del sistema, inclusive a sí mismo.

Salvo estas restricciones ambos perfiles tendrán las mismas potestades en la aplicación.

#### **6.2.4 ARQUITECTURA DE QUE DA SOPORTE A LA BASE DE DATOS QUE CLASIFICA DATOS SOBRE LA INFORMACIÓN EXISTENTE EN LAS PÁGINAS WEB.**

Como se viene comentando a lo largo de la descripción del proyecto, la aplicación que se desarrolla consiste en dos partes una de ellas consistente en realizar una aplicación que consiga mantener actualizada la información relacionada con el patrimonio y que ha sido reunida de diferentes sitios Web y almacenada en una fuente de datos, y una segunda parte, más general que consiste en realizar una aplicación de búsqueda orientada sobre diversas bases de datos entre las que se encuentra la base de datos que almacena información sobre el patrimonio recogido en los diferentes sitios Web. En este punto lo que se pretende es presentar el diseño que se ha realizado para la construcción de la aplicación encargada de prestar servicios de mantenimiento de la información acerca del patrimonio que ha sido clasificada y almacenada en la base de datos “sitios Web”.

La aplicación, tal y como se observa en la Figura 25, consta de tres capas lógicas completamente diferenciadas entres sí. Estas capas son capa de presentación, capa de lógica de negocio y capa de acceso a datos. La capa de presentación contiene las interfaces de usuario, o lo que es lo mismo, es la capa en la que se introducen y muestran los datos. La capa de lógica de negocio se encarga de procesar los datos y las peticiones de los usuarios, es por esto que se suele referir a esta capa como el cerebro de la aplicación. Por último, la capa de acceso a datos se encarga de recuperar y almacenar los datos que la aplicación requiere para su correcto funcionamiento.

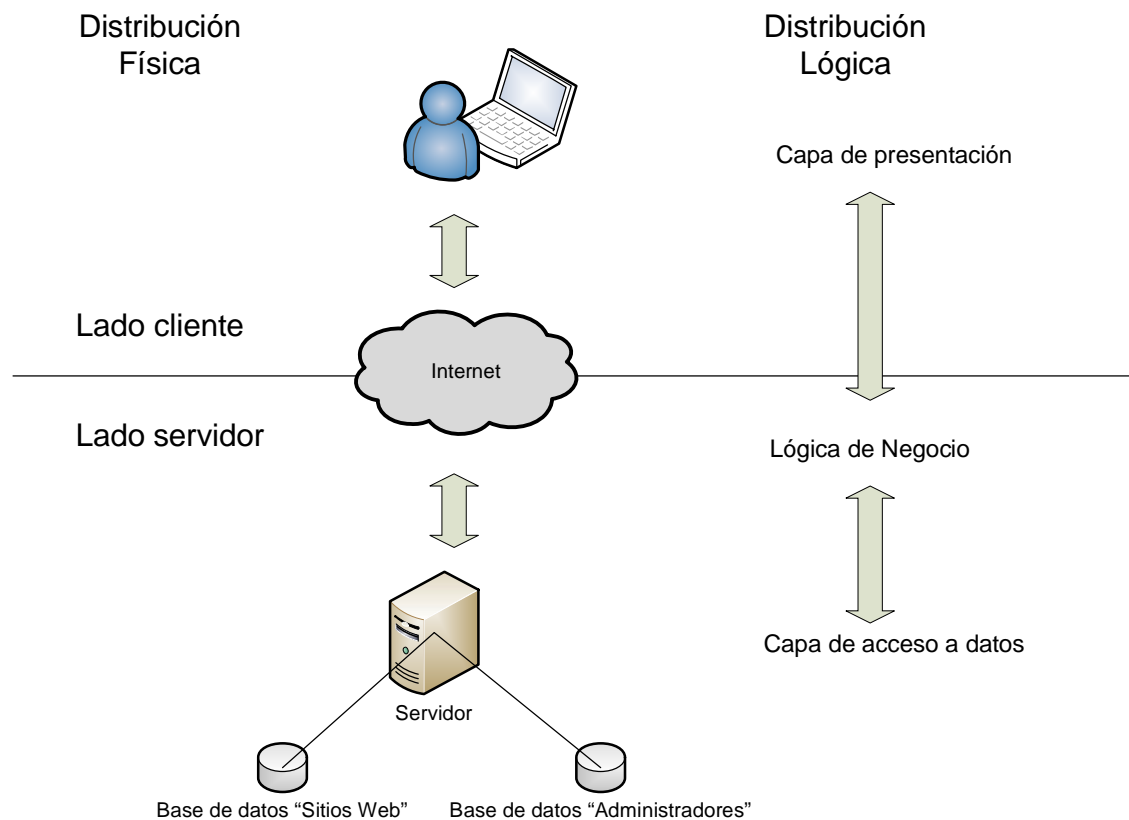


Figura 25. Arquitectura de la aplicación que trabaja contra las bases de datos “sitios Web” y “administradores”

Por otra parte, la aplicación está dividida en dos capas físicas, lo que implica que las tres capas lógicas comentadas anteriormente se ejecutan en dos máquinas físicas diferentes. Por una parte, el lado cliente alberga la capa lógica denominada “capa de presentación”, y por otra parte el servidor alberga las capas lógicas “lógica de negocio” y “capa de acceso a datos”.

#### 6.2.4.1 CAPA DE PRESENTACIÓN

La Capa de Presentación está situada físicamente en la parte cliente. Su función consiste en ofrecer al usuario las interfaces que le permitan interactuar con el sistema. Las interfaces recogen las peticiones de los clientes y muestran los resultados que son procesados internamente en el servidor. A pesar de que tiempo atrás se consideraba a la capa de presentación un simple acompañante a la aplicación que se desarrollaba, hoy en día, las empresas cada vez invierten más dinero en conseguir aplicaciones amigables, sencillas e innovadoras. Por lo que se debe considerar que las interfaces que se utilizan para interactuar con el usuario deben tener una serie de características o requisitos. Dichos requisitos han sido comentados anteriormente en el apartado 6.4.2.

#### **6.2.4.2 LÓGICA DE NEGOCIO**

La Lógica de Negocio, es considerada en numerosas ocasiones como el “motor de la aplicación”, esto es así, debido a que en esta capa se procesan todos los métodos que se desarrollan en la aplicación, salvo aquellos relacionados con el acceso a base de datos. En la aplicación que aquí se presenta, esta será la capa en donde residan los métodos que procesen datos, apoyándose únicamente en la capa de Presentación y en la capa de Acceso Datos para interactuar con el usuario, y para interactuar con la base de datos respectivamente. En la fase de implementación se presentarán los métodos que en esta capa se desarrollan.

#### **6.2.4.3 CAPA DE ACCESO A DATOS**

En esta capa en donde se encuentran las bases de datos de la aplicación, en este caso la base de datos “sitios Web” y “Administradores”, y los métodos que son utilizados para acceder a esas bases de datos. En cuanto a las bases de datos de la aplicación, ambas han sido explicadas detalladamente a lo largo de esta memoria (“sitios Web”-apartado 6.2- y “Administradores”-apartado 6.3-). Los métodos utilizados, y desarrollados para el acceso a los datos serán descritos posteriormente durante el apartado de implementación de la aplicación.

#### **6.2.5 FUNCIONALIDADES DEL SISTEMA**

En este apartado se definirán todas las funcionalidades que se pretenden desarrollar en la aplicación que va a servir de soporte a la base de datos que almacenará información sobre las páginas Web de las organizaciones e instituciones que realizan actividades con el patrimonio. Para definir las de una forma exhaustiva se ha hecho uso de diagramas de flujo, y diagramas de flujo de eventos. Además su explicación se completa con una descripción de cada una de las funcionalidades.

Esta exposición de las funcionalidades no hace más que completar el contenido y finalidad de las mismas que ya se inició en la fase de análisis con los casos de uso.

Como bien se ha explicado en el apartado 6.2.3 en la aplicación existen dos perfiles, administrador y usuario. Debido a que el administrador tiene potestad absoluta para desarrollar las funcionalidades que desee, y el usuario tiene restricción a la hora de llevar a cabo algunas de ellas, a lo largo de este apartado se hace referencia a las funcionalidades que pueden ser desarrolladas por el perfil administrador. Por tanto, de las funcionalidades que se describen a continuación habrá alguna que el usuario no pueda llevar a cabo. Para conocer cuáles son éstas, es suficiente con revisar los casos de uso que mencionábamos anteriormente, en los cuales se describe de forma detallada las funcionalidades que el usuario no puede llevar a cabo, o simplemente observar la información que aparece en la propia descripción de la funcionalidad, en donde indica los actores que pueden desarrollar la funcionalidad.

### 6.2.5.1 ACCEDER AL SISTEMA

**Nombre:** Acceder al sistema

**Actores:** Administrador o Usuario

**Descripción:** El administrador intenta acceder al sistema introduciendo su Nick y su contraseña. Si los datos introducidos son correctos, accede al sistema, si no, éste le muestra un mensaje de error.

**Diagrama de Flujo de la funcionalidad**

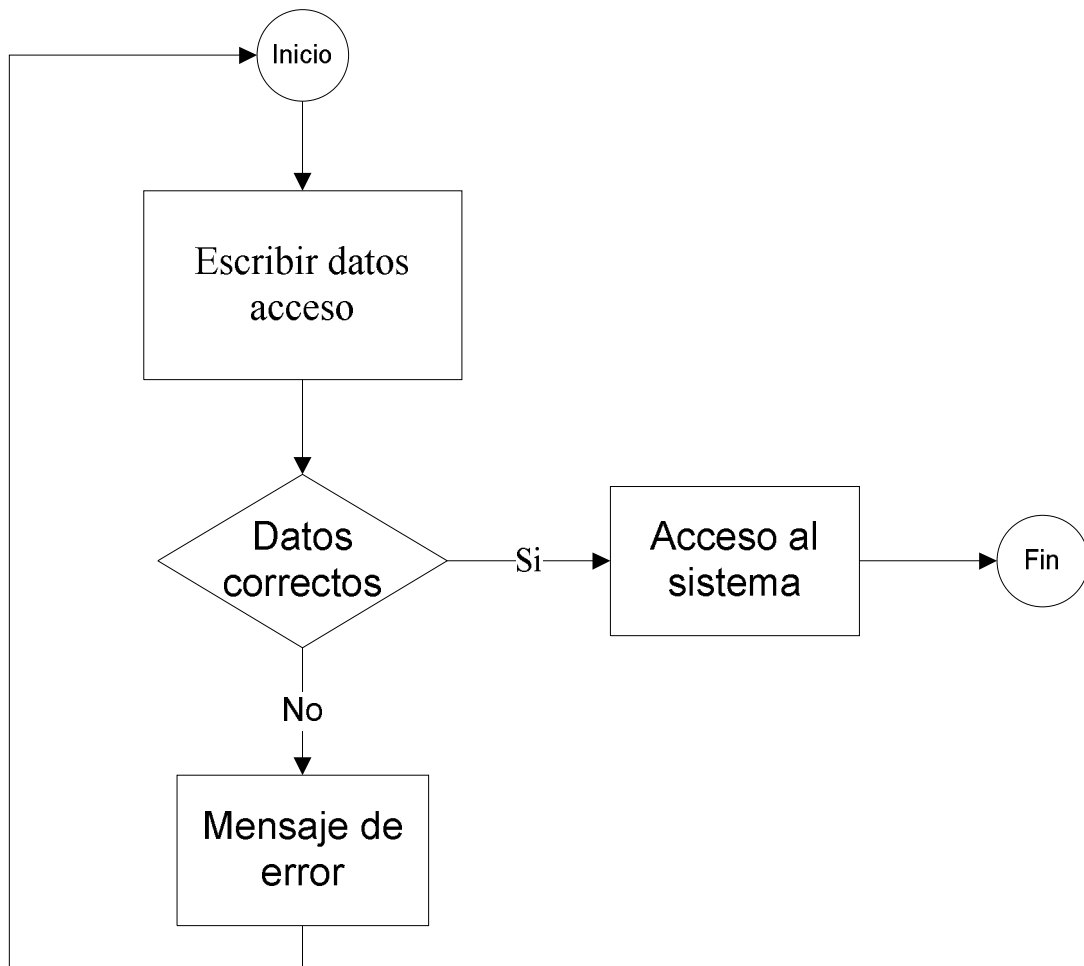


Figura 26. Diagrama de Flujo de la funcionalidad "Acceder al sistema".

## **Diagrama de Flujo de Eventos**

Flujo principal:

1. El administrador introduce los datos para acceder al sistema
2. El administrador accede al sistema. Fin

Flujo alternativo: Datos introducidos incorrectos

1. Los datos introducidos no son correctos. El usuario no puede acceder al sistema.
2. El sistema le muestra el mensaje de error correspondiente. Fin.



### 6.2.5.2 NUEVO SITIO WEB

**Nombre:** Insertar nuevo sitio Web

**Actores:** Administrador

**Descripción:** El administrador introduce un nuevo sitio Web en la aplicación. Si los datos son correctos el nuevo sitio Web queda registrado. Sin embargo, puede suceder que la dirección Web del sitio Web que se pretende introducir pertenezca a otro sitio Web que ya está registrado, en este caso el registro del nuevo sitio Web no es posible. Otro de los problemas que puede acontecer al insertar un nuevo sitio Web es que los campos que se van insertar no estén rellenos, por lo que el sistema obligará al administrador a rellenarlos.

#### Diagrama de Flujo de la funcionalidad

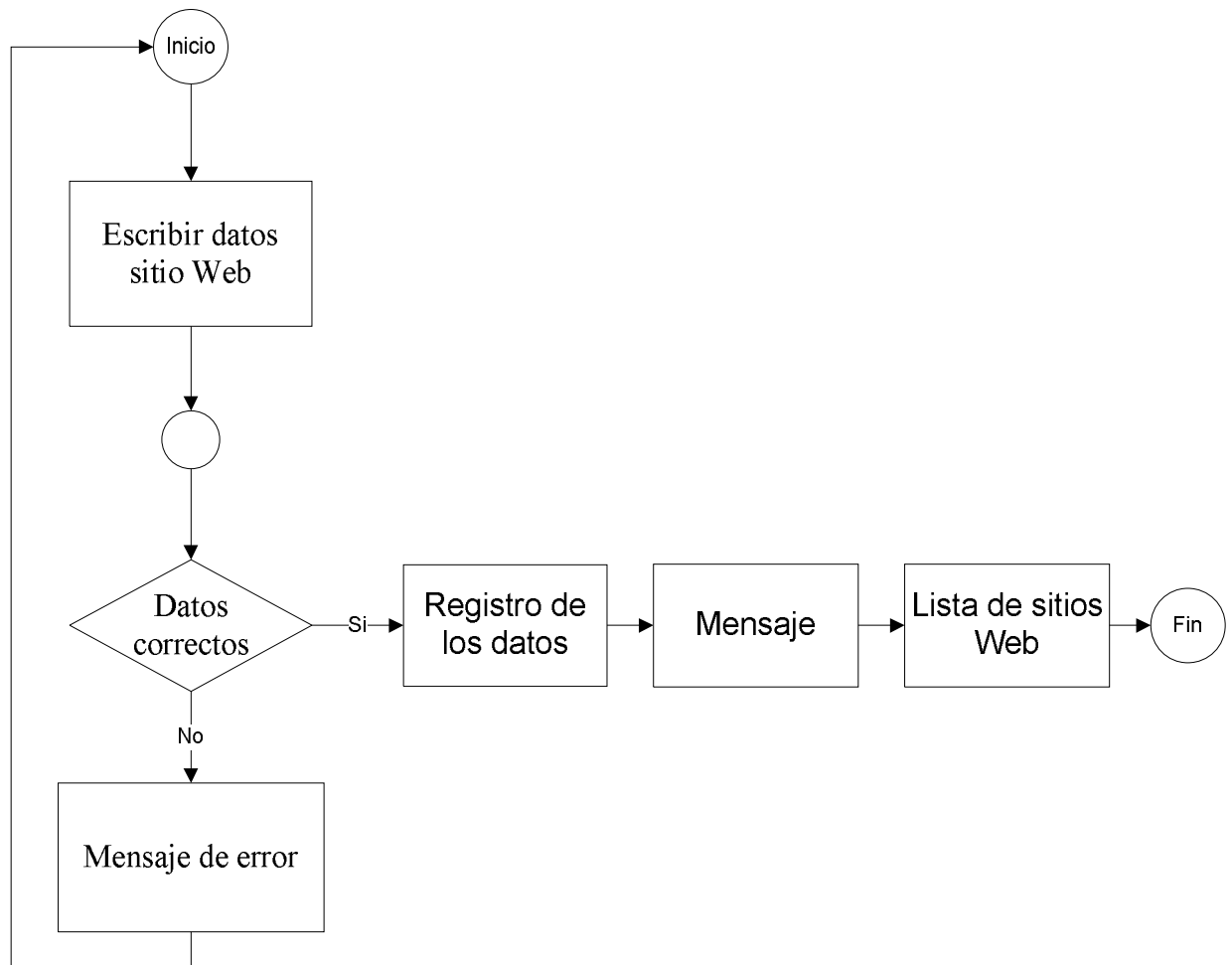


Figura 27. Diagrama de Flujo de la funcionalidad "Nuevo sitio Web".

## **Diagrama de Flujo de Eventos**

Flujo principal:

3. El administrador introduce los datos del nuevo sitio Web
4. El sistema registra el sitio Web.
5. El sistema muestra el mensaje de inserción.
6. El sistema muestra la lista de los sitios Web. Fin.

Flujo alternativo: Datos introducidos incorrectos

3. Los datos introducidos no son correctos.
4. El sistema le muestra el mensaje de error correspondiente. Fin.

### 6.2.5.3 MODIFICAR SITIO WEB

**Nombre:** Modificar sitio Web

**Actores:** Administrador

**Descripción:** El administrador modifica un sitio Web de la aplicación. El administrador obtiene de la base de datos una lista desplegable con la dirección de los sitios Web registrados, selecciona el que desee y modifica sus datos. Si los datos son correctos el sitio Web queda modificado. Sin embargo, puede suceder que el administrador pretenda modificar la dirección Web del sitio Web con una dirección Web que ya pertenezca a otro sitio Web que esté registrado, en este caso la modificación del sitio Web no es posible. Otro de los problemas que puede acontecer al introducir los nuevos datos del sitio Web es que los campos que se van insertar no estén rellenos, por lo que el sistema obligará al administrador a rellenarlos.

#### Diagrama de Flujo de la funcionalidad

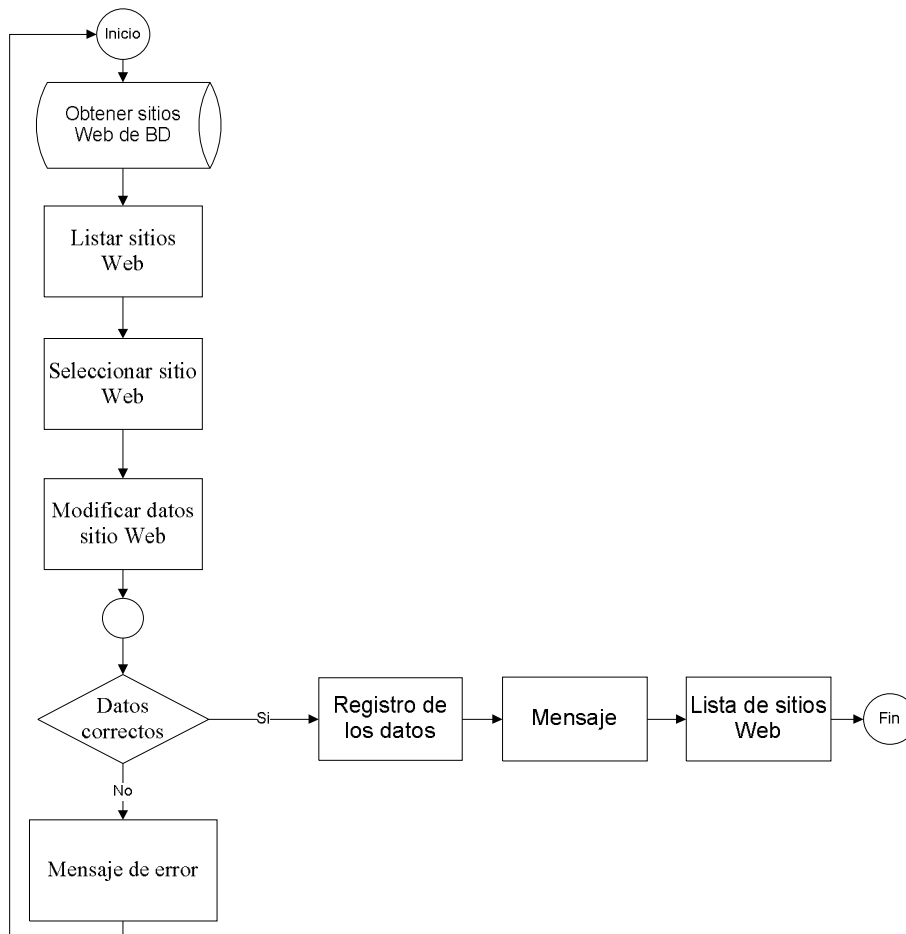


Figura 28. Diagrama de Flujo de la funcionalidad “Modificar sitio Web”.

## **Diagrama de Flujo de Eventos**

Flujo principal:

1. El administrador consulta al sistema los sitios Web que están registrados.
2. El administrador selecciona uno de esos sitios Web.
3. El administrador modifica los datos que crea convenientes.
4. El sistema registra la modificación.
5. El sistema muestra el mensaje.
6. El sistema muestra la lista de los sitios Web. Fin.

Flujo alternativo: No existen sitios Web registrados.

1. El sistema no muestra ningún sitio Web. Fin.

Flujo alternativo: Datos introducidos incorrectos.

3. Los datos introducidos no son correctos.
4. El sistema le muestra el mensaje de error correspondiente. Fin.

#### 6.2.5.4 ELIMINAR SITIO WEB

**Nombre:** Eliminar sitio Web

**Actores:** Administrador

**Descripción:** El administrador elimina un sitio Web de la aplicación. El administrador obtiene de la base de datos una lista con la dirección de los sitios Web registrados junto con una imagen que le permite eliminar el sitio Web que cree oportuno. El administrador selecciona uno, el sistema le muestra un mensaje de confirmación, éste lo acepta y elimina el sitio Web.

#### Diagrama de Flujo de la funcionalidad

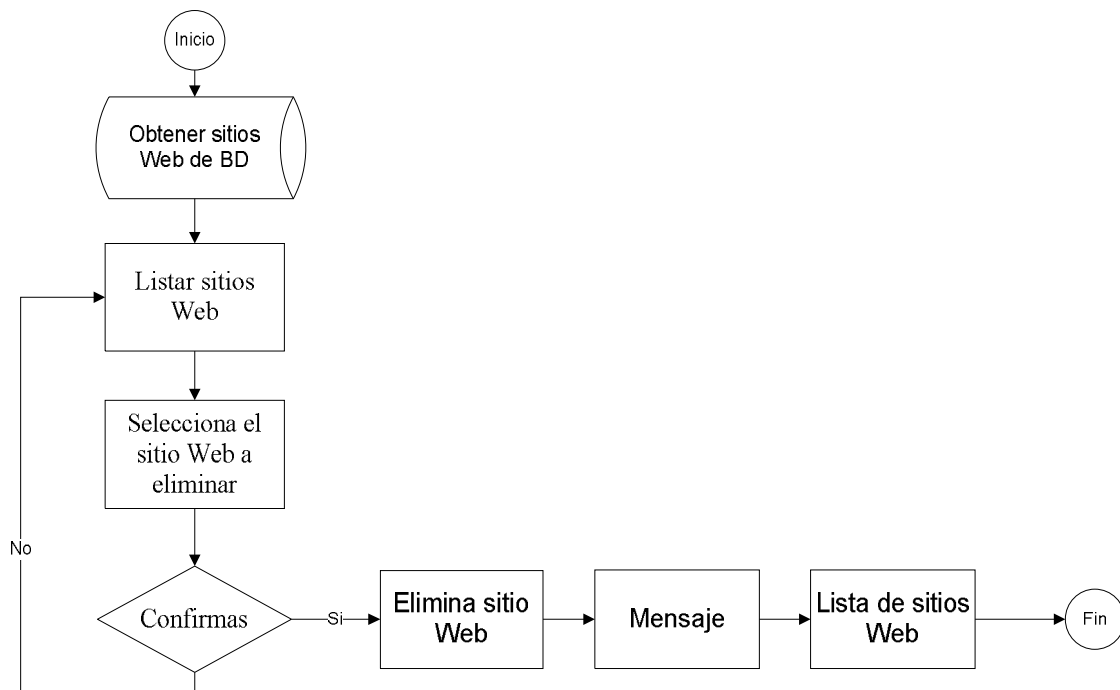


Figura 29. Diagrama de Flujo de la funcionalidad “Eliminar sitio Web”.

## **Diagrama de Flujo de Eventos**

Flujo principal:

1. El administrador consulta al sistema los sitios Web que están registrados.
2. El administrador selecciona el sitio Web que desea eliminar
3. El administrador confirma que quiere eliminar el sitio Web.
4. El sistema elimina el sitio Web.
5. El sistema muestra el mensaje.
6. El sistema muestra la lista de los sitios Web. Fin.

Flujo alternativo: No existen sitios Web registrados.

1. El sistema no muestra ningún sitio Web. Fin.

Flujo alternativo: El administrador no confirma la eliminación.

1. El administrador rechaza la eliminación.
2. El sistema le muestra el mensaje de error correspondiente. Fin.

### 6.2.5.5 NUEVA SUB-PÁGINA WEB

**Nombre:** Insertar nueva sub-página Web

**Actores:** Administrador

**Descripción:** El administrador introduce una nueva subpágina Web perteneciente a un sitio Web de la aplicación. El administrador obtiene una lista desplegable con todos los sitios Web que están registrados, selecciona aquel en el que desee introducir una nueva subpágina Web e introduce los datos. Si los datos son correctos la nueva subpágina Web queda registrada. Sin embargo, puede suceder que la dirección Web de la subpágina Web que se pretende introducir pertenezca ya a otra subpágina de dicho sitio Web, en este caso el registro de la nueva subpágina Web no es posible. Otro de los problemas que puede acontecer al insertar una nueva subpágina Web es que los campos que se van a insertar no estén rellenos, por lo que el sistema obligará al administrador a rellenarlos.

#### Diagrama de Flujo de la funcionalidad

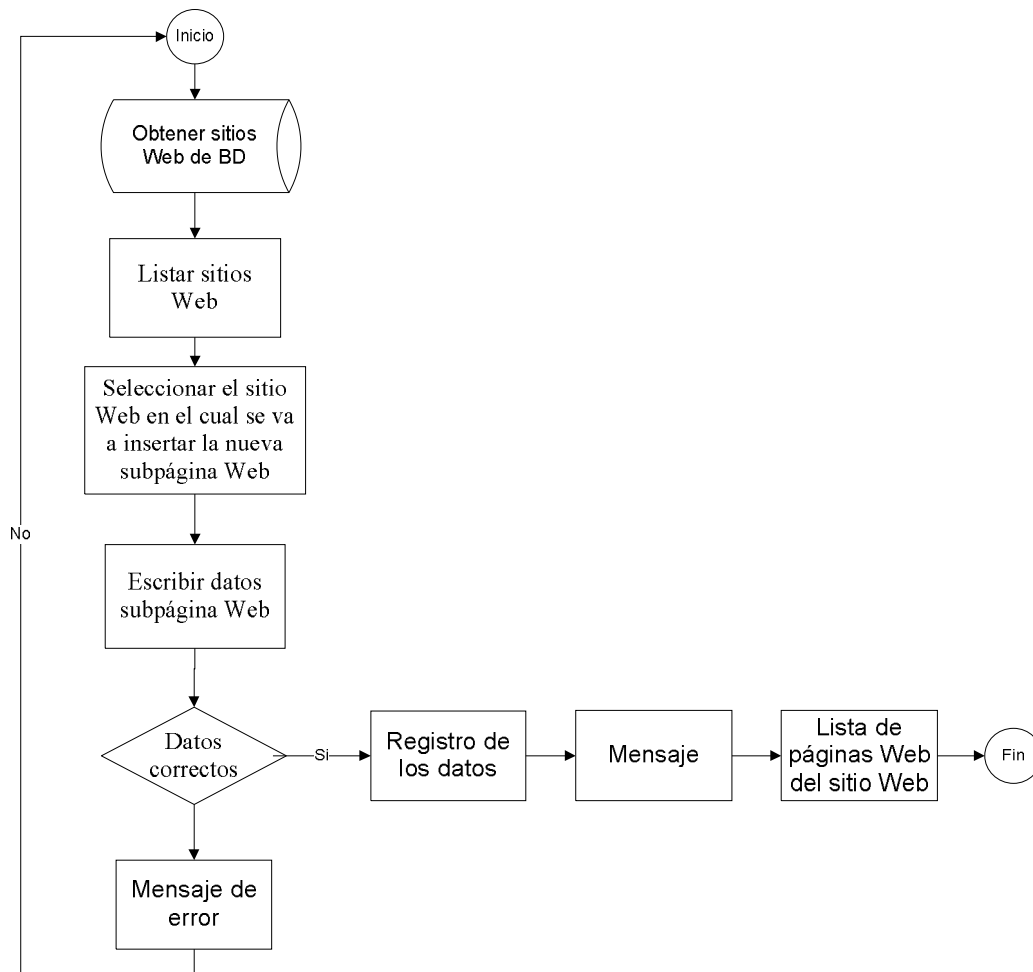


Figura 30. Diagrama de Flujo de la funcionalidad “Nueva subpágina Web”.

## **Diagrama de Flujo de Eventos**

Flujo principal:

1. El administrador consulta al sistema los sitios Web que están registrados.
2. El administrador selecciona el sitio Web en el que desea introducir la subpágina Web.
3. El administrador escribe los datos de la nueva subpágina Web.
4. El sistema registra la inserción.
5. El sistema muestra el mensaje de inserción.
6. El sistema muestra la lista de las subpáginas Web del sitio Web a la que ésta ahora pertenece. Fin.

Flujo alternativo: No existen sitios Web registrados.

1. El sistema no muestra ningún sitio Web. Para poder insertar una subpágina Web debe haber insertado al menos un sitio Web. Fin.

Flujo alternativo: Datos introducidos incorrectos

3. Los datos introducidos no son correctos.
4. El sistema le muestra el mensaje de error correspondiente. Fin.



### **6.2.5.6 MODIFICAR SUB-PÁGINA WEB**

**Nombre:** Modificar sub-página Web

**Actores:** Administrador

**Descripción:** El administrador modifica una subpágina Web perteneciente a un sitio Web de la aplicación. El administrador obtiene una lista desplegable con todos los sitios Web que están registrados, selecciona aquel del que desee modificar una subpágina Web y modifica sus datos. Si los datos son correctos la subpágina Web queda correctamente modificada. Sin embargo, puede suceder que la nueva dirección Web de la subpágina Web que se pretende modificar pertenezca ya a otra subpágina de dicho sitio Web, en este caso la modificación de la subpágina Web no es posible. Otro de los problemas que puede acontecer al modificar una subpágina Web es que alguno de los campos que se pretenden modificar no estén rellenos, por lo que el sistema obligará al administrador a rellenarlos.

## Diagrama de Flujo de la funcionalidad

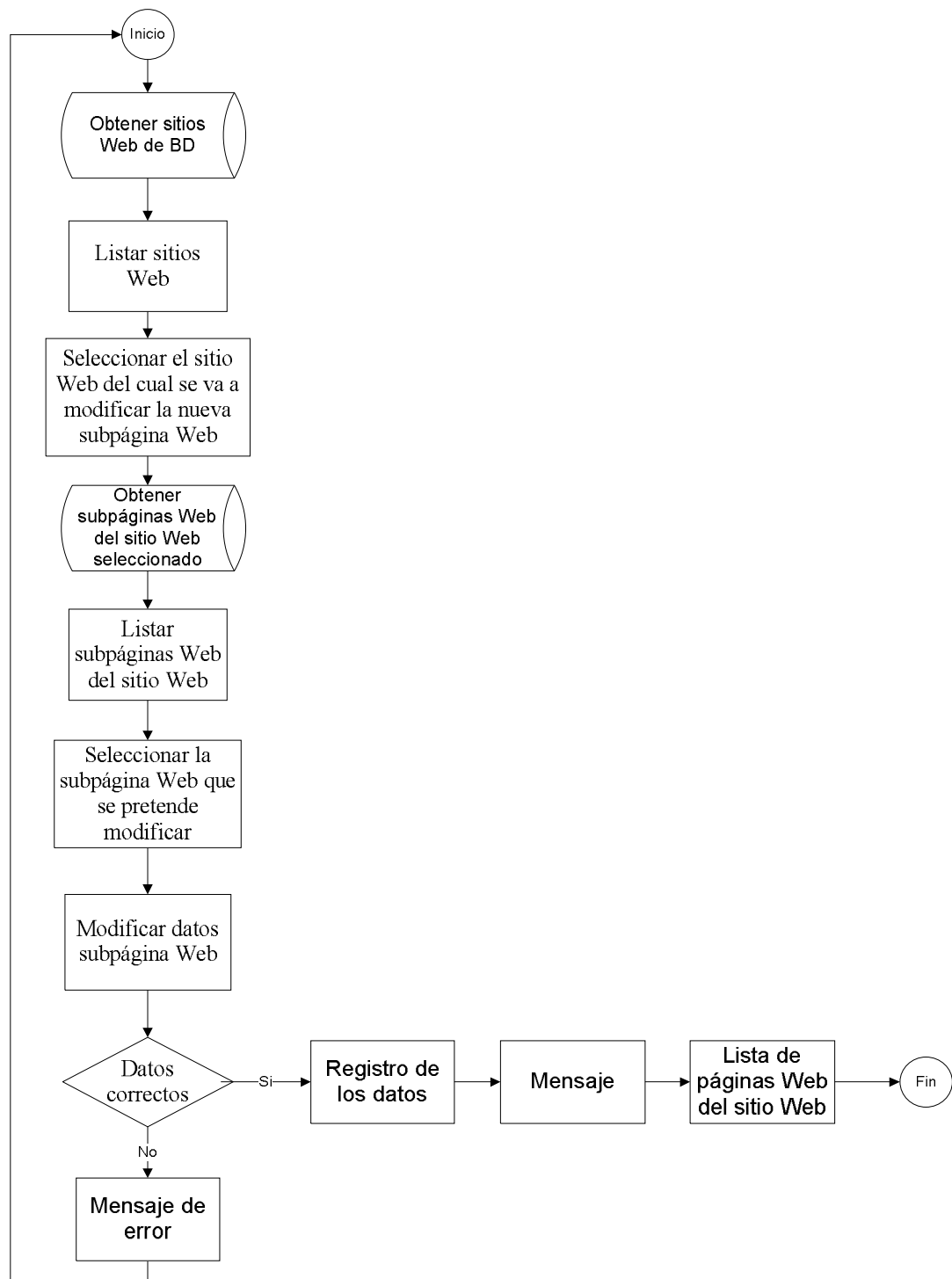


Figura 31. Diagrama de Flujo de la funcionalidad “Modificar subpágina Web”.

## Diagrama de Flujo de Eventos

Flujo principal:

1. El administrador consulta al sistema los sitios Web que están registrados.
2. El administrador selecciona el sitio Web en el que se encuentra la subpágina Web que desea modificar.
3. Automáticamente el sistema consulta las subpáginas Web del sitio Web seleccionado en la lista desplegable anterior.
4. El administrador selecciona la subpágina Web que desea modificar.
5. El administrador modifica los datos que desee de la subpágina Web.
6. El sistema registra la modificación de los datos.
7. El sistema muestra el mensaje de modificación.
8. El sistema muestra la lista de las subpáginas Web del sitio Web del cual se ha modificado la subpágina. Fin.

Flujo alternativo: No existen sitios Web registrados.

1. El sistema no muestra ningún sitio Web. Fin.

Flujo alternativo: No existen subpáginas Web en el sitio Web seleccionado.

3. El sistema no muestra ninguna subpágina Web. La subpágina Web que se pretende modificar no existe en el sitio Web seleccionado. Fin.

Flujo alternativo: Datos introducidos incorrectos

5. Los datos introducidos no son correctos.
6. El sistema le muestra el mensaje de error correspondiente. Fin.

### **6.2.5.7 ELIMINAR SUB-PÁGINA WEB**

**Nombre:** Eliminar sub-página Web

**Actores:** Administrador

**Descripción:** El administrador elimina una subpágina Web perteneciente a un sitio Web de la aplicación. El administrador obtiene de la base de datos una lista desplegable con la dirección de los sitios Web registrados. El administrador selecciona aquel sitio Web en el que se encuentra la subpágina Web que desea eliminar. Al seleccionar dicho sitio Web el sistema le muestra en una lista todas las subpáginas Web del mismo junto con una imagen que le permite eliminar el sitio Web que cree oportuno. El administrador selecciona una, el sistema le muestra un mensaje de confirmación, éste lo acepta y elimina la subpágina Web.

## Diagrama de Flujo de la funcionalidad

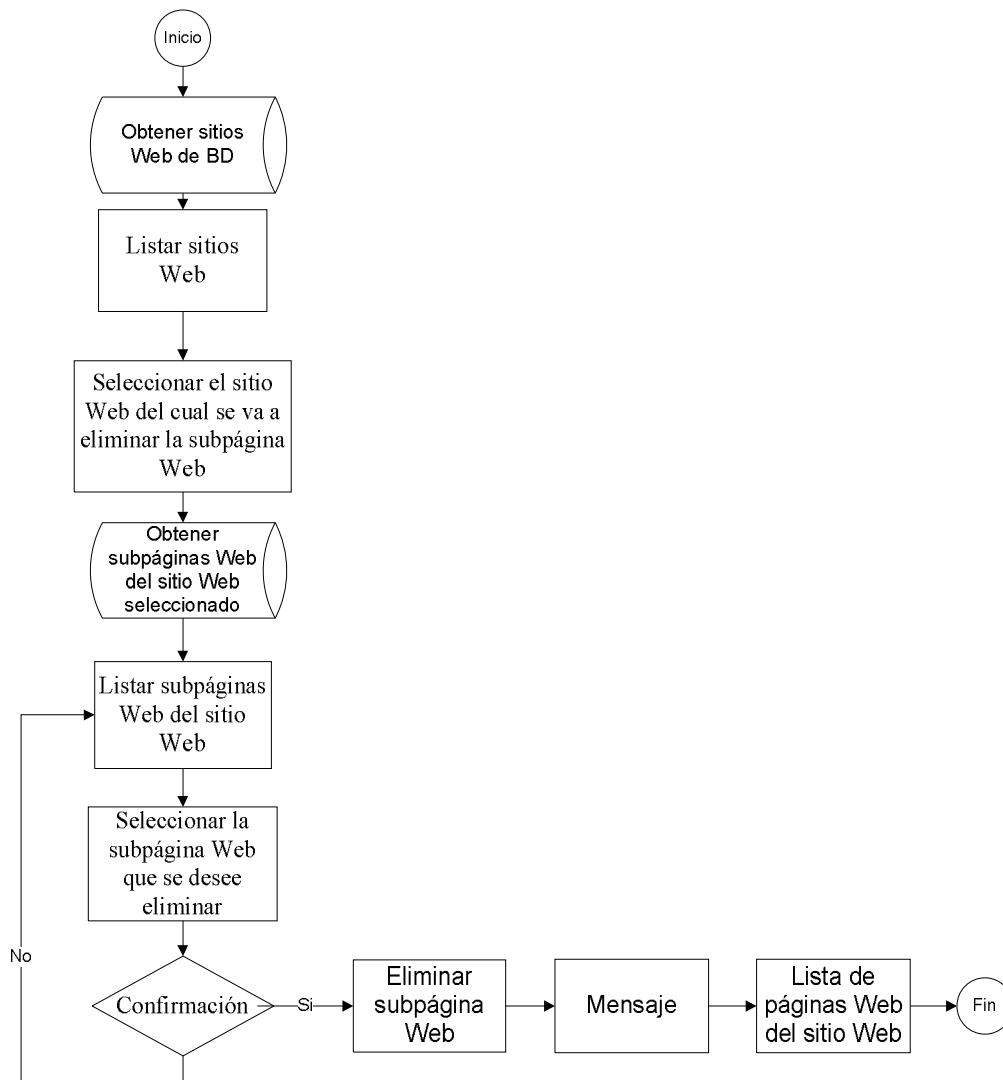


Figura 32. Diagrama de Flujo de la funcionalidad “Eliminar subpágina Web”.

## Diagrama de Flujo de Eventos

Flujo principal:

1. El administrador consulta al sistema los sitios Web que están registrados.
2. El administrador selecciona el sitio Web en el que se encuentra la subpágina Web que desea eliminar.
3. Automáticamente el sistema consulta las subpáginas Web del sitio Web seleccionado en la lista desplegable anterior.

4. El administrador selecciona la subpágina Web que desea eliminar.
5. El administrador confirma que quiere eliminar la subpágina Web seleccionada.
6. El sistema elimina la subpágina Web.
7. El sistema muestra el mensaje de eliminación.
8. El sistema muestra la lista de las subpáginas Web del sitio Web del cual se ha eliminado la subpágina. Fin.

Flujo alternativo: No existen sitios Web registrados.

2. El sistema no muestra ningún sitio Web. Fin.

Flujo alternativo: No existen subpáginas Web en el sitio Web seleccionado.

4. El sistema no muestra ninguna subpágina Web. La subpágina Web que se pretende eliminar no existe en el sitio Web seleccionado. Fin.

Flujo alternativo: El administrador rechaza la eliminación

5. El administrador rechaza la eliminación.
6. El sistema le muestra el mensaje de error correspondiente. Fin.

### **6.2.5.8 INTRODUCIR NUEVO PATRIMONIO**

**Nombre:** Introducir nuevo Patrimonio

**Actores:** Administrador

**Descripción:** El administrador introduce un nuevo elemento patrimonial en una subpágina Web. El administrador obtiene de la base de datos una lista desplegable con la dirección de los sitios Web registrados. El administrador selecciona aquel sitio Web en el que se encuentra la subpágina Web en la que desea introducir un nuevo elemento patrimonial. Al seleccionar dicho sitio Web el sistema le muestra en una lista todas las subpáginas Web del mismo. El administrador selecciona aquella en la que desea introducir el nuevo elemento patrimonial. Posteriormente, rellena los campos y el sistema registra el nuevo elemento patrimonial.

### Diagrama de Flujo de la funcionalidad

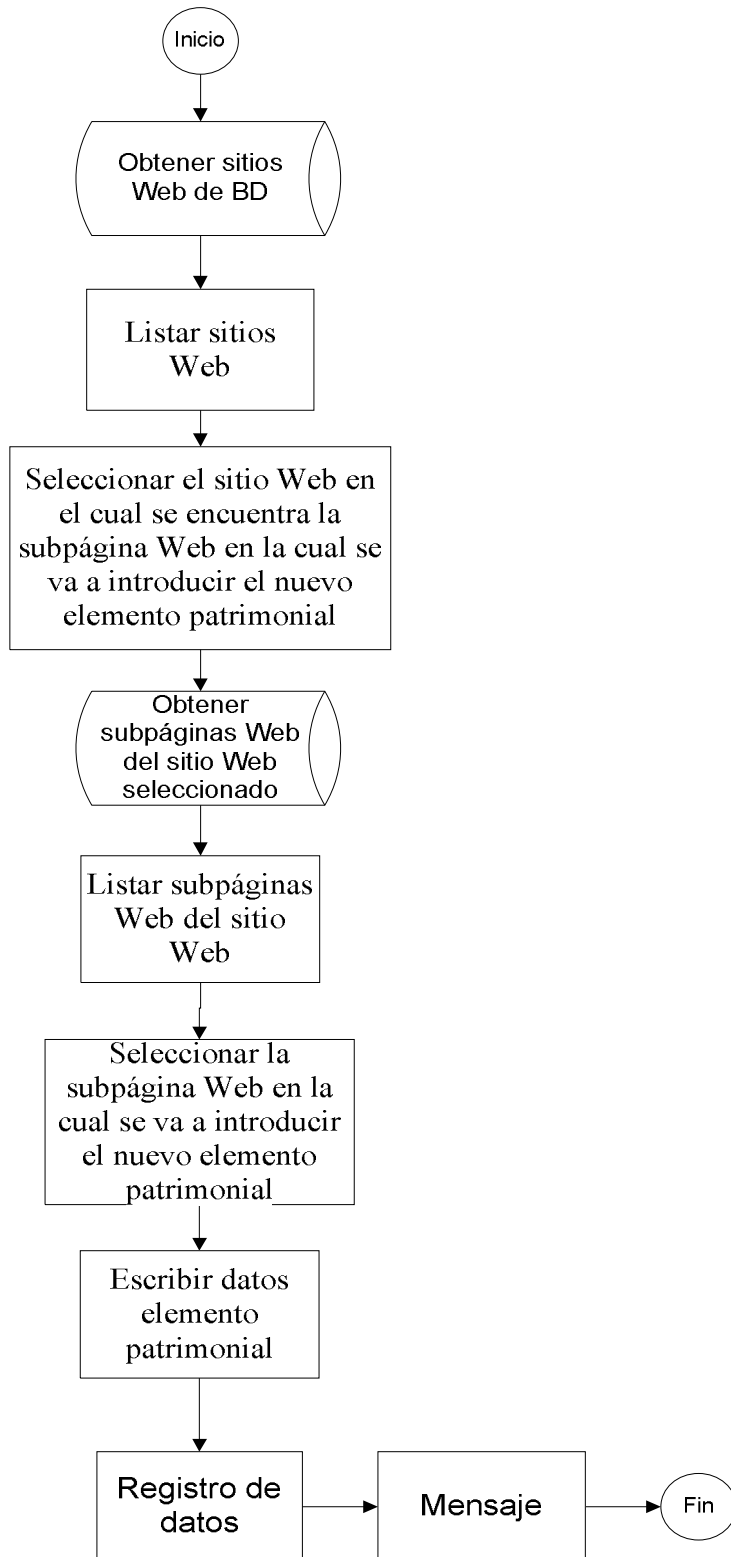


Figura 33. Diagrama de Flujo de la funcionalidad “Introducir nuevo patrimonio”.



## Diagrama de Flujo de Eventos

Flujo principal:

1. El administrador consulta al sistema los sitios Web que están registrados.
2. El administrador selecciona el sitio Web en el que se encuentra la subpágina Web en la cual se va a introducir el nuevo elemento patrimonial.
3. Automáticamente el sistema consulta las subpáginas Web del sitio Web seleccionado en la lista desplegable anterior.
4. El administrador selecciona la subpágina Web en la que pretende introducir el elemento patrimonial.
5. El administrador rellena los campos.
6. El sistema registra el elemento patrimonial.
7. El sistema muestra el mensaje de inserción. Fin

Flujo alternativo: No existen sitios Web registrados.

2. El sistema no muestra ningún sitio Web. Fin.

Flujo alternativo: No existen subpáginas Web en el sitio Web seleccionado.

5. El sistema no muestra ninguna subpágina Web. Fin.

### **6.2.5.9 MODIFICAR PATRIMONIO**

**Nombre:** Modificar Patrimonio

**Actores:** Administrador

**Descripción:** El administrador modifica un elemento patrimonial de una subpágina Web. El administrador obtiene de la base de datos una lista desplegable con la dirección de los sitios Web registrados. El administrador selecciona aquel sitio Web en el que se encuentra la subpágina Web de la que desea modificar un elemento patrimonial. Al seleccionar dicho sitio Web el sistema le muestra en una lista desplegable todas las subpáginas Web del mismo. El administrador selecciona aquella de la que desea modificar el elemento patrimonial. Una vez seleccionada la subpágina el sistema le lista todos los elementos patrimoniales de la misma junto con una imagen que le permite acceder a los datos del elemento patrimonial que pretende modificar. Al seleccionar el elemento que desea modificar, el sistema le muestra los datos del mismo, el administrador modifica los campos que crea oportunos y el sistema registra la modificación del elemento patrimonial.

## Diagrama de Flujo de la funcionalidad

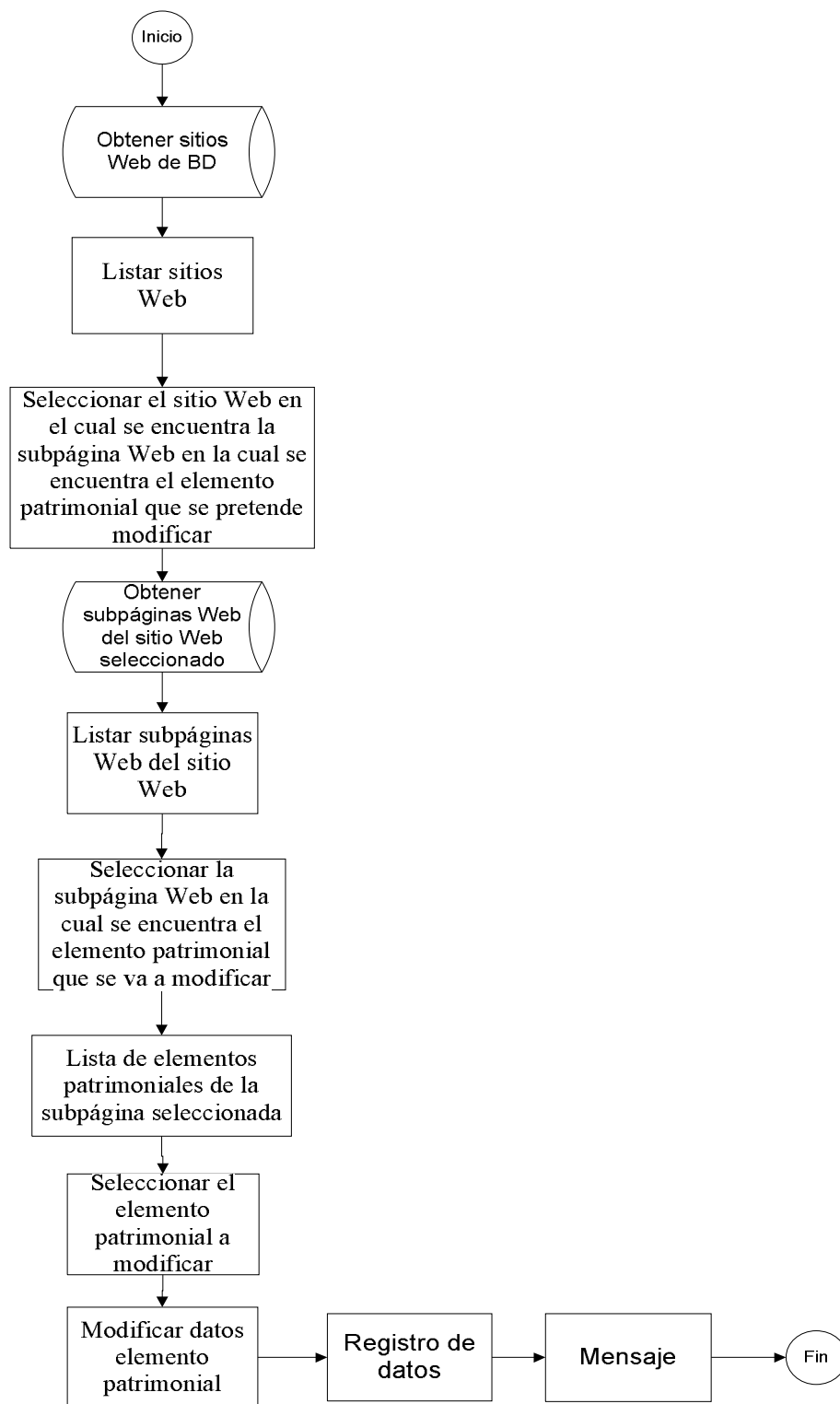


Figura 34. Diagrama de Flujo de la funcionalidad “Modificar patrimonio”.

## **Diagrama de Flujo de Eventos**

Flujo principal:

1. El administrador consulta al sistema los sitios Web que están registrados.
2. El administrador selecciona el sitio Web en el que se encuentra la subpágina Web en la cual está el elemento patrimonial que se pretende modificar.
3. Automáticamente el sistema consulta las subpáginas Web del sitio Web seleccionado en la lista desplegable anterior.
4. El administrador selecciona la subpágina Web en la que se encuentra el elemento patrimonial que se desea modificar.
5. Automáticamente el sistema consulta los elementos patrimoniales de la subpágina Web seleccionada y los muestra en una lista.
6. El administrador selecciona el elemento patrimonial que desee modificar.
7. El administrador modifica los campos que desee.
8. El sistema registra la modificación.
9. El sistema muestra el mensaje de modificación. Fin

Flujo alternativo: No existen sitios Web registrados.

2. El sistema no muestra ningún sitio Web. Fin.

Flujo alternativo: No existen subpáginas Web en el sitio Web seleccionado.

4. El sistema no muestra ninguna subpágina Web. Fin.

Flujo alternativo: No existen elementos patrimoniales en la subpágina Web seleccionada.

6. El sistema no muestra ningún elemento patrimonial. Fin.

#### **6.2.5.10 ELIMINAR PATRIMONIO**

**Nombre:** Eliminar Patrimonio

**Actores:** Administrador

**Descripción:** El administrador elimina un elemento patrimonial de una subpágina Web. El administrador obtiene de la base de datos una lista desplegable con la dirección de los sitios Web registrados. El administrador selecciona aquel sitio Web en el que se encuentra la subpágina Web de la que desea eliminar un elemento patrimonial. Al seleccionar dicho sitio Web el sistema le muestra en una lista desplegable todas las subpáginas Web del mismo. El administrador selecciona aquella de la que desee eliminar el elemento patrimonial. Una vez seleccionada la subpágina el sistema le lista todos los elementos patrimoniales de la misma junto con una imagen que le permite eliminarlo. Al seleccionar el elemento que desea eliminar, el sistema le pedirá una confirmación de eliminación, si ésta es aceptada el sistema eliminará el elemento patrimonial.

## Diagrama de Flujo de la funcionalidad

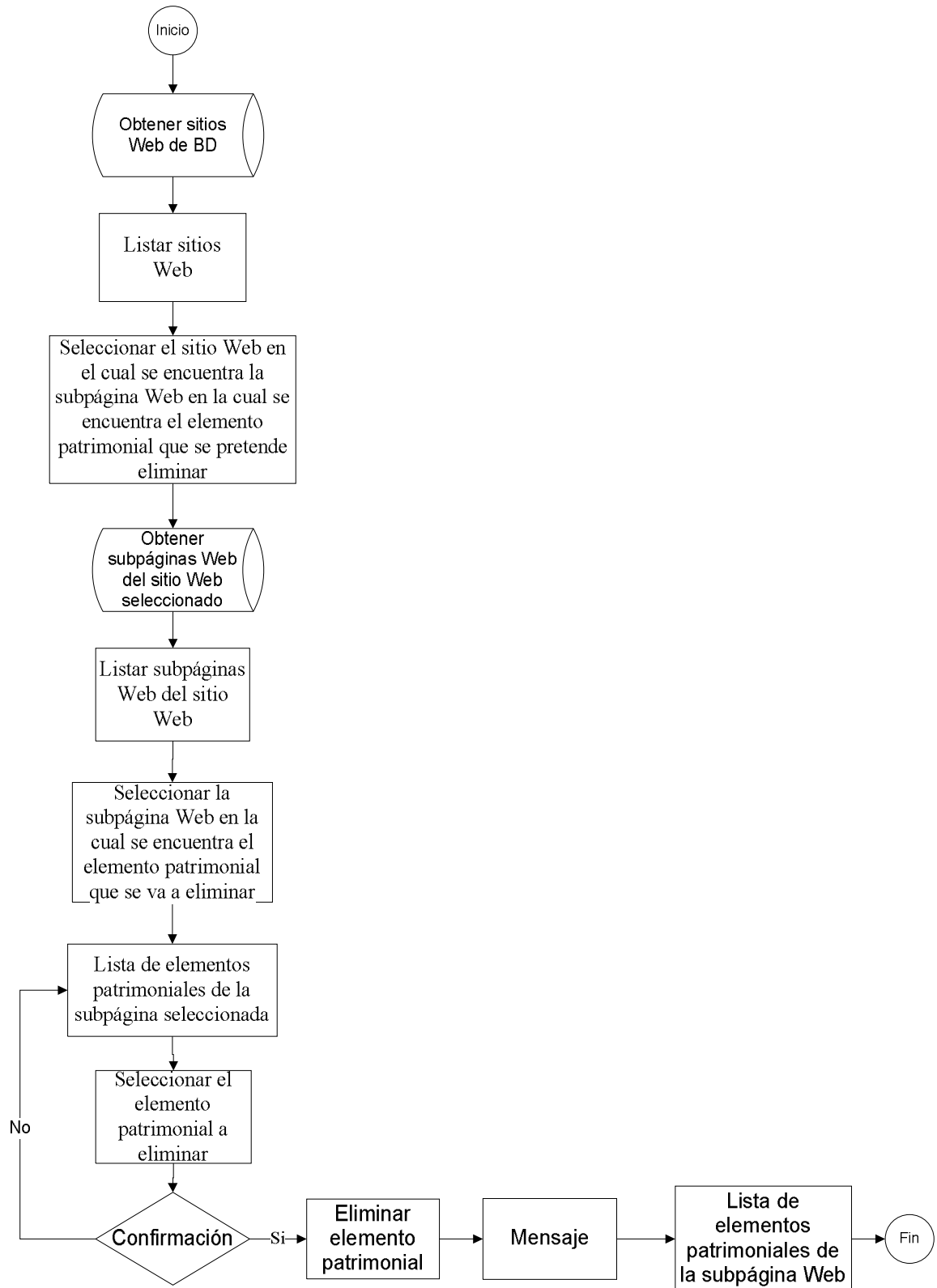


Figura 35. Diagrama de Flujo de la funcionalidad “Eliminar patrimonio”.

## Diagrama de Flujo de Eventos

Flujo principal:

1. El administrador consulta al sistema los sitios Web que están registrados.
2. El administrador selecciona el sitio Web en el que se encuentra la subpágina Web en la cual está el elemento patrimonial que se pretende eliminar.
3. Automáticamente el sistema consulta las subpáginas Web del sitio Web seleccionado en la lista desplegable anterior.
4. El administrador selecciona la subpágina Web en la que se encuentra el elemento patrimonial que se desea eliminar.
5. Automáticamente el sistema consulta los elementos patrimoniales de la subpágina Web seleccionada y los muestra en una lista.
6. El administrador selecciona el elemento patrimonial que desee eliminar.
7. El administrador confirma la eliminación del elemento patrimonial.
8. El sistema elimina el elemento patrimonial.
9. El sistema muestra el mensaje de eliminación. Fin

Flujo alternativo: No existen sitios Web registrados.

2. El sistema no muestra ningún sitio Web. Fin.

Flujo alternativo: No existen subpáginas Web en el sitio Web seleccionado.

4. El sistema no muestra ninguna subpágina Web. Fin.

Flujo alternativo: No existen elementos patrimoniales en la subpágina Web seleccionada.

6. El sistema no muestra ningún elemento patrimonial. Fin.

### 6.2.5.11 INTRODUCIR NUEVO ADMINISTRADOR

**Nombre:** Introducir nuevo Administrador

**Actores:** Administrador

**Descripción:** El administrador introduce un nuevo administrador en el sistema. Rellena los campos y el sistema registra el nuevo administrador. Si los datos son correctos el nuevo administrador queda registrado. Sin embargo, puede suceder que el Nick del administrador que se pretende introducir pertenezca a un administrador que ya está registrado, en este caso el registro del nuevo administrador no es posible. Otro de los problemas que puede acontecer al insertar un nuevo administrador es que los campos que se van insertar no estén rellenos, por lo que el sistema obligará al administrador a rellenarlos.

#### Diagrama de Flujo de la funcionalidad

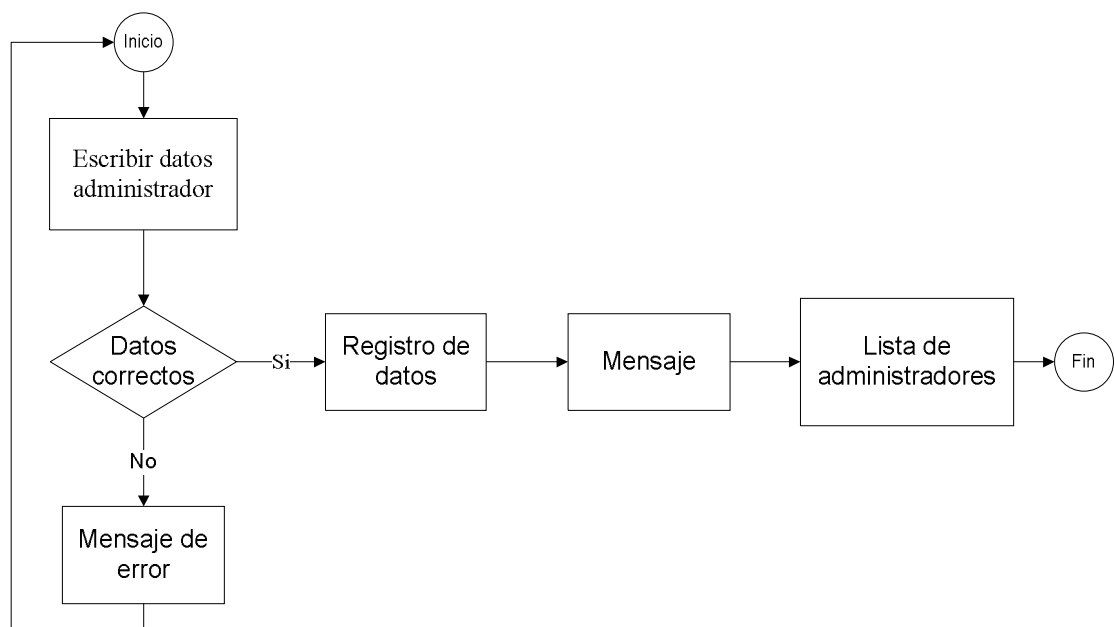


Figura 36. Diagrama de Flujo de la funcionalidad “Introducir nuevo administrador”.



## **Diagrama de Flujo de Eventos**

Flujo principal:

1. El administrador introduce los datos del nuevo administrador
2. El sistema registra el nuevo administrador.
3. El sistema muestra el mensaje de inserción.
4. El sistema muestra la lista de los administradores. Fin.

Flujo alternativo: Los datos no son correctos.

1. Los datos introducidos por el administrador no son correctos.
2. El sistema le muestra el mensaje de error correspondiente. Fin.

### 6.2.5.12 MODIFICAR ADMINISTRADOR

**Nombre:** Modificar Administrador

**Actores:** Administrador

**Descripción:** El administrador modifica los datos de un administrador del sistema. El sistema le muestra una lista desplegable con todos los administradores del sistema, el administrador escoge aquel cuyos datos desea modificar. El sistema le muestra los datos, y éste modifica aquellos que cree convenientes. Si los datos son correctos el administrador queda modificado. Sin embargo, puede suceder que el nuevo Nick que se le pretende asignar al administrador cuyos datos se están modificando pertenezca a otro administrador que ya está registrado, en este caso la modificación de los datos del administrador no es posible. Otro de los problemas que puede acontecer al modificar los datos de un administrador es que los campos que se van insertar no estén rellenos, por lo que el sistema obligará al administrador a rellenarlos.

#### Diagrama de Flujo de la funcionalidad

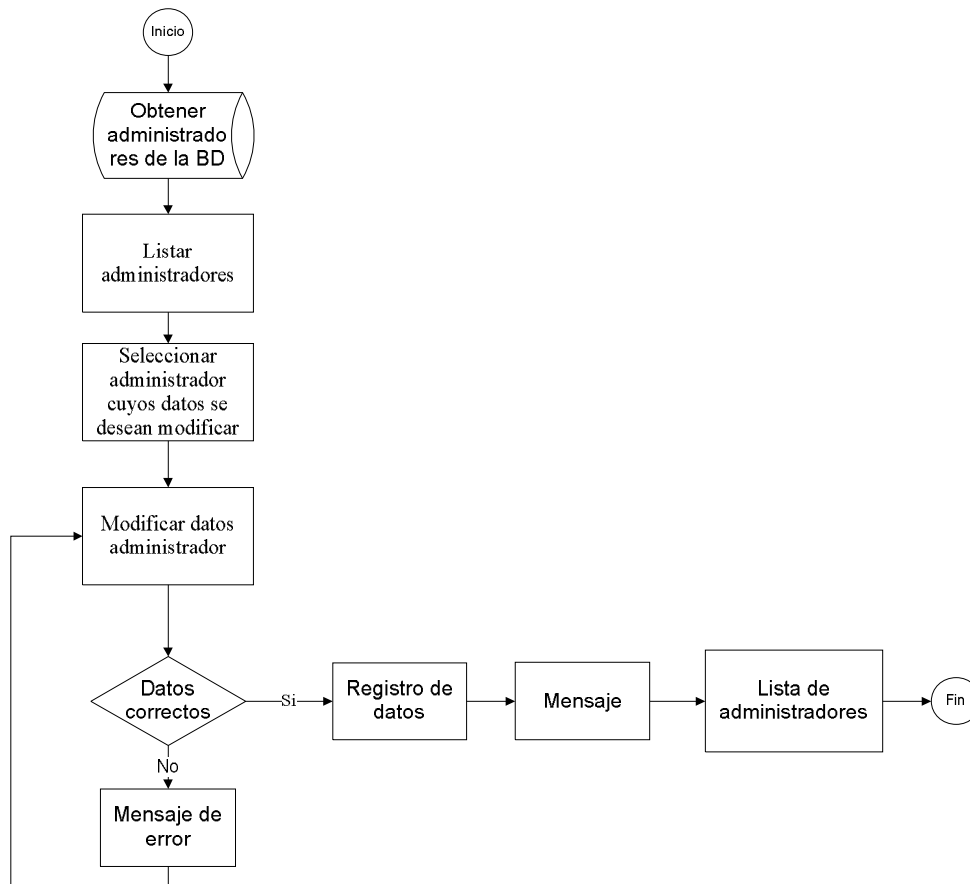


Figura 37. Diagrama de Flujo de la funcionalidad “Modificar Administrador”.

## **Diagrama de Flujo de Eventos**

Flujo principal:

1. El administrador consulta al sistema los administradores que están registrados.
2. El administrador selecciona el administrador que desea modificar.
3. El administrador modifica los datos del administrador que desee.
4. El sistema registra las modificaciones.
5. El sistema muestra el mensaje de modificación.
6. El sistema muestra la lista de los administradores. Fin.

Flujo alternativo: Los datos no son correctos.

3. Los datos introducidos por el administrador no son correctos.
4. El sistema le muestra el mensaje de error correspondiente. Fin.

### 6.2.5.13 ELIMINAR ADMINISTRADOR

**Nombre:** Eliminar Administrador

**Actores:** Administrador

**Descripción:** El administrador elimina un administrador del sistema. El sistema le muestra una lista con todos los administradores del sistema junto con una imagen que le permite eliminarlos. El administrador selecciona la imagen asociada al administrador que desea eliminar. El sistema le obliga a confirmar la eliminación, éste la confirma y el administrador queda eliminado.

**Diagrama de Flujo de la funcionalidad**

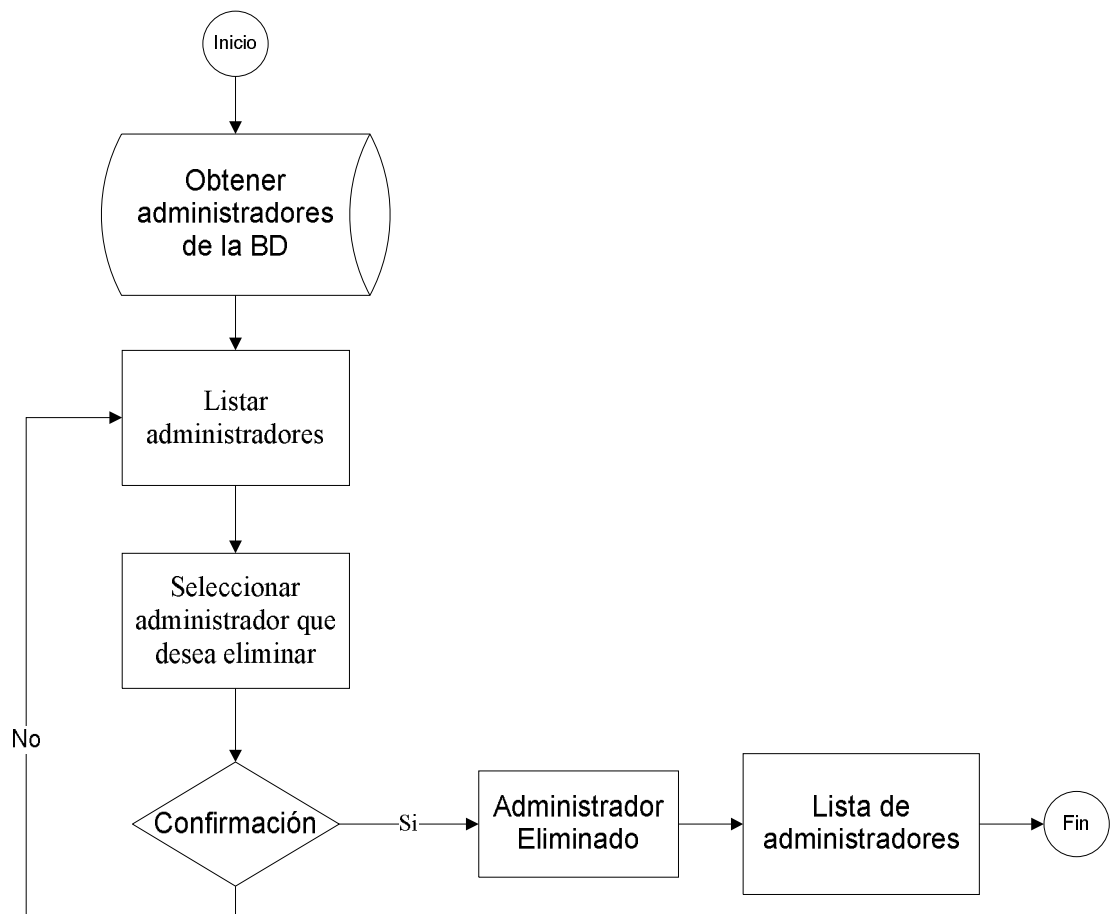


Figura 38. Diagrama de Flujo de la funcionalidad “Eliminar administrador”.

## **Diagrama de Flujo de Eventos**

Flujo principal:

1. El administrador consulta al sistema los administradores que están registrados.
2. El administrador selecciona el administrador que desea eliminar.
3. El administrador confirma la eliminación del administrador.
4. El sistema elimina el administrador.
5. El sistema muestra el mensaje de eliminación.
6. El sistema muestra la lista de los administradores. Fin.

Flujo alternativo: El administrador no elimina.

3. El administrador no confirma la eliminación del administrador. Fin

#### 6.2.5.14 VER MOVIMIENTOS

**Nombre:** Ver Movimientos

**Actores:** Administrador

**Descripción:** El administrador comprueba los movimientos que se han realizado a lo largo de la aplicación en orden cronológico.

**Diagrama de Flujo de la funcionalidad**

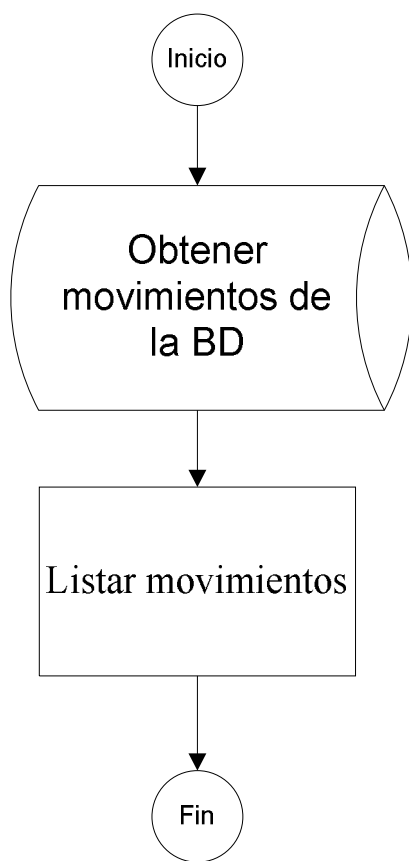


Figura 39. Diagrama de Flujo de la funcionalidad "Ver Movimientos".

**Diagrama de Flujo de Eventos**

Flujo principal:

1. El administrador consulta al sistema los movimientos que están registrados.
2. El sistema muestra dichos movimientos.

### 6.2.5.15 VER ADMINISTRADORES

**Nombre:** Ver Administradores

**Actores:** Administrador

**Descripción:** El administrador comprueba los administradores que participan en el sistema

#### Diagrama de Flujo de la funcionalidad

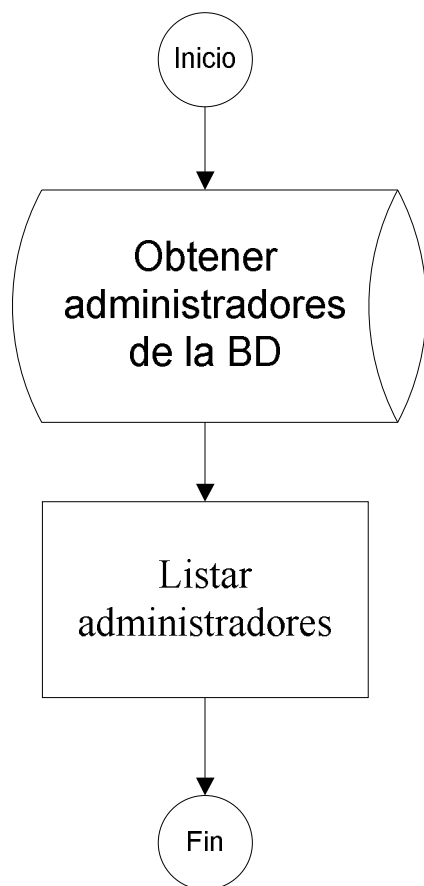


Figura 40. Diagrama de Flujo de la funcionalidad “Ver Movimientos”.

#### Diagrama de Flujo de Eventos

Flujo principal:

1. El administrador consulta al sistema los administradores que están registrados.
2. El sistema muestra dichos administradores.

### 6.2.5.16 MODIFICAR CONTRASEÑA

**Nombre:** Modificar contraseña

**Actores:** Administrador

**Descripción:** El administrador modifica su propia contraseña. Para ello rellena los campos y el sistema, si los datos son correctos, registra la nueva contraseña. Sin embargo, puede suceder que el campo contraseña y el campo confirmar contraseña sean diferentes, en este caso el registro de la nueva contraseña no es posible.

#### Diagrama de Flujo de la funcionalidad

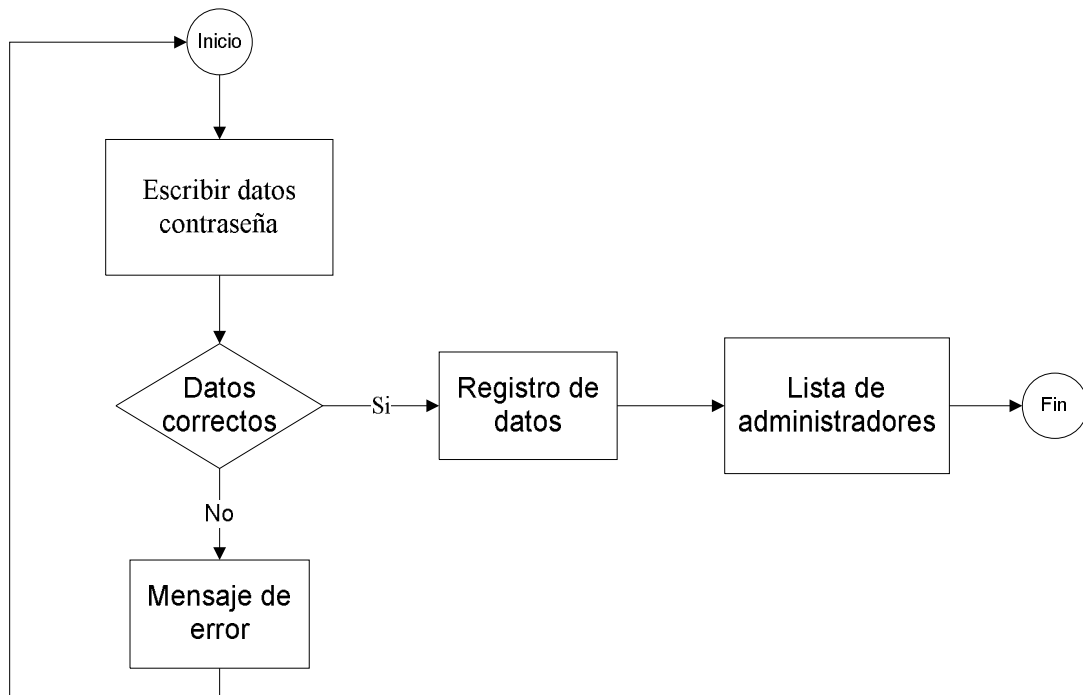


Figura 41. Diagrama de Flujo de la funcionalidad "Modificar subpágina Web".

#### Diagrama de Flujo de Eventos

Flujo principal:

1. El administrador introduce los datos de la nueva contraseña.
2. El sistema registra la nueva contraseña.
3. El sistema muestra la lista de los administradores. Fin.

Flujo alternativo: Los datos no son correctos.



3. Los datos introducidos por el administrador no son correctos. Los campos contraseña y confirmar contraseña no coinciden.
4. El sistema le muestra el mensaje de error correspondiente. Fin.

## 6.3 APLICACIÓN DE BÚSQUEDA ORIENTADA.

### 6.3.1 INTERFAZ DE APLICACIÓN DE BÚSQUEDA ORIENTADA

Fuente: Tesis Doctoral. Arquitectura para Federación de Bases de Datos Documentales basada en Ontologías (“Consultado el 12-XI-2009”)

*“Los sistemas Web son especialmente dependientes de la calidad de la Interfaz de Usuario que presenten. Es bien sabido que los usuarios Web no van a leer ningún manual de utilización y que apenas harán uso de las “ayudas”. Por tanto, es necesario que sepan usar el sistema desde el primer momento y, por ello, las interfaces deben ser muy sencillas e intuitivas.*

*A lo largo de este apartado lo que se pretende abordar es el diseño e implementación de Interfaces de Usuario intuitivas que propicien el éxito de cualquier sistema Web. En este sentido se presentan tres técnicas de diseño de interfaces de usuario que son útiles para construir Interfaces de Usuario (tanto de consulta como de respuesta) a bases de datos en general y, especialmente, a bases de datos similares a las de nuestro sistema.”*

La diferencia entre esta aplicación general del sistema y la aplicación que se pretende realizar para que el administrador pueda mantener actualizada la base de datos “sitios Web”, es que la segunda de estas va estar diseñada para que sea utilizada por los administradores de la base de datos, es decir, por un número muy limitado de personas, sin embargo, la aplicación general que se pretende desarrollar, se trata de una aplicación que va a estar colocada en la Web, y a la cual puede acceder todo aquel que requiera información sobre patrimonio. Con esto se quiere decir, que la importancia de conseguir unas interfaces amigables y funcionales se multiplica en cualquier aplicación que va a ser de libre acceso, ya que probablemente en función del nivel que se consiga en cuanto a interfaces, el éxito de la página puede variar.

Para conseguir el máximo nivel posible en cuanto a interfaces de usuario, se ha investigado acerca de tres técnicas de diseño de interfaces, las cuales serán citadas a continuación, para ser desarrolladas más adelante.

Fuente: Tesis Doctoral. Arquitectura para Federación de Bases de Datos Documentales basada en Ontologías (“Consultado el 12-XI-2009”)

- *“Uso de **Metáforas Cognitivas**. Consiste en construir páginas Web usando escenarios del mundo real, de manera que los elementos representados en la página Web sean similares, tanto en aspecto como en funcionamiento, a los elementos del mundo real. Esta técnica es muy conocida y utilizada, no sólo en Interfaces de Usuario Web, sino en cualquier sistema informático. Un buen ejemplo, es la Calculadora de Windows.*

- *Uso de **Lenguaje Natural Acotado**. Esta técnica consiste en ofrecer al usuario un conjunto de frases (en lenguaje natural) con huecos. El usuario debe elegir las frases en las que está interesado y llenar los huecos. Finalmente, el conjunto de frases seleccionadas con los huecos cubiertos representa la consulta del usuario.*
- ***Aproximación Navegacional**. Esta aproximación es útil tanto en las Interfaces de Consulta como en las de Respuesta. En las Interfaces de Consulta, en vez de presentar al usuario formularios en los que, para expresar condiciones sobre cada atributo, tenga que introducir el valor que ha de tener dicho atributo, la Aproximación Navegacional permite expresar dichas restricciones a través de simples clics de ratón sobre una pantalla (normalmente creada a partir de una Metáfora Cognitiva) en la que están explícitos los posibles valores que puede tomar un atributo o conjunto de atributos. Del mismo modo, al usar esta aproximación, se puede evitar que el usuario introduzca el valor de un cierto atributo si las respuestas se presentan ordenadas por dicho atributo, de modo que el usuario pueda fácilmente navegar por el conjunto de resultados y localizar aquellos de su interés.”*

Con el fin de adentrar al lector un poco más en las técnicas que se han comentado, a continuación se explica el uso de cada una de ellas, así como sus ventajas y desventajas. Para esto nos basaremos en el modo en que han sido utilizadas dichas técnicas en nuestro proyecto.

Son muchas las opciones que se presentan para el diseño de las interfaces de esta aplicación, sin embargo, debido sobre todo a las exigencias del cliente se ha optado en su mayoría por utilizar Lenguaje Natural Acotado, sin embargo con el fin de que el lector pueda entender mejor el uso de las otras dos técnicas vamos ejemplificar cada una de ellas.

### **Uso de Metáforas Cognitivas**

*Esta técnica está basada en el uso de algo conocido por el usuario que es trasladado a otro dominio. En algunos casos está basado en la similitud del aspecto físico y en otros casos en la similitud de los objetivos y tareas que lleva a cabo. Los primeros ejemplos de Metáforas Cognitivas fueron los procesadores de textos usando como metáfora una máquina de escribir o las carpetas o directorios de ficheros que usan una carpeta de archivador como metáfora. Una de las metáforas más conocidas y usadas es la calculadora que está incluida en todos los sistemas operativos Windows. En cuanto a la Web, una de las metáforas más exitosas es el extendido carrito de la compra que se usa en la mayoría –sino en todas– las tiendas online.*



Figura 42. Calculadora de Windows.

Tal y como se ha comentado en el párrafo anterior la calculadora de Windows (Figura 42) es, quizás, el ejemplo más utilizado de Metáfora Cognitiva. En la cual en vez de escribir los números, con los que se desea operar, en la pantalla, dichos números se escriben por sí solos en la misma únicamente con la acción de un “clic” en el número correspondiente. Su uso es completamente igual que el de una calculadora física normal y corriente.

### Lenguaje Natural Acotado

Fuente: Tesis Doctoral. Arquitectura para Federación de Bases de Datos Documentales basada en Ontologías (“Consultado el 13-XI-2009”)

*“Las Interfaces Web se construyen usando desde simples cuadros de texto hasta técnicas mucho más complejas y potentes. Algunos buscadores Web ofrecen la posibilidad de introducir una pregunta que es contestada por el sistema mostrando una lista de páginas Web que, supuestamente, pueden ayudar al usuario a encontrar la respuesta a su pregunta. Es decir, la interfaz está basada en el uso de lenguaje natural para definir la consulta. El uso de lenguaje natural es, indudablemente, muy potente. Sin embargo, tiene también muchos inconvenientes, debidos a la propia naturaleza del lenguaje natural. Los lenguajes naturales no siguen una gramática estricta (como lo hacen los lenguajes de programación), tienen sinónimos, frases hechas, y, además, las frases pueden ser ambiguas. Por lo tanto, es muy difícil encontrar una técnica que transforme correctamente una frase en lenguaje natural a un lenguaje formal en el que se pueda interrogar la base de datos en la que está almacenada la información. Parece claro, por tanto, que el uso de lenguaje natural puro para construir interfaces Web presenta muchos problemas para gestionar las consultas.*

*Nuestra propuesta es usar una técnica diferente que llamamos Lenguaje Natural Acotado (LNA). Esta técnica pretende aprovechar la expresividad y la facilidad de uso para los usuarios del lenguaje natural, evitando los inconvenientes que su*

*decodificación lleva implícitos. Una frase en Lenguaje Natural Acotado (LNA) es una frase en lenguaje natural con huecos que el usuario debe rellenar para expresar las condiciones de búsqueda. La técnica consiste en presentarle al usuario varias frases para poner restricciones. El usuario debe elegir qué frases va a usar y llenar los huecos. Finalmente, el conjunto de frases seleccionadas y completadas expresará, en lenguaje natural, la consulta completa que el usuario está haciendo.”*

Una vez realizada una introducción sobre la técnica de Lenguaje Natural Acotado se muestran varios ejemplos de dicha técnica que son utilizados a lo largo del proyecto.

- **Atributos de tipo cadenas de caracteres**

Este tipo de frase de LNA (Lenguaje Natural Acotado) es muy común en numerosos atributos, sobre todo en aquellos en los que el usuario debe escribir una palabra que restrinja la búsqueda que se va a llevar a cabo.



Figura 43. Frase de LNA para Atributos de tipo cadenas de caracteres.

Como se aprecia en la Figura 43 el atributo nombre será indicado por el usuario en el cuadro de texto que aparece en la imagen. Además la aplicación proporciona al usuario la posibilidad de realizar una búsqueda exacta, si por ejemplo está seguro de lo que busca, o por el contrario realizar una búsqueda simple indicando únicamente a la aplicación una sola palabra, si el usuario no está seguro del nombre del elemento patrimonial que busca.

- **Atributos de tipo fecha**

Para el desarrollo de este proyecto se ha utilizado, además, otro tipo de frases de LNA, como por ejemplo las relacionadas con los atributo de tipo fecha. Estos atributos son idóneos para indicar periodos de tiempo ya que ofrecen la posibilidad de buscar elementos patrimoniales que se han producido entre dos periodos, o que sean más recientes a un periodo concreto...

Por favor indique la restricción que desee

El atributo Periodo {

debe de ser exactamente Romano ▾  
 debe ser mayor a Romano ▾  
 debe ser menor a Romano ▾  
 debe estar comprendido entre SXV ▾ y SXVIII ▾

Aceptar

Figura 44. Frase de LNA para Atributos de tipo cadenas de caracteres

La Figura 44 muestra perfectamente lo que explicado en el párrafo anterior. En dicha figura se puede observar cuáles son las posibilidades que se le ofrecen al usuario para que pueda restringir valores sobre el atributo que, finalmente, den lugar a la consulta que quiera desarrollar.

### Aproximación Navegacional

Fuente: Tesis Doctoral. Arquitectura para Federación de Bases de Datos Documentales basada en Ontologías (“Consultado el 13-XI-2009”)

*“Esta técnica está íntimamente ligada a las interfaces de usuario desarrolladas para los buscadores de páginas en Internet. En este aspecto debemos indicar que existen amplios estudios en el campo de recuperación de información y también en el diseño de sistemas de búsqueda de información documental y Bibliotecas Virtuales que faciliten el acceso a grandes colecciones de documentos.*

*La Aproximación Navegacional es una técnica de diseño de Interfaces de Usuario que puede usarse tanto en Interfaces de Consulta como interfaces de presentación de respuestas.*

*Por ejemplo, en consulta, si un atributo sobre el que el usuario puede querer expresar restricciones tiene pocos valores posibles, es mejor usar la aproximación de presentar todas las alternativas en pantalla y dejarle elegir, que pedirle que restrinja la búsqueda por ese atributo en un formulario típico, escribiendo el valor que el atributo debe tener.*

*En la Fig. XXX, se muestra un ejemplo de uso de la Aproximación Navegacional para una Interfaz de Consulta. Como se puede ver, para restringir el “Género Literario” de las obras buscadas sólo es necesario pinchar sobre el cajón que corresponda.”*



Figura 45. Ejemplo de Aproximación Navegacional.

### 6.3.2 ARQUITECTURA DE LA APLICACIÓN DE BÚSQUEDA ORIENTADA

En este apartado se describe de forma general la arquitectura, que se ha construido, para el Sistema de Acceso Integrado a las bases de datos relacionadas con el patrimonio de Álava. La descripción de la arquitectura está basada en el análisis de cada una de las capas en las que ésta está compuesta, y la descripción de los módulos funcionales que existen en cada una de ellas.

Fuente: Tesis Doctoral. Arquitectura para Federación de Bases de Datos Documentales basada en Ontologías (“Consultado el 12-XI-2009”)

*“La arquitectura se divide en cuatro capas lógicas, las cuales se presentan en la Figura 46, y se describen más adelante.*

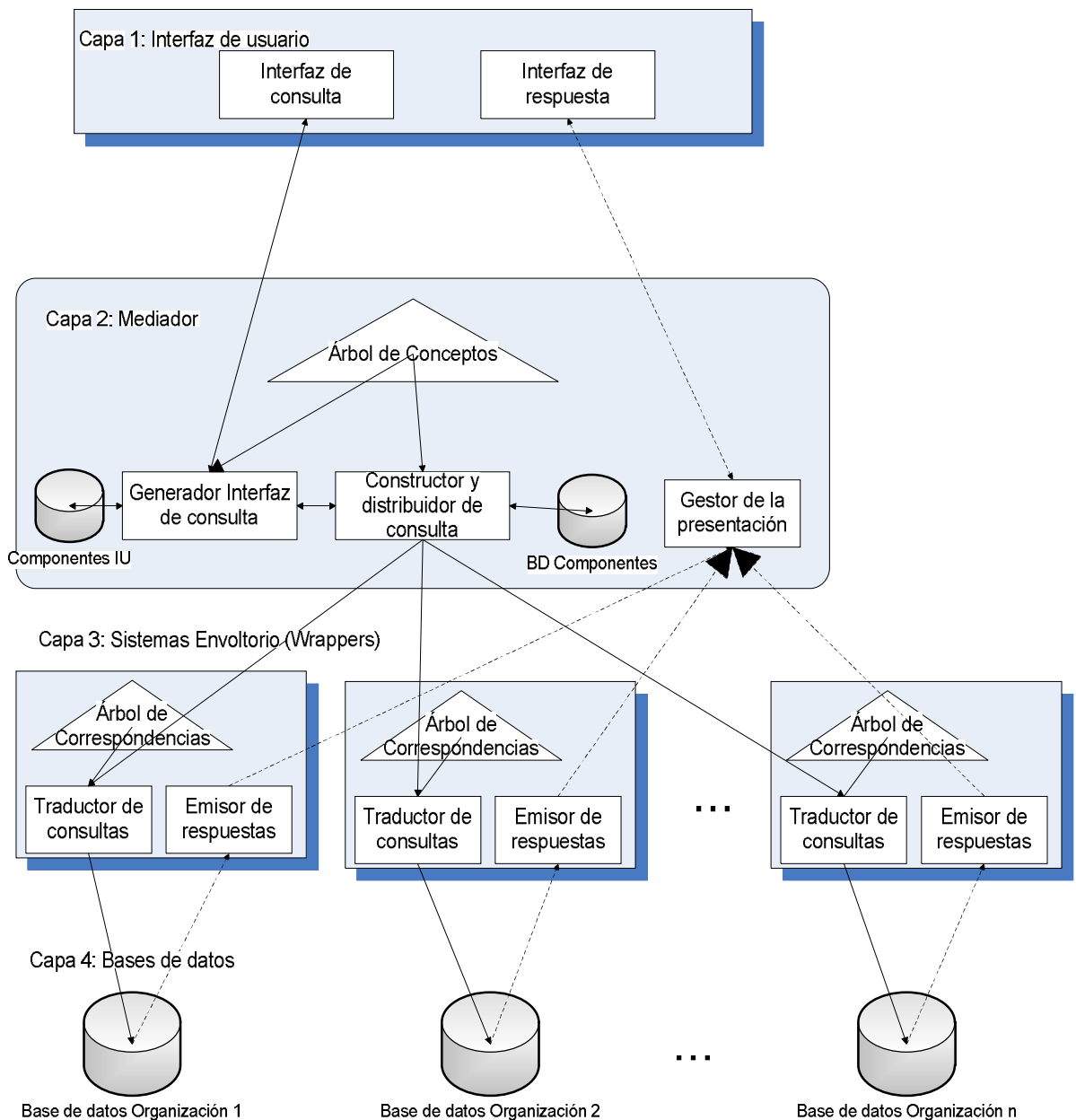


Figura 46. Arquitectura de la aplicación.

*Tal y como se puede observar, la arquitectura está dividida en cuatro capas lógicas, las cuáles son: Interfaz de usuario, Mediador, Sistemas Envoltorios o Wrappers y Bases de datos. Cada una de las capas será descrita de forma detallada a lo largo de este apartado.*

- **Capa 1: Interfaz de Usuario:** El sistema la genera automáticamente cada vez que se accede al sistema.
- **Capa 2: Mediador (Mediator):** Es el núcleo del sistema. Es lo que en bases de datos federadas corresponde al llamado Sistema Federado.



- **Capa 3: Sistemas Envoltorio (Wrappers):** Existe un Sistema Envoltorio por cada base de datos componente. Estos sistemas actúan de puente entre las bases de datos y el Mediador. Su función es “envolver” a la base de datos haciendo transparentes para el Mediador los detalles de implementación de dicha base de datos.
- **Capa 4: Bases de Datos:** las fuentes que se integran en el sistema son por un lado, las bases de datos documentales, heterogéneas y autónomas que contienen su información acerca del patrimonio alavés, y por el otro lado, la base de datos en donde se clasifica la información obtenida de los sitios Web de las diferentes organizaciones que disponen información sobre el patrimonio de Álava.”

Por otra parte, existen tres capas físicas que implican que las cuatro capas lógicas descritas en el párrafo anterior se ejecutan en tres máquinas físicas diferentes. Dichas máquinas serían, una, la máquina desde donde el usuario realiza la consulta (está máquina incluiría la capa “Interfaz del usuario” de la división lógica), dos, la máquina en donde se desarrolla la aplicación (esta máquina incluiría la capa “Mediador” de la división lógica), y tres, la/s máquina/s en donde se encuentran los datos. El número de máquinas de esta capa puede variar considerablemente, en función, por un lado, del número de bases de datos que formen parte del sistema, y por el otro, de la consulta que realice el usuario. Esto es debido, a que dependiendo del número de bases de datos existentes en el sistema, el número de máquinas que participen en él será mayor o menor. Por otro lado, en función de la consulta que realice el usuario, el sistema realizará una búsqueda en un determinado número de bases de datos, es decir, si el usuario realiza una búsqueda muy general, la aplicación, probablemente, elaborará una búsqueda por todas las bases de datos del sistema, sin embargo, si la consulta es sobre un atributo concreto, la búsqueda se realizará únicamente en aquellas bases de datos que almacenen información sobre dicho atributo, por lo que casi seguro, el número de máquinas que participen en la consulta será menor (esta/s máquina/s incluiría la capas “Sistema Envoltorio” y “Bases de datos” de la división lógica).

### 6.3.2.1 INTERFAZ DE USUARIO

Fuente: Tesis Doctoral. Arquitectura para Federación de Bases de Datos Documentales basada en Ontologías (“Consultado el 14-XI-2009”)

*“Se trata de una capa conceptual, ya que no existen módulos de software propiamente dichos en la Interfaz de Usuario. Es el Mediador el que la genera cada vez que un usuario se conecta al sistema.*

*El diseño de la interfaz se basa en una utilización combinada de las tres técnicas de diseño de Interfaces de Usuario, Lenguaje Natural Acotado (LNA), metáforas cognitivas y aproximación navegacional.*

*Se consigue así que el Mediador, a través del Generador de Interfaz de Consulta, sea capaz de generar una interfaz realmente fácil de usar, amigable y flexible de manera que el usuario pueda aprovechar las ventajas que ofrece el Sistema de Acceso Integrado que se ha diseñado y construido. Además, la Interfaz de Usuario presenta una parte sencilla, que permite hacer consultas muy simples a usuarios “comunes”, y otra parte que, sin incorporar demasiada dificultad, permite realizar consultas mucho más complejas a usuarios “expertos”.*

*Las principales características de la Interfaz de Usuario que es capaz de generar automáticamente nuestro Sistema de Acceso Integrado son las siguientes:*

- *Proporciona los medios para expresar consultas. A través de una serie de interfaces completamente intuitivas, permite al usuario formar la consulta que será enviada a las distintas bases de datos.*
- *Proporciona los mecanismos necesarios para acceder a las bases de datos, tanto a través de consultas globales a todo el conjunto de bases de datos, como consultas específicas a bases de datos de un dominio específico. Este es también un requisito imprescindible debido a que las bases de datos son heterogéneas, en cuanto a los textos y datos que almacenan.*
- *Es extremadamente intuitiva y fácil de usar. Requisito fundamental debido a que los usuarios del sistema van a ser usuarios web, en su mayoría, no informáticos, y, los usuarios web no podrán recibir entrenamiento alguno antes de acceder por primera vez al sistema.*
- *Gracias a sus posibilidades de consulta, la Interfaz de Usuario da la posibilidad de expresar consultas sencillas, a usuarios generales, y consultas más complejas y sofisticadas a usuarios expertos.”*

#### **6.3.2.1.1 COMPONENTES DE LA “INTERFAZ DE USUARIO”**

Tal y como se ha comentado anteriormente, esta capa carece de módulos de software. Sin embargo, en la misma, se pueden distinguir dos componentes, que se forman gracias a las acciones de los elementos de la capa “Mediador”.

- **Interfaz de Consulta:** Se encarga de mostrar al usuario la página correspondiente en cada caso, para que éste pueda elaborar una consulta. La interfaz de Consulta es construida por el “Generador de Interfaz de Consulta”.

- **Interfaz de Respuesta:** Muestra al usuario los resultados obtenidos de la consulta realizada. La interfaz de Respuesta es construida por el “Gestor de Presentación”.

### 6.3.2.2 MEDIADOR

Fuente: Tesis Doctoral. Arquitectura para Federación de Bases de Datos Documentales basada en Ontologías (“Consultado el 17-XI-2009”)

*“Esta capa engloba los módulos (y los Almacenes de Datos necesarios para soportar el funcionamiento de dichos módulos) encargados de hacer de nexo entre el usuario y las bases de datos, haciendo transparente para el usuario las diferencias entre las mismas y su dispersión. Como se puede observar en la Figura 46, esta capa también incluye los Árboles de Conceptos, en los cuales aparecen todos los conceptos que existen en las bases de datos y que son relevantes para realizar consultas.”* El Mediador se encarga de interactuar con el usuario, mostrando las páginas solicitadas por éste y recogiendo sus restricciones, a través de la “Interfaz de Consulta”, situada en la capa “Interfaz de Usuario”. Además de esto, una vez recogidas las restricciones dadas por el usuario, construye la consulta en lenguaje de consulta, y con el apoyo de “BD componentes” envía la consulta a todas las bases de datos que tengan información sobre los atributos que componen la consulta. Por último, una vez que son recogidas las soluciones a la consulta de cada una de las bases de datos, el “Mediador”, a través del “Gestor de la Presentación”, presenta la solución al usuario.

#### 6.3.2.2.1 COMPONENTES DEL MEDIADOR

Probablemente sea la capa más compleja de todas en cuanto a la variedad de componentes se refiere. Esta capa es la encargada de interactuar con el usuario a través de los componentes de la capa “Interfaz de Usuario”, además de encargarse de elaborar la consulta, en lenguaje consulta, para que sea enviada a las bases de datos. A continuación se muestran los diferentes componentes de esta capa.

Fuente: Tesis Doctoral. Arquitectura para Federación de Bases de Datos Documentales basada en Ontologías (“Consultado el 17-XI-2009”)

- **“Generador de la Interfaz de Consulta:** se encarga de generar de forma automática la interfaz Web de consulta, cada vez que un usuario se conecta al sistema. Para ello se basa en el contenido del Árbol de Conceptos.
- **Constructor y Distribuidor de Consultas:** recibe las restricciones que expresa el usuario y construye el documento XML de la consulta (XML que sigue el DTD del Lenguaje de Consulta del sistema, el cual está descrito en el Anexo B.3). Además, decide qué bases de datos están implicadas en la consulta y la redirige a dichas bases de datos.

- **Gestor de la Presentación:** *construye la interfaz Web que presenta un resumen de las respuestas obtenidas de cada base de datos y gestiona la navegación a través de los datos y documentos, accediendo a la Interfaz de Respuesta implementada para cada base de datos.*
- **Componentes IU:** *es almacén en el que están las Frases en Lenguaje Natural Acotado y las Metáforas Cognitivas necesarias para generar la Interfaz de Consulta.*
- **BDs Componentes:** *almacena datos (nombre, descripción, parámetros de acceso, etc.) sobre las bases de datos integradas en el sistema.”*

### 6.3.2.3 SISTEMAS ENVOLTORIOS (WRAPPERS)

Fuente: Tesis Doctoral. Arquitectura para Federación de Bases de Datos Documentales basada en Ontologías (“Consultado el 17-XI-2009”)

*“Existe un Sistema Envoltorio por cada base de datos componente. Su función es ocultar al Mediador las diferencias entre las bases de datos. Este sistema gestiona todas las peticiones que el Mediador hace a las bases de datos. Es decir, traduce todas las peticiones que le envía el Mediador al lenguaje propio de la base de datos a la que está asociado, para que puedan ser ejecutadas.”* Los Sistemas Envoltorio reciben las consultas del módulo Constructor y Distribuidor de Consultas, situado en la capa “Mediador”, en formato XML (Lenguaje de Consulta). Basándose en el árbol de Correspondencias que incluye cada Sistema Envoltorio (cada Sistema Envoltorio tiene el suyo propio), se traduce el Lenguaje de Consulta al lenguaje propio de la base de datos. Una vez realizada la traducción, la consulta se envía a la base de datos, y los resultados son recogidos por el Emisor de respuestas, situado en el propio Sistema Envoltorio, quien los envía al módulo encargado de mostrar la respuesta al usuario: El “Gestor de la Presentación” situado en la capa “Mediador”.

Tal y como hemos comentado anteriormente, cada Sistema Envoltorio incluye un Árbol de Correspondencias, el cual contiene la información necesaria para llevar a cabo la traducción de las consultas (expresadas en el Lenguaje de Consulta) al lenguaje propio de la base de datos que tiene asociada.

#### 6.3.2.3.1 COMPONENTES DE LOS SISTEMAS ENVOLTORIO (WRAPPERS)

Cada base de datos a la cual accede el sistema tiene un Sistema Envoltorio, cuya misión es conseguir adaptar la consulta, que llega en Lenguaje de Consulta, al lenguaje propio de la base de datos de cada sistema. Para ello cuenta con los siguientes componentes.

- **Traductor de Consultas:** Es el encargado de transformar el Lenguaje de Consulta en el lenguaje propio de la base de datos. Para ello recorre el Lenguaje de Consulta, y asigna a cada línea su correspondiente valor que está indicado en

el Árbol de Correspondencias. Una vez realizada la transformación, envía la consulta a la base de datos correspondiente.

- **Emisor de Respuestas:** Recoge la solución de la consulta que se realiza frente a la base de datos, y se le envía al módulo “Gestor de la Presentación” situado en la capa de “Mediador”.

#### 6.3.2.4 BASES DE DATOS

Las bases de datos forman la cuarta capa de la arquitectura. En principio el sistema dispone de dos bases de datos. Una de las bases de datos, es aquella que recoge la clasificación de los datos realizada en la fase de análisis. En ella se encuentran recogidos los datos que están disponibles en las páginas Web de las diferentes organizaciones. La segunda es la base de datos del Laboratorio de Documentación Geométrica del Patrimonio de la UPV/EHU. En ella se almacena información referente a investigaciones realizadas sobre el patrimonio de Álava, aunque también incluye investigaciones llevadas a cabo en la comunidad autónoma de La Rioja, incluso alguna realizada en Italia. En un futuro, se espera incorporar un número mayor de bases de datos al sistema.

Tanto las bases de datos con la que se cuenta para el desarrollo del proyecto, como las bases de datos que en un futuro se pretende incorporar al sistema, tienen las siguientes características:

Fuente: Tesis Doctoral. Arquitectura para Federación de Bases de Datos Documentales basada en Ontologías (“Consultado el 17-XI-2009”)

- *“Heterogéneas, en cuanto a las máquinas que las albergan, al sistema operativo de dichas máquinas y a su estructura.*
- *Autónomas, es decir, totalmente independientes en funcionamiento. Esto implica, por un lado, que no es posible cambiar el esquema de ninguna de las bases de datos para adaptarlo al esquema global del Sistema de Acceso Integrado. Por otro lado, es posible que el esquema de las bases de las bases de datos cambie para satisfacer las necesidades de las aplicaciones propietarias. Las bases de datos pueden tener su propia Interfaz de Usuario vía Web. El hecho de que formen parte de nuestro Sistema de Acceso Integrado, no significa que éstas no puedan contestar consultas provenientes de su propia interfaz.”*

##### 6.3.2.4.1 COMPONENTES DE LAS “BASES DE DATOS”

El único componente existente en esta última capa es la base de datos de una organización determinada.

- **Base de datos:** La consulta se realiza sobre ella. La elección del SGBD de la base de datos no afecta al desarrollo de la consulta.

### **6.3.3 ÁRBOL DE CONCEPTOS Y ÁRBOLES DE CORRESPONDENCIAS**

Tal y como se ha podido comprobar a lo largo de la descripción de la arquitectura de la aplicación, existen dos elementos sobre los que gira dicha arquitectura. Estos elementos son el Árbol de Conceptos, situado en la capa “Mediador”, y los Árboles de Correspondencias, situados en cada uno de los “Sistemas Envoltorio”. Existe un Árbol de Correspondencias para cada Sistema Envoltorio. A lo largo de este apartado se explicarán las funciones de estos dos árboles, así como su estructura y su forma. Para ello comenzaremos con una explicación general sobre los árboles, para posteriormente analizar en detalle cada uno de ellos.

#### **6.3.3.1 DESCRIPCIÓN DE LOS ÁRBOLES**

Fuente: Tesis Doctoral. Arquitectura para Federación de Bases de Datos Documentales basada en Ontologías (“Consultado el 20-XI-2009”)

*“Tanto los Árboles de Conceptos como los Árboles de Correspondencias tienen características en común que son las que vamos a describir en este apartado, refiriéndonos a ambos como Árboles.”*

Un Árbol en nuestro sistema es un conjunto de conceptos y relaciones (entre ellos) que describen la información almacenada tanto en la base de datos del Laboratorio de Documentación Geométrica del Patrimonio de la UPV/EHU, como la base de datos que contiene información existente en las páginas Web de las organizaciones.

Fuente: Tesis Doctoral. Arquitectura para Federación de Bases de Datos Documentales basada en Ontologías (“Consultado el 20-XI-2009”)

*“Los Árboles son estructuras, en donde los nodos son conceptos. Los conceptos tienen atributos, es decir, propiedades atómicas de dichos conceptos. Tanto los conceptos como sus atributos, son nociones usadas en el mundo real, en los dominios de las bases de datos”. Por ejemplo, un posible concepto de un Árbol podría ser “Iglesias”. Entre los posibles atributos de este concepto podrían estar: “Nombre”, “Siglo de creación”, “Tipo de Iglesia”, o “Localidad en la que se encuentra”.*

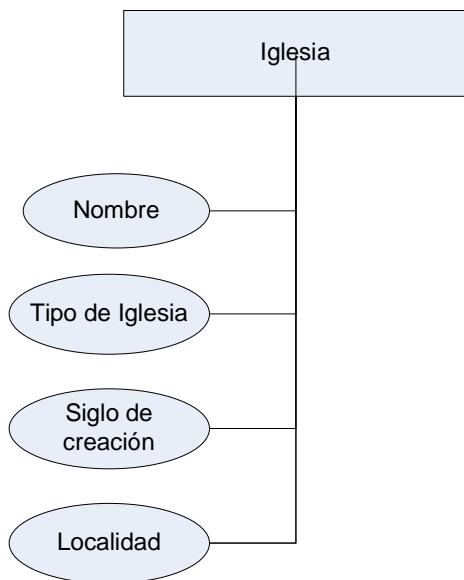


Figura 47. Un concepto y sus atributos.

En Figura 47, aparece la representación gráfica de un concepto con sus atributos. De aquí en adelante, hablaremos de “términos” cuando nos estemos refiriendo tanto a conceptos como a atributos de un Árbol.

Fuente: Tesis Doctoral. Arquitectura para Federación de Bases de Datos Documentales basada en Ontologías (“Consultado el 20-XI-2009”)

*“Las ramas de un Árbol representan las relaciones entre los conceptos. Para el desarrollo de nuestro proyecto hemos considerado relevantes dos tipos de relaciones: Relaciones de Generalización /Especialización y Relaciones de Descripción.”*

### 6.3.3.1.1 RELACIONES DE DESCRIPCIÓN

Un concepto se describe, no sólo mediante sus atributos, sino también a través de otros conceptos. En nuestra aplicación únicamente vamos a encontrar la descripción de los conceptos a partir de atributos. Sin embargo, en un futuro, si el sistema es ampliado, puede aparecer este tipo de relación de Descripción, por lo que se considera necesario mencionarla.

### 6.3.3.1.2 RELACIONES DE GENERALIZACIÓN/ESPECIALIZACIÓN

Fuente: Tesis Doctoral. Arquitectura para Federación de Bases de Datos Documentales basada en Ontologías (“Consultado el 20-XI-2009”)

*“Decimos que existe una relación de Generalización / Especialización entre un concepto general y un conjunto de conceptos especializados cuando estos últimos se pueden definir a partir de una propiedad distintiva del concepto general. O, lo que es*

equivalente, cuando el concepto general se genera para englobar las propiedades comunes de un conjunto de conceptos (especializados).

*Hay dos razones principales para incluir relaciones de Generalización / Especialización en un Árbol. La primera es que pueden existir ciertos atributos aplicables únicamente a ciertos tipos de instancias del concepto. La segunda razón es que cada especialización del concepto puede establecer una Relación de Descripción con conceptos distintos.”*

Por ejemplo, el concepto “electrodomésticos”, según el valor de su atributo “clase” se especializaría en “Televisor” y en “Lavavajillas”. El “Electrodoméstico televisor” tendría atributos como “pulgadas” y el “Electrodoméstico Lavavajillas” tendría atributos como “capacidad”.

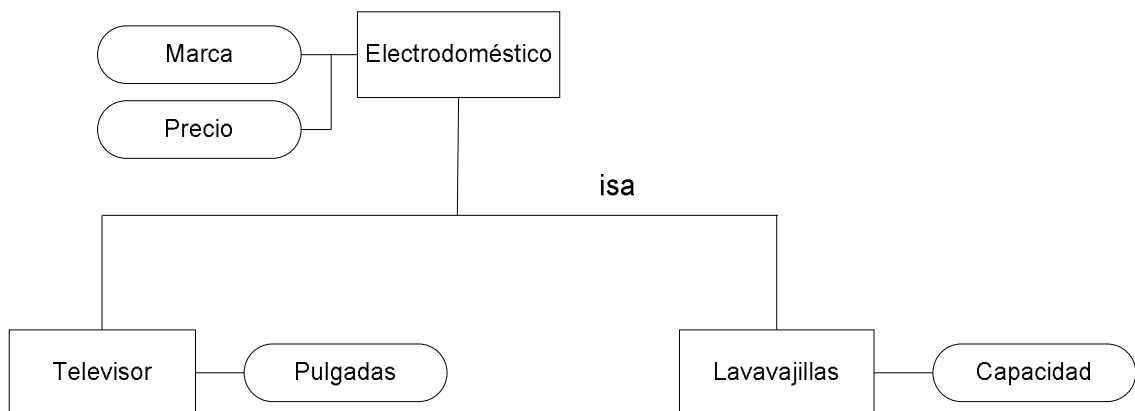


Figura 48. Ejemplo de una relación de Generalización / Especialización.

Las relaciones de Generalización / Especialización se representan, como en el ejemplo de la Figura 48, etiquetadas con la cadena “is a”.

Fuente: Tesis Doctoral. Arquitectura para Federación de Bases de Datos Documentales basada en Ontologías (“Consultado el 20-XI-2009”)

*“Las relaciones de Generalización / Especialización de nuestro sistema son:*

- **Disjuntas:** una instancia del concepto general sólo se especializa en uno de los conceptos especializados.
- **Totales o completas:** todas las instancias del concepto general se especializan en alguno de los conceptos especializados.”

En nuestro sistema, un elemento patrimonial o “patrimonio” es forzosamente, o un “Edificio”, o una “Excavación”, o bien, un “Objeto”, y además solamente uno de esos tipos.



Además, en nuestro sistema, existe otra relación de Generalización/ Especialización. Un “Edificio” es forzosamente o una “Arquitectura Religiosa”, o una “Arquitectura Civil” o un “Elemento Menor”.

La existencia de la Relaciones de Generalización / Especialización provocan que se pueda distinguir dentro de nuestra aplicación, una parte experta y una parte general. Para aclarar esta nueva división, a continuación se explicarán cada una de las partes.

### Parte General

La parte general es principalmente utilizada por aquellos usuarios que tienen conocimientos escasos sobre el patrimonio. Se trata de una parte, un subárbol, del Árbol de Conceptos, que incluye únicamente los atributos comunes de todas las especializaciones del concepto Patrimonio. Es decir, si el usuario desea realizar una búsqueda sobre la iglesia de San Martín, no deberá llegar hasta los atributos del concepto Arquitectura Religiosa, sino que sobre el concepto general de Patrimonio, indicando el nombre de la iglesia, el sistema le mostrará la solución que buscaba.

Además de esta parte general, en nuestro sistema existe otra parte general del subárbol “Edificios”, ya que en vez de realizar una búsqueda concreta se puede conseguir la información deseada, realizando, simplemente, una descripción de los atributos del concepto “Edificio”.



Figura 49. Parte general del concepto Patrimonio.



isa

Figura 50. Parte general del concepto Edificios.

En la Figura 49, y en la Figura 50 se puede observar qué partes del Árbol de Concepto conforman la parte general de dicho Árbol de Conceptos, y la parte general del subárbol del concepto “Edificios” respectivamente.

### Parte Experta

La parte experta podría ser definida, como una búsqueda realizada sobre los conceptos más especializados del Árbol de Conceptos. Lo que supone, que para realizar búsquedas denominadas “expertas” es necesario tener unos conceptos básicos sobre patrimonio, o bien, saber con claridad qué es aquello que se está buscando. En el Árbol de Conceptos que se desarrolla en esta aplicación (Figura 51), la parte experta estaría formada, por los sub-árboles “Excavaciones”, “Edificios” y “Objetos”. Además dentro del concepto “Edificios” existe otra parte aún más especializada que reúne los conceptos “Arquitectura Religiosa”, “Arquitectura Civil” y “Elementos Menores”.

El motivo por el cual se diferencian estas dos partes, experta y general, no es otra que intentar conseguir adaptar una interfaz concreta, a los conocimientos de un usuario determinado sobre el patrimonio de la provincia de Álava. Esto dará la posibilidad a los diferentes usuarios, de llevar a cabo una consulta específica y elaborada, por un lado, y una consulta general y simple, por el otro lado.

### **6.3.3.2 ÁRBOLES DE CONCEPTOS**

#### **6.3.3.2.1 CONCEPTOS BÁSICOS**

Si existe un elemento que puede denominarse imprescindible en la arquitectura que se está desarrollando, este es el Árbol de Conceptos. Los Árboles de conceptos se sitúan en la capa “Mediador”, y en ellos se almacenan todos los conceptos y atributos sobre los que el usuario puede realizar una búsqueda, además de otra información relevante que será analizada más adelante. En esta arquitectura se ha optado por el lenguaje XML para implementar el Árbol de Conceptos.

#### **6.3.3.2.2 UTILIDADES DE LOS ÁRBOLES DE CONCEPTOS**

Los Árboles de Conceptos, en general, pueden desempeñar diferentes funciones, sin embargo es el encargado del diseño de la arquitectura quién decide las funciones que va a desempeñar su Árbol de Conceptos concreto. En la arquitectura que aquí se desarrolla, el Árbol de Conceptos, desarrolla o ayuda a desarrollar las siguientes funciones.

##### **Información sobre la búsqueda del sistema**

A pesar de que, obviamente, la información sobre el patrimonio alavés es obtenida de las fuentes de datos, no son ellas, quienes de manera directa, deciden que información es accesible y cual no. El encargado de decidir que conceptos son mostrados al usuario para que éste exprese sus restricciones, es el Árbol de Conceptos. Es obvio que cada fuente de datos cuenta con sus datos, y que estos son diferentes a los de las demás fuentes de datos, por esto, previo al comienzo del desarrollo de la arquitectura se realiza un estudio de las fuentes de datos existentes, y se decide cuál va a ser la estructura de búsqueda, los conceptos, atributos y relaciones, tanto de descripción, como de generalización/especificación, en los que se va a basar la búsqueda. Todo esto queda reflejado en el Árbol de Conceptos.

##### **Información para construir la Interfaz de Consulta**

Fuente: Tesis Doctoral. Arquitectura para Federación de Bases de Datos Documentales basada en Ontologías (“Consultado el 21-XI-2009”)

*“Una consulta del usuario puede definirse como una descripción de las propiedades que deben tener los objetos que se obtengan en la respuesta. Un usuario puede describir los objetos que está buscando describiendo las características de los mismos. Dichas características o propiedades que el usuario puede “describir” no son más que los atributos asociados a dicho objeto en el Árbol de Conceptos del sistema. Por lo tanto, podemos decir que las consultas permitidas por el sistema son conjuntos de condiciones sobre los atributos del Árbol de Conceptos.*

*El sistema, por tanto, ofrece un entorno (Interfaz de Consulta) en el que el usuario puede, por un lado, seleccionar los atributos del Árbol de Conceptos sobre los que tiene interés en expresar restricciones, es decir, puede recorrer el Árbol de Conceptos y seleccionar aquellos atributos que le interesan; y, por otro lado, en el que puede, finalmente, expresar las restricciones que desee sobre los atributos que ha seleccionado. Además, cuando los valores que puede tomar un atributo son un conjunto pequeño, dichos valores se almacenan en el Árbol de Conceptos asociados a dicho atributo, facilitando así que el usuario sólo escoja valores válidos para restringir dichos atributos.”*

De esta manera, el Árbol de Conceptos nos ayuda a presentar los conceptos, atributos y relaciones de generalización/especialización, de forma que el usuario pueda, además, aportar las restricciones que desee convenientes, y que darán lugar a la consulta que será enviada a las diferentes fuentes de datos.

### **Información relativa a las bases de datos**

Los Árboles de Conceptos contienen los conceptos, útiles para consulta, que existen en las bases de datos. Sin embargo, puede ocurrir que algunos de los conceptos y/o atributos del Árbol de Conceptos no existan en alguna de las bases de datos componentes. Por lo tanto, a efectos de saber a qué bases de datos redirigir las consultas expresadas sobre un cierto conjunto de conceptos y atributos, es necesario saber qué atributos existen en cada base de datos componente.

En el Árbol de Conceptos de nuestro sistema, cada atributo tiene asociada una lista de identificadores. Se trata de los identificadores de las bases de datos en las que de dicho atributo existe. De esta manera, una consulta expresada sobre un conjunto de atributos será redirigida únicamente a las bases de datos cuyos identificadores estén asociados a dichos atributos. Es decir, a aquellas bases de datos que realmente pueden contestar la consulta (aunque sea parcialmente).

### 6.3.3.2.3 ÁRBOL DE CONCEPTOS DE NUESTRO SISTEMA

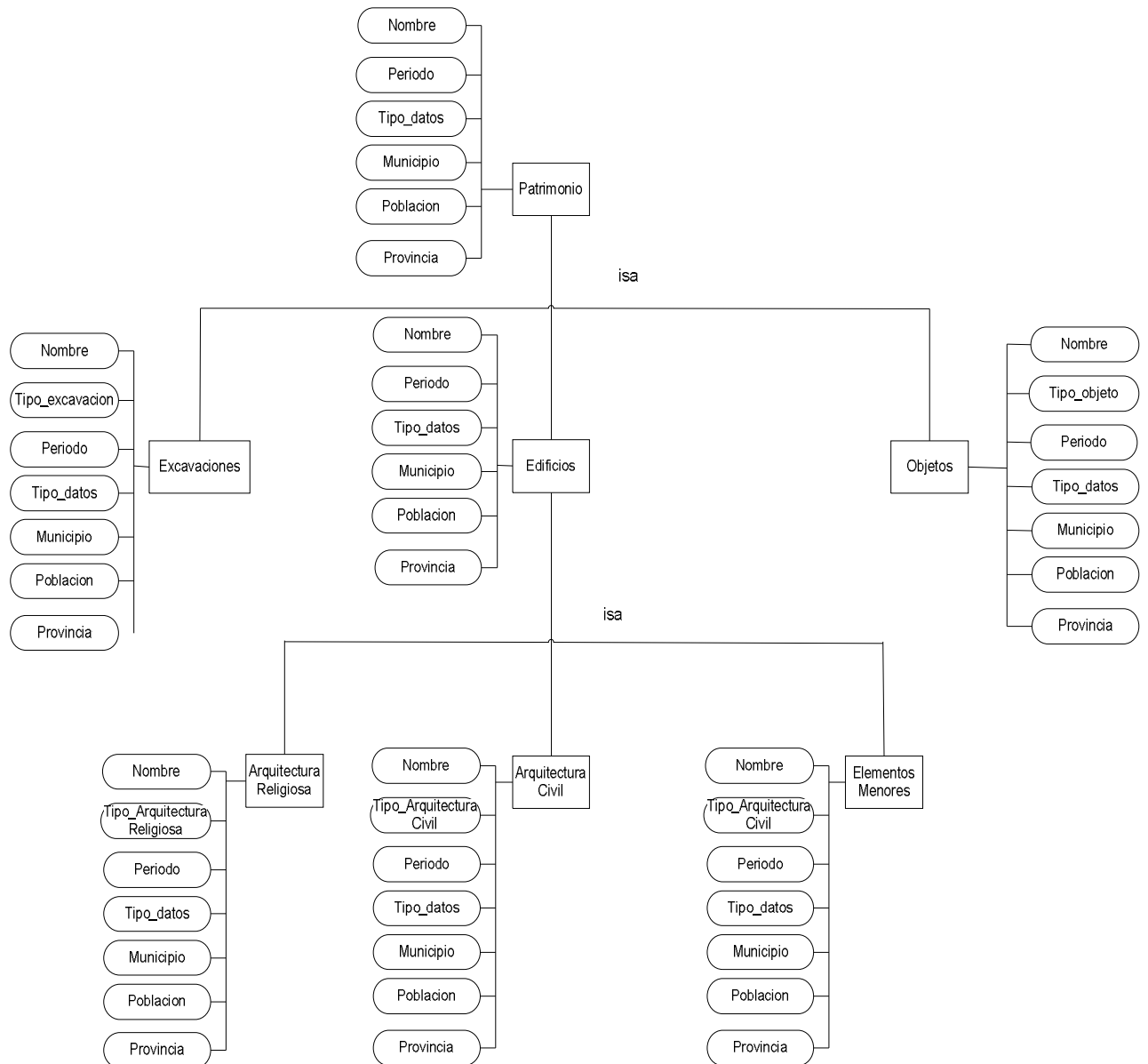


Figura 51. Árbol de Conceptos del Sistema.

### **6.3.3.3 ÁRBOLES DE CORRESPONDENCIAS**

#### **6.3.3.3.1 CONCEPTOS BÁSICOS**

Fuente: Tesis Doctoral. Arquitectura para Federación de Bases de Datos Documentales basada en Ontologías (“Consultado el 21-XI-2009”)

*“Los Árboles de Correspondencias tienen una estructura idéntica a la de los Árboles de Conceptos. Es una estructura en forma de árbol en la que los nodos son conceptos, con propiedades que llamamos atributos, y las ramas relaciones. Existen los mismos tipos de relaciones y la semántica de los conceptos y los atributos es la misma. Sin embargo, el propósito de los Árboles de Conceptos y los Árboles de Correspondencias son muy distintos. Los Árboles de Correspondencias servirán, como hemos dicho, para traducir las consultas que plantean los usuarios sobre los atributos del Árbol de Conceptos (y que el sistema almacena en XML) al lenguaje de la base de datos a la que está asociado dicho Árbol de Correspondencias.*

*Durante la traducción, para cada concepto que aparezca en el documento XML, se lee del Árbol de Correspondencias la Información de Correspondencia que tiene asociado dicho concepto y se añade a la sentencia SQL final. Asimismo, para cada atributo que exista en el documento XML de la consulta, se lee la Información de Correspondencia de dicho atributo y, antes de completar la sentencia SQL final con la nueva condición, se sustituye las etiquetas entre # por los valores concretos que haya establecido el usuario y que están recogidos en el documento XML. En nuestro sistema contemplamos también los casos en los que los valores de algún atributo, no estén almacenados tal cual en la base de datos, sino que se almacene su código correspondiente. En este caso, la Información de Correspondencia del atributo también incluye la tabla de traducción de cada posible valor al código que se almacena en la base de datos.”*

#### **6.3.3.3.2 ÁRBOLES DE CORRESPONDENCIAS EXISTENTES EN NUESTRO SISTEMA**

El número de Árboles de Correspondencias del sistema dependerá del número de fuentes de datos a las que se acceda, ya que para cada fuente de datos es necesario un Árbol de Correspondencia que ayude a realizar la traducción de Lenguaje de Consulta al lenguaje propio de la base de datos. En el caso del sistema que se desea implantar, en principio, va a contar con dos bases de datos. Una de las dos bases de datos está compuesta por una clasificación de la información existente en diversas páginas Web de diversas organizaciones que realizan alguna actividad relacionada con el patrimonio. Esta clasificación también ha sido llevada a cabo durante el desarrollo de este proyecto, al igual que su posterior informatización. Una vez llevada a cabo esta informatización, la fuente de datos resultante ha sido añadida a la arquitectura que se quiere desarrollar como si fuera una fuente de datos más. La otra base de datos es la del Laboratorio de Documentación Geométrica del Patrimonio de la UPV/EHU, en donde se almacena la información relativa a las investigaciones llevadas a cabo por dicho laboratorio. Sin

embargo, se espera que en un futuro, cuando las organizaciones comprueben el buen funcionamiento del buscador, comiencen a incorporar sus bases de datos al sistema.

#### 6.3.3.3.2.1 BASE DE DATOS “SITIOS WEB”

La base de datos “Sitios Web” almacena la información recogida de diferentes páginas Web relacionadas con el patrimonio. Esta base de datos está explicada a lo largo del apartado 6.2 de esta memoria. En la fase de implementación se podrá observar cómo se lleva a cabo la traducción de Lenguaje de Consulta al lenguaje natural de esta base de datos.

#### 6.3.3.3.2.2 BASE DE DATOS DEL LABORATORIO DE DOCUMENTACIÓN GEOMÉTRICA DEL PATRIMONIO DE LA UPV/EHU

La base de datos del Laboratorio de Documentación Geométrica del Patrimonio de la UPV/EHU almacena información referente a los proyectos realizados a lo largo de estos últimos años por el grupo de investigación del propio Laboratorio de Documentación Geométrica del Patrimonio de la UPV/EHU. Se trata de una base de datos sencilla y de uso común. Con el fin de conocer un poco más la segunda base de datos contra la que trabajará la aplicación de búsqueda orientada a continuación se mostrarán su diagrama Entidad-Relación, así como su diagrama Relacional.

#### Modelo Entidad-Relación base de datos “Proyectos”

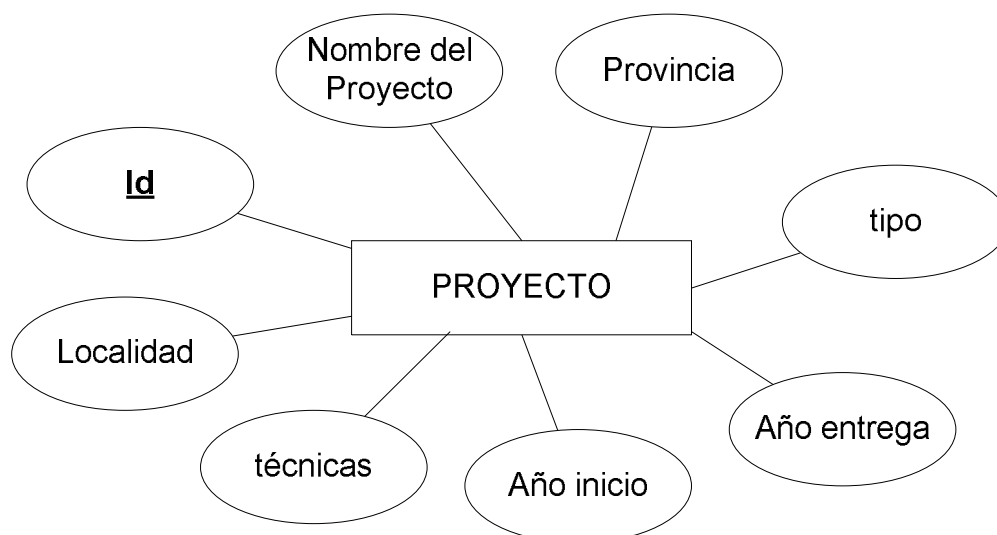


Figura 52. Diagrama Entidad-Relación de la Base de Datos Proyectos.

Como se puede observar en la Figura 52 la base de datos “Proyectos” cuenta únicamente con una tabla denominada “Proyecto”. Esta tabla contiene una serie de atributos que ayudan a almacenar la información relacionada con los proyectos de investigación llevados a cabo por el Laboratorio de Documentación Geométrica del Patrimonio de la UPV/EHU.

## Modelo Relacional base de datos “Sitios Web”

El modelo Relacional de la base de datos “Proyectos” se consigue, tal y como se ha mencionado a lo largo de esta memoria siguiendo, para cada tabla, una serie de reglas, en este caso la transformación es aún más directa debido a la simplicidad de la base de datos.

### Tabla PROYECTO

Id	Nombre del Proyecto	Localidad	Provincia	Año inicio	Año entrega	tipo	técnicas
----	---------------------	-----------	-----------	------------	-------------	------	----------

Tabla 12. Tabla Proyecto.

### Relaciones de la base de datos

- **Tabla Proyecto**

En esta tabla se almacenarán todos los proyectos llevados a cabo por el grupo de investigación del Laboratorio de Documentación Geométrica del Patrimonio de la UPV/EHU. Tal y como se puede observar en el Modelo Relacional, en esta tabla se almacenarán el id del proyecto, el nombre del mismo la localidad y provincia en los que tiene lugar la investigación así como el año de inicio y el año de entrega del proyecto, además del tipo y las técnicas del mismo.



#### 6.3.4 LENGUAJES DE COMUNICACIÓN ENTRE CAPAS

Como ya se ha comentado, se han definido dos lenguajes para establecer la comunicación entre las capas de la arquitectura. Estos lenguajes son:

- *Lenguaje de Consulta*: define el formato de las consultas del usuario.
- *Lenguaje de Respuestas*: proporciona un formato para enviar el número de respuestas obtenidas en una consulta.

Estos dos lenguajes de intercambio de datos hacen independiente el Mediador, no sólo de las diferencias entre los lenguajes de las bases de datos sino también entre los tipos de datos de los documentos almacenados en la base de datos.

Al igual que los Árboles de Conceptos y Los Árboles de Correspondencias, estos lenguajes se definen en XML.

## **7. IMPLEMENTACIÓN DE LA APLICACIÓN**

### **7.1 INTRODUCCIÓN**

Una vez desarrolladas ambas aplicaciones, tanto la aplicación de búsqueda orientada, como la aplicación que da soporte a la base de datos que clasifica datos sobre la información existente en las páginas Web, se va a explicar, a lo largo de este apartado, cómo se han implementado éstas, profundizando en cada una de ellas.

Siguiendo los patrones del apartado de diseño, para que el lector pueda diferenciar de manera clara cada una de las dos aplicaciones desarrolladas, primero se llevará a cabo la explicación detallada de la aplicación que da soporte a la base de datos que almacena los datos clasificados de la información existente en las páginas Web, y posteriormente tendrá lugar la explicación de la aplicación de búsqueda orientada.

### **7.2. APLICACIÓN QUE DA SOPORTE A LA BASE DE DATOS QUE CLASIFICA DATOS SOBRE LA INFORMACIÓN EXISTENTE EN LAS PÁGINAS WEB**

#### **7.2.1 INTRODUCCIÓN**

La aplicación que se ha desarrollado tiene como fin dar soporte a la base de datos que ha sido construida para almacenar datos referentes a la información que nos podemos encontrar en las páginas Web de organizaciones o instituciones que trabajan con el patrimonio, especialmente en Álava.

Al ser una aplicación de esta índole, lo que se pretende a lo largo de su desarrollo, es conseguir un producto sencillo pero eficiente. Para ello se pretendía contar con un diseño sencillo e intuitivo, en el que el encargado de mantener la base de datos actualizada se pudiera manejar de forma ágil y sencilla. A su vez se pretendía contar con todas las funcionalidades necesarias para conseguir mantener actualizada la fuente de información donde se encontraban los datos obtenidos de las páginas Web. Para conseguir esto, la aplicación se basa en un menú dinámico situado en la parte izquierda de la pantalla, en el cual aparecen ordenadas y agrupadas cada una de las funcionalidades del sistema, de tal forma que sea lo más sencillo e intuitivo posible conseguir llevar una funcionalidad a cabo. Además, los formularios que el sistema expone al usuario son formularios de diseño simple, lo cual le va a facilitar tarea de inserción, modificación y borrado de datos.

Cabe destacar que en la aplicación existen dos perfiles: administrador y usuario. El administrador tiene completa potestad para realizar cualquier tarea de las que presenta la aplicación, mientras que el usuario tiene funciones limitadas. El perfil lo determina uno de los administradores del sistema al incorporar al nuevo miembro al sistema.

Además, cuando el usuario (ya tenga perfil de usuario o administrador) intenta acceder al sistema debe autenticarse otorgando su Nick y su contraseña. Si estas no son correctas el sistema no le permitirá el acceso, si por el contrario la información otorgada por el usuario coincide con la registrada en la base de datos el usuario podrá acceder al sistema, mientras que éste almacenará información sobre el propio usuario.

Las contraseñas de los usuarios (ya tenga perfil de usuario o administrador) quedan almacenadas en la base de datos de forma encriptada con el algoritmo md5.

Con el fin de facilitar la comprensión de la aplicación que da soporte a la base de datos que clasifica datos sobre la información existente en las páginas Web, desarrollada en el proyecto, a lo largo de este apartado se recogerá las interfaces que provee el sistema para que el usuario pueda realizar sus tareas, así como su distribución física y una serie de detalles que se consideran importantes para la aplicación.

### **7.2.2 DISTRIBUCIÓN FÍSICA**

La aplicación está dividida en dos directorios, además de un directorio central, donde reside la información relativa a la página de inicio, el menú y la página de autenticación del usuario. Los dos directorios responden al nombre de “Administradores”, y “Páginas Web”. El primero de ellos almacena toda la información acerca de las funcionalidades relacionadas con los administradores, como son a modo de ejemplo, añadir administrador, eliminar administrador, modificar contraseña, ver movimientos... El segundo directorio, “Páginas Webs”, almacena toda la información relativa a las operaciones que se realizan frente a la base de datos “sitios Web”, es decir todas aquellas operaciones relacionadas con el patrimonio, como son, por ejemplo añadir sitio Web, eliminar subpágina Web, introducir patrimonio, eliminar patrimonio...

Esta distribución tiene como único fin ayudar a la modificación de alguno de los elementos o a la inserción de una nueva página .php3.

### **7.2.3 INTERFACES DE LA APLICACIÓN**

Aunque, tal y como se ha comentado en el apartado de diseño, el sistema que almacena la clasificación de datos consta de dos bases de datos, una de ellas la propia base de datos que recoge la información, y la otra, aquella que almacena los movimientos que se producen en la primera, la aplicación que trabaje, en constante funcionamiento, contra ambas bases de datos será la misma, por lo que, obviamente, sus interfaces serán las mismas en muchos casos, y similares en otros tantos.

Obviamente cada interfaz difiere del resto en función de la funcionalidad para la que es utilizada, sin embargo, podemos distinguir diferentes modelos de interfaces globales que se explicarán a continuación.

## Interfaz de Autenticación

The wireframe shows a login form titled "VALIDAR USUARIO". It features two input fields: "Nick" with the value "ion" and "Password" with masked characters "••••". An "Aceptar" button is located at the bottom right. A red line connects the "Nick" label to the "Aceptar" button, indicating a visual flow or relationship.

Figura 53. Interfaz de Autenticación

La figura superior muestra la interfaz que utiliza el sistema para la autenticación de usuarios. Se trata de una interfaz completamente intuitiva y de fácil manejo, para cualquier tipo de usuario.

## Interfaz Formulario común

The wireframe shows a form titled "NUEVO SITIO WEB" under the heading "Datos del Sitio Web". It contains three text input fields: "Dirección Web:", "Nombre de la organización:", and "Descripción:". A large "Introducir" button is positioned at the bottom center of the form.

Figura 54. Interfaz tipo formulario común

Se trata de un ejemplo de formulario común, en este caso es utilizada para la funcionalidad Nuevo Sitio Web. A lo largo de la aplicación nos encontramos numerosas interfaces de este modelo, el cual está formado básicamente por cuadros de texto los cuales deben ser rellenos por el usuario para actualizar las bases de datos del sistema. Como se aprecia se trata de una interfaz sencilla, intuitiva y de uso común, utilizable para cualquier usuario.

## Interfaz Lista desplegable

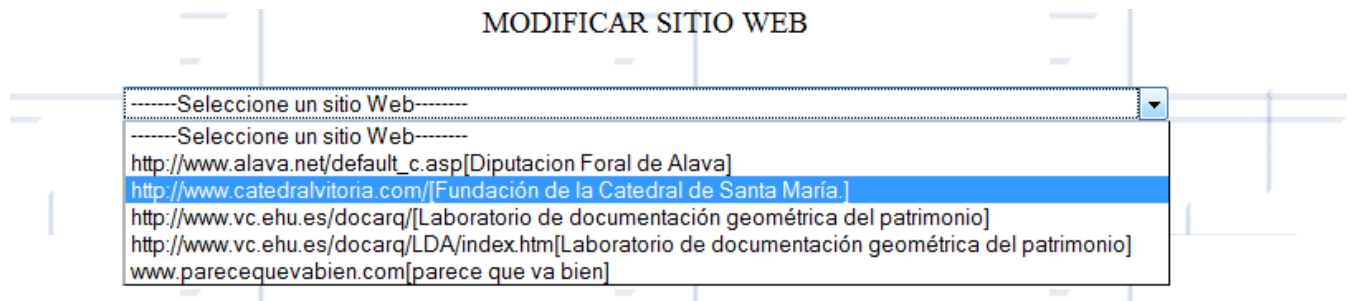


Figura 55. Lista desplegable.

Aunque realmente no está considerada una interfaz, ya que en muchos casos forma parte de otras interfaces más complejas, las listas desplegables son muy utilizadas en la aplicación. El sistema las requiere en numerosas ocasiones cuando necesita que el usuario escoja entre diferentes opciones. Es muy común en numerosos frames.

Se trata también de una interfaz fácil de utilizar, ya que su simplicidad es máxima. Es adecuada para todo tipo de usuarios.

## Interfaz de borrado

Lista de elementos patrimoniales de <a href="http://www.vc.ehu.es/docarq/LDA/index.htm">http://www.vc.ehu.es/docarq/LDA/index.htm</a>					
Clase	Nombre	Municipio	Poblacion	Provincia	Modificar
excavaciones	Catedral de Santa María	Vitoria-Gasteiz	Vitoria-Gasteiz	Alava	
excavaciones	Antiguo Instituto	Vitoria-Gasteiz	Vitoria-Gasteiz	Alava	
excavaciones	Santuario de la Encina	Vitoria-Gasteiz	Vitoria-Gasteiz	Alava	
excavaciones	Gracurris	Alfaro	Alfaro	La Rioja	
excavaciones	Castillo de los Monjes	Lumbreras	Lumbreras	La Rioja	
excavaciones	Convento de Valbuena	Logroño	Logroño	La Rioja	
arquitecturacivil	Nombre	Municipio	Poblacion	Provincia	

Figura 56. Interfaz de borrado

Interfaz que muestra una lista de elementos a los que acompaña con en su última columna con una imagen que permite el borrado de cada uno de los elementos. En la figura superior (Figura 56) se puede ver un ejemplo de este modelo de interfaz de borrado, concretamente el que hace referencia a los elementos patrimoniales.

Una vez más se observa otro caso de interfaz de fácil manejo, apta para cualquier usuario.

## Interfaz de información al usuario

Lista sitio Web	
Dirección Web	Nombre
http://www.alava.net/default_c.asp	Diputacion Foral de Alava
http://www.catedralvitoria.com/	Fundación de la Catedral de Santa María.
http://www.vc.ehu.es/docarq/	Laboratorio de documentación geométrica del patrimonio
http://www.vc.ehu.es/docarq/LDA/index.htm	Laboratorio de documentación geométrica del patrimonio
www.arabarri.com	arabarri

Figura 57. Ejemplo de Interfaz de información al usuario.

Su verdadera finalidad es informar al usuario que el elemento ha sido bien, insertado, o bien, modificado. Para esto la interfaz nos muestra todos los elementos de la misma índole al elemento actualizado, y éste último aparece en un color rojizo.

Se trata de una interfaz meramente informativa para el usuario. Se considera útil para la aplicación, ya que informa al usuario de la tarea realizada.

## Interfaz que muestra los movimientos realizados en el sistema

Lista de Movimientos		
Responsable	Acción	Fecha
ion	Borrar sitio Web www.todopatrimonio.com	2009-01-23 17:55:28
ion	Eliminar subpágina Web www.todopatrimonio.com/excavaciones del sitio Web www.todopatrimonio.com	2009-01-23 17:55:23
ion	Eliminar patrimonio: Figuras de marmol/ acero de la subpágina www.todopatrimonio.com/excavaciones	2009-01-23 17:54:18
esti	Modificar sitio Web: http://www.alava.net/default_c.asp	2009-01-23 17:52:56
ion	Eliminar administrador: peter	2009-01-23 17:48:13
ion	Modificar Administrador: martuki	2009-01-23 17:47:57
ion	Insertar Administrador: martuki	2009-01-23 17:47:40
ion	Modificar patrimonio: Figuras de marmol/ acero en la subpágina www.todopatrimonio.com/excavaciones	2009-01-23 17:46:45
ion	Insertar patrimonio: Figuras de marmol en la subpágina Web www.todopatrimonio.com/excavaciones	2009-01-23 17:46:27
ion	Insertar subpagina Web: www.todopatrimonio.com/excavaciones en el sitio Web www.todopatrimonio.com	2009-01-23 17:45:16
		Ver siguientes

Figura 58. Interfaz de Movimientos en el sistema.

Por último, es importante citar la interfaz en la que se muestran los movimientos que se llevan a cabo en el sistema. Esta interfaz consigue mostrar de forma clara y ordenada los movimientos que se producen en la aplicación en cada momento.

## 7.2.4 CONTROLES

En una aplicación que trabaja actualizando información de una base de datos es importante realizar un control sobre determinados campos, ya que una actualización inadecuada de la base de datos puede producir graves problemas de consistencia en la misma. Por esto, en las diferentes funcionalidades de la aplicación desarrollada, existen distintos controles de campos que se desarrollan en la parte del cliente, y controladores que son llevados a cabo en la parte servidor. Con el objetivo de conocer, sin entrar al máximo detalle, los controles desarrollados, tanto en la parte cliente como en la parte servidor, se llevará a cabo una explicación de los mismos.

### 7.2.4.1 CONTROLES DE ACCESO

Cuando un usuario accede al sistema lo primero que se encuentra es una pantalla de autenticación, en la cual escribe su Nick y su contraseña, el sistema las valida y si todo es correcto el usuario accede al sistema.

#### Parte cliente:

- El sistema controla que el usuario rellena tanto el campo correspondiente al Nick como el correspondiente a la contraseña. En ambos casos, si uno de los dos campos no se ha completado mostrará un mensaje de error indicando el mismo.
- Además la contraseña es cifrada con el algoritmo de cifrado md5.

#### Parte servidor:

- El sistema valida por un lado, que el usuario existe en la base de datos (es decir, ya está registrado) y por otro que la contraseña introducida por el mismo coincide con la que está registrada en la base de datos.

### 7.2.4.2 CONTROLES EN LAS INSERCIONES

Tanto cuando se introduce un sitio Web, una subpágina Web, un elemento patrimonial o un nuevo administrador se utilizan los siguientes controladores:

#### Parte cliente:

- Se comprueba por medio de javascript, que todos los campos que son importantes para la base de datos hayan sido rellenos por el usuario, mostrando en su defecto, un mensaje de aviso y no permitiendo al usuario continuar con la funcionalidad hasta que corrija el error. En determinados casos, por ejemplo el campo teléfono, no aceptará caracteres propios del campo al que hace referencia, en el caso del teléfono sólo aceptará números.

#### Parte servidor:

- Salvo para la inserción de un elemento patrimonial, que se considera que puede introducirse con el mismo nombre y siendo distinguido por el resto de sus atributos, para el resto de las inserciones el sistema comprueba que el elemento que se pretende introducir no utilice bien una dirección Web o bien un Nick, en función del elemento que se esté introduciendo, que ya esté registrado en la base de datos.

#### **7.2.4.3 CONTROLES EN LAS MODIFICACIONES**

Al igual que sucedía con las inserciones, en las modificaciones, ya sea cuando se modifica un sitio Web, una subpágina Web, un elemento patrimonial, un administrador, o la contraseña del propio usuario se utilizan los siguientes controladores:

#### Parte cliente

- Se comprueba por medio de javaScript que antes de modificar los datos, todos los campos que se consideran importantes para la base de datos hayan sido rellenados por el usuario, mostrando en su defecto, un mensaje de aviso y no permitiendo al usuario continuar con la funcionalidad hasta que corrija el error. En determinados casos, por ejemplo el campo teléfono, no aceptará caracteres propios del campo al que hace referencia, en el caso del teléfono sólo aceptará números.
- En el caso concreto de la modificación de la contraseña, los campos escribir contraseña y confirmar contraseña deben coincidir, en el caso contrario, el sistema no permitirá continuar con la modificación y mostrará un mensaje de error.

#### Parte servidor:

- Al igual que sucede con las inserciones, salvo para la modificación de un elemento patrimonial, que se considera, que al modificarse, puede introducirse con el mismo nombre y siendo distinguido por el resto de sus atributos, para el resto de las modificaciones el sistema comprueba que el elemento que se pretende modificar no utilice bien una dirección Web o bien un Nick, en función del elemento que se esté modificando, que ya esté registrado en la base de datos.



#### **7.2.4.4 CONTROLES EN LAS ELIMINACIONES**

Tal y como sucede en los dos casos anteriores, en los casos en los que se requiere una eliminación de algún elemento, ya sea un sitio Web, una subpágina Web, un elemento patrimonial, o un administrador, el sistema utiliza los siguientes controladores:

##### Parte cliente

- Tal y como se ha podido observar en el apartado anterior dedicado a la explicación de las interfaces utilizadas en el sistema, en las funcionalidades de eliminación de elementos el sistema muestra listas de elementos junto con una imagen situada al lado de cada uno de los elementos que permite la eliminación de éstos. El sistema, para evitar un error de selección del usuario a la hora del borrado de un elemento pide, mediante un mensaje, la confirmación al usuario de dicha eliminación.

##### Parte servidor:

- No existen controles.

#### **7.2.5 CONTRASEÑA**

Para otorgar mayor seguridad al sistema la contraseña de cada usuario se almacena en la base de datos encriptada mediante el algoritmo md5. Para que, cuando el usuario acceda al sistema, pueda autenticarse, la contraseña introducida por el propio usuario es, también, encriptada mediante el algoritmo md5, y comparada con la contraseña que está almacenada en la base de datos (que está también encriptada).

#### **7.2.6 SESIONES**

Antes de comenzar con la explicación de las funcionalidades que ofrece la aplicación, es importante mencionar que el sistema desarrollado trabaja con sesiones. En el proyecto llevado a cabo, el trabajo con sesiones toma un papel bastante importante. En primer lugar, existen dos perfiles en la aplicación: administrador y usuario. Dependiendo del perfil que tenga asociado el individuo que accede al sistema, su potestad a la hora de realizar tareas será ilimitada o restringida. Para que el sistema conozca qué es exactamente lo que le debe permitir llevar a cabo al inquilino de la aplicación, consultará su perfil. Este dato, el perfil, es proporcionado por un atributo de la sesión, ya que cuando el inquilino se ha registrado en la aplicación la sesión ha almacenado automáticamente su perfil.

Además de almacenar el perfil del inquilino de la aplicación, almacena su Nick y su nombre, de tal forma que cuando dicho inquilino realiza algún tipo de operación en el sistema, ésta queda registrada en la base de datos, lo que permite conocer en cada

momento, qué operaciones de han llevado a cabo, y quién es el responsable de las mismas.

### **7.2.7 PERFILES**

Tal y como se ha indicado en el apartado de sesiones, y se ha explicado con mayor detenimiento en el apartado de diseño de la aplicación, el sistema cuenta con dos perfiles: administrador y usuario.

En principio salvo una serie de restricciones que aparecieron en el apartado de diseño, y que serán repetidas posteriormente ambos perfiles iban a tener potestad similar en la aplicación. Sin embargo, debido a la petición del cliente el poder del usuario en la aplicación se ha visto restringido. A continuación se indican las restricciones que tiene el usuario en la aplicación. Antes de mostrar dichas restricciones, recordad que el administrador tiene potestad para llevar a cabo cualquiera de las funcionalidades que aparecen en el sistema. Restricciones del usuario:

- No se le permite introducir ningún sitio Web, así como eliminar uno ya existente.
- A cada usuario se le asigna una lista de sitios Webs en los que puede llevar a cabo ciertas funcionalidades, al resto de los sitios Web no tendrá acceso para realizar ninguna tarea. En los sitios Web en los que tiene potestad para desempeñar tareas, no tendrá restricción ninguna.

A éstas se les unen las que ya conocíamos de la fase de diseño.

- El usuario no tendrá potestad para crear nuevos individuos en el sistema, ya sean administradores o usuarios.
- El usuario tampoco podrá modificar los datos de ningún administrador ni usuario inclusive sus datos, salvo su contraseña que sí podrá ser modificada.
- El sistema no le otorga la posibilidad al usuario de eliminar a ningún miembro del sistema, inclusive a sí mismo.

## 7.3. APLICACIÓN DE BÚSQUEDA ORIENTADA

### 7.3.1 EXPLICACIÓN DE LA APLICACIÓN DE BÚSQUEDA ORIENTADA

Para una mejor comprensión de la aplicación de búsqueda orientada que se ha llevado a cabo, se pretende describir cada uno de los módulos que aparecen a lo largo de la aplicación. Los módulos se irán explicando en el momento exacto en el que aparecen en el proceso de consulta. Con el fin de que el lector tenga un esquema visual de la arquitectura de la aplicación, y poder así seguir de manera más sencilla la explicación de los diferentes módulos de la misma, a continuación (Figura 59) se muestra un esquema de la arquitectura de la aplicación.

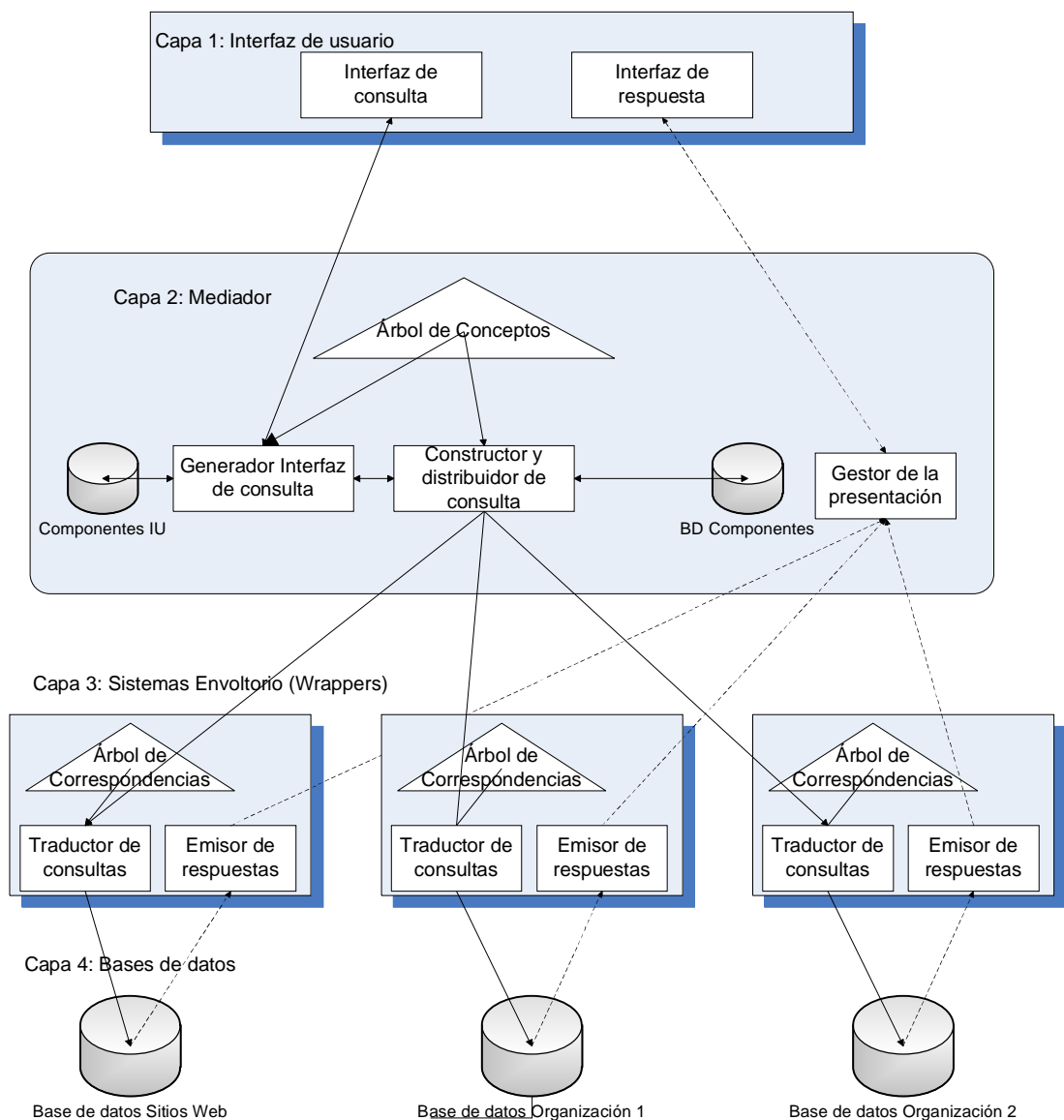


Figura 59. Arquitectura de la aplicación.

El primer módulo que aparece en la arquitectura, es el Generador de Interfaz de Consulta, módulo que se encarga, entre otras cosas, de interactuar con el usuario.

### 7.3.1.1 GENERADOR DE INTERFAZ DE CONSULTA

Este módulo, situado en capa “Mediador”, realiza diversas funciones entre las que destaca la interacción con el usuario. El Generador de Interfaz de Consulta interactúa con el usuario a través de las interfaces de consulta, las cuáles genera el propio módulo dinámicamente en función de las exigencias del usuario en cada momento. Para generar la interfaz se usa, tal y como se ha visto en el apartado de diseño, la técnica del Lenguaje Natural Acotado, una de las tres técnicas de diseño de Interfaces de Usuario. Sin embargo las otras dos, Metáforas Cognitivas y la Aproximación Navegacional, explicadas también en el apartado de diseño, debido a peticiones del cliente, no se han tenido en cuenta, aunque pueden ser aplicables en un futuro sin problema alguno.

A Continuación se puede observar (Figura 60) las capas “Interfaz de Usuario” y “Mediador”, en las cuales se sitúan los módulos Interfaz de Consulta, y Generador de Interfaz de Consulta, Árbol de Conceptos y Componentes IU respectivamente. Estos módulos son piezas básicas en esta primera fase de la arquitectura de la aplicación de búsqueda orientada.

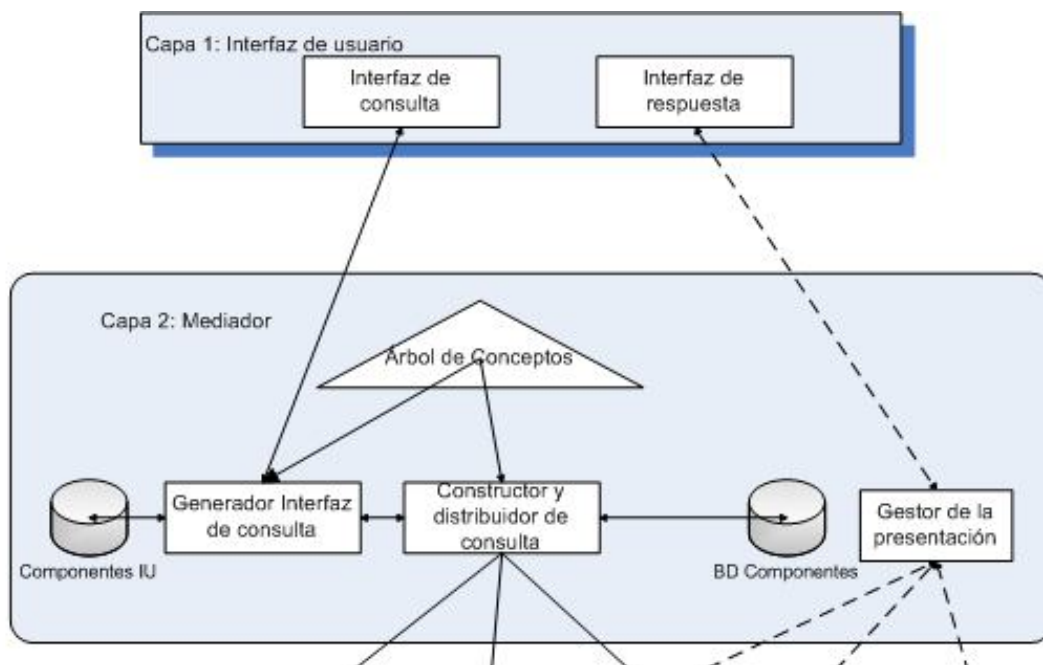


Figura 60. Capas “Interfaz de Usuario” y “Mediador” de la arquitectura general.

El Generador de Interfaz de Consulta genera las interfaces de manera dinámica basándose en el Árbol de Conceptos de la aplicación, situado también en la capa

“Mediador”, y utilizando los esqueletos de frase de Lenguaje Natural Acotado que se almacenan en el módulo IU. Cada atributo existente en el Árbol de Conceptos tiene asociado el identificador de un elemento del almacén de Componentes IU.

El Generador de Interfaz de Consulta realiza las siguientes funciones:

- Recorre el Árbol de Conceptos, mostrando en cada momento la interfaz adecuada para las peticiones del usuario.
- Recoge las restricciones del usuario sobre los atributos de la consulta. Estas restricciones son enviadas, por parte también del Generador de Interfaz de Consulta, al módulo “Constructor y Distribuidor de Consultas” quien es el encargado de construir la consulta y distribuirla.

### **Recorre el Árbol de Conceptos**

El árbol de Conceptos representa un pilar básico para nuestra aplicación, ya que en él se almacenan datos tan importantes como conceptos, relaciones de especialización entre esos conceptos, atributos... Pero además de esto, el árbol de Conceptos almacena un identificador que está relacionado con el almacén de Componentes IU, quien almacena las diferentes interfaces que son mostradas al usuario. Cabe citar que además de esto, también puede almacenar posibles valores de los atributos.

Demostrada la importancia que el árbol de Conceptos tiene en nuestra aplicación, es necesario construir un módulo que recorra dicho Árbol, recogiendo la información que en un momento dado necesite el usuario. Esto es exactamente lo que realiza el Generador de Interfaz de Consulta.

Centrándonos en la iteración del sistema con el usuario, vamos a explicar los diferentes esqueletos de frase que se han utilizado para construir las interfaces que se muestran a los usuarios. Si bien es cierto que algunos de estos esqueletos, como son aquellos que se utilizan para anotar restricciones sobre atributos, fueron explicados en la fase de diseño, existen algunos más que aún no han aparecido.

En la aplicación que se ha desarrollado se pueden realizar consultas muy variadas, ya que éstas pueden ser muy generales, por ejemplo una búsqueda general de patrimonio, o muy precisas, por ejemplo una búsqueda sobre un elemento de arquitectura civil. Debido a esto, el sistema tiene que ser capaz de otorgar al usuario la posibilidad de especializar el concepto Patrimonio (concepto general de nuestro sistema). De esa posibilidad de especialización surge la relación de Generalización/Especialización. Cada relación de Generalización/Especialización que aparece en el sistema puede llevar asociado un identificador del almacén de Componentes. Sin embargo, puede darse el caso que éste no exista, por lo que automáticamente será el esqueleto de Frase de Especialización quien trate dicha especialización.

## Esqueleto de Frase de Especialización

Este tipo de Esqueleto de Frase, tal y como se ha mencionado anteriormente, se usa con los conceptos que tienen una relación de tipo Generalización/Especialización con otros conceptos.

A continuación se muestran dos figuras que representan la explicación que se ha dado. En la primera de ellas (Figura 61) podemos observar el Esqueleto de Frase de Especialización general que se utiliza para la aplicación, y en la segunda (Figura 62) se puede apreciar un caso particular de dicho Esqueleto de Frase de Especialización general, en concreto el Esqueleto de Frase de Especialización para el concepto patrimonio.

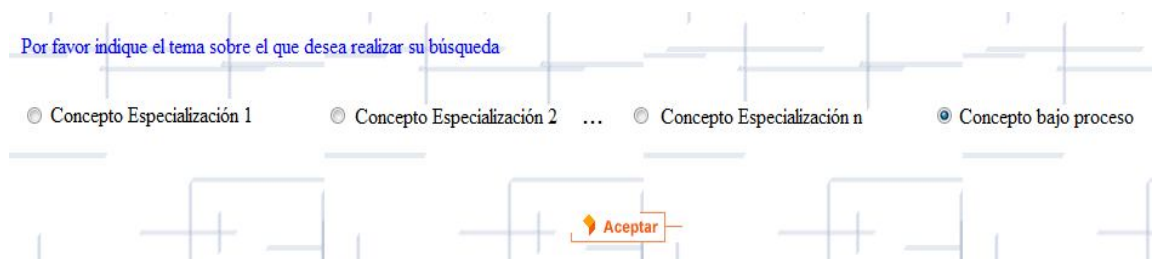


Figura 61. Esqueleto de Frase de Especialización general



Figura 62. Esqueleto de Frase de Especialización para el concepto patrimonio

Aunque se trata de un recurso bastante útil, ya que se aplica de forma automática a todos los conceptos que tienen una relación de tipo Generalización/Especialización con otros conceptos, si éstos no tienen, como hemos comentado anteriormente, asociado un identificador del almacén de Componentes puede resultar en términos de diseño demasiado sencillo, por lo que asociándole un identificador de almacén de Componentes IU se le puede asociar una Metáfora Cognitiva que nos permita elegir entre el concepto general y alguna de sus especializaciones. Si una relación de tipo Generalización/Especialización tiene asociada una Metáfora Cognitiva el esqueleto de Frase no se ejecutará.

## Esqueleto de Frase de Descripción

Además de las relaciones de Generalización/Especialización, existen otro tipo de relaciones conocidas como relaciones de Descripción. Las relaciones de Descripción

pueden darse entre conceptos o entre un concepto y diversos atributos. En nuestra aplicación se da este último caso. La relación de Descripción permite al usuario decidir sobre que atributos de los que describen el concepto, desea expresar restricciones.

Como explicación visual se presentan a continuación el Esqueleto de Frase de Descripción global que sigue la aplicación de búsqueda orientada de Patrimonio (Figura 63), y como ejemplo del mismo se muestra Esqueleto de Frase de Descripción asociado al concepto “Arquitectura civil” (Figura 64).

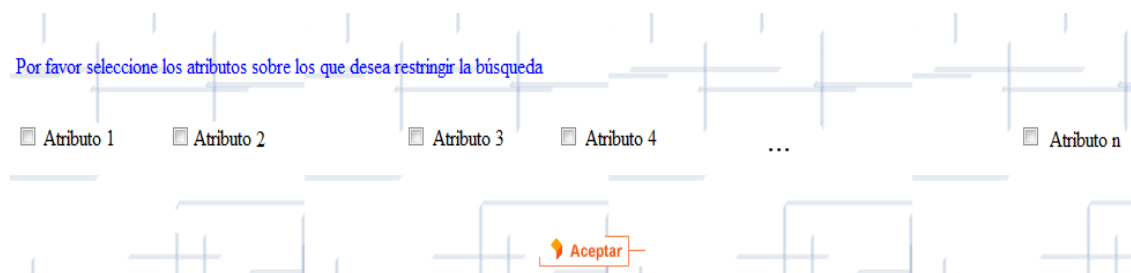


Figura 63. Esqueleto de Frase de Descripción global

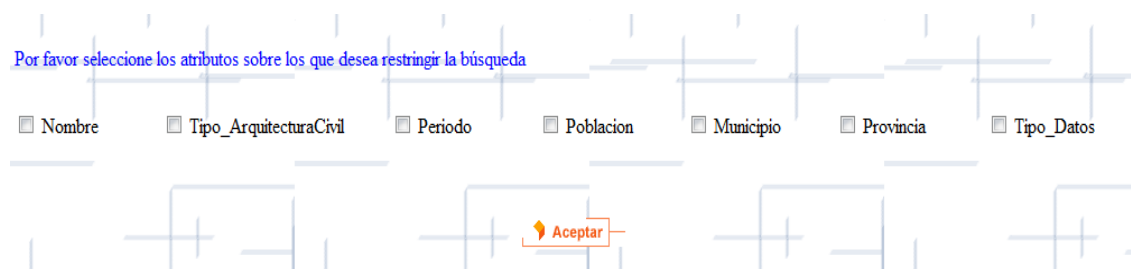


Figura 64. Esqueleto de Frase de Descripción para el concepto Arquitectura civil

### **Recoge las restricciones del usuario**

Como ya sabemos, todos y cada uno de los atributos del Árbol de Conceptos tiene asociado una frase en Lenguaje Natural Acotado. Las frases en Lenguaje Natural Acotado, ya han sido previamente explicadas en la fase de diseño. Cada una de estas frases permiten al usuario introducir las restricciones sobre el atributo que considere necesarias, y permite al sistema recoger esas restricciones, que serán almacenadas y utilizadas posteriormente para la elaboración de la consulta en lenguaje de consulta, pero esto será explicado más adelante.

### **Algoritmo del Generador de la Interfaz de Consulta**

Con el fin de detallar de manera clara y ordenada la acción que desempeña el Generador de la Interfaz de Consulta, a lo largo de este apartado se va a describir el algoritmo que sigue dicho módulo (Figura 65). En él se podrá apreciar claramente las tareas que desempeña el módulo y que ya han sido descritas en el apartado anterior.

El Generador de la Interfaz de Consulta actúa recorriendo el Árbol de Conceptos y en cada caso concreto, apoyándose en éste, va mostrando al usuario lo que éste último solicita. Para mostrar a éste lo que pide en cada momento, se apoya en el almacén de Componentes IU, en donde están diseñadas las interfaces siguiendo las técnicas ya comentadas de Lenguaje Natural Acotado y Metáforas Cognitivas.

Nada más comenzar la ejecución del módulo Generador de la Interfaz de Consulta, el concepto raíz, en nuestro caso Patrimonio, se sitúa como “Concepto Actual”. El Concepto Actual es aquel que está siendo tratado en ese momento. Tal y como se ha explicado a lo largo de este apartado de implementación de la aplicación, el concepto actual puede tener una relación de Generalización/Especialización con otros conceptos. Si el Concepto Actual no tiene esta relación, el sistema pasará a mostrarle la página de descripción asociada al mismo. Si, por el contrario, el Concepto Actual tiene una relación de Generalización/Especialización con otros conceptos, el sistema debe mostrar al usuario la página de especialización correspondiente al mismo. Esta página de especialización puede estar creada bien por Metáforas Cognitivas, o bien por el Esqueleto de Frase de Especialización que ya ha sido analizado durante este apartado de “Generador de Interfaz de Consulta”. El concepto que seleccione el usuario de los mostrados en la página de especialización será el nuevo Concepto Actual. Cabe recordar que el usuario puede escoger en la página de especialización el concepto general en vez de seleccionar uno de los conceptos de la especialización. Si el concepto seleccionado por el usuario es el concepto general, entonces éste pasará a ser el Concepto Actual, y el sistema le mostrará entonces su página de descripción. Sin embargo si el concepto seleccionado por el usuario es uno de los conceptos de especialización, dicho concepto será el nuevo Concepto Actual, y el proceso se repetirá hasta que, bien el Concepto Actual no tenga una relación de Generalización/Especialización con otros conceptos, o bien, hasta que el usuario seleccione el concepto general en la página de Generalización/Especialización.

Una vez que el usuario ya ha escogido el concepto sobre el cual quiere realizar la búsqueda, el sistema le muestra la página de descripción de dicho concepto. En ella aparecen todos y cada uno de los atributos pertenecientes al concepto seleccionado que están descritos en el Árbol de Conceptos. El usuario selecciona aquellos sobre los que quiere expresar restricciones. Estos atributos forman una lista de atributos, la cual irá vaciándose dinámicamente en función de los atributos que van siendo tratados por el usuario. Mientras la lista de atributos no esté vacía, se irá obteniendo el primer atributo de la misma, al cual se le otorga el nombre de Atributo Actual, mostrándole al usuario la página de restricción del atributo actual. Esta página de restricción está indicada en el Árbol de Conceptos y desarrolla en el almacén de Componentes IU. El usuario indica las restricciones sobre el Atributo Actual, y el sistema las captura. Este proceso se repetirá hasta que la lista de atributos esté vacía, que será cuando el algoritmo del Generador de la Interfaz de Consulta termine.



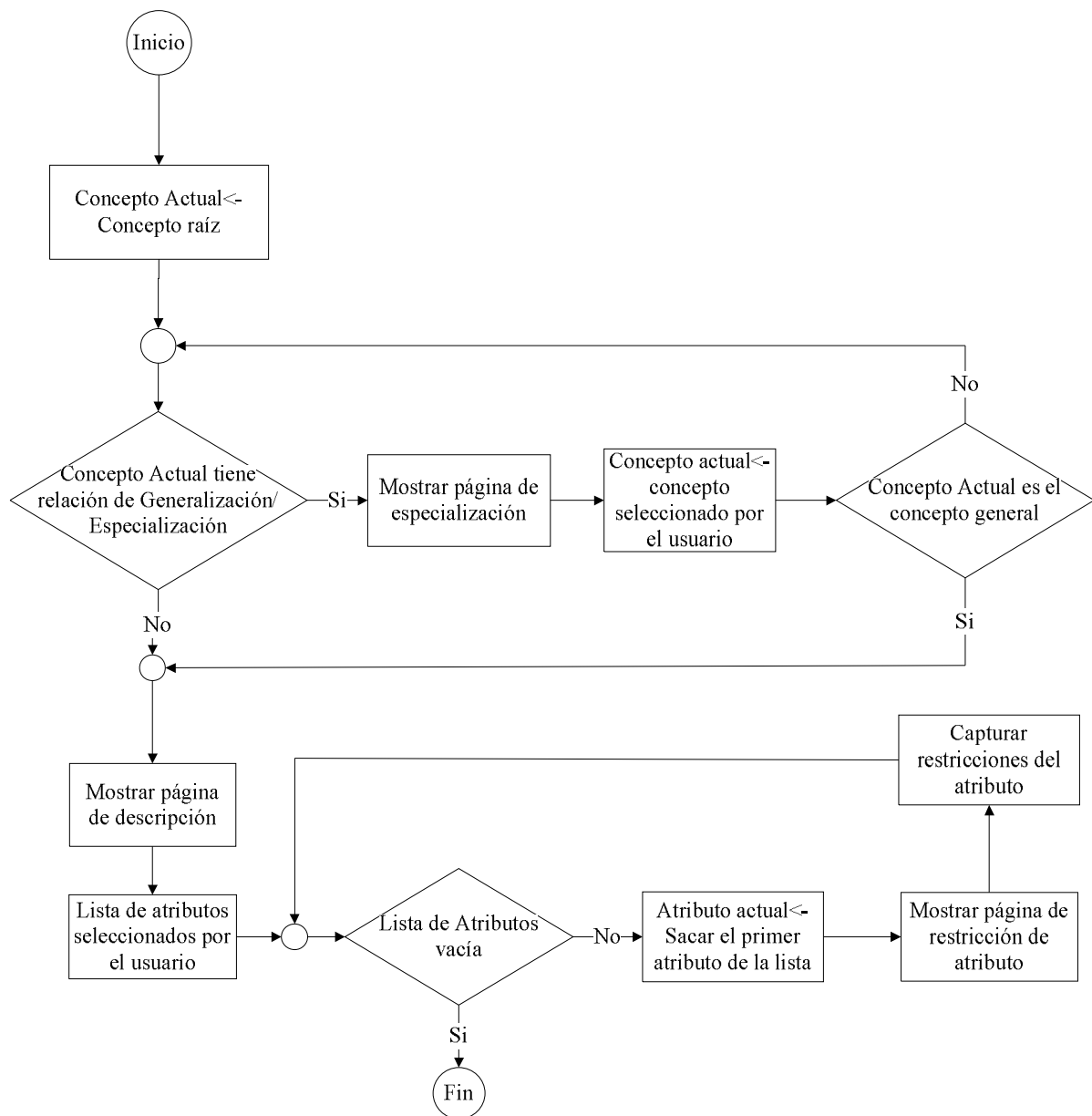


Figura 65. Diagrama de Flujo del Algoritmo del Generador de la Interfaz de Consulta

A lo largo de esta memoria se podrá observar un ejemplo de un proceso global de consulta lo que nos ayudará a comprender mejor todo el proceso, incluido también este algoritmo.

### 7.3.1.2 LENGUAJE CONSULTA, CONSTRUCTOR Y DISTRIBUIDOR DE CONSULTAS

El siguiente módulo que aparece en el proceso de consulta es el módulo Constructor y Distribuidor de consultas. Dicho módulo recibe las restricciones capturadas por el Generador de Interfaz de Consulta, las procesa y obtiene como

resultado una consulta expresada en Lenguaje Consulta, lenguaje que será explicado a continuación. La consulta obtenida es enviada a aquellos sistemas envoltorio del sistema que recojan información sobre los atributos sobre los cuales el usuario ha expresado algún tipo de restricción.

Durante este apartado se llevará a cabo una explicación de las funciones que realiza el módulo Constructor y Distribuidor de consultas, así como un análisis con detenimiento de lo que hemos denominado Lenguaje Consulta.

### **7.3.1.2.1 LENGUAJE CONSULTA**

La aplicación que se desarrolla puede estar compuesta por varios sistemas envoltorio, cada uno de ellos con su fuente de datos. Estas fuentes de datos son, obviamente, diferentes entre ellas, por lo que es imposible generar un lenguaje de base de datos común para todas ellas. Debido a esto, es necesario generar un lenguaje intermedio para todos los sistemas envoltorio, de tal forma que cada sistema envoltorio sea capaz de transformar dicho lenguaje, al lenguaje natural de la base de datos existente dentro de su sistema envoltorio.

El lenguaje general que se necesita generar, es el que a lo largo de esta aplicación se denomina Lenguaje de Consulta. Para obtener el Lenguaje Consulta nos basamos en las decisiones tomadas por el usuario en el módulo Generador de Interfaz de Consulta. De éstas obtenemos el concepto o los conceptos, en función si existe, o no, en la consulta solicitada por el usuario una relación de Generalización/Especialización, sobre los que realiza la consulta, los atributos que han sido seleccionados para expresar condiciones sobre dichos conceptos y las restricciones de dichos atributos. Es decir, el usuario describe conceptos a partir de los atributos que dicho concepto tiene asociado en el árbol de Conceptos.

El Lenguaje Consulta queda recogido en un documento XML, que mantiene la posición relativa que tienen los conceptos y atributos dentro del Árbol de Conceptos. Esto es bastante obvio, ya que la consulta realizada es un subárbol del Árbol de Conceptos. En dicho subárbol sólo aparecen aquellos conceptos y atributos que han sido seleccionados por el usuario para elaborar su consulta.

Este lenguaje de Consultase formaliza a través del DTD Consulta que se presenta en Anexo B.3.

En definitiva, el Lenguaje Consulta es un lenguaje intermedio que consigue dar forma a las condiciones expresadas por el usuario sobre los conceptos y atributos del Árbol de Conceptos, y es interpretado por cada sistema envoltorio, quien, como veremos más adelante, gracias a su módulo traductor, traduce dicho lenguaje Consulta a el lenguaje propio de su base de datos.

### **7.3.1.2.2 MÓDULO CONSTRUCTOR Y DISTRIBUIDOR DE CONSULTAS**

Módulo situado, al igual que el Generador de Interfaz de Consulta, en la capa “Mediador”. Este módulo recibe del propio Generador de Interfaz de Consulta las condiciones de consulta expuestas por el usuario, y, a partir de ellas, con ayuda del propio Árbol de Conceptos genera la consulta en Lenguaje Consulta, la cual será enviada a los diferentes sistemas envoltorios siempre y cuando estos tengan en su base de datos alguno de los atributos implicados en la consulta.

Para construir el Lenguaje Consulta, el módulo Constructor y Distribuidor de Consultas recibe del módulo Generador de Interfaz de Consulta una lista con las condiciones de consulta expresadas por el usuario. Una vez, que recibe dicha lista, obtiene dinámicamente el primer elemento de la lista y lo procesa, es decir, analiza si el elemento de la lista es un concepto, una relación de Generalización/Especialización, o un atributo, y en función del elemento que sea actúa de una forma u otra. El proceso se repite hasta que la lista queda completamente vacía. En la figura inferior (Figura 66) se puede observar el algoritmo que sigue el módulo Constructor y Distribuidor de consultas para generar la consulta en lenguaje consulta.

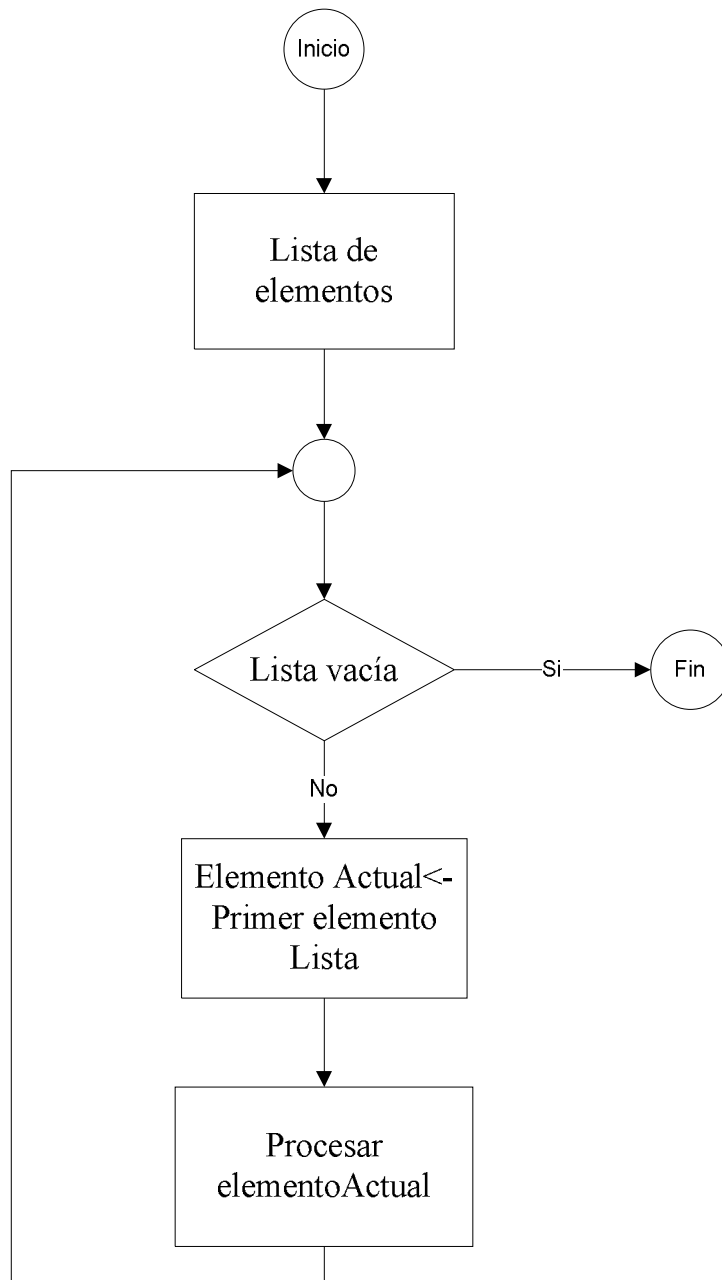


Figura 66. Diagrama de Flujo Algoritmo del módulo Constructor y Distribuidor de consulta.

Una vez que el módulo ha construido la consulta en Lenguaje Consulta, ésta debe ser enviada a los sistemas envoltorio de la aplicación. Sin embargo, no será enviada a todos los sistemas envoltorio, sino a aquellos cuya base de datos contenga alguno de los atributos implicados en la consulta. Tal y como se avanzó en el apartado de diseño, cada atributo del Árbol de Conceptos tiene asociado en dicho Árbol una lista con los identificadores de las bases de datos en las que está presente. Debido a esto, mientras el módulo Constructor y Distribuidor de consultas va construyendo la consulta en lenguaje consulta, a su vez anota para cada atributo las bases de datos en las que está

presente. Una vez que todos los atributos son procesados, la consulta será enviada a aquellas bases de datos que tengan al menos un atributo de los implicados en la consulta. Para conocer esto basta con unir la lista de bases de datos de cada atributo, y enviársela a aquellas que al menos tengan un elemento en la lista. Podemos ejemplificar esto de la siguiente forma. Si existen tres bases de datos en el sistema y la consulta expresada por el usuario es la búsqueda de edificios con nombre “San Martín” y población “Vitoria-Gasteiz”. Supongamos, sobre atributos del concepto general edificios, que la primera base de datos BD1, tiene información sobre ambos atributos, la segunda base de datos BD2 tiene información únicamente sobre el atributo nombre, mientras que la tercera BD3 no tiene información sobre ninguno de los atributos. Por lo tanto, atributo nombre {BD1, BD2} y atributo población {BD1}. En definitiva, el módulo Constructor y Distribuidor de consultas enviará la consulta a las bases de datos BD1, Y BD2, en BD1 la consulta sobre edificios se realizará con las restricciones dadas por los atributos nombre y población, y el BD2 dicha consulta se llevará a cabo únicamente con la restricción del atributo nombre.

Cabe citar en este punto, que existe otra posibilidad en el envío de la consulta a los sistemas envoltorios, y esta no es otra que enviar la consulta únicamente a aquellas bases de datos que tengan todos y cada unos de los atributos que están implicados en la consulta. Siguiendo con el ejemplo anterior, con esta posibilidad de envío la consulta sería enviada únicamente a la bases de datos BD1. Esta opción es más restrictiva que la anterior y se considera que en vez de favorecer una búsqueda más eficiente de un elemento patrimonial, lo que va a hacer es limitar el resultado de la consulta expresada por el usuario, por lo que puede perderse información realmente interesante acerca de la misma.

### **7.3.1.3 TRADUCCIÓN DE CONSULTAS**

Una vez que el sistema envía la consulta, en Lenguaje Consulta, a las bases de datos implicadas en la misma, el proceso siguiente es el de traducir, en cada sistema envoltorio, la consulta de lenguaje consulta al lenguaje natural de cada base de datos. Para ello contaremos básicamente con dos módulos: el Árbol de Correspondencias y el módulo Traductor. A lo largo de este apartado se explicará el proceso de traducción de la consulta, indicando los módulos que aparecen en él y la tarea que desempeñan.

#### **7.3.1.3.1 ALGORITMO DE TRADUCCIÓN**

La traducción de la consulta se lleva a cabo en cada uno de los sistemas envoltorios de las diferentes bases de datos implicadas en la consulta. Para recordar los módulos existentes en un sistema envoltorio determinado, a continuación se muestra una imagen (Figura 67) de un sistema envoltorio del sistema con sus módulos.

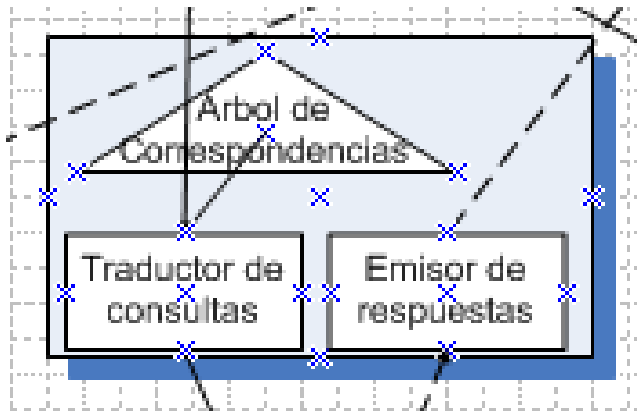
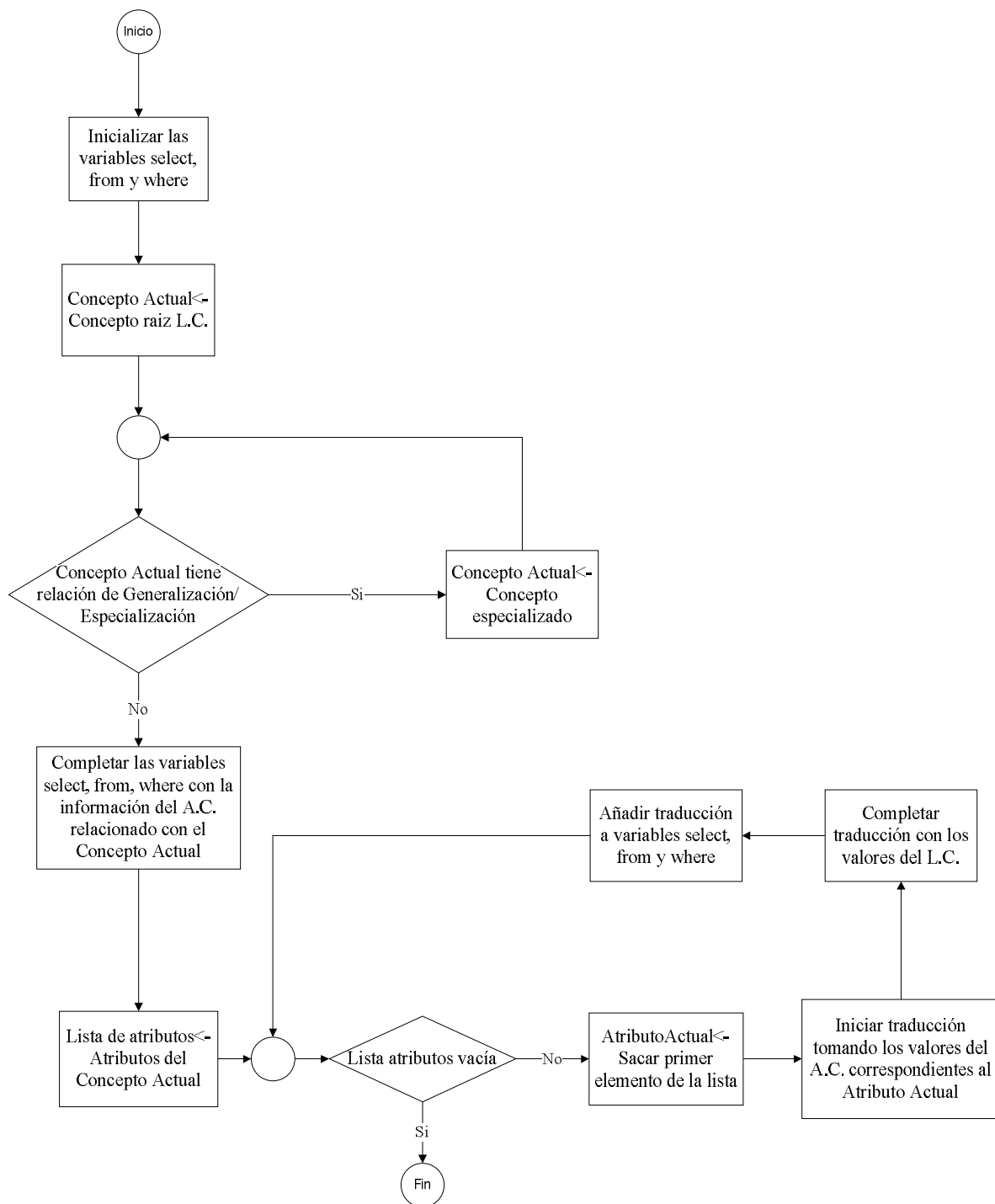


Figura 67. Sistema envoltorio

El módulo encargado de transformar la consulta XML al lenguaje propio de la base de datos del sistema envoltorio es el traductor de consultas. Para llevar a cabo esta traducción, el traductor de consultas toma información de la consulta que llega a su módulo en lenguaje de consulta, y apoyándose en el Árbol de Correspondencias consigue transformar dicha consulta, en otra legible para su base de datos.

A continuación se muestra el algoritmo de traducción de consultas, Figura 68, que posteriormente será explicado con mayor detenimiento. El algoritmo nos permite comprobar que la traducción se realiza mediante la información que llega en Lenguaje Consulta y su correspondiente valor tomado del Árbol de Correspondencia



\*L.C.-> Lenguaje consulta \*A.C.-> Árbol de Correspondencias

Figura 68. Diagrama de flujo del algoritmo de traducción de consultas

Para conseguir la traducción, el traductor de consultas deberá recorrer el documento XML que contiene la consulta en Lenguaje Consulta, y asignar para cada elemento su correspondencia, que está escrita en el Árbol de Correspondencias. Por esto, la primera tarea que lleva a cabo el traductor es buscar el concepto sobre el que el

usuario ha realizado la consulta. Una vez encontrado éste, toma del Árbol de Correspondencias la información que tiene asociada dicho concepto (Concepto Actual) y completa las variables select, from y where que serán utilizadas para realizar la consulta. Además recoge del lenguaje consulta todos los atributos del Concepto Actual sobre los que el usuario ha hecho referencia en su consulta.

Una vez que ha formado una lista con dichos atributos, los procesa uno a uno. El proceso de los atributos es similar al seguido por el Concepto Actual, es decir, se busca la información asociada al atributo en el Árbol de Correspondencias. La diferencia existente entre el Concepto Actual y los atributos, es que en estos últimos el usuario ha expresado una condición que, obviamente, no está registrada en el Árbol de Correspondencia y habrá que procesarla.

Además de procesar los datos introducidos por el usuario como condiciones de los atributos, es necesario también procesar el operador escogido por éste, ya que tal y como se ha indicado en la explicación de las interfaces del sistema, la aplicación proporciona al usuario la posibilidad de escoger entre diferentes opciones de consulta (datos comprendidos entre dos valores, datos que sean exactamente iguales que las palabras introducidos, o que simplemente las contengan, datos cuyo periodo sea mayor a uno indicado...). Esta información se obtiene del lenguaje de consulta, se procesa y la información se añade a las variables select, from y where.

Terminados todos los atributos, la consulta está preparada para ser enviada a la base de datos del sistema envoltorio.

## **Sinónimos**

Con el fin de facilitar la búsqueda al usuario se ha desarrollado un documento XML, que almacena posibles sinónimos de condiciones expresadas por el usuario sobre los atributos durante el proceso de consulta. Se trata de una funcionalidad muy útil sobre todo para listas desplegables ya que para un posible valor de la lista pueden asociarse diversos valores en función de lo que se quiera expresar. Por ejemplo, para el atributo “tipo de datos” nuestro sistema muestra una lista desplegable con posibles valores, entre los que se encuentra imágenes. Sin embargo, es obvio que dependiendo de la base de datos, el campo “tipo de datos” puede tomar valores como imagen, fotografía, foto, incluso dibujo... Todas ellas palabras válidas, que sin embargo, nuestro buscador no las encontraría. En cambio, teniendo en cuenta la lista de sinónimos, el buscador sí conseguiría mostrar al usuario esas soluciones.

### **7.3.1.4 PRESENTACIÓN DE RESULTADOS**

Una vez conseguidos todos los resultados de las diferentes bases de datos, es el momento de poner la información en común y presentársela al usuario.

Debido a que los datos proceden de fuentes de datos diferentes, es necesario, antes de mostrar los datos obtenidos, presentar dichas fuentes. La información de las



fuentes de datos fue obtenida en el proceso de construcción y distribución de consultas cuando se decidió los sistemas envoltorio a los que había que enviar la consulta expresada por el usuario.

Por lo tanto en esta primera página de presentación se mostrará al usuario todas las bases de datos que han participado en la consulta, describiendo para cada una de ellas:

- Nombre de la organización a la que pertenece la base de datos.
- Su dirección de página Web, si es que existe una página que albergue dichos datos.
- La descripción de la base de datos. Qué datos contiene y en qué consisten esos datos.
- Los atributos que han participado en la consulta sobre dicha base de datos. Esto es, si por ejemplo existen atributos sobre los que el usuario ha expresado condiciones en su consulta, que no existen en la base de datos, en vez de no procesar la consulta, se suprime dicho atributo y se realiza la consulta teniendo en cuenta el resto de los atributos seleccionados por el usuario.

Además por cada solución de base de datos el sistema indica el número de soluciones que se ha obtenido, a las que a través de un enlace se podrá acceder.

La página donde se muestra las soluciones obtenidas de cada base de datos, es completamente diferente en función de la base de datos seleccionada. Los dos modelos desarrollados ya, para este proyecto son completamente diferentes. Mientras que en uno, el de las páginas Web, describe la organización y la subpágina Web de donde se obtiene la información en cada caso, además del sitio Web que recoge dicha subpágina Web y una descripción del mismo, en el otro, base de datos de LDGP, se muestran los proyectos relacionados con la consulta y un enlace a la página Web donde se encuentran dichos proyectos.

Por último cabe citar que no se pueden dar resultados duplicados en las consultas realizadas por el usuario.

### 7.3.2 EJEMPLO DE PROCESO DE CONSULTA

Una vez llevada a cabo una explicación detallada de los módulos más importantes que aparecen en el proceso de consulta es el momento de ejemplificar dicha explicación con un ejemplo de proceso de consulta. El proceso de consulta comienza cuando el usuario accede al sistema y termina cuando los datos son mostrados al mismo.

Por lo tanto, manteniendo la estructura del apartado anterior (Implementación de la solución) a continuación se desarrollará un ejemplo completo del proceso de consulta.

La consulta que llevará el usuario a cabo será la siguiente: *“Elementos de arquitectura religiosa cuyo nombre contiene las palabras ‘santa maría’, cuya población es Vitoria y provincia es Álava”*

#### 7.3.2.1 LENGUAJE CONSULTA, CONSTRUCTOR Y DISTRIBUIDOR DE CONSULTAS

Cuando el usuario accede al sistema, tal y como se muestra en el algoritmo de la Figura 65, el Generador de Interfaz de Consulta comienza leyendo el concepto raíz del Árbol de Conceptos (en nuestro sistema es el concepto Patrimonio). Siguiendo el algoritmo del Generador de Interfaz de Consulta el sistema comprueba en el árbol de concepto si este concepto tiene una relación de Generalización/ Especialización. Como el concepto patrimonio tiene una relación de este tipo, el sistema le muestra la página de Generalización/Especialización asociada al concepto. En este caso, el concepto Patrimonio tiene asociada una página que ha sido diseñada exclusivamente para él (ver identificador asociado al concepto Patrimonio en el Árbol de Concepto, en el anexo B.1.2), sin embargo si no existiera tal página, el sistema le mostraría por defecto el esqueleto de frase asociado a las relaciones de Generalización/ Especialización, Figura 69.

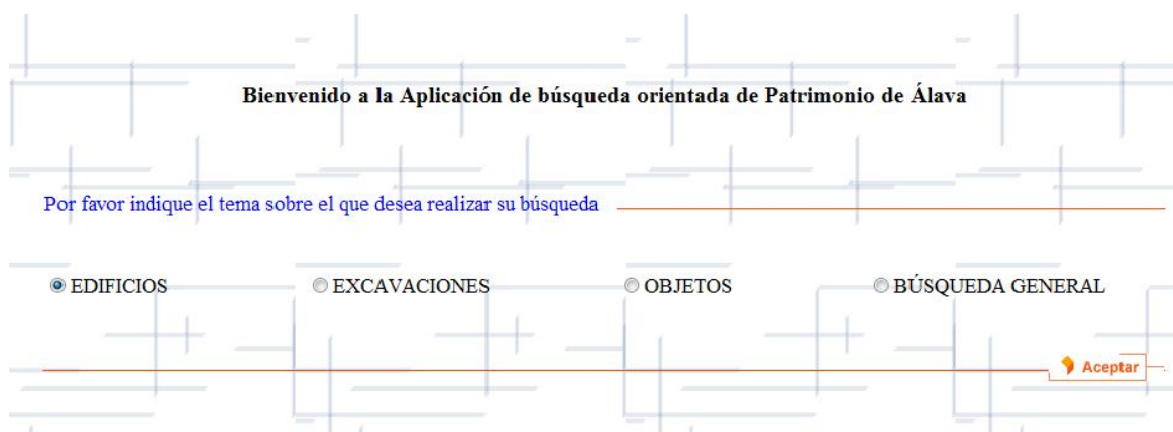


Figura 69. Página asociada a la relación de Generalización/ Especialización del concepto Patrimonio.

Cuando el sistema muestra la Figura 69 al usuario, éste debe decidir si escoger el concepto general, o elegir una de sus especializaciones. En el caso concreto de la consulta que estamos ejemplificando, el usuario escogería el concepto “Edificios”, ya que en él se encuentra la especialización “Arquitectura Religiosa”, la cual será el concepto escogido por el usuario para llevar a cabo la consulta.

Una vez que el usuario ha seleccionado el concepto “Edificios” el proceso, como puede observarse en el algoritmo de Generador de Interfaz de Consulta, se repite con el único cambio de que el Concepto Actual es el concepto seleccionado por el usuario (en este caso “Edificios”). Al igual que en el caso anterior, el sistema busca en el árbol de Concepto si el Concepto Actual tiene una relación de Generalización/ Especialización. Al ser así, el sistema busca en dicho árbol de Conceptos si dicho concepto tiene un identificador asociado a una página concreta. En caso afirmativo, como es nuestro caso, el sistema muestra la página asociada (Figura 70).

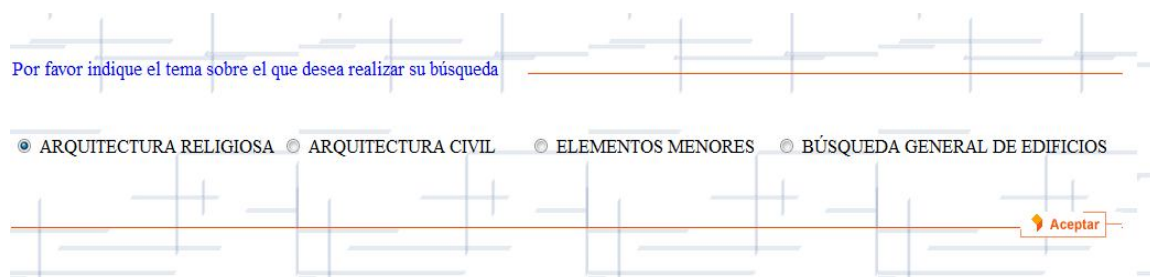


Figura 70. Página asociada a la relación de Generalización/ Especialización del concepto Edificios.

El sistema recoge la elección del usuario. Ésta, como en el caso anterior, puede ser bien el concepto general, o una de las especializaciones. En el caso de nuestra consulta, el usuario elige la especialización “Arquitectura Religiosa”, y como no es el concepto general, el proceso se repite siendo “Arquitectura Religiosa” el Concepto Actual.

El proceso continúa y el sistema busca en el árbol de Concepto si el Concepto Actual tiene una relación de Generalización/ Especialización. En este caso “Arquitectura Religiosa” no tiene relación de ese tipo, por lo que el sistema directamente muestra la página de descripción del Concepto Actual, Figura 71.

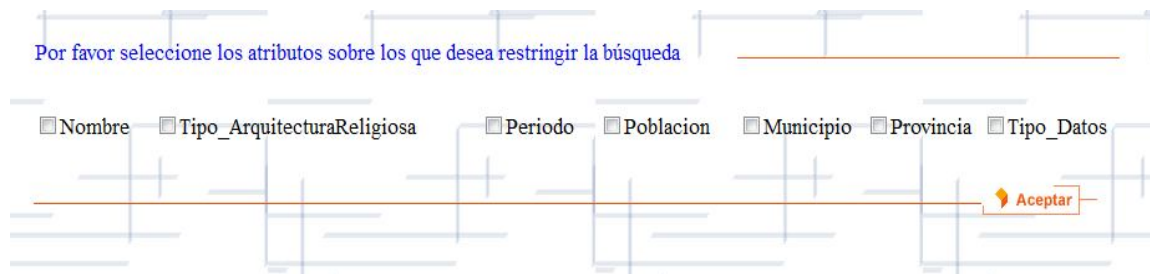


Figura 71. Página de descripción del concepto “Arquitectura Religiosa”.

Esta página es construida utilizando el Esqueleto de Frase de Descripción (Figura 63) que ha sido explicado a lo largo del apartado anterior. Esta página ofrece al usuario la posibilidad de seleccionar aquellos atributos con los que pretende describir el concepto “Arquitectura Religiosa”. Los atributos seleccionados son los que permitirán al usuario expresar las características del concepto “Arquitectura Religiosa” que quiere restringir para describir los elementos de arquitectura religiosa que está buscando.

En el caso de la consulta que se está desarrollando, los atributos seleccionados por el usuario serán: “Nombre”, “Población” y “Provincia”.

Siguiendo con el algoritmo del Generador de Interfaz de Consulta, una vez que el usuario ha seleccionado los atributos sobre los que va a mostrar restricciones, éstos son almacenados en una lista, y serán procesados uno por uno hasta que dicha lista quede vacía.

Continuando con nuestro ejemplo, el primer atributo en sacar de la lista es el atributo “Nombre”, el cual será el Atributo Actual. El sistema busca en el Árbol de Conceptos el identificador que tiene asociado el atributo “Nombre” para mostrar al usuario la página correspondiente. Para dicho atributo la página de descripción es la que se muestra en la Figura 72.

The image shows a web form for setting restrictions on the 'Nombre' attribute. At the top, there is a blue instruction: "Por favor indique la restricción que desee". Below this, the text "El atributo Nombre {" is followed by two radio button options: "debe de ser exactamente" (unselected) and "debe contener la siguiente serie de caracteres" (selected). To the right of the second option is a text input field containing "Santa María". At the bottom center of the form is a red button with a white arrow and the text "Aceptar".

Figura 72. Página de descripción del atributo “Nombre”.

Como se observa en la Figura 72, el usuario ha rellenado la restricción del atributo “Nombre”, indicando que el nombre debe contener los caracteres “Santa María”. Una vez que el usuario expresa las restricciones sobre el atributo, el sistema las almacena, y el proceso vuelve a repetirse con el siguiente atributo, pasando a ser el atributo “Población” el Atributo Actual. Como el proceso es exactamente igual para los otros dos atributos, únicamente se van a mostrar la página de descripción de cada uno de ellos.

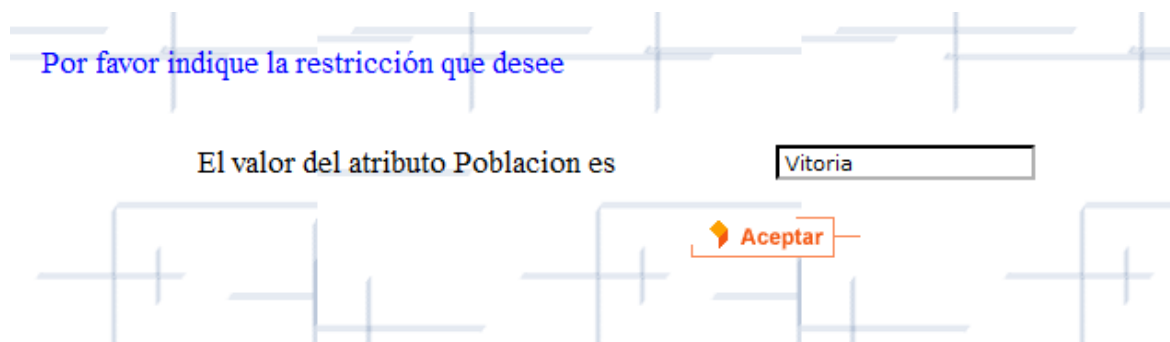


Figura 73. Página de descripción del atributo “Población”.

En este caso, tal y como se observa en la Figura 73, el usuario ha rellenado la restricción del atributo “Población”, indicando que su valor es “Vitoria”.

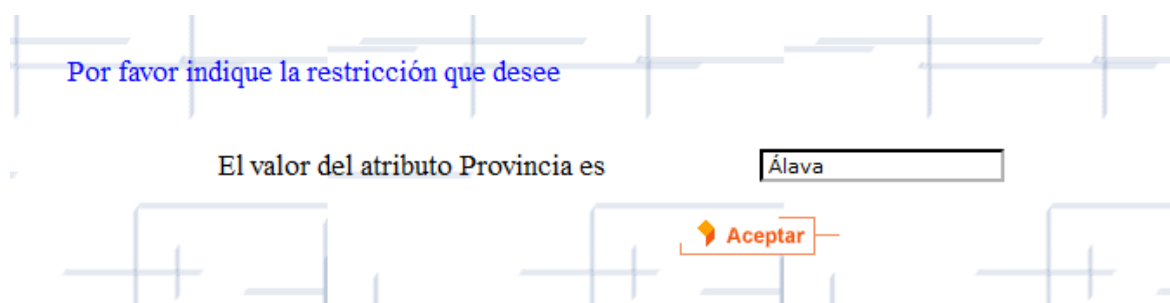


Figura 74. Página de descripción del atributo “Provincia”.

Finalmente, el último atributo, “Provincia” ha sido restringido con el valor “Álava”, tal y como se observa en la Figura 74.

Una vez que todos los atributos han sido procesados y la lista está vacía, el sistema envía todos los datos recogidos al módulo constructor y distribuidor de consulta.

### 7.3.2.2 CONSTRUCTOR Y DISTRIBUIDOR DE CONSULTAS

La información recogida por el sistema sobre la consulta diseñada por el usuario llega al módulo “Constructor y Distribuidor de consultas”. Dicho módulo será el encargado de procesar esa información y plasmarla en una nueva consulta en lo que hemos denominado “Lenguaje Consulta”. Durante este apartado se explicará, para nuestro caso particular, cómo el sistema lleva a cabo este proceso, y consigue, a partir de la información recogida, elaborar una consulta y enviarla a las bases de datos que tengan atributos que estén implicados en la misma.

Siguiendo el algoritmo del módulo Constructor y Distribuidor de consulta (Figura 66). El módulo procesa, de manera individual, cada elemento de la información que recibe del módulo Generador de Interfaz de Consulta. Lo primero que procesa son los conceptos, en nuestro caso concreto, procesaría el concepto “Patrimonio” obteniendo el siguiente resultado.

```

<consulta>
  <concepto nombre="Patrimonio">
    ...
  </concepto>
</consulta>

```

Figura 75. Primer paso de la construcción de la consulta.

Continuando con el algoritmo, el siguiente elemento que nos encontraríamos sería la relación de Generalización/ Especialización, y posteriormente el concepto “Edificios”. El sistema procesaría ambos casos presentando el siguiente resultado.

```

<consulta>
  <concepto nombre="Patrimonio">
    <isa>
      <concepto nombre="Edificios">
        ...
      </concepto>
    </isa>
  </concepto>
</consulta>

```

Figura 76. Segundo paso de la construcción de la consulta.

Siguiendo el mismo proceso, el sistema procesará la relación de Generalización/ Especialización del concepto Edificios, y posteriormente el concepto “Arquitectura Religiosa”, dando lugar al siguiente resultado.

```

<consulta>
  <concepto nombre="Patrimonio">
    <isa>
      <concepto nombre="Edificios">
        <isa>
          <concepto nombre="Arquitectura Religiosa">
            ...
          </concepto>
        </isa>
      </concepto>
    </isa>
  </concepto>
</consulta>

```

Figura 77. Tercer paso de la construcción de la consulta.

Una vez procesados los conceptos, es el turno de los atributos. El proceso es similar ya que se obtiene un atributo de la lista y se procesa, y así sucesivamente hasta que no queden más atributos en la lista. El primer atributo en procesarse será el atributo "Nombre", cuyo resultado del proceso será el que aparece en la figura XXXXX.

```
<consulta>
  <concepto nombre="Patrimonio">
    <isa>
      <concepto nombre="Edificios">
        <isa>
          <concepto nombre="Arquitectura Religiosa">
            <atributo nombre="Nombre">
              <operacionbinario op="like"/>
              <valor constante="Santa Maria"/>
            </atributo>
            ...
          </concepto>
        </isa>
      </concepto>
    </isa>
  </concepto>
</consulta>
```

Figura 78. Cuarto paso de la construcción de la consulta.

El mismo proceso se repetirá para los atributos restantes ("Población" y "Provincia"), con el siguiente resultado.

```

<consulta>
  <concepto nombre="Patrimonio">
    <isa>
      <concepto nombre="Edificios">
        <isa>
          <concepto nombre="Arquitectura Religiosa">
            <atributo nombre="Nombre">
              <operacionbinario op="like"/>
              <valor constante="Santa Maria"/>
            </atributo>
            <atributo nombre="Poblacion">
              <operacionbinario op="like"/>
              <valor constante="Vitoria"/>
            </atributo>
            <atributo nombre="Provincia">
              <operacionbinario op="like"/>
              <valor constante="Alava"/>
            </atributo>
          </concepto>
        </isa>
      </concepto>
    </isa>
  </concepto>
</consulta>

```

Figura 79. Quinto paso de la construcción de la consulta.

Una vez procesada toda la información recogida por el sistema, en el módulo de Generador de Interfaz de Consulta sobre la selección de consulta por parte del usuario, la consulta (en Lenguaje Consulta) está preparada para ser enviada a los diferentes sistemas envoltorio. Sin embargo, la consulta será enviada únicamente a aquellos sistemas envoltorio cuyas bases de datos contengan al menos uno de los atributos que están implicados en la consulta.

### **Envío de la consulta a los sistemas envoltorio**

Con el fin de acelerar el proceso de consulta, y evitar perder tiempo procesando consultas en sistemas envoltorios que no tienen ningún atributo implicado en la consulta, durante el proceso de construcción de la consulta, el sistema va anotando por para cada atributo las bases de datos en las que aparece.

Para cada atributo repite el mismo proceso, de tal forma que una vez hayan sido procesados todos los atributos, conoceremos para cada base de datos los atributos que están implicados en la consulta. Basta con que una base de datos determinada tenga un único atributo de los implicados en la consulta, para que la consulta sea enviada también a su sistema envoltorio.



En nuestro caso particular de consulta, tenemos, por un lado dos bases de datos relacionales en el sistema, y tres atributos implicados en la consulta, “Nombre”, “Población” y “Provincia”. Si observamos el Árbol de Conceptos de nuestro sistema (Anexo B.1.2) nos damos cuenta que los atributos “Nombre”, “Población” y “Provincia” del concepto “Arquitectura Religiosa”, están presentes en las dos bases de datos del sistema, por lo que la consulta será enviada a los sistemas envoltorios de ambas bases de datos.

### Otra posible consulta

La consulta desarrollada como ejemplo, ha sido una consulta de cierta complejidad ya que se ha especializado dos conceptos como son “Patrimonio” y “Edificios”. Por esto podemos decir que el usuario ha decidido realizar una consulta apoyándose en la “parte experta” del Árbol de Conceptos. La parte experta albergaría todas aquellas consultas que se produjeran mediante la especialización del concepto “Patrimonio”.

Sin embargo, el sistema permite realizar búsquedas más globales sobre patrimonio sin la necesidad de especializar dicho concepto. Esta sería la parte que denominamos, “parte general” del Árbol de Conceptos. Una posible consulta general sería buscar todas las *“imágenes que hay sobre el patrimonio en la población de Ocio”*. La consulta quedaría de la siguiente forma.

```
<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
  <consulta>
    <concepto nombre="Patrimonio">
      <atributo nombre="Poblacion">
        <operacionbinario op="like"/>
        <valor constante="Ocio"/>
      </atributo>
      <atributo nombre="Tipo_Datos">
        <operacionbinario op="like"/>
        <valor constante="imagenes"/>
      </atributo>
    </concepto>
  </consulta>
```

Figura 80. Consulta alternativa.

### 7.3.2.3 TRADUCCIÓN DE CONSULTAS

El módulo Constructor y Distribuidor de consultas envía la consulta a los sistemas envoltorio que tengan en la base de datos al menos un atributo implicado en la consulta. Dentro del sistema envoltorio es el módulo Traductor de Consultas quien recoge la consulta, y con la ayuda del Árbol de Correspondencias, será quien la procese. La tarea de dicho módulo es traducir la consulta en Lenguaje Consulta al lenguaje

propio de la base de datos existente en su sistema envoltorio. En nuestro caso contamos con dos bases de datos relacionales, por lo que el lenguaje de la base de datos en ambos casos será SQL. Para llevar a cabo la tarea de la traducción el Traductor lee secuencialmente la consulta en lenguaje consulta, y va asignando para cada concepto o atributo que aparece en ella su correspondencia, que está almacenada en el Árbol de Correspondencias del sistema envoltorio.

Para nuestro caso particular, se va a analizar la traducción que se lleva a cabo en el sistema envoltorio que contiene la base de datos del Laboratorio de Documentación Geométrica del Patrimonio de la Universidad del País Vasco (UPV/EHU). El Árbol de Correspondencias para esta base de datos está situado en el Anexo B.2.2.2.

Para llevar a cabo la traducción de la consulta en lenguaje consulta al lenguaje de la base de datos del Laboratorio de Documentación Geométrica del Patrimonio de la Universidad del País Vasco (UPV/EHU) nos basaremos en el algoritmo de traducción de consultas de la Figura 68. El primer paso es inicializar las variables “select”, “from” y “where” que serán utilizadas durante el proceso de traducción. El siguiente paso es encontrar el concepto exacto sobre el que se hace la consulta. Para esto, el sistema lee el primer elemento del lenguaje consulta (Concepto Actual), en el caso de nuestra consulta “Patrimonio”, y comprueba si éste tiene una relación de Generalización/Especialización. Al ser así, el concepto especializado pasa a ser el Concepto Actual, en nuestro caso “Edificios”. El proceso se repite, y el Concepto Actual pasa a ser “Arquitectura Religiosa”. Dicho concepto no tiene ninguna relación de Generalización/Especialización, por lo que es procesado por el sistema.

Para procesarlo el módulo Traductor asigna para el concepto “Arquitectura Religiosa” el valor correspondiente asociado a dicho concepto en el Árbol de Correspondencias. Citar que Información de Correspondencia del concepto “Arquitectura Religiosa” está dividida en la parte que hay que añadir a la variable select, a la variable from y a la variable where.

LENGUAJE DE CONSULTA	LENGUAJE SQL
<pre data-bbox="331 1570 826 1653">&lt;concepto nombre="Arquitectura Religiosa"&gt; ... &lt;/concepto&gt;</pre>	<pre data-bbox="858 1570 1418 1675">SELECT DISTINCT Proyectos.'Nombre del proyecto' FROM Proyectos WHERE (Proyectos.tipo="ermita") or (Proyectos.tipo="iglesia") or (Proyectos.tipo="monasterio")</pre>

Figura 81. Paso 1. Traducción del concepto “Arquitectura Religiosa”.

Una vez procesado el Concepto Actual, “Arquitectura Religiosa” en nuestro caso, el siguiente paso es procesar cada uno de los atributos. A diferencia de los conceptos, el proceso de traducción de los atributos consta de tres pasos.

- En primer lugar, al igual que en la traducción de los conceptos, el sistema toma del Árbol de Correspondencias el valor asociado al atributo que está procesando, Atributo Actual.
- En segundo lugar se sustituyen los parámetros que aparezcan en la Información de Correspondencia del atributo por los valores concretos que estén almacenados en la consulta en lenguaje de consulta. En el caso concreto de nuestra consulta, #op# y %#valor#%.
- Por último, una vez completados correctamente los pasos anteriores, se completa la variable where con información del Atributo Actual.

El primer atributo que se procesa es el atributo “Nombre”, el cual sigue el proceso descrito anteriormente, otorgando el siguiente resultado.

LENGUAJE DE CONSULTA	LENGUAJE SQL
<pre> &lt;concepto nombre="Arquitectura Religiosa"&gt;   &lt;atributo nombre="Nombre"&gt;     &lt;operacionbinario op="like"/&gt;     &lt;valor constante="Santa Maria"/&gt;   &lt;/atributo&gt;   ... &lt;/concepto&gt; </pre>	<pre> SELECT DISTINCT Proyectos.'Nombre del proyecto' FROM Proyectos WHERE ((Proyectos.tipo="ermita") or (Proyectos.tipo="iglesia") or (Proyectos.tipo="monasterio")) and (Proyectos.'Nombre del proyecto' like 'Santa Maria' ) </pre>

Figura 82. Paso 2. Traducción del atributo “Nombre”.

El proceso se repite para cada uno de los atributos de la consulta, en nuestro caso para el atributo “Población”, y para el atributo “Provincia”. En la Figura 83 se muestra el resultado del proceso de los dos atributos.

LENGUAJE DE CONSULTA	LENGUAJE SQL
<pre> &lt;concepto nombre="Arquitectura Religiosa"&gt;   &lt;atributo nombre="Nombre"&gt;     &lt;operacionbinario op="like"/&gt;     &lt;valor constante="Santa Maria"/&gt;   &lt;/atributo&gt;   &lt;atributo nombre="Poblacion"&gt;     &lt;operacionbinario op="like"/&gt;     &lt;valor constante="Vitoria"/&gt;   &lt;/atributo&gt;   &lt;atributo nombre="Provincia"&gt;     &lt;operacionbinario op="like"/&gt;     &lt;valor constante="Alava"/&gt;   &lt;/atributo&gt; &lt;/concepto&gt; </pre>	<pre> SELECT DISTINCT Proyectos.'Nombre del proyecto' FROM Proyectos WHERE (((Proyectos.tipo="ermita") or (Proyectos.tipo="iglesia") or (Proyectos.tipo="monasterio"))) and (Proyectos.'Nombre del proyecto' like 'Santa Maria' ) and (Proyectos.Localidad like 'Vitoria' ) and (Proyectos.Provincia like 'Alava' )) </pre>

Figura 83. Paso 3 y 4. Traducción de los atributos “Población” y “Provincia”.

Una vez procesados los atributos, la consulta está preparada para ser ejecutada en la base de datos correspondiente. Siguiendo el ejemplo que estamos abordando a lo largo de esta explicación, a continuación se van a mostrar las dos consultas, una para cada sistema envoltorio, que serían ejecutadas en sus diferentes bases de datos.

```
SELECT DISTINCT ar.Direccion_Web
FROM Arquitecturareligiosa ar, subpaginaweb sub
WHERE (ar.Direccion_Web=sub.Direccion_Web) and (ar.Nombre like 'Santa Maria')
and (ar.Poblacion like 'Vitoria') and (ar.Provincia like 'Alava')
```

BASE DE DATOS LDGP
<pre>SELECT DISTINCT Proyectos.`Nombre del proyecto` FROM Proyectos WHERE (((Proyectos.tipo="ermita") or (Proyectos.tipo="iglesia") or (Proyectos.tipo="monasterio"))) and (Proyectos.`Nombre del proyecto` like 'Santa Maria' ) and (Proyectos.Localidad like 'Vitoria' ) and (Proyectos.Provincia like 'Alava'))</pre>

Figura 84. Consulta para Base de Datos LDGP.

BASE DE DATOS PÁGINAS WEB
<pre>SELECT DISTINCT ar.Direccion_Web FROM Arquitecturareligiosa ar, subpaginaweb sub WHERE (ar.Direccion_Web=sub.Direccion_Web) and (ar.Nombre like 'Santa Maria') and (ar.Poblacion like 'Vitoria') and (ar.Provincia like 'Alava')</pre>

Figura 85. Consulta para Base de Datos Páginas Web.

#### 7.3.2.4 PRESENTACIÓN DE RESULTADOS

Cada consulta es procesada en la base de datos correspondiente, y una vez que los resultados están disponibles, el siguiente paso es el de ponerlos en común y

presentárselos al usuario. Tal y como se ha explicado anteriormente, debido a que los datos proceden de fuentes de datos diferentes, es necesario, antes de mostrar los datos obtenidos, presentar dichas fuentes. La información de las fuentes de datos fue obtenida en el proceso de construcción y distribución de consultas cuando se decidió los sistemas envoltorio a los que había que enviar la consulta expresada por el usuario.

Por lo tanto en esta primera página de presentación se mostrará al usuario todas las bases de datos que han participado en la consulta, describiendo para cada una de ellas:

- Nombre de la organización a la que pertenece la base de datos.
- Su dirección de página Web, si es que existe una página que albergue dichos datos.
- La descripción de la base de datos. Qué datos contiene y en qué consisten esos datos.
- Los atributos que han participado en la consulta sobre dicha base de datos.

En nuestro caso concreto, tal y como se ha explicado anteriormente, participan las dos bases de datos del sistema y ambas con los mismos atributos (“Nombre”, “Población”, y “Provincia”). A continuación se muestra la página de descripción de ambas bases de datos (Figura 86).

### Información Paginas Web

Resultados Obtenidos 3  
[Aceptar](#)

**Organización**  
Paginas Web

Datos de la organización

**Dirección Página Web**  
Información no disponible.

**Descripción**  
Recoge toda la información existente en diferentes sitios Web

**Atributos que aparecen en la consulta:**  
Nombre, Poblacion, Provincia

---

### Información LDGP

Resultados Obtenidos 0  
[Aceptar](#)

**Organización**  
LDGP

Datos de la organización

**Dirección Página Web**  
<http://web.alava.net/PatrimonioHistorico/patrimonio/patrimoniohome.htm>

**Descripción**  
Recoge la información de los diferentes trabajos de investigación llevados a cabo por el Laboratorio de Documentación Geométrica del Patrimonio

**Atributos que aparecen en la consulta:**  
Nombre, Poblacion, Provincia

Figura 86. Presentación al usuario de las Base de Datos implicadas en la consulta.

Como se aprecia en la Figura 86, para ambas bases de datos, el sistema nos muestra el nombre de la organización que soporta dicha base de datos, una pequeña descripción de dicha organización, que incluye información como la dirección principal de la página Web, si es que dicha organización tiene página Web, una descripción de lo que recoge la base de datos, y los atributos que aparecen en la consulta. Además, para cada base de datos muestra el número de resultados obtenidos en la consulta, y junto a este dato aparece un botón que permite al usuario, pinchando sobre él, visitar los resultados de la base de datos. En nuestro caso particular, únicamente la base de datos de las páginas Web ha obtenido resultados en su consulta por lo que accederemos a ella para conocer cómo muestra el contenido de sus resultados.

Información Páginas Web	
<p><b>Organización</b>  <b>Diputación Foral de Alava</b>  Datos de la organización</p> <p><b>Página Web</b>  <a href="http://www.alava.net/default_c.asp">http://www.alava.net/default_c.asp</a></p> <p><b>Descripción</b>  Página oficial de la Diputación Foral de Álava</p> <p><b>Subpáginas Web:</b>  <a href="http://web.alava.net/PatrimonioHistorico/patrimonio/patrimonihome.htm">http://web.alava.net/PatrimonioHistorico/patrimonio/patrimonihome.htm</a></p> <p style="text-align: right;"><a href="#">Más Información</a></p>	
<p><b>Organización</b>  <b>Fundación de la Catedral de Santa María.</b>  Datos de la organización</p> <p><b>Página Web</b>  <a href="http://www.catedralvitoria.com/">http://www.catedralvitoria.com/</a></p> <p><b>Descripción</b>  Página Web de la Fundación de la Catedral de Santa María.</p> <p><b>Subpáginas Web:</b>  <a href="http://www.catedralvitoria.com/contenido.asp?pos=1_1_1&amp;op=op4&amp;url=pdfs/vari0s/catedral_1.pdf&amp;target=blank&amp;bzFranja=1">http://www.catedralvitoria.com/contenido.asp?pos=1_1_1&amp;op=op4&amp;url=pdfs/vari0s/catedral_1.pdf&amp;target= blank&amp;bzFranja=1</a></p> <p style="text-align: right;"><a href="#">Más Información</a></p>	
<p><b>Organización</b>  <b>Laboratorio de documentación geométrica del patrimonio</b>  Datos de la organización</p> <p><b>Página Web</b>  <a href="http://www.vc.ehu.es/docarq/">http://www.vc.ehu.es/docarq/</a></p> <p><b>Descripción</b>  Página oficial del laboratorio de documentación geométrica del patrimonio</p> <p><b>Subpáginas Web:</b>  <a href="http://www.vc.ehu.es/docarq/Proyectos">http://www.vc.ehu.es/docarq/Proyectos</a></p> <p style="text-align: right;"><a href="#">Más Información</a></p>	

Figura 87. Presentación al usuario de los datos la Base de Datos “Páginas Web”

Además esta base de datos nos ofrece también la posibilidad de conocer la información que existe en cada una de las páginas Web que forman la solución, bastará con pulsar sobre el link “Más Información”, y el sistema le mostrará una página como esta (Figura 88).

Descripción	
Dirección de la subpágina	<a href="http://web.alava.net/PatrimonioHistorico/patrimonio/patrimonihome.htm">http://web.alava.net/PatrimonioHistorico/patrimonio/patrimonihome.htm</a>
Datos de la subpágina	
Descripción de subpáginaWeb	Patrimonio HistÁrico ArquitectÁnico de la Diputaci³n Foral de Álava
¿Qué información nos ofrece esta página Web?	
¿Qué información podemos obtener en esta institución?	No disponible

Figura 88. Descripción de una de las páginas Web de la solución.

Finalmente para acceder a la página donde se encuentra la información bastará pinchar sobre los hipervínculos de cada una de las subpáginas Web.





## **8. IMPLANTACIÓN DE LA SOLUCIÓN EN EL ORDENADOR DE LA UNIVERSIDAD DEL PAÍS VASCO**

Una vez terminado el proyecto a modo local, utilizando la herramienta Xampp, tal y como se indica en apartados anteriores, era el momento de implantar las dos aplicaciones en el servidor de la Universidad del País Vasco (el cliente). El proceso resultó más complicado de lo que al principio se suponía, lo que implicó que la implantación del Proyecto de Fin de Carrera realizado durara más días de lo previsto.

En primer lugar recordar que la herramienta Xampp dispone de PHP (en su versión 5) como lenguaje de programación Web, MySQL como sistema gestor de bases de datos, y Apache como servidor. En mi caso particular he utilizado Xampp desde Windows Vista, que según me informado, aún siendo la misma herramienta, su funcionamiento es mínimamente diferente. En un principio quedé con el cliente en que íbamos a necesitar tres bases de datos, para colocar allí las tres bases de datos de nuestro sistema. La base de datos en la cual se guarda información sobre las páginas Web, la base de datos “administradores”, y la base de datos “proyectos”, la cual el Laboratorio de Documentación Geométrica del Patrimonio había creado para almacenar sus proyectos, y que no sirve como segunda base de datos para la aplicación de búsqueda orientada. La Universidad del País Vasco nos concedió únicamente una base de datos, ya que según se nos informó desde el C.I.D.I.R. por cada cuenta únicamente sólo proporcionan una base de datos. Por tanto el propio C.I.D.I.R. nos sugirió introducir todas las bases de datos de nuestra aplicación en la base de datos que ellos nos habían concedido, y diferenciar estas por prefijos. Al final no tuvimos problemas en hacerlo así. El segundo día en el que se intentó la instalación nos dimos cuenta que el servidor no albergaba PHP5 necesario para las aplicaciones desarrolladas. Nos pusimos en contacto de nuevo con el C.I.D.I.R, y le transmitimos nuestro problema. Desde allí, nos dijeron que el problema era el servidor que albergaba la página Web del Laboratorio de Documentación Geométrica del Patrimonio, en donde se iba a colocar las aplicaciones, era un servidor antiguo y que sólo contenía hasta la versión 4 de PHP. Sin embargo nos dijeron que estaban progresivamente cambiando todos los datos del servidor antiguo al servidor nuevo, que si albergaba PHP5, así que la página web del Laboratorio de Documentación Geométrica del Patrimonio sería cambiada inmediatamente.

Así fue, en cuestión de pocos días ya teníamos la página Web en el nuevo servidor. Este nuevo servidor, hasta el punto que yo conozco, es un servidor Apache, que utiliza MySQL como sistema gestor de bases de datos, y alberga PHP5, que es exactamente lo que yo necesitaba. Sin embargo la instalación de las aplicaciones fue más compleja de lo que esperaba. Si bien es cierto que la importación de las tres bases de datos a una única base de datos fue sencilla, la configuración de las aplicaciones fue una tarea un tanto difícil.

Desconozco si fue debido a mi falta de experiencia, ya que nunca había trabajado con un servidor anteriormente, pero los errores se sucedieron. En primer lugar mencionar, que a diferencia del sistema en modo local, el sistema de la universidad no

me mostraba los errores por pantalla, únicamente salía la pantalla en blanco, en vez de indicarte la línea en la que se producía el fallo. Esto desencadenó en que corregir dichos fallos me costará mucho más, ya que en primer lugar debía conocer cuál era el fallo.

Aunque me costó bastante conseguí solucionar todos los problemas que iban apareciendo. El primer error fue la extensión de los archivos. Personalmente desconocía porqué mi editor de texto por defecto me guardaba los ficheros como .php3, lo cual entendí que era una nueva extensión otorgada por PHP (ya que las veces anteriores que había trabajado con PHP, la extensión había sido .php). Sin embargo resultó ser una versión más antigua. Todo esto se remonta a cuando apareció PHP3, que a modo de distinción con PHP2, utilizó la extensión .php3. Esto me causó los primeros problemas, ya que aunque a modo local funcionaba perfectamente, el servidor de la universidad no leía el código PHP y se limitaba únicamente a imprimirlo por pantalla. Una vez que me di cuenta del error cambié todas sus extensiones, y con lo que ello conlleva, todas las rutas existentes en cada fichero. Otro de los errores importantes fue las consultas en las bases de datos. Mi sistema no hacía ninguna distinción entre las mayúsculas y las minúsculas en lo que al contenido de las cláusulas SQL se refiere. Sin embargo el sistema de la Universidad del País Vasco, sí. Esto implicó que en todas aquellas consultas que había un letra mayúscula que en realidad debería ser minúscula o viceversa, la consulta no se ejecutaba, y el sistema no mostraba nada excepto una página en blanco. Para solucionar esto se analizaron cada una de las consultas y se fueron corrigiendo los posibles fallos. Otro de los errores a mencionar fue lo relativo al fichero XML consulta.xml, que almacena la consulta en Lenguaje Consulta. Este fichero debe ser modificado cada vez que un usuario entra al sistema y realiza una consulta. Sin embargo el archivo nunca se modificaba y el sistema siempre ejecutaba la misma consulta. Para solucionar esto bastaba con dar permisos de escritura a dicho fichero XML. Aunque parezca sencillo darse cuenta una vez que lo sabes, personalmente me costó un rato saber porqué sucedía lo comentado. Además de estos errores más significativos, sucedieron otros como el cambio de las conexiones a las bases de datos, ya que pasamos de un sistema local con tres a bases de datos a un servidor con una única base de datos que reúne las tres anteriores. Aunque realmente lo que más costó y desesperó a los allí presentes, alumno y cliente, fue no saber, en un principio, a que se debían los errores.

Finalmente y a falta de la elaboración de las pruebas por parte del cliente podemos decir que ambas aplicaciones han quedado perfectamente instaladas en el servidor de la Universidad del País Vasco, y que además éstas son accesibles a través de la Web. De momento las direcciones Web en donde las podemos encontrar son:

#### **Aplicación de búsqueda orientada:**

<http://www.ehu.es/docarq/webbuscador/Aplicacion/Index.php>

**Aplicación de soporte a la base de datos que amacena información de las páginas Webs.**

<http://www.ehu.es/docarq/webgestion/AplicacionNormal/Index.php>

Si bien es cierto que una vez que estas estén bien preparadas con los datos necesarios, se podrá acceder a ellas a través de un hipervínculo en la página principal del Laboratorio de Documentación Geométrica del Patrimonio.

Concluir que, al parecer en un principio, todo ha salido bien, y el producto ha sido entregado e instalado perfectamente.



## 9. FASE DE PRUEBAS

Durante este apartado se van a definir una serie de pruebas cuya finalidad es comprobar el correcto funcionamiento de las aplicaciones que se han desarrollado a lo largo del Proyecto de Fin de Carrera. Con el fin de facilitar al lector la comprensión de dichas pruebas se ha optado por dividir las pruebas en función de la aplicación a la que hacen referencia. Comenzaremos con la aplicación que da soporte a la base de datos que almacena información sobre las páginas Web, para posteriormente describir las pruebas referentes a la aplicación de búsqueda orientada. En ambos casos, contaremos para el desarrollo de las pruebas con tres personas con diferentes conocimientos sobre la informática. Sus datos son los siguientes.

Nombre	Edad	Nivel de formación en informática
Irene Martín	50	Ninguno.
Marta López-Davalillo	23	Nivel de usuario común.
Beatriz López-Davalillo	23	Ingeniera Técnica en Informática de Gestión.

Tabla 13. Responsables de las pruebas.

### 9.1 PRUEBAS DE LA APLICACIÓN QUE DA SOPORTE A LA BASE DE DATOS QUE ALMACENA INFORMACIÓN SOBRE LAS PÁGINAS WEB.

#### 9.1.1 PRUEBA 1

- **Nombre de la prueba**

Acceso a la aplicación. Contraseña no válida.

- **Descripción de la prueba**

Intento de acceso a la aplicación con una contraseña que no corresponde con la del usuario.

- **Resultado esperado**

El sistema no permite el acceso mostrando un aviso de error.

- **Resultado real**

Irene: No deja entrar si los datos no son correctos. Bien

Marta: No permite el acceso. Correcto.

Beatriz: Correcto.

### 9.1.2 PRUEBA 2

- **Nombre de la prueba**

Insertar un nuevo Sitio Web.

- **Descripción de la prueba**

El usuario inserta un nuevo sitio Web en la base de datos.

- **Variante 1.** El usuario intenta introducir un sitio Web que ya existe en la base de datos.

- **Resultado esperado**

El sitio Web queda correctamente registrado.

- **Variante 1.** El sistema no se lo permite, le muestra un mensaje de error.

- **Resultado real**

Irene: El nuevo sitio Web ha sido introducido correctamente.

- **Variante 1.** No he podido introducir un sitio Web cuya dirección pertenecía a otro sitio Web.

Marta: He introducido un nuevo sitio web fácilmente, sin ningún problema.

- **Variante 1.** El sistema no me permite introducir una página web ya existente, para avisarme de esto aparece un mensaje.

Beatriz: Se inserta perfectamente.

- **Variante 1.** No me permite la inserción y me muestra un mensaje para avisarme.

### 9.1.3 PRUEBA 3

- **Nombre de la prueba**

Modificar un Sitio Web.

- **Descripción de la prueba**

El usuario modifica un nuevo sitio Web ya existente en la aplicación.

- **Variante 1.** El usuario intenta modificar un sitio Web asignándole una dirección Web que pertenece a otro sitio Web ya registrado.
- **Variante 2.** El usuario intenta modificar un sitio Web asignándole una dirección Web nueva.

- **Resultado esperado**

El sitio Web queda correctamente modificado.

- **Variante 1.** El sistema no se lo permite, le muestra un mensaje de error.
- **Variante 2.** La aplicación se modifica correctamente. Además todas las sub-páginas Web que pertenecen a dicho sitio Web se deben mover perfectamente a la nueva dirección Web del sitio Web.

- **Resultado real**

Irene: Resultado correcto. No permite la operación cuando intento asignarle una dirección ya existente, pero me lo permite si la dirección Web es la que ya tenía o una nueva.

Marta: Si modifico la dirección web por otra existente, efectivamente aparece mensaje de error. En cambio, modifico dirección web por otra no existente, y el cambio se efectúa correctamente.

Beatriz: Si se intenta asignarle la dirección de un sitio Web ya existente no se deja, si los datos son correctos se modifica bien.

#### 9.1.4 PRUEBA 4

- **Nombre de la prueba**

Eliminar un Sitio Web.

- **Descripción de la prueba**

El usuario elimina un sitio Web ya existente en la aplicación.

- **Resultado esperado**

El sitio Web queda correctamente eliminado. Además todas las sub-páginas Web que pertenecen a dicho sitio Web así como los elementos patrimoniales que se encuentran situados en las sub-páginas Webs que pertenecen al sitio Web que se va eliminar quedarán a su vez eliminadas. Eliminación en cascada.

- **Resultado real**



Irene: Si elimino un sitio Web determinado se elimina dicho sitio Web, todas las sub-paginas que “cuelgan” de él y todos los elementos patrimoniales que “cuelgan” de dichas sub-páginas.

Marta: Al eliminar un sitio Web, se eliminan todas sus subpáginas Web, así como sus elementos patrimoniales correspondientes.

Beatriz: La eliminación en cascada funciona perfectamente.

### 9.1.5 PRUEBA 5

- **Nombre de la prueba**

Introducir una nueva subpágina Web.

- **Descripción de la prueba**

El usuario introduce una subpágina Web en un sitio Web ya existente.

- **Variante 1.** El usuario intenta introducir una subpágina en un sitio Web, asignándole una dirección a dicha subpágina, que ya pertenece a otra subpágina Web de ese sitio Web.

- **Resultado esperado**

La nueva subpágina Web queda correctamente registrada.

- **Variante 1.** El sistema no se lo permite, le muestra un mensaje de error.

- **Resultado real**

Irene: Permite la introducción de una nueva página Web, siempre y cuando esta no exista ya, en cuyo caso nos muestra un mensaje de error.

Marta: Se introduce correctamente la nueva subpágina Web asociada a la dirección Web. Y efectivamente si esa subpágina ya existe, aparece mensaje de error que no permite continuar.

Beatriz: Se inserta bien si no se intenta introducir una dirección Web que pertenezca a otra página, en cuyo caso muestra un mensaje.

### 9.1.6 PRUEBA 6

- **Nombre de la prueba**

Modificar una subpágina Web.

- **Descripción de la prueba**

El usuario modifica una subpágina Web de un sitio Web ya existente.

- **Variante 1.** El usuario intenta modificar una subpágina Web en un sitio Web, asignándole una dirección Web a dicha subpágina Web que ya pertenece a otra subpágina Web de ese sitio Web.

- **Resultado esperado**

La subpágina Web queda correctamente modificada.

- **Variante 1.** El sistema no se lo permite, le muestra un mensaje de error.

- **Resultado real**

Irene: Se puede modificar los datos de la subpágina Web perfectamente, a no ser que la dirección Web que se le asocie pertenezca ya a otra subpágina de dicho sitio Web.

Marta: Se puede modificar la subpágina perfectamente siempre y cuando ésta no exista ya dentro del sitio Web, en cuyo caso saldrá un mensaje.

Beatriz: Funciona bien tanto la prueba como la variante.

### 9.1.7 PRUEBA 7

- **Nombre de la prueba**

Eliminar una subpágina Web.

- **Descripción de la prueba**

El usuario elimina una subpágina Web de un sitio Web ya existente.

- **Resultado esperado**

La subpágina Web queda correctamente eliminada. Además todos los elementos patrimoniales pertenecientes a dicha subpágina Web serán también eliminados. Eliminación en cascada.

- **Resultado real**

Irene: Se elimina la subpágina y todos sus elementos patrimoniales.

Marta: Efectivamente al eliminar una subpágina web se eliminan sus elementos patrimoniales.

Beatriz: La eliminación en cascada funciona bien.

### 9.1.8 PRUEBA 8

- **Nombre de la prueba**

Introducir un nuevo elemento patrimonial.

- **Descripción de la prueba**

El usuario introduce un nuevo elemento patrimonial en una subpágina Web ya existente.

- **Resultado esperado**

El nuevo elemento patrimonial queda correctamente registrado.

- **Resultado real**

Irene: Insertado correctamente.

Marta: Se introduce perfectamente y fácilmente el nuevo elemento patrimonial.

Beatriz: Bien insertado.

### 9.1.9 PRUEBA 9

- **Nombre de la prueba**

Modificar un elemento patrimonial.

- **Descripción de la prueba**

El usuario modifica un elemento patrimonial de una subpágina Web ya existente.

- **Resultado esperado**

El elemento patrimonial queda correctamente modificado.

- **Resultado real**

Irene: Modificación intuitiva y fácil de llevar a cabo.

Marta: Fácil y rápido de modificar.

Beatriz: Se modifica fácilmente.

#### 9.1.10 PRUEBA 10

- **Nombre de la prueba**

Eliminar un elemento patrimonial.

- **Descripción de la prueba**

El usuario elimina un elemento patrimonial de una subpágina Web ya existente.

- **Resultado esperado**

El elemento patrimonial queda correctamente eliminado.

- **Resultado real**

Irene: Se elimina el elemento patrimonial correctamente.

Marta: Igualmente ningún problema para eliminar el elemento patrimonial.

Beatriz: Eliminación correcta.

#### 9.1.11 PRUEBA 11

- **Nombre de la prueba**

Introducir un nuevo administrador.

- **Descripción de la prueba**

El usuario introduce un nuevo administrador.

- **Variante 1.** El usuario intenta introducir un administrador/usuario nuevo asignándole como “Nick”, un “Nick” de un administrador/usuario ya registrado.

- **Resultado esperado**

El nuevo administrador queda correctamente registrado.

- **Variante 1.** El sistema no se lo permite, le muestra un mensaje de error.

- **Resultado real**

Irene: Si los datos introducidos son correctos, se inserta perfectamente, si no nos avisa mediante un mensaje de error.

Marta: Se queda bien registrado siempre y cuando no exista ese Nick, si existe, salta un cuadro de error.

Beatriz: Inserta bien si el Nick que se pretende introducir no pertenece ya a otra persona, en cuyo caso muestra un mensaje de error.

#### 9.1.12 PRUEBA 12

- **Nombre de la prueba**

Modificar un administrador.

- **Descripción de la prueba**

El usuario modifica los datos de un administrador.

- **Variante 1.** El usuario intenta modificar un administrador asignándole como “Nick”, un “Nick” de un administrador ya registrado.

- **Resultado esperado**

Los datos del administrador quedan correctamente modificados.

- **Variante 1.** El sistema no se lo permite, le muestra un mensaje de error.

- **Resultado real**

Irene: Si el Nick que se pretende modificar no corresponde con uno ya existente en el sistema, lo modifica correctamente.

Marta: No lo permite si el nuevo Nick existe, sino se modifica correctamente.

Beatriz: Funciona bien, tanto la variante como la prueba general.

### 9.1.13 PRUEBA 13

- **Nombre de la prueba**

Eliminar un administrador.

- **Descripción de la prueba**

El usuario elimina un administrador.

- **Resultado esperado**

El administrador queda correctamente eliminado.

- **Resultado real**

Irene: Se elimina perfectamente.

Marta: Eliminado sin ningún problema.

Beatriz: Se elimina bien.

### 9.1.14 PRUEBA 14

- **Nombre de la prueba**

Ver movimientos.

- **Descripción de la prueba**

El usuario comprueba los movimientos que ha realizado, con el fin de comprobar que se almacenan correctamente.

- **Resultado esperado**

Los movimientos realizados se almacenan correctamente.

- **Resultado real**

Irene: Se pueden ver todos los movimientos realizados en el sistema en orden comenzando por el más reciente.

Marta: Se visualizan todos los movimientos realizados en la aplicación.

Beatriz: Se pueden ver todos los movimientos en orden en que estos han acontecido.

#### **9.1.15 PRUEBA 15**

- **Nombre de la prueba**

Ver administradores.

- **Descripción de la prueba**

El usuario comprueba los administradores que existen en la aplicación.

- **Resultado esperado**

Los administradores que se muestran son exactamente los que en realidad existen.

- **Resultado real**

Irene: Se ven todos los administradores del sistema.

Marta: Efectivamente se muestran todos los administradores y usuarios existentes.

Beatriz: Bien, funciona tal y como dice la prueba.

### **9.2 PRUEBAS DE LA APLICACIÓN DE BÚSQUEDA ORIENTADA.**

#### **9.2.1 PRUEBA 1**

- **Nombre de la prueba**

Realizar consulta general.

- **Descripción de la prueba**

El usuario realiza una búsqueda general seleccionando los atributos que desee.

- **Resultado esperado**

La consulta se realiza correctamente.

- **Resultado real**

Irene: La búsqueda se ha hecho correctamente.

Marta: Se ha llevado a cabo sin ningún problema.

Beatriz: La consulta se realiza perfectamente.

### 9.2.2 PRUEBA 2

- **Nombre de la prueba**

Realizar consulta general sobre “edificios”.

- **Descripción de la prueba**

El usuario realiza una búsqueda general sobre “edificios” seleccionando los atributos que desee.

- **Resultado esperado**

La consulta se realiza correctamente.

- **Resultado real**

Irene: Bien.

Marta: He encontrado perfectamente lo que quería.

Beatriz: La operación ha finalizado con éxito.

### 9.2.3 PRUEBA 3

- **Nombre de la prueba**

Realizar consulta dentro de “edificios” sobre “arquitectura religiosa”.

- **Descripción de la prueba**

El usuario realiza una búsqueda dentro de “edificios” sobre “arquitectura religiosa” seleccionando los atributos que desee.



- **Resultado esperado**

La consulta se realiza correctamente.

- **Resultado real**

Irene: Busca sin problemas.

Marta: La consulta se realiza sin ningún problema.

Beatriz: La búsqueda se ha hecho satisfactoriamente.

#### 9.2.4 PRUEBA 4

- **Nombre de la prueba**

Realizar consulta dentro de “edificios” sobre “arquitectura civil”.

- **Descripción de la prueba**

El usuario realiza una búsqueda dentro de “edificios” sobre “arquitectura civil” seleccionando los atributos que desee.

- **Resultado esperado**

La consulta se realiza correctamente.

- **Resultado real**

Irene: La búsqueda se realiza correctamente.

Marta: El resultado es el esperado.

Beatriz: Efectivamente la consulta finaliza como esperaba.

#### 9.2.5 PRUEBA 5

- **Nombre de la prueba**

Realizar consulta dentro de “edificios” sobre “elementos menores”.

- **Descripción de la prueba**

El usuario realiza una búsqueda dentro de “edificios” sobre “elementos menores” seleccionando los atributos que desee.

- **Resultado esperado**

La consulta se realiza correctamente.

- **Resultado real**

Irene: Sin problemas.

Marta: Se realiza la búsqueda de elementos menores correctamente.

Beatriz: Ningún problema al realizar la consulta.

### 9.2.6 PRUEBA 6

- **Nombre de la prueba**

Realizar consulta sobre “Excavaciones”.

- **Descripción de la prueba**

El usuario realiza una búsqueda sobre “Excavaciones” seleccionando los atributos que desee.

- **Resultado esperado**

La consulta se realiza correctamente.

- **Resultado real**

Irene: Búsqueda correcta.

Marta: El resultado es el esperado.

Beatriz: Se realiza la búsqueda correctamente, a través de determinados atributos seleccionados.

### 9.2.7 PRUEBA 7

- **Nombre de la prueba**

Realizar consulta sobre “Objetos”.

- **Descripción de la prueba**

El usuario realiza una búsqueda sobre “Objetos” seleccionando los atributos que desee.

- **Resultado esperado**

La consulta se realiza correctamente.

- **Resultado real**

Irene: Lo que se esperaba.

Marta: Búsqueda exitosa.

Beatriz: La consulta se ha realizado correctamente.

## 9.3 CONCLUSIONES

### 9.3.1 CONCLUSIONES DE LA APLICACIÓN QUE DA SOPORTE A LA BASE DE DATOS QUE ALMACENA INFORMACIÓN SOBRE LAS PÁGINAS WEB.

- ✓ **Marta**

Considero que es una aplicación de fácil utilidad para conocimientos medios de la informática. No creo que exista dificultad a la hora de llevar a cabo las operaciones deseadas, además las explicaciones me parecen claras y concisas, lo que ayuda a no tener dudas sobre su funcionamiento.

- ✓ **Irene**

Nunca antes había utilizado el ordenador en tanta profundidad como en esta ocasión, por lo que he de reconocer que al principio me ha costado un poco entender el funcionamiento del sistema. Sin embargo gracias a la ayuda del desarrollador del proyecto, Ion, he conseguido realizar todas las pruebas.

En cuanto a la aplicación he de comentar que me parece relativamente sencillo realizar cualquiera de las operaciones descritas, además por lo que he podido comprobar estas funcionan perfectamente.

- ✓ **Beatriz**

A mi modo de entender, considero que es una aplicación sencilla a la par que completa. En mi opinión cubre de forma exhaustiva la funcionalidades para la cual fue diseñada, además su diseño es sencillo permite al usuario trabajar con facilidad y de forma intuitiva.

### **9.3.2 CONCLUSIONES DE LA APLICACIÓN DE BÚSQUEDA ORIENTADA.**

✓ **Marta**

Es un buscador fácil que permite realizar consultas sobre el patrimonio histórico, de manera tanto general como específica. Por lo que me permite a mí, como usuaria sin conocimientos de patrimonio realizar consultas generales del mismo.

✓ **Irene**

Considero que se trata de un buscador muy completo que en comparación con otros más generales nos da la posibilidad de buscar por “cosas” más concretas.

✓ **Beatriz**

Desde el punto de vista informático me parece una aplicación bastante completa, ya que a su dificultad de búsqueda en bases de datos heterogéneas, hay que unir su profundidad en las búsquedas ya que permite a cualquier persona realizar búsquedas en función de su nivel de conocimientos sobre el patrimonio.



## **10. SEGUIMIENTO DEL PROYECTO**

### **10.1 CAMBIOS EN EL PROYECTO**

Debido, quizás, al tipo de proyecto desarrollado se podía prever desde un principio que el proyecto podía verse alterado en numerosas fases debido, especialmente, a posibles cambios solicitados por el cliente.

Sin embargo, y a pesar de que el proyecto era completamente innovador y carecía de unas pautas determinadas para conseguir su éxito, la relación constante mantenida entre el alumno y el cliente ha logrado que, salvo en una ocasión, la cual se comentará más adelante, el alumno no haya tenido que realizar modificaciones en su Proyecto de Fin de Carrera.

Desde el principio del desarrollo del proyecto, alumno y cliente se mantuvieron en contacto, exponiendo sus ideas por un lado y necesidades por el otro. Esto desembocó en que cualquier paso que daba el alumno era contrastado por el cliente, quien lo aprobaba, solicitaba alguna mejora, o directamente lo rechazaba. A sí mismo cada nueva idea o necesidad expuesta por el cliente, era analizada por el desarrollador del proyecto, y puesta en marcha o rechazada en función de la posibilidad de adaptación de dicha idea o necesidad al proyecto.

A pesar de esta buena relación entre ambas partes, durante la fase de implementación, el cliente solicitó una posible mejora que consistía en la restricción de las funcionalidades de uno de los perfiles de la aplicación por un lado, y en posibilidad de adaptar la aplicación de soporte a la base de datos que almacena información sobre las páginas Web, a posibles cambios en la base de datos.

Este cambio supuso un incremento de horas en las fases de diseño e implementación. Debido a la implicación en el proyecto por parte del alumno, este aceptó dicha modificación consciente de que ambos cambios podían suponerle un incremento de 30 horas en el desarrollo del proyecto.

## 10.2 SEGUIMIENTO DE LA GESTIÓN DEL PROYECTO

A lo largo del desarrollo del proyecto se han producido diversas alteraciones en términos de tiempo en relación a la planificación inicial del proyecto elaborada al comienzo del mismo, allá por el mes de Julio. Estas variaciones de tiempo han sido consecuencia de diferentes aspectos, entre los que cabe destacar la estimación inapropiada en alguna de las tareas en las se compone el proyecto. Con este apartado lo que se pretende es adentrar al lector en lo que ha sido la gestión del proyecto, para ello cronológicamente se irán comparando para cada una de las fases la duración estimada al comienzo del proyecto y la duración real de la misma. Además para cada caso concreto se analizarán las causas, las medidas adoptadas ante cada imprevisto, y las consecuencias que ha tenido éste en el desarrollo del proyecto.

Al comenzar el proyecto, y con el fin de fijar una serie de instrucciones a llevar cabo en el caso de que ocurriera algún imprevisto durante el desarrollo del proyecto, se elaboró un documento titulado “Dirección de Riesgos del Proyecto” (Ver apartado 4.6). Este documento ha sido tomado en cuenta a lo largo del proyecto en aquellos momentos en los que ha surgido algún tipo de imprevisto. Durante la descripción de cada fase del Proyecto de Fin de Carrera se podrá comprobar que medidas han sido desarrolladas para cada imprevisto, y como consecuencia sabremos si el documento elaborado al principio del proyecto “Dirección de Riesgos del Proyecto” ha sido utilizado como medida para solventar estos.

### Seguimiento de la gestión por fases

#### 10.2.1 FASE “PREVIO”

La fase “Previo” está compuesta principalmente por la elaboración del documento de propuesta de proyecto. A continuación se muestran dos figuras (Figura 89 y Figura 90) en las que se muestra una parte del Diagrama de Gantt correspondiente a la planificación inicial y al resultado real respectivamente.

Nombre de tarea	Duración	Comienzo	Fin
<input checked="" type="checkbox"/> Previo	6 días	mar 08/07/08	mar 15/07/08
Reunión previa del proyecto	1 día	mar 08/07/08	mar 08/07/08
Elaboración de propuesta de proyecto	5 días	mié 09/07/08	mar 15/07/08

Figura 89. Planificación inicial de la fase “Previo”

Nombre de tarea	Duración	Comienzo	Fin
<b>Previo</b>	<b>8 días</b>	<b>mar 08/07/08</b>	<b>jue 17/07/08</b>
Reunión previa del proyecto	1 día	mar 08/07/08	mar 08/07/08
Elaboración de propuesta de proyectos	7 días	mié 09/07/08	jue 17/07/08

Figura 90. Resultado real de la fase “Previo”

### Comparación entre planificación inicial y resultado real

	Planificación inicial		Resultado real	
	Días	Horas	Días	Horas
<b>Reunión previa del proyecto</b>	1 día	1 hora	1 día	1 hora
<b>Elaboración de propuesta de proyecto</b>	5 días	7 horas	7 días	6 horas
<b>Fase completa</b>	6 días	8 horas	8 días	7 horas

Tabla 14. Fase “Previo”. Comparativa planificación inicial y resultado real.

Tal y como aparece en la Tabla 13, la desviación es mínima, por lo que podemos considerar que la planificación es correcta y que no va a tener consecuencias de ningún tipo en el desarrollo del proyecto.

Otro de los datos que llama la atención en la tabla es que en la “Elaboración de propuesta de proyecto” se invierten más días que horas. Esto es debido a que la propuesta fue redactada por Ion, el desarrollador del proyecto y corregida en varias ocasiones por el cliente, José Manuel Valle, y finalmente corregida y aprobada por el tutor del proyecto José Miguel Blanco.

### 10.2.2 FASE “INICIACIÓN”

Esta fase tiene como actividad principal el desarrollo del D.O.P, además de la planificación y gestión del proyecto que irá incluida en el propio D.O.P. A continuación se muestran dos figuras (Figura 91 y Figura 92) en las que se muestra una parte del Diagrama de Gantt correspondiente a la planificación inicial y al resultado real respectivamente.



Nombre de tarea	Duración	Comienzo	Fin
<b>Iniciación</b>	<b>23 días</b>	<b>mié 16/07/08</b>	<b>lun 01/09/08</b>
Planificación y gestión del proyecto	9 días	mié 16/07/08	lun 28/07/08
Interés y Viabilidad del Proyecto	4 días	mar 29/07/08	lun 11/08/08
<b>DOP</b>	<b>10 días</b>	<b>mar 12/08/08</b>	<b>lun 01/09/08</b>
Ánalysis de la estructura del DOP	2 días	mar 12/08/08	mié 13/08/08
Desarrollo del DOP	8 días	jue 14/08/08	lun 01/09/08

Figura 91. Planificación inicial de la fase “Iniciación”

Nombre de tarea	Duración	Comienzo	Fin
<b>Iniciación</b>	<b>27 días</b>	<b>vie 18/07/08</b>	<b>mié 10/09/08</b>
Planificación y gestión del proyecto	4 días	vie 18/07/08	mié 23/07/08
Interés y Viabilidad del Proyecto	2 días	jue 24/07/08	vie 25/07/08
<b>DOP</b>	<b>21 días</b>	<b>lun 28/07/08</b>	<b>mié 10/09/08</b>
Ánalysis de la estructura del DOP	6 días	lun 28/07/08	mié 13/08/08
Desarrollo del DOP	15 días	jue 14/08/08	mié 10/09/08

Figura 92. Resultado real de la fase “Iniciación”

### Comparación entre planificación inicial y resultado real

	Planificación inicial		Resultado real	
<b>Planificación y gestión del proyecto</b>	9 días	10 horas	4 días	14 horas
<b>Interés y viabilidad del proyecto</b>	4 días	4 horas	2 días	7 horas
<b>(D.O.P.)Análisis de la estructura del D.O.P.</b>	2 días	5 horas	4 días	7 horas
<b>(D.O.P)Desarrollo del D.O.P.</b>	8 días	30 horas	15 días	36 horas
<b>Fase completa</b>	23 días	49 horas	26 días	64 horas

Tabla 15. Fase “Iniciación”. Comparativa planificación inicial y resultado real.

Como se observa en la Tabla 14, se ha producido una desviación considerable en el número de horas invertidas como resultado de una mala planificación. Siguiendo el procedimiento indicado en el documento “Dirección de Riesgos del Proyecto” para corregir este margen negativo de tiempo, en las siguientes fases se deberá “apretar” un poco el ritmo para conseguir equilibrar la planificación inicial y el resultado real. Como se verá en la siguiente fase, esto es precisamente lo que lleva a cabo el desarrollador ya que comienza el estudio de las diferentes posibilidades de solución del proyecto desde que prácticamente éste comienza.

Sin embargo desde el punto de vista del desarrollador del proyecto, no se considera este retraso como una mala planificación sino como consecuencia del comienzo de un proyecto abstracto y sin apenas definición. Además, considera que los principios son difíciles y más el de este proyecto, donde en un principio la idea es demasiado general.

De momento las consecuencias de este retraso no son muy visibles, ya que Ion, el desarrollador del proyecto ha invertido tiempo en avanzar tareas con el fin de mitigar el error.

### 10.2.3 FASE “DESARROLLO”

Debido a la extensión, tanto en términos de tiempo, como en número de tareas dividiremos la fase desarrollo en sub-fases de la misma.

#### 10.2.3.1 SUB-FASE “ESTUDIO PREVIO”

El estudio de la materia que se va a albergar en un proyecto es el primer paso hacia el éxito en el desarrollo del mismo. En este apartado se recogen las duraciones tanto en términos de días como de horas de la planificación inicial del proyecto y el resultado real del mismo relacionadas con esta fase. En la Figura 93 y Figura 94 podemos observar la parte del Diagrama de Gantt correspondiente a la planificación inicial y al resultado real respectivamente.

Nombre de tarea	Duración	Comienzo	Fin
<input type="checkbox"/> Estudio previo	7 días	mar 02/09/08	mié 10/09/08
<input type="checkbox"/> Profundización en conocimientos en bases de datos	7 días	mar 02/09/08	mié 10/09/08
Repaso de conocimientos ya adquiridos	3 días	mar 02/09/08	jue 04/09/08
Estudio de nuevos conocimientos	4 días	vie 05/09/08	mié 10/09/08

Figura 93. Planificación inicial de la sub-fase “Estudio previo”

Nombre de tarea	Duración	Comienzo	Fin
<input type="checkbox"/> Estudio previo	33 días	lun 28/07/08	vie 03/10/08
<input type="checkbox"/> Profundización en conocimientos en bases de datos	33 días	lun 28/07/08	vie 03/10/08
Repaso de conocimientos ya adquiridos	3 días	lun 28/07/08	mié 30/07/08
Estudio de nuevos conocimientos	30 días	jue 31/07/08	vie 03/10/08

Figura 94. Resultado real de la sub-fase “Estudio previo”

### Comparación entre planificación inicial y resultado real

	Planificación inicial		Resultado real	
<b>Repaso de conocimientos ya adquiridos</b>	3 días	20 horas	3 días	23 horas
<b>Estudio de nuevos conocimientos</b>	4 días	25 horas	30 días	187 horas
<b>Sub-fase completa</b>	7 días	45 horas	33 días	210 horas

Tabla 16. Sub-Fase “Estudio previo”. Comparativa planificación inicial y resultado real.

Esta sub-fase, tal y como se observa en la Tabla 15, produce la mayor desviación temporal del proyecto. La causa de esta desviación temporal tan notable no es otra que una planificación inadecuada. Siguiendo el proceso indicado en el documento “Dirección de Riesgos del Proyecto” para corregir este margen negativo de tiempo, en las fases posteriores se deberá trabajar de manera más productiva.

Aunque Ion comenzara con el estudio de diferentes soluciones posibles antes de lo previsto con el fin de adelantar trabajo, el hecho de que el proyecto no tuviera unos patrones claros y definidos desde el primer momento, obliga a Ion a realizar un estudio profundo sobre diferentes posibilidades de elaborar una solución, lo cual lleva al alumno a invertir muchas más horas de las que en un principio el propio alumno había estipulado. Sin embargo, Ion es consciente de que este estudio tan exhaustivo puede permitirle desarrollar las fases de análisis, diseño e implementación en menos tiempo del estipulado. Puede ser que lo que ahora el alumno está sembrando lo pueda recoger más adelante.

Las consecuencias de este retraso en términos de tiempo, puede ser un retraso en la entrega del proyecto al cliente. Sin embargo, Ion decide no replanificar, siendo consciente de que todavía queda tiempo para recuperar el retraso, siendo consciente además de que puede invertir más horas diarias e incluso días sueltos de su periodo vacacional.

#### 10.2.3.2 SUB-FASE “ANÁLISIS”

Aquí se tratan todas las actividades relacionadas con el análisis. Se trata de una fase vital del proyecto, ya que basándose en ella se tomarán una serie de decisiones en el diseño del proyecto y en la posterior implementación del mismo. En la Figura 95 y la Figura 96 podemos observar la parte del Diagrama de Gantt correspondiente a la sub-fase análisis de la planificación inicial y del resultado real.

Nombre de tarea	Duración	Comienzo	Fin
<input type="checkbox"/> <b>Fase de Análisis</b>	<b>15 días</b>	<b>jue 11/09/08</b>	<b>mié 08/10/08</b>
Análisis de los requerimientos	2 días	jue 11/09/08	vie 12/09/08
<input type="checkbox"/> <b>Análisis de los datos</b>	<b>3 días</b>	<b>lun 22/09/08</b>	<b>mié 24/09/08</b>
Análisis de las instituciones que participan	1 día	lun 22/09/08	lun 22/09/08
Clasificación de los datos	2 días	mar 23/09/08	mié 24/09/08
Análisis de las herramientas	2 días	jue 25/09/08	vie 26/09/08
Análisis de cada una de las propuestas	6 días	lun 29/09/08	lun 06/10/08
Selección de la mejor propuesta	1 día	mar 07/10/08	mar 07/10/08
Elaboración de los casos de uso	1 día	mié 08/10/08	mié 08/10/08

Figura 95. Planificación inicial de la sub-fase “Análisis”

Nombre de tarea	Duración	Comienzo	Fin
<input type="checkbox"/> <b>Fase de Análisis</b>	<b>14 días</b>	<b>lun 06/10/08</b>	<b>jue 23/10/08</b>
Análisis de los requerimientos	1 día	lun 06/10/08	lun 06/10/08
<input type="checkbox"/> <b>Análisis de los datos</b>	<b>6 días</b>	<b>mar 07/10/08</b>	<b>mar 14/10/08</b>
Análisis de las instituciones que participan	2 días	mar 07/10/08	mié 08/10/08
Clasificación de los datos	4 días	jue 09/10/08	mar 14/10/08
Análisis de las herramientas	2 días	mié 15/10/08	jue 16/10/08
Análisis de cada una de las propuestas	3 días	vie 17/10/08	mar 21/10/08
Selección de la mejor propuesta	1 día	mié 22/10/08	mié 22/10/08
Elaboración de los casos de uso	1 día	jue 23/10/08	jue 23/10/08

Figura 96. Resultado real de la sub-fase “Análisis”

### Comparación entre planificación inicial y resultado real

	Planificación inicial		Resultado real	
<b>Análisis de requerimientos</b>	2 día	5 horas	1 día	1 hora
<b>(Análisis de los datos) Análisis de las instituciones que participan</b>	1 días	5 horas	2 días	12 horas
<b>(Análisis de los datos) Clasificación de los datos</b>	2 días	12 horas	4 días	20 horas
<b>Análisis de las herramientas</b>	2 días	15 horas	2 días	14 horas
<b>Análisis de cada una de las propuestas</b>	6 días	50 horas	3 días	24 horas
<b>Selección de la mejor propuesta</b>	1 días	5 horas	1 días	3 horas
<b>Elaboración de los Casos de Uso</b>	1 días	6 horas	1 días	4 horas
<b>Sub-fase completa</b>	15 días	98 horas	14 días	68 horas

Tabla 17. Sub-Fase “Análisis”. Comparativa planificación inicial y resultado real.

La sub-fase “análisis” nos ofrece una nueva versión de una planificación inadecuada. Esta ocasión la previsión ha sido demasiado pesimista, ya que en un menor número de horas se ha podido llevar a cabo la fase de análisis. Sin embargo ¿Será que a partir de ahora comenzaremos a recoger los frutos de un estudio tan exhaustivo?

Lo sucedido en esta fase no puede asociarse a un riesgo, ya que es todo lo contrario, se ha conseguido realizar la tarea en menos tiempo de lo estimado.

#### 10.2.3.3 SUB-FASE DISEÑO

La sub-fase de diseño incluye todas las tareas relacionadas con el diseño de las bases de datos, de las aplicaciones y de las pruebas. . En la Figura 97 y la Figura 98 podemos observar la parte del Diagrama de Gantt correspondiente a la sub-fase diseño de la planificación inicial y del resultado real.

Nombre de tarea	Duración	Comienzo	Fin
<input type="checkbox"/> Fase de Diseño	32 días	jue 09/10/08	vie 21/11/08
<input type="checkbox"/> Diseño Aplicación soporte páginas Web	11 días	jue 09/10/08	jue 23/10/08
Diseño de la(s) estructura(s) de las base(s) de datos	2 días	jue 09/10/08	vie 10/10/08
<input type="checkbox"/> Diseño de la base de datos "Sitios Web"	2 días	lun 13/10/08	mar 14/10/08
Diseño de los metadatos	2 días	lun 13/10/08	mar 14/10/08
Elaboración del diagrama Entidad-Relación	2 días	lun 13/10/08	mar 14/10/08
Elaboración del diagrama Relacional	2 días	lun 13/10/08	mar 14/10/08
Diseño de las relaciones entre tablas	2 días	lun 13/10/08	mar 14/10/08
<input type="checkbox"/> Diseño de la base de datos "Administradores"	2 días	mié 15/10/08	jue 16/10/08
Diseño de los metadatos	2 días	mié 15/10/08	jue 16/10/08
Elaboración del diagrama Entidad-Relación	2 días	mié 15/10/08	jue 16/10/08
Elaboración del diagrama Relacional	2 días	mié 15/10/08	jue 16/10/08
Diseño de las relaciones entre tablas	2 días	mié 15/10/08	jue 16/10/08
<input type="checkbox"/> Diseño de la estructura de la Aplicación	5 días	vie 17/10/08	jue 23/10/08
Diseño de los perfiles	1 día	vie 17/10/08	vie 17/10/08
Diseño de las interfaces	1 día	lun 20/10/08	lun 20/10/08
Diseño de la funcionabilidad de la aplicación	2 días	mar 21/10/08	mié 22/10/08
Diseño de la arquitectura de la aplicación	1 día	jue 23/10/08	jue 23/10/08
<input type="checkbox"/> Diseño Aplicación de búsqueda orientada	20 días	vie 24/10/08	jue 20/11/08
<input type="checkbox"/> Diseño de la Arquitectura de la aplicación	20 días	vie 24/10/08	jue 20/11/08
Diseño de las interfaces	2 días	vie 24/10/08	lun 27/10/08
Diseño del Árbol de Conceptos	2 días	mar 28/10/08	mié 29/10/08
Diseño de la estructura de la aplicación	13 días	jue 30/10/08	lun 17/11/08
Diseño de los árboles de correspondencias necesarios	3 días	mar 18/11/08	jue 20/11/08
Diseño de las pruebas a realizar en la fase de Implementación	1 día	vie 21/11/08	vie 21/11/08
Diseño de las pruebas a realizar en la fase de Implantación	1 día	vie 21/11/08	vie 21/11/08

Figura 97. Planificación inicial de la sub-fase “Diseño”

Nombre de tarea	Duración	Comienzo	Fin
<input type="checkbox"/> Fase de Diseño	22 días	vie 24/10/08	lun 24/11/08
<input type="checkbox"/> Diseño Aplicación soporte páginas Web	8 días	vie 24/10/08	mar 04/11/08
Diseño de la(s) estructura(s) de las base(s) de datos	1 día	vie 24/10/08	vie 24/10/08
<input type="checkbox"/> Diseño de la base de datos "Sitios Web"	1 día	lun 27/10/08	lun 27/10/08
Diseño de los metadatos	1 día	lun 27/10/08	lun 27/10/08
Elaboración del diagrama Entidad-Relación	1 día	lun 27/10/08	lun 27/10/08
Elaboración del diagrama Relacional	1 día	lun 27/10/08	lun 27/10/08
Diseño de las relaciones entre tablas	1 día	lun 27/10/08	lun 27/10/08
<input type="checkbox"/> Diseño de la base de datos "Administradores"	1 día	mar 28/10/08	mar 28/10/08
Diseño de los metadatos	1 día	mar 28/10/08	mar 28/10/08
Elaboración del diagrama Entidad-Relación	1 día	mar 28/10/08	mar 28/10/08
Elaboración del diagrama Relacional	1 día	mar 28/10/08	mar 28/10/08
Diseño de las relaciones entre tablas	1 día	mar 28/10/08	mar 28/10/08
<input type="checkbox"/> Diseño de la estructura de la Aplicación	5 días	mié 29/10/08	mar 04/11/08
Diseño de los perfiles	1 día	mié 29/10/08	mié 29/10/08
Diseño de las interfaces	1 día	jue 30/10/08	jue 30/10/08
Diseño de la funcionabilidad de la aplicación	2 días	vie 31/10/08	lun 03/11/08
Diseño de la arquitectura de la aplicación	1 día	mar 04/11/08	mar 04/11/08
<input type="checkbox"/> Diseño Aplicación de búsqueda orientada	13 días	mié 05/11/08	vie 21/11/08
<input type="checkbox"/> Diseño de la Arquitectura de la aplicación	13 días	mié 05/11/08	vie 21/11/08
Diseño de las interfaces	2 días	mié 05/11/08	jue 06/11/08
Diseño del Árbol de Conceptos	2 días	vie 07/11/08	lun 10/11/08
Diseño de la estructura de la aplicación	7 días	mar 11/11/08	mié 19/11/08
Diseño de los árboles de correspondencias necesarios	2 días	jue 20/11/08	vie 21/11/08
Diseño de las pruebas a realizar en la fase de Implementación	1 día	lun 24/11/08	lun 24/11/08
Diseño de las pruebas a realizar en la fase de Implantación	1 día	lun 24/11/08	lun 24/11/08

Figura 98. Resultado real de la sub-fase “Diseño”

Para facilitar la comparación entre la planificación inicial y la planificación real, se van a mostrar tres tablas comparativas como la de apartados anteriores. En la primera de ellas aparecerá la comparativa entre la planificación inicial y el resultado real en la tarea “Diseño de la Aplicación soporte páginas Web”, en la segunda aparecerá dicha comparativa para la tarea “Diseño de la Aplicación de búsqueda orientada”, y en la tercera y última aparecerán los datos de las tablas anteriores junto a las pruebas.

**Tarea “Diseño de la Aplicación soporte páginas Web”.**

	Planificación inicial		Resultado real	
<b>Diseño de la estructura(s) de la base(s) de dato(s)</b>	2 días	8 horas	1 día	5 horas
<b>(Diseño de la base de datos “Sitios Web”) Diseño de los metadatos</b>	2 días(*1)	3 horas	1 día(*1)	2 horas
<b>(Diseño de la base de datos “Sitios Web”) Elaboración del diagrama Entidad-Relación</b>	2 días(*1)	5 horas	1 día(*1)	2 horas
<b>(Diseño de la base de datos “Sitios Web”) Elaboración del diagrama Relacional</b>	2 días(*1)	3 horas	1 día(*1)	2 horas
<b>(Diseño de la base de datos “Sitios Web”) Diseño de las relaciones entre tablas</b>	2 días(*1)	3 horas	1 día(*1)	1 horas
<b>(Diseño de la base de datos “Administradores”) Diseño de los metadatos</b>	2 días(*2)	3 horas	1 día(*2)	2 horas
<b>(Diseño de la base de datos “Administradores”) Elaboración del diagrama Entidad-Relación</b>	2 días(*2)	5 horas	1 día(*2)	2 horas
<b>(Diseño de la base de datos “Administradores”) Elaboración del diagrama</b>	2 días(*2)	3 horas	1 día(*2)	2 horas

<b>Relacional</b>				
<b>(Diseño de la base de datos “Administradores”) Diseño de las relaciones entre tablas</b>	2 días(*2)	3 horas	1 día(*2)	1 horas
<b>(Diseño de la estructura de la aplicación) Diseño de los perfiles</b>	1 día	7 horas	1 día	10 horas
<b>(Diseño de la estructura de la aplicación) Diseño de las interfaces</b>	1 día	7 horas	1 día	4 horas
<b>(Diseño de la estructura de la aplicación) Diseño de la funcionalidad de la aplicación</b>	2 días	14 horas	2 días	13 horas
<b>(Diseño de la estructura de la aplicación) Diseño de la arquitectura de la aplicación</b>	1 día	7 horas	1 días	3 horas
<b>Total tarea</b>	11 días	71 horas	8 días	49 horas

\*1. Estas tareas se realizan a la vez en el plazo estipulado (número de días).

\*2. Estas tareas se realizan a la vez en el plazo estipulado (número de días).

Tabla 18. Sub-Fase (parte 1) “Diseño”. Comparativa planificación inicial y resultado real.



**Tarea “Diseño de la Aplicación de búsqueda orientada”.**

	<b>Planificación inicial</b>		<b>Resultado real</b>	
<b>(Diseño de la arquitectura de la aplicación) Diseño de las interfaces</b>	2 días	15 horas	2 día	13 horas
<b>(Diseño de la arquitectura de la aplicación) Diseño del Árbol de Conceptos</b>	2 días	15 horas	2 día	10 horas
<b>(Diseño de la arquitectura de la aplicación) Diseño de la estructura de la aplicación</b>	13 días	80 horas	7 día	65 horas
<b>(Diseño de la arquitectura de la aplicación) Diseño de los Árboles de Correspondencias necesarios</b>	3 días	22 horas	2 día	18 horas
<b>Total tarea</b>	20 días	132 horas	13 días	106 horas

Tabla 19. Sub-Fase (parte 2) “Diseño”. Comparativa planificación inicial y resultado real.

### Sub-fase diseño completa

	Planificación inicial		Resultado real	
Tarea “Diseño de la Aplicación soporte páginas Web”.	11 días	71 horas	8 días	49 horas
Tarea “Diseño de la Aplicación de búsqueda orientada”.	20 días	132 horas	13 días	106 horas
Diseño de las pruebas a realizar en la fase de implementación	1 día(*1)	4 horas	1 día(*1)	3 horas
Diseño de las pruebas a realizar en la fase de implantación	1 día(*1)	4 horas	1 día(*1)	3 horas
<b>Sub-fase completa</b>	<b>32 días</b>	<b>211 horas</b>	<b>22 días</b>	<b>161 horas</b>

\*1. Estas tareas se realizan a la vez en el plazo estipulado (número de días).

Tabla 20. Sub-Fase “Diseño”. Comparativa planificación inicial y resultado real.

Analizando la sub-fase “diseño” nos damos cuenta que una vez más estamos ante una planificación inadecuada. Como ocurría en el caso anterior la previsión ha sido demasiado pesimista, ya que en un menor número de horas el trabajo podría haberse realizado.

A pesar del incremento de horas que ha implicado el cambio exigido por el cliente y que ha sido descrito en el apartado 8.1 Cambios en el proyecto, aproximadamente 5 horas, se ha conseguido reducir el tiempo estimado considerablemente.

Parece que el estudio realizado en fases anteriores está provocando un efecto positivo en fases posteriores, como en la sub-fase de análisis o en esta sub-fase de diseño. Se puede decir que el estudio exhaustivo realizado está permitiendo al desarrollador del proyecto avanzar con pasos más firmes en espacios de tiempo más reducidos.

#### 10.2.3.4 SUB-FASE IMPLEMENTACIÓN

La sub-fase de implementación incluye todas las tareas relacionadas con la implementación de las aplicaciones diseñadas. Esta sub-fase consta básicamente de implementar todo aquello que fue diseñado en la sub-fase anterior. En la Figura 99 y la Figura 100 podemos observar la parte del Diagrama de Gantt correspondiente a la sub-fase implementación de la planificación inicial y del resultado real.

Nombre de tarea	Duración	Comienzo	Fin
<input type="checkbox"/> <b>Fase de Implementación</b>	<b>47 días</b>	<b>lun 24/11/08</b>	<b>jue 12/02/09</b>
<input type="checkbox"/> <b>Implementación de la Aplicación soporte páginas Web</b>	<b>22 días</b>	<b>lun 24/11/08</b>	<b>jue 08/01/09</b>
<input type="checkbox"/> <b>Implementación de la base de datos "Sitios Web"</b>	<b>1 día</b>	<b>lun 24/11/08</b>	<b>lun 24/11/08</b>
Implementación de los metadatos	1 día	lun 24/11/08	lun 24/11/08
Implementación de la estructura de la base de datos	1 día	lun 24/11/08	lun 24/11/08
<input type="checkbox"/> <b>Implementación de la base de datos "Administradores"</b>	<b>1 día</b>	<b>mar 25/11/08</b>	<b>mar 25/11/08</b>
Implementación de los metadatos	1 día	mar 25/11/08	mar 25/11/08
Implementación de la estructura de la base de datos	1 día	mar 25/11/08	mar 25/11/08
<input type="checkbox"/> <b>Implementación de la estructura de la Aplicación</b>	<b>20 días</b>	<b>mié 26/11/08</b>	<b>jue 08/01/09</b>
Implementación de los perfiles	2 días	mié 26/11/08	jue 27/11/08
Implementación de las interfaces	2 días	vie 28/11/08	lun 01/12/08
Implementación de la funcionalidad de la aplicación	13 días	mar 02/12/08	lun 05/01/09
Implementación de la estructura de la funcionalidad	3 días	mar 06/01/09	jue 08/01/09
<input type="checkbox"/> <b>Implementación Aplicación de búsqueda orientada</b>	<b>21 días</b>	<b>vie 09/01/09</b>	<b>vie 06/02/09</b>
<input type="checkbox"/> <b>Implementación de la Arquitectura de la aplicación</b>	<b>21 días</b>	<b>vie 09/01/09</b>	<b>vie 06/02/09</b>
Implementación de las interfaces	3 días	vie 09/01/09	mar 13/01/09
Implementación del Árbol de Conceptos	1 día	mié 14/01/09	mié 14/01/09
Implementación de la estructura de la aplicación	15 días	jue 15/01/09	mié 04/02/09
Implementación de los árboles de correspondencias necesarios	2 días	jue 05/02/09	vie 06/02/09
Realización de las pruebas diseñadas	2 días	lun 09/02/09	mar 10/02/09
Documentación de errores y correcciones realizadas	2 días	mié 11/02/09	jue 12/02/09

Figura 99. Planificación inicial de la sub-fase “Implementación”

Nombre de tarea	Duración	Comienzo	Fin
<b>[-] Fase de Implementación</b>	<b>39 días</b>	<b>mar 25/11/08</b>	<b>mar 03/02/09</b>
<b>[-] Implementación de la Aplicación soporte páginas Web</b>	<b>18 días</b>	<b>mar 25/11/08</b>	<b>lun 05/01/09</b>
<b>[-] Implementación de la base de datos "Sitios Web"</b>	<b>1 día</b>	<b>mar 25/11/08</b>	<b>mar 25/11/08</b>
Implementación de los metadatos	1 día	mar 25/11/08	mar 25/11/08
Implementación de la estructura de la base de datos	1 día	mar 25/11/08	mar 25/11/08
<b>[-] Implementación de la base de datos "Administradores"</b>	<b>1 día</b>	<b>mié 26/11/08</b>	<b>mié 26/11/08</b>
Implementación de los metadatos	1 día	mié 26/11/08	mié 26/11/08
Implementación de la estructura de la base de datos	1 día	mié 26/11/08	mié 26/11/08
<b>[-] Implementación de la estructura de la Aplicación</b>	<b>16 días</b>	<b>jue 27/11/08</b>	<b>lun 05/01/09</b>
Implementación de los perfiles	1 día	jue 27/11/08	jue 27/11/08
Implementación de las interfaces	3 días	vie 28/11/08	mar 02/12/08
Implementación de la funcionabilidad de la aplicación	9 días	mié 03/12/08	mar 16/12/08
Implementación de la estructura de la funcionabilidad	3 días	mié 17/12/08	lun 05/01/09
<b>[-] Implementación Aplicación de búsqueda orientada</b>	<b>18 días</b>	<b>mar 06/01/09</b>	<b>jue 29/01/09</b>
<b>[-] Implementación de la Arquitectura de la aplicación</b>	<b>18 días</b>	<b>mar 06/01/09</b>	<b>jue 29/01/09</b>
Implementación de las interfaces	1 día	mar 06/01/09	mar 06/01/09
Implementación del Árbol de Conceptos	1 día	mié 07/01/09	mié 07/01/09
Implementación de la estructura de la aplicación	14 días	jue 08/01/09	mar 27/01/09
Implementación de los árboles de correspondencias necesarios	2 días	mié 28/01/09	jue 29/01/09
Realización de las pruebas diseñadas	2 días	vie 30/01/09	lun 02/02/09
Documentación de errores y correcciones realizadas	1 día	mar 03/02/09	mar 03/02/09

Figura 100. Resultado real de la sub-fase “Implementación”

Siguiendo el proceso de la sub-fase de diseño, y para facilitar la comparación entre la planificación inicial y la planificación real, se van a mostrar tres tablas comparativas como la de apartados anteriores. En la primera de ellas aparecerá la comparativa entre la planificación inicial y el resultado real en la tarea “Implementación de la Aplicación soporte páginas Web”, en la segunda aparecerá dicha comparativa para la tarea “Implementación de la Aplicación de búsqueda orientada”, y en la tercera y última aparecerán los datos de las tablas anteriores junto a las pruebas.

#### Tarea “Implementación de la Aplicación soporte páginas Web”.

	Planificación inicial		Resultado real	
<b>(Implementación de la base de datos “Sitios Web”) Implementación de los metadatos</b>	1 día(*1)	1 hora	1 día(*1)	1 hora
<b>(Implementación de la base de datos “Sitios Web”) Implementación de la estructura de la base de datos</b>	1 días(*1)	7 horas	1 día(*1)	5 horas

<b>(Implementación de la base de datos “Administradores”) Implementación de los metadatos</b>	1 día(*2)	1 hora	1 día(*2)	1 hora
<b>(Implementación de la base de datos “Administradores”) Implementación de la estructura de la base de datos</b>	1 día(*2)	7 horas	1 día(*2)	3 horas
<b>(Implementación de la estructura de la aplicación) Implementación de los perfiles</b>	2 días	15 horas	1 día	10 horas
<b>(Implementación de la estructura de la aplicación) Implementación de las interfaces</b>	2 días	14 horas	3 día	20 horas
<b>(Implementación de la estructura de la aplicación) Implementación de la funcionalidad de la aplicación</b>	13 días	100 horas	9 días	105 horas
<b>(Implementación de la estructura de la aplicación) Implementación de la arquitectura de la aplicación</b>	3 días	22 horas	3 días	20 horas
<b>Total tarea</b>	22 días	167 horas	18 días	167 horas

\*1. Estas tareas se realizan a la vez en el plazo estipulado (número de días).

\*2. Estas tareas se realizan a la vez en el plazo estipulado (número de días).

Tabla 21. Sub-Fase (parte 1) “Implementación”. Comparativa planificación inicial y resultado real.

**Tarea “Implementación de la Aplicación de búsqueda orientada”.**

	<b>Planificación inicial</b>		<b>Resultado real</b>	
<b>(Implementación de la arquitectura de la aplicación) Implementación de las interfaces</b>	3 días	21 horas	1 día	10 horas
<b>(Implementación de la arquitectura de la aplicación) Implementación del Árbol de Conceptos</b>	1 día	8 horas	1 día	7 horas
<b>(Implementación de la arquitectura de la aplicación) Implementación de la estructura de la aplicación</b>	15 días	110 horas	14 días	130 horas
<b>(Implementación de la arquitectura de la aplicación) Implementación de los Árboles de Correspondencias necesarios</b>	2 días	12 horas	2 días	11 horas
<b>Total tarea</b>	21 días	151 horas	18 días	158 horas

Tabla 22. Sub-Fase (parte 2) “Implementación”. Comparativa planificación inicial y resultado real.

### Sub-fase implementación completa

	Planificación inicial		Resultado real	
<b>Tarea “implementación de la Aplicación soporte páginas Web”.</b>	22 días	167 horas	18 días	167 horas
<b>Tarea “implementación de la Aplicación de búsqueda orientada”.</b>	21 días	151 horas	18 días	158 horas
<b>Realización de las pruebas diseñadas</b>	2 días	3 horas	2 días	5 horas
<b>Documentación de errores y correcciones realizadas</b>	2 días	11 horas	1 día	7 horas
<b>Sub-fase completa</b>	47 días	332 horas	39 días	337 horas

Tabla 23. Sub-Fase “Implementación”. Comparativa planificación inicial y resultado real.

Tal y como se puede observar en la Tabla 22, la desviación en cuanto a horas no es muy significativa. Estudiando dicha tabla se puede observar que el alumno ha apretado un poco más y ha conseguido rebajar el número de días estimados en siete.

El esfuerzo del alumno por rebajar el número de días estimados ha sido máximo lo cual le ha llevado a rebajar el número de días aún siendo superior el número de horas reales al número de horas estimadas. Este esfuerzo es debido en parte al desarrollo de la petición realizada por el cliente en el apartado 8.1 Cambios en el proyecto, ya que dicho cambio ha supuesto aproximadamente un plus de 25 horas en esta fase.

#### 10.2.3.5 SUB-FASE IMPLANTACIÓN

Esta sub-fase consta básicamente de la instalación de la aplicación en los servidores de la Universidad del País Vasco, así como la puesta en marcha de las aplicaciones. En la Figura 101 y la Figura 102 podemos observar la parte del Diagrama de Gantt correspondiente a la sub-fase implantación de la planificación inicial y del resultado real.

Nombre de tarea	Duración	Comienzo	Fin
<input type="checkbox"/> Fase de Implantación	5 días	vie 13/02/09	jue 19/02/09
Instalación de la(s) bases de datos realizadas en el lo(s) servidor(es)	2 días	vie 13/02/09	lun 16/02/09
Configuración de la comunicación entre bases de datos	2 días	mar 17/02/09	mié 18/02/09
Puesta en marcha del buscador	1 día	jue 19/02/09	jue 19/02/09

Figura 101. Planificación inicial de la sub-fase “Implantación”.

Nombre de tarea	Duración	Comienzo	Fin
<input type="checkbox"/> Fase de Implantación	8 días	mié 04/02/09	vie 13/02/09
Instalación de la(s) bases de datos realizadas en el lo(s) servidor(es)	2 días	mié 04/02/09	jue 05/02/09
Configuración de la comunicación entre bases de datos	1 día	vie 06/02/09	vie 06/02/09
Puesta en marcha del buscador	5 días	lun 09/02/09	vie 13/02/09

Figura 102. Resultado real de la sub-fase “Implantación”.

### Comparación entre planificación inicial y resultado real

	Planificación inicial		Resultado real	
	Días	Horas	Días	Horas
<b>Instalación de la(s) bases de datos realizadas en el/lo(s) servidor(es)</b>	2 días	12 horas	2 días	1 hora
<b>Configuración de la comunicación entre bases de datos</b>	2 día	12 horas	1 día	3 horas
<b>Puesta en marcha del buscador</b>	1 día	7 horas	5 días	16 horas
<b>Total Fase</b>	5 días	31 horas	8 días	19 horas

Tabla 24. Sub-Fase “Implantación”. Comparativa planificación inicial y resultado real.

En esta fase existen dos factores que explican los resultados. El primero de ellos es el buen funcionamiento de la aplicación a nivel local, lo que ha permitido al alumno instalar la aplicación sin ningún problema exceptuando los del propio servidor de la Universidad del País Vasco, que han sido muchos. Por otra parte la inexperiencia del alumno en lo que se refiere al tratamiento con servidores en general, y con el de la Universidad del País Vasco en particular ha hecho que el tiempo se excediera más de lo que la propia tarea debería haber llevado. Sin embargo considerando ambos factores, el trabajo real es inferior (en número de horas) a la estimación realizada.



### 10.2.3.6 SUB-FASE PRUEBAS

Esta sub-fase recoge las pruebas realizadas por el cliente sobre la aplicación. En la Figura 103 y la Figura 104 podemos observar la parte del Diagrama de Gantt correspondiente a la sub-fase pruebas de la planificación inicial y del resultado real.

Nombre de tarea	Duración	Comienzo	Fin
<input type="checkbox"/> <b>Pruebas</b>	<b>4 días</b>	<b>vie 20/02/09</b>	<b>mié 25/02/09</b>
Definición de las pruebas	1 día	vie 20/02/09	vie 20/02/09
Ejecución de las pruebas definidas	1 día	lun 23/02/09	lun 23/02/09
Documentación de errores y correcciones realizadas	2 días	mar 24/02/09	mié 25/02/09

Figura 103. Planificación inicial de la sub-fase “Pruebas”

Nombre de tarea	Duración	Comienzo	Fin
<input type="checkbox"/> <b>Pruebas</b>	<b>4 días</b>	<b>lun 16/02/09</b>	<b>jue 19/02/09</b>
Definición de las pruebas	1 día	lun 16/02/09	lun 16/02/09
Ejecución de las pruebas definidas	1 día	mar 17/02/09	mar 17/02/09
Documentación de errores y correcciones realizadas	2 días	mié 18/02/09	jue 19/02/09

Figura 104. Resultado real de la sub-fase “Pruebas”

#### Comparación entre planificación inicial y resultado real

	Planificación inicial		Resultado real	
<b>Definición de las pruebas</b>	1 día	7 horas	1 días	1 horas
<b>Ejecución de las pruebas definidas</b>	1 día	8 horas	1 días	1 horas
<b>Documentación de errores y correcciones realizadas</b>	2 días	9 horas	2 días	2 horas
<b>Total Fase</b>	4 días	24 horas	4 días	5 horas

Tabla 25. Sub-Fase “Pruebas”. Comparativa planificación inicial y resultado real.

En la sub-tarea “Pruebas” vuelve aparecer una pequeña desviación en el número de horas. Esto es debido a que el alumno delega la ejecución de las pruebas en el cliente, por lo que la única actividad que realiza en esta sub-fase es una pequeña explicación acerca del funcionamiento de la aplicación, y el coste de tiempo asociado a los viajes hasta el lugar de trabajo del cliente y el tiempo invertido allí.

### 10.2.3.7 TOTAL FASE DESARROLLO

	Planificación inicial		Resultado real	
<b>Sub-fase estudio previo</b>	7 días	45 horas	33 días	210 horas
<b>Sub-fase análisis</b>	15 días	98 horas	14 días	68 horas
<b>Sub-fase diseño</b>	32 días	211 horas	22 días	161 horas
<b>Sub-fase implementación</b>	47 días	332 horas	39 días	337 horas
<b>Sub-fase implantación</b>	5 días	31 horas	8 días	19 horas
<b>Sub-fase pruebas</b>	4 días	24 horas	4 días	5 horas
<b>Total Fase</b>	110 días	741 horas	120 días	800 horas

Tabla 26. Fase “Desarrollo” Comparativa planificación inicial y resultado real.

### 10.2.4 TOTAL PROYECTO

	Planificación inicial		Resultado real	
<b>Fase Previo</b>	6 días	8 horas	8 días	7 horas
<b>Fase Iniciación</b>	23 días	49 horas	30 días	64 horas
<b>Fase Desarrollo</b>	110 días	741 horas	120 días	800 horas
<b>Total Fase</b>	129 días	798 horas	158 días	871 horas

Tabla 27. Fase “Desarrollo” Comparativa planificación inicial y resultado real.

## Diagrama de Gantt correspondiente a la planificación inicial

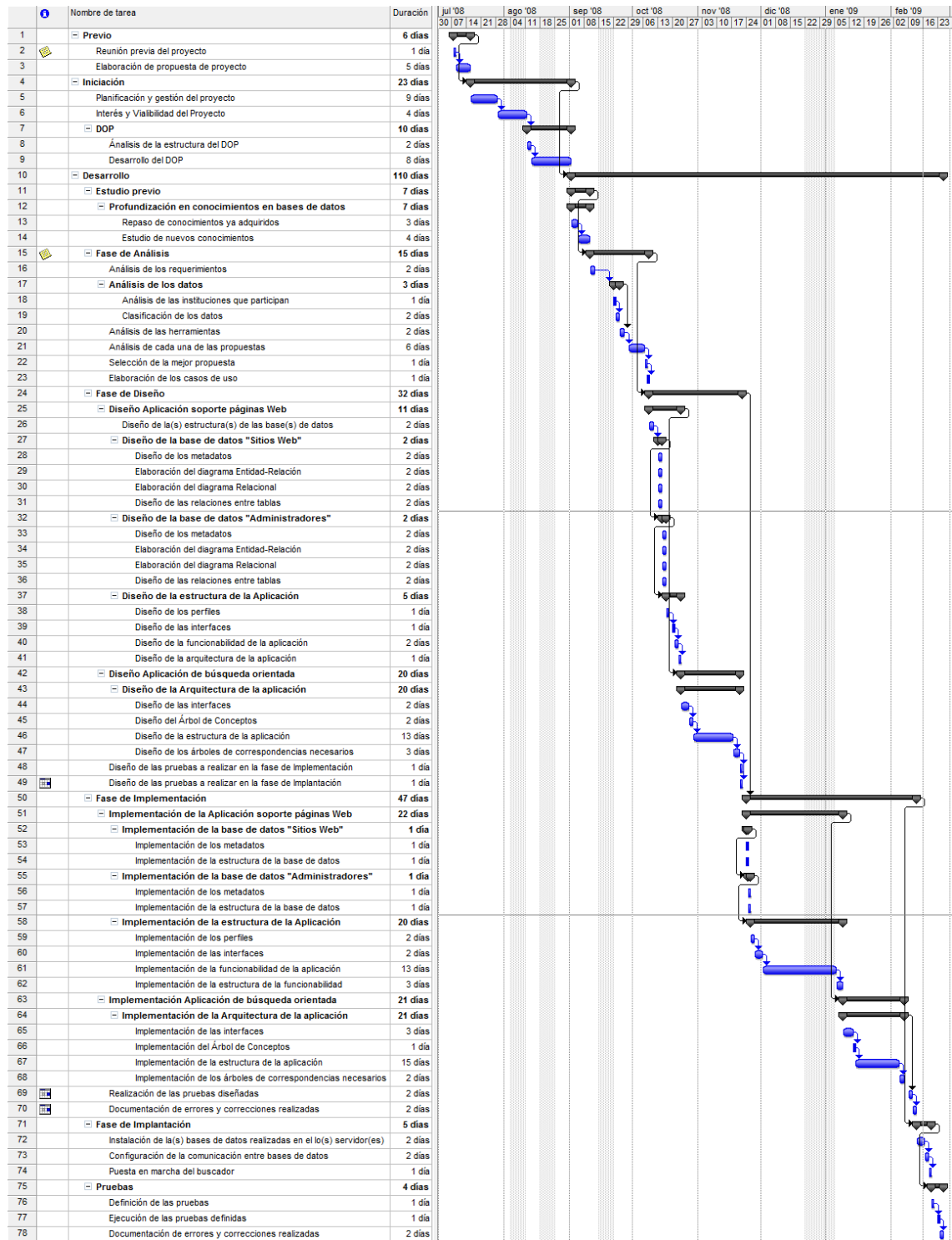


Figura 105. Diagrama de Gantt correspondiente a la planificación inicial.

## Diagrama de Gantt correspondiente al resultado real

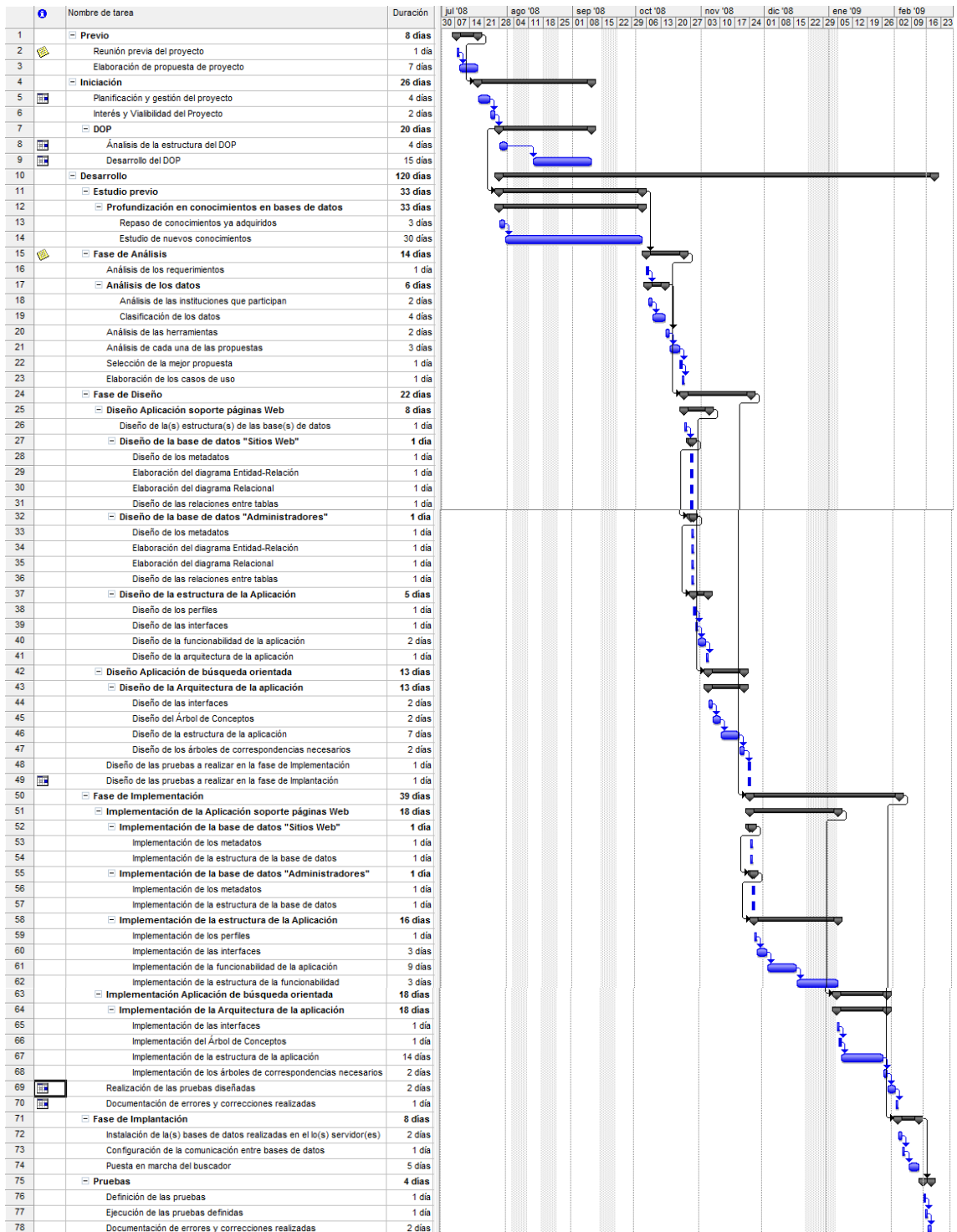


Figura 106. Diagrama de Gantt correspondiente a la resultado real.

### 10.3 CONCLUSIONES

Una vez desarrolladas y analizadas todas las fases del proyecto, es el momento de analizar el proyecto en su conjunto, y obtener conclusiones. Para ello se ha representado en modo gráfico la comparación entre, la planificación inicial y el resultado real del proyecto completo (en horas) (Figura 107), del proyecto dividido en fases (en horas) (Figura 108), y de la descomposición de la fase de desarrollo (Figura 109).

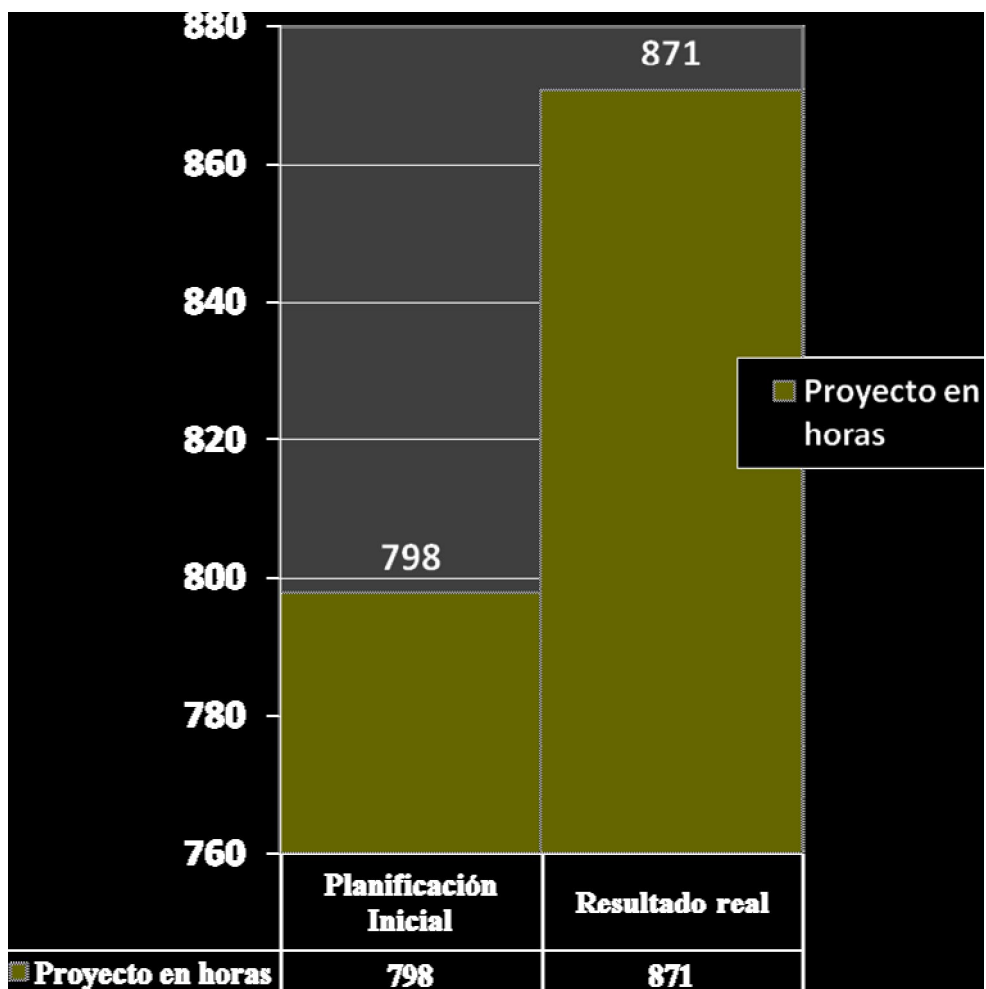


Figura 107. Comparación (en horas) entre la planificación inicial (1ª barra) y el resultado real (2ª barra)

En esta primera figura (Figura 107) podemos comprobar cómo en número de horas el alumno ha necesitado invertir más de 70 horas más para poder completar el

proyecto con éxito. Para un proyecto que nació de una idea abstracta, y que ha ido tomando forma según avanzaba el propio proyecto podríamos decir que tampoco es una desviación temporal muy importante. Si bien es cierto, el alumno calificó su planificación inicial, como pesimista, es decir consideraba que el proyecto se podía llevar a cabo en menos tiempo del que el mismo planificó, quizás esta planificación “pesimista” ha permitido que el resultado final en lo que a la comparación entre la planificación inicial y el resultado real se refiere sea bastante positivo.

En la siguiente figura (Figura 108) podemos observar como en la fase desarrollo es donde recae todo el peso de la aplicación. Esta fase se desarrollan todas aquellas funciones importantes, desde el estudio previo hasta la implantación en el servidor de la Universidad del País Vasco, pasando por la fase de diseño, implementación... Además en términos comparativos podemos decir que todas las fases son bastante semejantes en lo que a duración en horas se refiere.

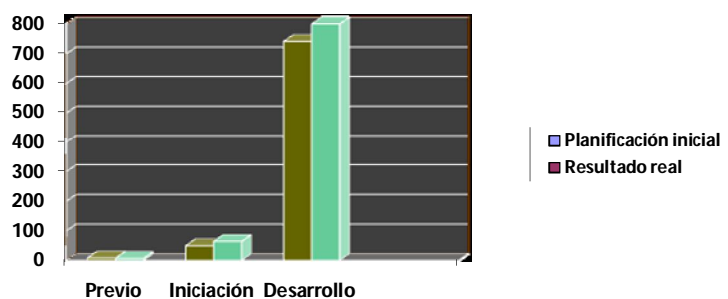


Figura 108. Comparación por fases (en horas) entre la planificación inicial y el resultado real

Si analizamos la Figura 109 nos damos cuenta que existe una variación notable en muchas sub-fases de la fase desarrollo entre lo estimado y lo real. Sin embargo esto únicamente da lugar a una variación apenas notable que queda reflejada en la Figura 108. La sub-fase donde más se aprecia una gran variación entre lo estimado y lo real es en la sub-fase “Estudio”, en la que el alumno invierte mucho más tiempo de lo esperado. Sin embargo el aprendizaje desarrollado en esta fase da lugar a que en sub-fases posteriores como el análisis y el diseño el tiempo real pueda ser disminuido respecto al tiempo estimado, equilibrando la comparación entre tiempo estimado y tiempo real de la fase desarrollo.

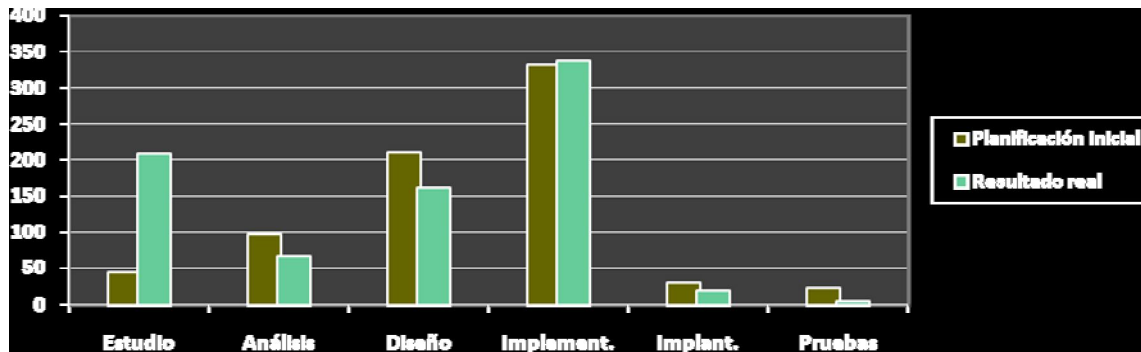


Figura 109. Comparación por tareas de la fase desarrollo (en horas) entre la planificación inicial y el resultado real

En conclusión se considera que la planificación ha sido bastante buena pese a la inexperiencia del alumno en proyectos de este tipo. Quizás la experiencia del proyecto realizado como Proyecto de Fin de Carrera para la Ingeniería Técnica en Informática de Gestión, y proyectos realizados en asignaturas como Planificación y Gestión de Proyectos hayan sido culpables de que el alumno haya realizado una planificación que él mismo consideraba “pesimista”, porque se creía capaz de terminarlo antes de la fecha indicada.

## 11. CONCORDANCIA ENTRE LOS RESULTADOS OBTENIDOS Y LOS OBJETIVOS PLANTEADOS

Una vez terminadas, y documentadas las dos aplicaciones desarrolladas durante el Proyecto de Fin de Carrera, es el momento de echar la vista atrás y analizar si se han cumplido los objetivos solicitados por el cliente al inicio del proyecto. Para ellos recordaremos el análisis de requerimientos realizado por el cliente al comienzo del proyecto desarrollador, e iremos analizando punto por punto qué apartado han sido cubiertos, y cuáles por el contrario no ha sido posible cubrir.

A continuación se listan los requerimientos exigidos por el cliente para obtener un producto satisfactorio.

- ✓ Al tratarse de un desarrollo experimental, se pide obtener una solución y realizar una valoración crítica de la misma. La solución no será óptima debido a que se desconocen varios factores transcendentales y no se cuenta con la colaboración del conjunto de instituciones, pero será un primer paso fundamental sobre el cual se podrá empezar a recabar esta colaboración y diseñar soluciones más adecuadas.

**Requerimiento cumplido.** Se han desarrollado dos aplicaciones, la cual una de ellas, la aplicación de búsqueda orientada, es aquella que se pedía como una solución al problema. Además todo el proceso de elaboración de la solución está perfectamente documentado. Desde el comienzo del proyecto hasta su instalación, pasando por el estudio de los datos, de las posibles soluciones, el diseño de la aplicación, y una explicación de su implementación. Además la memoria incluye aspectos de gestión de la planificación, relaciones con el cliente, y un apartado final de conclusiones.

- ✓ La solución adoptada debe funcionar como un buscador web convencional de tal manera que el usuario pueda realizar las búsquedas de la manera más intuitiva posible.

**Requerimiento cumplido.** La aplicación de búsqueda orientada permite al usuario desarrollar cualquier tipo de búsqueda acerca del patrimonio histórico de una manera intuitiva y sencilla.

- ✓ Las búsquedas podrán realizarse directamente por palabras clave desde el nivel inicial o ir desplegando jerarquías para ir limitando el ámbito de búsqueda, por ejemplo [Patrimonio] -> [Patrimonio edificado] -> [Arquitectura religiosa].

**Requerimiento cumplido.** La aplicación desarrollada va a permitir al usuario realizar consultas globales por un lado y realizar consultas mas especializadas por otro lado. Esto es debido a que el sistema ofrece la posibilidad al usuario de



realizar la búsqueda sobre un concepto general, o por el contrario especializar dicho concepto en uno más concreto.

- ✓ Las búsquedas deben realizarse tanto sobre información web previamente seleccionada como sobre bases de datos de instituciones relacionadas con el patrimonio alavés.

**Requerimiento cumplido.** Para lo primero se ha informatizado la información existente en las páginas Web de las diferentes organizaciones en una base de datos que ahora pertenece a nuestro sistema de búsqueda orientada. Además como comienzo también se cuenta con otra base de datos heterogénea que almacena información sobre los proyectos llevados a cabo por el Laboratorio de Documentación Geométrica del Patrimonio. En un futuro se espera la incorporación de nuevas fuentes de datos. Para ello además se ha elaborado un manual que permita que su incorporación sea lo más sencilla posible.

- ✓ Debido al carácter dinámico de la información que se va a catalogar y ofrecer, debe proporcionarse la manera de ir actualizando los enlaces a los contenidos y este proceso debe realizarse de la manera más simple y ágil posible.

**Requerimiento cumplido.** Ambas aplicaciones han sido desarrolladas con vistas a una posible ampliación y variación de contenidos. Por esto podemos decir que ambas aplicaciones son aplicaciones escalables. Además se ha desarrollado un manual de mantenimiento para cada una de las aplicaciones en los cuales se explica cada una de las modificaciones y ampliaciones que se pueden llevar a cabo.

## **Conclusión**

Como se observa, todos los requerimientos han sido abarcados completamente. Además de esto es necesario recordar que el cliente solicitó una mejora en uno de los contenidos de la aplicación, cuyo coste asociado en término de horas era de aproximadamente 30 horas, y cuya mejora fue también desarrollada perfectamente.

Ante este resultado podemos calificar a este proyecto desarrollado como un éxito, ya que ha cumplimentado todos los requisitos expuestos en un principio por el cliente del propio proyecto.

## **12. RELACIONES CON EL CLIENTE**

A lo largo de este apartado titulado “Relaciones con el cliente” se analizarán para cada una de las fases del proyecto la relación mantenida entre el alumno y encargado de llevar a cabo el proyecto, Ion Marín, y el cliente, el Laboratorio de Documentación Geométrica del Patrimonio.

### **12.1 FASE DEL PRE-PROYECTO**

Hacia mediados de Junio de 2008, a través de una serie de intermediarios llega a Ion Marín, alumno de la Universidad del País Vasco, una propuesta de un posible Proyecto de Fin de Carrera. Se trataba de un proyecto propuesto por el Laboratorio de Documentación Geométrica del Patrimonio de la UPV/EHU, cuyo objetivo principal era conseguir reunir toda la información del patrimonio, que en ese momento poseían cada una de las instituciones u organizaciones de Álava. La parte cliente iba a estar formada por los responsables del propio Laboratorio de Documentación Geométrica del Patrimonio de la UPV/EHU, José Manuel Valle Melón, y Álvaro Rodríguez. Ambos profesores de la Universidad del País Vasco y responsables, además, del propio laboratorio. El encargado de llevar a cabo el proyecto sería Ion Marín, tal y como se ha comentado anteriormente, estudiante de Ingeniería Informática en la propia Universidad del País Vasco, al cual le falta únicamente el Proyecto de Fin de Carrera para terminar la carrera.

Aproximadamente sobre el 10 de Julio se produce el primer encuentro entre las dos partes: cliente y alumnos, ya que en un principio, eran dos los alumnos que iban a encargarse de llevar el proyecto a cabo, quedando al final únicamente uno de ellos. En este encuentro, el cliente expone, de manera muy superficial, a los alumnos cuál es la situación actual y qué es lo que se pretende conseguir. El proyecto constaría de dos partes básicas, una de ellas una fase de investigación, en la cual se analizarían diferentes soluciones posibles, y una segunda en la que se llevaría a cabo una de las posibles soluciones estudiadas en la primera fase. El cliente en esta primera toma de contacto asegura que no es necesaria la implementación de una aplicación idónea para el problema que se plantea. Siendo consciente de la poca información sobre el estado de los datos en las diferentes instituciones u organizaciones de Álava y alrededores, y siendo consciente, además, de la inexistencia de aplicaciones similares en el mercado, lo que pretende conseguir es un primer paso para lograr en un futuro un producto final. Por esto, el objetivo principal de este proyecto es conseguir desarrollar una aplicación base que sirva de enganche para que las organizaciones dedicadas a temas relacionados con el patrimonio, se interesen por el proyecto y puedan en un futuro colaborar con el mismo.

En esta primera toma de contacto el cliente ha expuesto sus necesidades, y su objetivo final, sin embargo carece de información tan vital como ¿Qué datos están disponibles? ¿Qué organizaciones están dispuestas a colaborar con el proyecto? ¿Dónde están almacenados los datos?, ¿Cómo están almacenados los mismos? ¿Serían accesibles? Y si esto fuera así, ¿Cómo serían accesibles? Todas estas preguntas nos

hacen, desde un principio, suponer que se va a tratar de un proyecto de larga duración, en el cual las preguntas escritas anteriormente irán siendo contestadas en la medida que el propio proyecto vaya desarrollándose. Suponemos desde un principio que las reuniones que van a tomar parte durante la elaboración del proyecto, con el cliente, van a ser de vital importancia para ir construyendo el objetivo final entre ambas partes, cliente y desarrollador del proyecto.

Ambas partes, cliente y alumno, en un principio, acuerdan llevar a cabo el proyecto. Como comienzo, el primer paso será enviar una propuesta de proyecto al tutor del mismo José Miguel Blanco. Dicha propuesta puede observarse en el apartado de la memoria denominado “*Propuesta del proyecto*”.

## **12.2 FASE DE ACUERDO**

Una vez que el tutor, José Miguel Blanco, acepta la propuesta de Proyecto de Fin de Carrera, aproximadamente el 20 de julio, la relación entre el cliente y el alumno, Ion Marín, comienza a acentuarse, sobre todo a través de correo electrónico. Aunque el mes de Agosto es considerado tanto por Ion como por el cliente como un mes vacacional, Ion comienza a llevar a cabo un estudio superficial sobre diferentes propuestas que pueden ser desarrolladas para resolver el problema planteado en el Proyecto Fin de Carrera.

A la vuelta de las vacaciones de verano, concretamente el día 2 de septiembre, ambas partes, cliente y alumno, se reúnen con el fin de fijar el alcance del proyecto, y definir el mismo de manera más concreta. Durante la reunión José Manuel Valle expone una serie de páginas Web que pueden servir de guía para el desarrollo del proyecto. Estas páginas Web son del estilo a [www.edreams.com](http://www.edreams.com), [www.muchoviaje.com](http://www.muchoviaje.com), [www.atrapalo.com..](http://www.atrapalo.com..) Sin embargo, pese a esta información todavía a 2 de septiembre de 2008, el proyecto sigue sin estar totalmente definido. Desde el inicio del proyecto, se puede prever que se va a tratar de un proyecto “dinámico”, es decir, que el proyecto va a estar sujeto a exigencias o peticiones por parte del cliente, en función de determinados factores, como pueden ser, los datos que se van a disponer, la viabilidad de implementar una determinada solución, el coste de la misma... Por tanto, a 2 de septiembre contamos con un objetivo global, el cual se define como “Elaboración de un sistema de datos que reúna toda la información sobre el Patrimonio Histórico en Álava, y desarrollo de una aplicación que trabaje contra dicho sistema de datos y la cual sea accesible desde la Web (ejemplo un buscador)”, sin embargo desconocemos las organizaciones que van a colaborar con nuestro proyecto, los datos de los que vamos a disponer, las fuentes de datos que almacenan los mismos, las facilidades de acceso que nos van a proporcionar las diferentes organizaciones a dichas fuentes de datos ... A pesar de esto, Ion sigue trabajando con la información que José Manuel le va proporcionando con el fin de encontrar una orientación adecuada al proyecto. El primer paso es desarrollar el alcance del proyecto, en lo que hemos denominado “*Documento de Objetivos del Proyecto (D.O.P.)*”.

Una vez desarrollado el D.O.P, y siendo éste revisado por el tutor del proyecto, el proyecto queda oficialmente puesto en marcha.

El primer paso una vez terminado el D.O.P., es llevar a cabo una fase de análisis que reúna el análisis de requerimientos expuesto por el cliente, un análisis de los datos y un análisis profundo de las diferentes propuestas de solución. El análisis de requerimientos es desarrollado por el cliente con el fin de orientar al alumno los pasos que debe seguir para conseguir cumplir dichos requerimientos. Gracias a la información proporcionada tanto por el cliente, como por el tutor del proyecto, Ion comienza un estudio bastante profundo de diferentes alternativas que pueden ser desarrolladas como solución a la problemática expuesta en este proyecto. Este estudio queda reflejado en el apartado de “*Análisis de las diferentes propuestas*”. Para la elaboración de este estudio se apoya en un análisis previo de los datos disponibles. Éste, se puede contemplar en el apartado de la memoria denominado “*Análisis de los datos*”.

### **12.3 FASE DE SEGUIMIENTO**

A lo largo de esta fase se describirán aspectos como la metodología de las reuniones entre cliente y alumno, los temas más importantes derivados de las mismas, y las consecuencias que han tenido las conclusiones obtenidas en dichas reuniones en el desarrollo del proyecto.

La relación entre el cliente y el alumno se puede definir como una relación constante, ya que durante los meses que ha durado el proyecto ambas partes se han mantenido en contacto constantemente, ya bien sea mediante reuniones (las cuales se pueden ver en el anexo D.1), mediante el correo electrónico, o simplemente debido a que Ion, el desarrollador del proyecto, se trasladaba al propio Laboratorio de Documentación Geométrica del Patrimonio a realizar alguna parte concreta del propio proyecto.

Esta relación constante entre ambas partes ha facilitado mucho las cosas a la hora de conseguir un producto final adecuado para cliente. Cada paso que se avanzaba en el proyecto era contrariado por parte del cliente con el fin de que éste pudiera decidir si el proyecto estaba siendo bien encaminado, o por el contrario había ciertos aspectos que era necesario mejorar. Esto permitía al alumno conocer en todo momento las necesidades del cliente, y saber con seguridad al terminar cada una de las fases del proyecto, si el trabajo realizado hasta el momento era correcto.

A lo largo del desarrollo del proyecto se han mantenido nueve reuniones entre el cliente y el alumno (Ver reuniones anexo D.1), en estas reuniones se han debatido diferentes puntos de interés en distintos momentos del proyecto. Con el fin de llevar a cabo una explicación, a modo de resumen, de las reuniones mantenidas entre ambas partes, y conocer apoyándonos en las reuniones y en las conclusiones obtenidas de éstas, como se ha ido forjando el producto final, producto que será utilizado por el cliente en un futuro inmediato, a continuación aparece, en orden cronológico, los

cambios más significativo y las ideas más importantes que se han obtenido de las diferentes reuniones.

Como bien se ha comentado en el primer punto de este apartado “*Relaciones con el cliente*” el proyecto nace de una idea del Laboratorio de Documentación Geométrica del Patrimonio a mediados del mes de julio. Tras la idea transmitida por el cliente al alumno (Se puede ver en el apartado propuesta de proyecto), éste último comienza a buscar soluciones posibles para conseguir el objetivo del proyecto. La primera reunión del proyecto se produce el día 2 de septiembre de 2008 (ver dicha reunión en el anexo D.1.1). En esta reunión José Manuel Valle, cliente, proporciona a Ion una serie de posibles soluciones similares a la del proyecto que se va a desarrollar. Tras una serie de conclusiones obtenidas por parte del alumno, quien es ayudado por su tutor, Ion hace saber al cliente que necesario conocer cuáles son las organizaciones que están dispuestas a colaborar con el proyecto, qué datos puede aportar y dónde tienen almacenados esos datos. El cliente, representado por Álvaro Rodríguez en esta ocasión se pone en contacto con las diferentes instituciones y organizaciones cuya actividad principal está relacionada con el patrimonio histórico. Las conclusiones obtenidas por Álvaro de las reuniones con dichas organizaciones se pueden ver en el acta de reunión del 3-10-2008. Además de lo acontecido en esta reunión Álvaro envía a Ion varios correos electrónicos en los que informa de la situación de las diferentes organizaciones. A modo de resumen, las organizaciones tienen pocos datos informatizados, además las que creen tener alguno no saben prácticamente ni dónde ni cómo lo tienen almacenado.

En vista de los resultados obtenidos, el alumno, tras una reunión con José Miguel Blanco, tutor del proyecto (Ver reuniones con el tutor anexo D.2 reunión (7-10-2008)), decide prescindir de la información de las diferentes organizaciones, y comenzar las fases de diseño del proyecto e implementación del mismo basándose en la información que a partir de ese momento pueda conseguir el propio alumno de las diferentes páginas Web de las organizaciones anteriormente mencionadas. Además, como resulta obvio, se podrá contar con los datos del propio Laboratorio de Documentación Geométrica del Patrimonio.

Ion transmite esta decisión al cliente (ver reunión con el cliente 9-10-2007), quien aprueba la decisión. Por tanto se acuerda, que el proyecto constará de dos aplicaciones, una de ellas servirá de soporte para la base de datos que almacenará información sobre la clasificación de los datos que se realizará sobre la información existente en las diferentes organizaciones e instituciones que trabajan con el patrimonio histórico, y una segunda que consistirá en una aplicación de búsqueda orientada en bases de datos heterogéneas. Dichas bases de datos heterogéneas serán en un principio, la base de datos que almacena información existentes en páginas Webs relacionadas con el patrimonio, y la base de datos del Laboratorio de Documentación Geométrica del Patrimonio, pudiendo ser incluidas nuevas bases de datos en un futuro, tanto cercano (durante el desarrollo de la aplicación), como lejano (una vez terminada la aplicación) Ambas partes consideran vital empezar cuando antes con el análisis y la clasificación de los datos existentes en las diferentes páginas Web de las distintas organizaciones relacionadas con el patrimonio histórico.

Durante la fase de análisis de datos y clasificación de los mismos, Ion traslada su lugar de trabajo al propio Laboratorio de Documentación Geométrica del Patrimonio, con el fin de ayudarse de los conocimientos del cliente acerca de la materia que se está abordando. Durante estos días Álvaro Rodríguez se convierte en un gran apoyo para el alumno, ya que ayuda a éste a realizar una clasificación coherente de los datos, lo cual será de gran ayuda a Ion para el estudio de las diferentes soluciones alterativas del proyecto.

Una vez terminada la parte de análisis de los datos y clasificación de los mismos (Ver análisis de datos), Ion comienza con la fase de diseño de la aplicación, la cual una vez terminada es presentada y explicada en una reunión al cliente (ver reuniones con el cliente 12-11-2008). Finalizada la fase de diseño el alumno comienza la fase de implementación, la cual como era de suponer le abarca más de dos meses de tiempo real (incluyendo las navidades). Durante el desarrollo de la implementación cliente y alumno mantienen una nueva reunión en la cual Ion explica al cliente la situación actual del proyecto, y por primera vez se comienza a hablar de finalizar el proyecto en una fecha no muy lejana (ver reuniones con el cliente 8-12-2008).

El 22 de diciembre se produce una nueva reunión en la cual José Manuel solicita a Ion posibles mejoras en la aplicación, mejoras que no estaban recogidas en el análisis de requerimientos (Ver análisis de requerimientos). Estas mejoras consisten en la restricción de actividades en uno de los perfiles, y aportar mayor escalabilidad a una de las aplicaciones. Esta escalabilidad consiste en que la aplicación sea capaz de funcionar correctamente si la clasificación de los elementos patrimoniales se modifica, es decir, la aplicación debe responder correctamente a cambios tanto en las tablas como en los atributos de la base de datos.

El 15 de enero de 2009 (ver reuniones con el cliente 15-1-2009) se produce la siguiente reunión, en la cual cabe destacar que se comienza hablar de la instalación de la aplicación en la Universidad en el País Vasco. El cliente solicita un espacio en el servidor de la universidad para instalar las tres bases de datos del nuevo sistema. Además Ion muestra a José Manuel y a Álvaro los cambios realizados debido a las sugerencias de mejora solicitadas por éstos. Ambos dan el visto bueno a dichos cambios.

#### **12.4 FASE DE ENTREGA, PRUEBAS Y ACEPTACIÓN DE RESULTADOS.**

El jueves 12 de febrero de 2009, el producto es instalado en el servidor de la Universidad del País Vasco. Tras un pequeño intento fallido, ya que el servidor no albergaba PHP5, el cual es necesario para el funcionamiento del proyecto, el proyecto queda formalmente instalado y disponible para su uso en cuanto el cliente lo crea oportuno.

Una vez instalado el producto, y como cierre de las relaciones mantenidas entre cliente y alumno, se considera necesario el desarrollo de una serie de pruebas que

verifiquen el correcto funcionamiento de ambas aplicaciones, así como una valoración del proyecto desarrollado por parte del cliente.

#### **12.4.1 PRUEBAS**

Para completar la finalización del producto, el Cliente desarrollará una serie de pruebas que servirán para comprobar el funcionamiento de la aplicación. A continuación aparecen una serie de pruebas, todas ellas desarrolladas por el Cliente.

#### **CLIENTE: Laboratorio de Documentación Geométrica del Patrimonio**

Pruebas realizadas en la aplicación cargada en el servidor de la UPV-EHU, dentro de la página web del Laboratorio.

**Pruebas de la aplicación que da soporte a la base de datos que almacena información sobre las páginas Web.**

##### **Prueba 1**

- **Nombre de la prueba**

Acceso a la aplicación. Contraseña no válida.

- **Descripción de la prueba**

Intento de acceso a la aplicación con una contraseña que no corresponde con la del usuario.

- **Resultado esperado**

El sistema no permite el acceso mostrando un aviso de error.

- **Resultado real**

Cliente: CORRECTO

##### **Prueba 2**

- **Nombre de la prueba**

Insertar un nuevo Sitio Web.

- **Descripción de la prueba**

El usuario inserta un nuevo sitio Web en la base de datos.

- **Variante 1.** El usuario intenta introducir un sitio Web que ya existe en la base de datos.
- **Resultado esperado**

El sitio Web queda correctamente registrado.

  - **Variante 1.** El sistema no se lo permite, le muestra un mensaje de error.

- **Resultado real**

Cliente: Tras introducir un nuevo sitio web, aparece correctamente registrado. Si el sitio ya existe se presenta un mensaje de error advirtiéndolo y se vuelve a la pantalla de introducción del sitio web para seguir añadiendo datos.

### Prueba 3

- **Nombre de la prueba**

Modificar un Sitio Web.

- **Descripción de la prueba**

El usuario modifica un nuevo sitio Web ya existente en la aplicación.

- **Variante 1.** El usuario intenta modificar un sitio Web asignándole una dirección Web que pertenece a otro sitio Web ya registrado.
- **Variante 2.** El usuario intenta modificar un sitio Web asignándole una dirección Web nueva.

- **Resultado esperado**

El sitio Web queda correctamente modificado.

- **Variante 1.** El sistema no se lo permite, le muestra un mensaje de error.
- **Variante 2.** La aplicación se modifica correctamente. Además todas las sub-páginas Web que pertenecen a dicho sitio Web se deben mover perfectamente a la nueva dirección Web del sitio Web.

- **Resultado real**

Cliente: Las modificaciones en los campos de nombre y descripción se realizan correctamente. Si se intenta modificar la dirección utilizando una ya existente da un mensaje de error y no lo permite. El cambio se realiza correctamente, al



modificar una dirección web nueva, aunque en el menú desplegable para seleccionar la página web que se quiere modificar no se actualiza directamente (sigue apareciendo la dirección antigua). Las sub-páginas web no se actualizan y continúan dependiendo de la dirección antigua.

#### **Prueba 4**

- **Nombre de la prueba**

Eliminar un Sitio Web.

- **Descripción de la prueba**

El usuario elimina un sitio Web ya existente en la aplicación.

- **Resultado esperado**

El sitio Web queda correctamente eliminado. Además todas las sub-páginas Web que pertenecen a dicho sitio Web así como los elementos patrimoniales que se encuentran situados en las sub-páginas Webs que pertenecen al sitio Web que se va eliminar quedarán a su vez eliminadas. Eliminación en cascada.

- **Resultado real**

Cliente: CORRECTO

#### **Prueba 5**

- **Nombre de la prueba**

Introducir una nueva subpágina Web.

- **Descripción de la prueba**

El usuario introduce una subpágina Web en un sitio Web ya existente.

- **Variante 1.** El usuario intenta introducir una subpágina en un sitio Web, asignándole una dirección a dicha subpágina, que ya pertenece a otra subpágina Web de ese sitio Web.

- **Resultado esperado**

La nueva subpágina Web queda correctamente registrada.

- **Variante 1.** El sistema no se lo permite, le muestra un mensaje de error.

- **Resultado real**

Cliente: CORRECTO tanto la acción principal como la variante.

### **Prueba 6**

- **Nombre de la prueba**

Modificar una subpágina Web.

- **Descripción de la prueba**

El usuario modifica una subpágina Web de un sitio Web ya existente.

- **Variante 1.** El usuario intenta modificar una subpágina Web en un sitio Web, asignándole una dirección Web a dicha subpágina Web que ya pertenece a otra subpágina Web de ese sitio Web.

- **Resultado esperado**

La subpágina Web queda correctamente modificada.

- **Variante 1.** El sistema no se lo permite, le muestra un mensaje de error.

- **Resultado real**

Cliente: CORRECTO tanto la acción principal como la variante.

### **Prueba 7**

- **Nombre de la prueba**

Eliminar una subpágina Web.

- **Descripción de la prueba**

El usuario elimina una subpágina Web de un sitio Web ya existente.

- **Resultado esperado**

La subpágina Web queda correctamente eliminada. Además todos los elementos patrimoniales pertenecientes a dicha subpágina Web serán también eliminados. Eliminación en cascada.

- **Resultado real**

Cliente: CORRECTO

## **Prueba 8**

- **Nombre de la prueba**

Introducir un nuevo elemento patrimonial.

- **Descripción de la prueba**

El usuario introduce un nuevo elemento patrimonial en una subpágina Web ya existente.

- **Resultado esperado**

El nuevo elemento patrimonial queda correctamente registrado.

- **Resultado real**

Cliente: CORRECTO

## **Prueba 9**

- **Nombre de la prueba**

Modificar un elemento patrimonial.

- **Descripción de la prueba**

El usuario modifica un elemento patrimonial de una subpágina Web ya existente.

- **Resultado esperado**

El elemento patrimonial queda correctamente modificado.

- **Resultado real**

Cliente: CORRECTO

## **Prueba 10**

- **Nombre de la prueba**

Eliminar un elemento patrimonial.

- **Descripción de la prueba**

El usuario elimina un elemento patrimonial de una subpágina Web ya existente.

- **Resultado esperado**

El elemento patrimonial queda correctamente eliminado.

- **Resultado real**

Cliente: CORRECTO

### **Prueba 11**

- **Nombre de la prueba**

Introducir un nuevo administrador.

- **Descripción de la prueba**

El usuario introduce un nuevo administrador.

- **Variante 1.** El usuario intenta introducir un administrador/usuario nuevo asignándole como “Nick”, un “Nick” de un administrador/usuario ya registrado.

- **Resultado esperado**

El nuevo administrador queda correctamente registrado.

- **Variante 1.** El sistema no se lo permite, le muestra un mensaje de error.

- **Resultado real**

Cliente: CORRECTO

### **Prueba 12**

- **Nombre de la prueba**

Modificar un administrador.

- **Descripción de la prueba**

El usuario modifica los datos de un administrador.

- **Variante 1.** El usuario intenta modificar un administrador asignándole como “Nick”, un “Nick” de un administrador ya registrado.

- **Resultado esperado**

Los datos del administrador quedan correctamente modificados.

- **Variante 1.** El sistema no se lo permite, le muestra un mensaje de error.

- **Resultado real**

Cliente: Las modificaciones en los campos se realizan correctamente, no se permite utilizar un Nick ya en uso. CORRECTO

### Prueba 13

- **Nombre de la prueba**

Eliminar un administrador.

- **Descripción de la prueba**

El usuario elimina un administrador.

- **Resultado esperado**

El administrador queda correctamente eliminado.

- **Resultado real**

Cliente: Correcto al eliminar administradores.

### Prueba 14

- **Nombre de la prueba**

Ver movimientos.

- **Descripción de la prueba**

El usuario comprueba los movimientos que ha realizado, con el fin de comprobar que se almacenan correctamente.

- **Resultado esperado**

Los movimientos realizados se almacenan correctamente.

- **Resultado real**

Cliente: CORRECTO

## **Prueba 15**

- **Nombre de la prueba**

Ver administradores.

- **Descripción de la prueba**

El usuario comprueba los administradores que existen en la aplicación.

- **Resultado esperado**

Los administradores que se muestran son exactamente los que en realidad existen.

- **Resultado real**

Cliente: CORRECTO.

## **Pruebas de la aplicación de búsqueda orientada.**

### **Prueba 1**

- **Nombre de la prueba**

Realizar consulta general.

- **Descripción de la prueba**

El usuario realiza una búsqueda general seleccionando los atributos que desee.

- **Resultado esperado**

La consulta se realiza correctamente.

- **Resultado real**

Cliente: CORRECTO

### **Prueba 2**

- **Nombre de la prueba**

Realizar consulta general sobre “edificios”.

- **Descripción de la prueba**

El usuario realiza una búsqueda general sobre “edificios” seleccionando los atributos que desee.

- **Resultado esperado**

La consulta se realiza correctamente.

- **Resultado real**

Cliente: Error: en algunos casos aparece el mensaje “sintaxis XML no válida”, aunque algunas búsquedas las realiza correctamente, creo que el problema es introducir nombres con “ñ”. CORREGIDO Y CORRECTO.

### Prueba 3

- **Nombre de la prueba**

Realizar consulta dentro de “edificios” sobre “arquitectura religiosa”.

- **Descripción de la prueba**

El usuario realiza una búsqueda dentro de “edificios” sobre “arquitectura religiosa” seleccionando los atributos que desee.

- **Resultado esperado**

La consulta se realiza correctamente.

- **Resultado real**

Cliente: En apariencia es correcto.

### Prueba 4

- **Nombre de la prueba**

Realizar consulta dentro de “edificios” sobre “arquitectura civil”.

- **Descripción de la prueba**

El usuario realiza una búsqueda dentro de “edificios” sobre “arquitectura civil” seleccionando los atributos que desee.

- **Resultado esperado**

La consulta se realiza correctamente.

- **Resultado real**

Cliente: En apariencia es correcto.

#### **Prueba 5**

- **Nombre de la prueba**

Realizar consulta dentro de “edificios” sobre “elementos menores”.

- **Descripción de la prueba**

El usuario realiza una búsqueda dentro de “edificios” sobre “elementos menores” seleccionando los atributos que desee.

- **Resultado esperado**

La consulta se realiza correctamente.

- **Resultado real**

Cliente: En apariencia es correcto.

#### **Prueba 6**

- **Nombre de la prueba**

Realizar consulta sobre “Excavaciones”.

- **Descripción de la prueba**

El usuario realiza una búsqueda sobre “Excavaciones” seleccionando los atributos que desee.

- **Resultado esperado**

La consulta se realiza correctamente.

- **Resultado real**

Cliente: En apariencia es correcto.

#### **Prueba 7**

- **Nombre de la prueba**

Realizar consulta sobre “Objetos”.



- **Descripción de la prueba**

El usuario realiza una búsqueda sobre “Objetos” seleccionando los atributos que desee.

- **Resultado esperado**

La consulta se realiza correctamente.

- **Resultado real**

Cliente: En apariencia es correcto.

#### **12.4.2 CONCLUSIONES POR PARTE DEL CLIENTE**

##### **Conclusiones del Grupo de Investigación para el Proyecto PFA07/03 “Desarrollo de Estándares y Estrategias para el Establecimiento de un archivo vivo del Patrimonio Digital Alavés”**

Al establecer los objetivos del proyecto, se detectó la necesidad de implementar una solución informática que hiciera operativa la búsqueda de la información patrimonial en formato digital a partir de un entorno web. El profesor Pablo Navarro Ullés, miembro del grupo investigador, se puso en contacto con la Facultad de Informática de San Sebastián para recabar colaboración en esta tarea. Como resultado de esta consulta, el profesor José Miguel Blanco propuso al alumno Ion Marín Martín, con DNI: 72.725.250-D, para colaborar en este desarrollo.

El trabajo se organizó mediante reuniones periódicas en las que, al comienzo, se fue perfilando la solución mediante un intercambio de necesidades (por parte del grupo de investigación) y posibilidades técnicas (por parte del alumno), el cometido del alumno en esta primera fase fue investigar dichas posibilidades con el fin de llegar a una propuesta metodológica válida que pudiera responder al reto planteado.

En la siguiente fase, las reuniones se alternaban con los periodos de trabajo individual del alumno que, en este caso, consistía en el desarrollo de la aplicación. Las reuniones suponían la validación del estado de los trabajos por parte del equipo de investigación.

La última fase, ha consistido en cargar la aplicación dentro de la web de la Universidad del País Vasco, realizando las pruebas necesarias para dejarla en condiciones de funcionamiento.

Queremos destacar, en primer lugar, la buena disposición para el trabajo del alumno, tanto por su constancia y saber hacer como por la facilidad de integrarse en el entorno con el resto de compañeros.

Respecto al proyecto en sí, reconocemos que ha sido un reto para todos ya que desde el grupo de investigación partíamos de una idea clara de lo que se pretendía pero desconocíamos la posibilidad real de implementación. Vista en retrospectiva, la tarea de traducir estas ideas en una aplicación concreta, nos parece de una complejidad muy superior a la inicialmente prevista y hemos de afirmar que la actitud del alumno ha sido excelente en todo momento: entendiendo el problema que se pretendía abordar, aportando las soluciones técnicas que, en la medida de sus conocimientos, consideraba apropiadas y recabando información para poder evaluar nuevas opciones.

Analizando el resultado presentado, comprobamos que no sólo cumple con las expectativas iniciales sino que, además, nos ha permitido ver nuevas e interesantes posibilidades que pretendemos explorar próximamente.

Por todo ello, afirmamos que tanto la solución informática desarrollada como la relación personal con Ion Marín han sido plenamente satisfactorias.



## 13. CONCLUSIONES

### 13.1 LINEAS FUTURAS

El proyecto realizado, en un principio, es considerado como un punto de partida para conseguir en un futuro cercano, un sistema robusto que almacene información sobre el patrimonio histórico de la provincia de Álava y sus alrededores. Por tanto ante un proyecto de estas características se me ocurren numerosas líneas de futuro, sin embargo voy a comentar únicamente algunas de ellas, aquellas que considero más importantes.

**Aumento del número de bases de datos.** Siendo positivos, consideramos que la aplicación de búsqueda desarrollada puede atraer a nuevas organizaciones relacionadas con el patrimonio, sugiriendo que los datos de su fuente de datos sean accesibles también desde nuestra aplicación de búsqueda. Para ello se han elaborado unos manuales de mantenimiento de la aplicación en los cuales se explica de forma detallada los pasos a seguir.

**Cambio de las interfaces.** Tal vez en un futuro cercano, cuando el buscador pueda convertirse en una referencia a nivel local del patrimonio histórico, quizás, por temas de diseño se crea conveniente cambiar la apariencia de las interfaces de la aplicación. Al igual que en el caso anterior, existen unos manuales de mantenimiento que indican paso a paso las indicaciones a seguir para conseguir desarrollar este cambio.

**Mejora de la búsqueda.** Si bien es cierto que, esta aplicación permite a través de la introducción de una palabra realizar la búsqueda con sinónimos de la misma con el fin de que la consulta sea más exacta, además de realizar dicha búsqueda eliminando artículos, preposiciones, etc. y permitiendo al usuario indicar únicamente palabras que sean necesarias en la búsqueda, hoy en día están apareciendo filtros de búsqueda muy avanzados, y que no han sido tomados en cuenta en el desarrollo del proyecto, básicamente por la duración finita del mismo.

**Crecimiento del sistema.** En el caso de que el sistema se convirtiera en un sistema muy importante a nivel local y que numerosas bases de datos se incorporaran al mismo, es posible que éste requiera una modificación global, ya que hay que tener en cuenta que la aplicación está desarrollada como un Proyecto de Fin de Carrera desarrollado por un único alumno, a lo que hay que añadir que dicho proyecto consta además de una parte de investigación y en el cual se ha desarrollado otra aplicación aparte de la aplicación de búsqueda orientada. Todo esto teniendo en cuenta que el proyecto tiene una duración finita, la cual según el desarrollador del proyecto considera incluso excesiva.

## 13.2 CONCLUSIONES SOBRE EL PROYECTO

Una vez terminado el proyecto que ha sido desarrollado entre los meses de julio del año 2007 y febrero del 2008 es el momento de valorar el trabajo desarrollado.

### Desarrollo de las aplicaciones

Tal y como se planteó en el Documento de Objetivos del Proyecto (D.O.P.) este proyecto ha sido realizado con vistas a cumplir dos objetivos.

- Conseguir “reunir” la información relacionada con el patrimonio digital existente en las diferentes instituciones u organizaciones de la provincia como pueden ser Diputación, Ayuntamiento o el propio Laboratorio de Documentación Geométrica del Patrimonio de la UPV/EHU.
- Elaborar una aplicación de búsqueda que permitiera al usuario consultar dicha información.

Con respecto al primero de los objetivos, pese a la carencia de recursos (bases de datos, documentos escritos...), se ha conseguido reunir toda la información existente en las diferentes páginas Webs asociadas a organizaciones e instituciones de la provincia. Esta meta-información (no se almacena la información en sí) es almacenada en una base de datos que ha sido creada durante el desarrollo del Proyecto de Fin de Carrera. Para mantener actualizada dicha base de datos se ha desarrollado, también, una aplicación que sirve de soporte para la misma, y que permite a un determinado usuario llevar a cabo diferentes funciones que permitan mantener la base de datos actualizada.

Además de la base de datos comentada en el párrafo anterior el Laboratorio de Documentación Geométrica del Patrimonio nos ha permitido utilizar su base de datos propia. Dicha base de datos contiene información acerca de los proyectos desarrollados por el propio laboratorio.

En lo que al segundo objetivo se refiere, se ha desarrollado una aplicación de búsqueda orientada a bases de datos heterogéneas. Esta aplicación de búsqueda tiene asociadas en un principio dos bases de datos, que son las comentadas anteriormente, aquella que almacena meta-información sobre páginas Webs y la del Laboratorio de Documentación Geométrica del Patrimonio. Sin embargo este sistema de acceso integrado es un sistema escalable que es capaz de adaptarse perfectamente a los cambios que puedan, en un futuro, producirse. Se puede asegurar que el sistema cumple los requisitos solicitados por el cliente (apartado 5.1), debido a los siguientes motivos.

#### Escalabilidad del sistema

El sistema permite incorporar nuevas fuentes de datos al sistema con la simple construcción de un Árbol de Correspondencias que sirva para transformar el Lenguaje

Consulta en el lenguaje propio de la nueva fuente de datos. Debido a esto, el sistema puede tener asociadas un número de fuentes de datos indeterminadas. El apartado negativo de la escalabilidad, es que a mayor número de bases de datos, mayor será el tiempo que tarde en ejecutarse la consulta, ya que la búsqueda se realiza en serie.

#### Facilidad para adaptarse a los cambios

La arquitectura diseñada como solución al proyecto cuenta con un elemento esencial, el Árbol de Conceptos. El Árbol de Conceptos almacena información sobre los conceptos y atributos del sistema además de datos referentes a las bases de datos y a las interfaces que aparecen en el mismo. El sistema cuenta con un módulo que recorre dicho Árbol de Conceptos y va obteniendo de él la información que necesita en cada momento. Esto nos permite, siempre y cuando se respeten las normas establecidas en el DTD del Árbol de Conceptos (ver anexos B.1.1), realizar los cambios que se estimen oportunos en cuanto a:

- Las interfaces del sistema. Es posible cambiar las interfaces existentes por otras nuevas. Bastará con indicárselo al Árbol de Concepto en el campo referente a las interfaces.
- Modificar la estructura del árbol. Nos permite modificar las relaciones existentes entre los conceptos.
- Añadir, borrar o modificar conceptos. Por ejemplo, si introducimos una nueva base de datos que aporta información sobre un concepto que no aparece en la aplicación de búsqueda, éste puede ser incluido sin problema alguno.
- Añadir, borrar o modificar atributos.

#### Interfaces sencillas

Cabe resaltar además que se ha conseguido elaborar una serie de interfaces sencillas e intuitivas que van a permitir a cualquier usuario realizar una consulta de una manera fácil y cómoda. Para ello, se ha utilizado la técnica del Lenguaje Natural Acotado.

#### **Gestión del proyecto**

Aparte del desarrollo de las dos aplicaciones elaboradas a lo largo de este Proyecto Fin de Carrera, otro de los apartados que es conveniente analizar es el relativo a la gestión del mismo. Personalmente, considero que la gestión de este proyecto puede considerarse como bastante aceptable, no sólo a nivel de estimación de tiempo en cada fase, si no en cuanto al día a día del mismo. Pienso que cada una de las decisiones, y cada uno de los pasos que se han tomado a lo largo de estos casi ocho meses han quedado reflejados en la memoria. Si revisamos cada una de las fases debemos destacar la fase de análisis, donde se desarrollo una parte de investigación exhaustiva que quedo reflejada en los apartados 5.4 y 5.5 de la memoria. Desde mi punto de vista, además,

considero que quedan reflejados perfectamente las dedicaciones, en términos de tiempo, invertidas en el desarrollo de cada fase, y su comparación con las estimaciones iniciales.

### **Relaciones con el cliente**

Aunque existe ya un apartado dedicado a las relaciones mantenidas con el cliente, considero necesario elaborar una conclusión personal que indique cómo ha sido la relación el mismo. Basándonos en las conclusiones redactadas por el propio cliente, se puede afirmar que la relación entre cliente y alumno ha sido buena. Desde el primer momento ambas partes hemos mantenido una relación cordial, basada en la confianza y en la continua comunicación, lo que considero ha sido un factor importante para la culminación del proyecto de manera satisfactoria. Tanto Álvaro, como José Manuel han mostrado una buena disposición para solucionar cualquier problema que me ha ido surgiendo a lo largo del proyecto.

### **13.2 CONCLUSIONES PERSONALES**

El Proyecto de Fin de Carrera es considerado como el mayor reto al que un estudiante se enfrenta en su carrera formativa, en mi caso no ha sido diferente. Si bien es cierto que este no es mi primer Proyecto de Fin de Carrera, ya que anteriormente superé otro con el cual concluía la Ingeniería Técnica en Informática de Gestión, este último ha sido diferente, tanto en volumen de trabajo como en el contexto que se ha desarrollado.

Ambos proyectos tenían características en común, abarcaban un campo desconocido de la informática para mí en ese momento, y el proyecto era desarrollado para un determinado cliente, quién solicitaba una serie de requisitos, los que yo me limitaba a cumplimentar de la mejor forma posible. Sin embargo, el proyecto desarrollado durante estos meses atrás ha sido más completo en todos los aspectos. A diferencia del proyecto de la Ingeniería Técnica, he llevado a cabo un proceso de investigación, concretamente acerca de las diferentes posibles soluciones del proyecto, la gestión del mismo ha sido bastante completa, documentando cada fase, y explicando cada decisión tomada. El trato con el cliente ha sido diferente, ya que la solución generada durante este proyecto está funcionando en la Web actualmente, lo cual refleja la importancia del mismo para el cliente. Además considero que el proyecto desarrollado pertenece a un campo del cual no existe numerosa información, como son las bases de datos heterogéneas, por lo que creo que esto debe ser considerado como un plus de dificultad en el desarrollo del mismo.

En cuanto a mi experiencia personal, una vez terminado el proyecto, puedo confesar que he experimentado diferentes estados de ánimo, desde la más absoluta satisfacción hasta la más profunda desesperación, todo ello provocado por los diferentes altibajos del proyecto. Echando la vista atrás puedo asegurar que he conseguido superar situaciones, en lo que a errores en las aplicaciones se refiere, que jamás me creía capaz de superar.

No sólo es que haya aprendido más sobre las herramientas utilizadas, o sobre la arquitectura desarrollada, sino que he aprendido a trabajar mejor, a buscar soluciones a problemas que creía que no la tenían. He aprendido a planificarme, a lo importante que es documentar cada fase, cada paso, cada decisión tomada. Creo que puedo afirmar que gracias a este proyecto he madurado en lo que al trabajo se refiere. Personalmente, y con una salida laboral inminente, considero que el Proyecto de Fin de Carrera realizado me va a ayudar un mi futuro puesto de trabajo.

Concluir expresando mi gran felicidad por el trabajo realizado. Creo que lo que se ha conseguido ha sido gracias al esfuerzo realizado, al empeño y las ganas con las que he afrontado desde el primer día este proyecto. Agradecer también el comportamiento del cliente del proyecto, José Manuel Valle y Álvaro Rodríguez quienes en todo momento se han mostrado amables conmigo y me han ayudado, en la manera de lo posible, en todos y cada uno de los aspectos del proyecto. Me alegro enormemente que el proyecto sea de su agrado y hayan quedado realmente satisfechos con él. Por último agradecer a mi tutor del Proyecto, José Miguel Blanco, su apoyo y orientación durante los meses del proyecto.





## 14. BIBLIOGRAFÍA

- Página Web “wikipedia”, la cual alberga contenidos interesantes acerca de temas relacionados con las herramientas utilizadas en el proyecto. En este caso sobre Java Server Pages. [http://es.wikipedia.org/wiki/Java\\_Server\\_Pages](http://es.wikipedia.org/wiki/Java_Server_Pages)
- Página Web “wikipedia”. La cual alberga información interesante acerca de diversos aspectos de la informática. En este caso respecto a EditPlus. <http://en.wikipedia.org/wiki/EditPlus>.
- Página Web “wikipedia”, la cual alberga contenidos interesantes acerca de temas relacionados con las herramientas utilizadas en el proyecto. En este caso sobre PHP. <http://es.wikipedia.org/wiki/Php>
- Página Web “wikipedia”, la cual alberga contenidos interesantes acerca de temas relacionados con las herramientas utilizadas en el proyecto. En este caso sobre .Net. <http://es.wikipedia.org/wiki/.NET>
- Página Web “wikipedia”, la cual alberga contenidos interesantes acerca de temas relacionados con las herramientas utilizadas en el proyecto. En este caso sobre la plataforma XAMPP. <http://es.wikipedia.org/wiki/XAMPP>
- Página Web “monografías”, la cual alberga información interesante acerca de las bases de datos. <http://www.monografias.com/trabajos35/comparativa-bases-datos/comparativa-bases-datos.shtml>
- Página Web “wikipedia”, la cual alberga contenidos interesantes acerca de temas de distintas índoles. En este caso sobre la Arquitectura de tres niveles. [http://es.wikipedia.org/wiki/Arquitectura\\_de\\_tres\\_niveles](http://es.wikipedia.org/wiki/Arquitectura_de_tres_niveles)
- Apuntes sobre bases de datos distribuidas obtenidos de la Web, cuya dirección es: <http://siul02.si.ehu.es/~jimena/ABD/fuentes/BDDistribuida.pdf>
- Contenidos obtenidos a través de la Web accediendo a ellos a través de la dirección <http://sacbeob.8m.com/tutoriales/bddistribuidas/index.htm>
- Contenidos obtenidos a través de la Web accediendo a ellos a través de la dirección <http://sacbeob.8m.com/tutoriales/bddistribuidas/cap1.html>
- Apuntes obtenidos de la Web sobre bases de datos que pertenecen a la Universidad de Castilla la Mancha- Arquitectura de Sistemas de Bases de Datos. <http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r23897.PDF>

- Tesis Doctoral de la cual se han obtenido numerosos conceptos vitales para el desarrollo de la aplicación. Tesis Doctoral. Arquitectura para Federación de Bases de Datos Documentales basada en Ontologías. [http://rosalia.dc.fi.udc.es/lbd/..%5CPlainConsulta%5CArchivosAdministracion%5Ctrabajos\\_investigadores%5Ctrabajos%5C576965170V.pdf](http://rosalia.dc.fi.udc.es/lbd/..%5CPlainConsulta%5CArchivosAdministracion%5Ctrabajos_investigadores%5Ctrabajos%5C576965170V.pdf)

## ÍNDICE DE CONTENIDOS –ANEXOS-

<b>A. ESTUDIO Y CLASIFICACIÓN DE LOS DATOS.....</b>	<b>1</b>
A.1 DESCRIPCIÓN DE LAS ENTIDADES CONTACTADAS.....	1
A.2 CLASIFICACIÓN DE LOS DATOS.....	7
<b>B. CONCEPTOS IMPORTANTES DE LA APLICACIÓN DE BÚSQUEDA ORIENTADA.....</b>	<b>9</b>
B.1 ÁRBOL DE CONCEPTOS.....	9
B.1.1 DTD ÁRBOL DE CONCEPTOS .....	9
B.1.2 ÁRBOL DE CONCEPTOS .....	9
B.2 ÁRBOLES DE CORRESPONDENCIAS .....	16
B.2.1 DTD DE LOS ÁRBOLES DE CORRESPONDENCIAS.....	16
B.2.2 ÁRBOLES DE CORRESPONDENCIAS.....	16
B.2.2.1 ÁRBOLES DE CORRESPONDENCIAS PÁGINAS WEB. ....	17
B.2.2.2 ÁRBOLES DE CORRESPONDENCIAS LDGP. ....	25
B.3 LENGUAJE CONSULTA.....	30
<b>C. MANUALES DE LA APLICACIÓN .....</b>	<b>31</b>
C.1 MANUALES DE MANTENIMIENTO .....	31
C.1.1 MANUAL DE MANTENIMIENTO-APLICACIÓN DE SOPORTE A LA BASE DE DATOS QUE ALMACENA INFORMACIÓN SOBRE PÁGINAS WEB RELACIONADAS CON EL PATRIMONIO-.....	31
C.1.1.1 BASE DE DATOS CON LAS QUE TRABAJA LA APLICACIÓN.....	31
C.1.1.1.2 BASE DE DATOS “SITIOSWEB”.....	31
C.1.1.1.3 BASE DE DATOS “ADMINISTRADORES”.....	32
C.1.1.2 MODIFICACIONES POSIBLES Y ESCALABILIDAD DE AMBAS BASES DE DATOS.....	32
C.1.1.2.1 BASE DE DATOS “SITIOS WEBS”.....	32

C.1.1.2.1.1 CAMBIAR DATOS REFERENTES AL SITIO WEB.....	33
C.1.1.2.1.2 CAMBIAR DATOS REFERENTES A LA SUBPÁGINA WEB.....	33
C.1.1.2.1.3 CAMBIAR DATOS REFERENTES AL PATRIMONIO.....	33
C.1.1.2.2 BASE DE DATOS “ADMINISTRADORES” .....	42
C.1.1.2.2.1 CAMBIAR DATOS REFERENTES A ADMINISTRADOR .....	43
C.1.1.2.2.2 CAMBIAR DATOS REFERENTES A MOVIMIENTOS.....	43
C.1.1.3 INFORMACIÓN REFERENTE A LAS INTERFACES DE LA APLICACIÓN .....	43
C.1.1.3.1 NUEVO PATRIMONIO .....	43
C.1.1.3.2 MODIFICAR PATRIMONIO .....	45
C.1.1.3.3 MODIFICAR PATRIMONIO Y ELIMINAR PATRIMONIO .....	45
C.1.1.4 INFORMACIÓN REFERENTE A LOS PERFILES DE LA APLICACIÓN .....	48
C.1.2 MODIFICACIONES DE LA APLICACIÓN DE BÚSQUEDA ORIENTADA .....	50
C.1.2.1 ÁRBOL DE CONCEPTOS .....	50
C.1.2.2 MODIFICACIÓN DEL ÁRBOL DE CONCEPTOS.....	52
C.1.2.2.1 MODIFICACIÓN DE LA ESTRUCTURA DEL ÁRBOL DE CONCEPTOS (AÑADIR/ELIMINAR CONCEPTOS, ATRIBUTOS, RELACIONES DE GENERALIZACIÓN/ESPECIALIZACIÓN ENTRE CONCEPTOS).....	52
C.1.2.2.2 MODIFICACIÓN DEL NOMBRE DE UN CONCEPTO .....	52
C.1.2.2.3 MODIFICACIÓN DEL NOMBRE DE UN CONCEPTO .....	53
C.1.2.2.4 MODIFICACIÓN DE LOS POSIBLES VALORES DE UN ATRIBUTO .....	54
C.1.2.2.5 AÑADIR UN POSIBLE VALOR A UN ATRIBUTO .....	56
C.1.2.2.6 ELIMINAR UN POSIBLE VALOR DE UN ATRIBUTO .....	57
C.1.2.2.7 MODIFICACIÓN EN LAS BASES DE DATOS EN LAS QUE APARECE UN ATRIBUTO.....	58

C.1.2.2.8 MODIFICACIÓN DE LAS INTERFACES DE USUARIO .....	59
C.1.2.3 MODIFICACIÓN DEL PROCESO DE CONSULTA .....	60
C.1.2.3.1 NUEVO SINÓNIMO DE UNA PALABRA .....	60
C.1.2.3.2 MODIFICACIÓN DE UN SINÓNIMO DE UNA PALABRA .....	61
C.1.2.3.3 ELIMINACIÓN DE UN SINÓNIMO DE UNA PALABRA.....	62
C.1.2.3.4 NUEVA PALABRA EN EL DOCUMENTO XML .....	62
C.1.2.3.5 MODIFICACIÓN DE UNA PALABRA EN EL DOCUMENTO XML.....	63
C.1.2.3.6 ELIMINACIÓN DE UNA PALABRA DEL DOCUMENTO XML .....	63
C.1.2.4 MODIFICACIÓN DE LA PRESENTACIÓN DE LOS RESULTADOS.....	64
C.1.2.5 PROCESO DE INSERCIÓN DE UNA NUEVA BASE DE DATOS EN EL SISTEMA .....	65
C.1.2.5.1 MODIFICACIÓN DEL FICHERO “BDsComponentes.xml” .....	66
C.1.2.5.2 CREACIÓN DIRECTORIO COMO “SISTEMA ENVOLTORIO”.....	67
C.1.2.5.3 MODIFICACIÓN DEL FICHERO “algoritmoTraductorConsulta2.php3” .....	67
C.1.2.5.4 MODIFICACIÓN DEL FICHERO “gestorBD2.php3” .....	69
C.1.2.5.5 MODIFICACIÓN DEL FICHERO “emisorRespuestas2.php3”.....	70
C.1.2.5.6 ÁRBOL DE CORRESPONDENCIAS.....	70
C.1.2.5.7 MODIFICACIÓN DEL FICHERO “capaPresentacion.php3” .....	70
C.2 MANUAL DE USUARIO DE LA APLICACIÓN QUE GESTIONA LA BASE DE DATOS “SITIOS WEB” Y “ADMINISTRADORES” .....	72
C.2.1 PERFIL ADMINISTRADOR .....	73
C.2.1.1 FUNCIONALIDADES SITIO WEB .....	73
C.2.1.2 FUNCIONALIDADES SUB-PÁGINAS WEB.....	79
C.2.1.3 FUNCIONALIDADES PATRIMONIO .....	85

C.2.1.4 FUNCIONALIDADES ADMINISTRADOR .....	92
C.2.1.5 CERRAR SESIÓN.....	102
C.2.2 PERFIL USUARIO .....	103
C.2.2.1 FUNCIONALIDADES SITIO WEB .....	103
C.2.2.2 FUNCIONALIDADES SUB-PÁGINAS WEB.....	106
C.2.2.3 FUNCIONALIDADES PATRIMONIO .....	112
C.2.2.4 FUNCIONALIDADES ADMINISTRADOR .....	119
C.2.2.5 CERRAR SESIÓN.....	122
<b>D. ACTAS DE REUNIÓN .....</b>	<b>123</b>
D.1. ACTAS DE REUNIÓN CLIENTE.....	123
D.1.1 Acta del martes 2 de septiembre de 2008.....	123
D.1.2 Acta del lunes 22 de septiembre de 2008 .....	125
D.1.3 Acta del viernes 3 de octubre de 2008 .....	127
D.1.4 Acta del jueves 9 de octubre de 2008 .....	130
D.1.5 Acta del miércoles 12 de noviembre de 2008 .....	132
D.1.6 Acta del martes 8 de diciembre de 2008 .....	134
D.1.7 Acta del lunes 22 de diciembre de 2008 .....	136
D.1.8 Acta del miércoles 15 de diciembre de 2009.....	138
D.2. ACTAS DE REUNIÓN TUTOR.....	140
D.2.1 Acta del viernes 12 de septiembre de 2008.....	140
D.2.2 Acta del martes 7 de octubre de 2008 .....	142
D.2.3 Acta del miércoles 14 de enero de 2009 .....	145

## ÍNDICE DE FIGURAS -ANEXOS-

Figura A.1. DTD Árbol de Conceptos .....	9
Figura A.2. Árbol de Conceptos .....	9
Figura A.3. DTD de los Árboles de Correspondencias.....	16
Figura A.4. Árbol de correspondencias para base de datos Webs .....	17
Figura A.5. Árbol de correspondencias para base de datos LDGP.....	25
Figura A.6. DTD del Lenguaje Consulta. ....	30
Figura A.7. Esquema de un sitio Web.....	32
Figura A.8. Parte inicial de datosbd.xml. Muestra la información relacionada con el concepto excavaciones.....	34
Figura A.9. Parte de datosbd.xml correspondiente a la tabla arquitecturacivil. Muestra la información relacionada con dicho concepto. ....	35
Figura A.10. Modificación realizada. Se cambia el nombre de la tabla.....	36
Figura A.11. Inserción de un nuevo concepto (tabla) realizada.....	37
Figura A.12. Eliminación de concepto (tabla).....	38
Figura A.13. Cambio del atributo “Tipo_Excavacion” por “tipo” .....	40
Figura A.14. Inserción del atributo “País” dentro del concepto “excavaciones” .....	41
Figura A.15 Eliminación del atributo “Periodo” dentro del concepto “excavaciones”.....	42
Figura A.16. Interfaz “Nuevo elemento Patrimonial”.....	44
Figura A.17. Información referente al concepto “arquitecturareligiosa” del documento XML. .	44
Figura A.18. Interfaz “Modificar elemento Patrimonial”. ....	45
Figura A.19. Lista de elemento patrimoniales que aparece en el proceso denominado como “Eliminar Patrimonio”.....	46
Figura A.20. Lista de elemento patrimoniales que aparece en el proceso denominado como “Modificar Patrimonio”.....	46
Figura A.21. Documento XML en el que se muestra el concepto “excavaciones”. ....	47



Figura A.22. Documento “usuarios.xml” .....	48
Figura A.23. Documento “usuarios.xml”. Nuevo usuario introducido. ....	48
Figura A.24. Documento “usuarios.xml”. Nueva página asociada a usuario. ....	49
Figura A.25. Una parte del Árbol de Conceptos.....	51
Figura A.26. Posible modificación del nombre del concepto “objetos”. ....	53
Figura A.27. El nombre del atributo “Población” va a ser sustituido. ....	54
Figura A.28. Posibles valores otorgados a los atributos Periodo y Tipo_Datos. ....	55
Figura A.29. Resultado de los valores otorgados a un atributo en la aplicación. ....	56
Figura A.30. Cambiando uno de los posibles valores del atributo. ....	56
Figura A.31. Se añade un posible valor al atributo “Periodo” .....	57
Figura A.32. Eliminar uno de los posibles valores del atributo.....	58
Figura A.33. Bases de datos en las que aparecen los atributos “Población” y “Municipio”. .....	58
Figura A.34. “Atributo Municipio”. Identificador asociado a una interfaz existente en el almacén de Componentes IU. ....	59
Figura A.35. “Relación de Generalización/Especialización”. Identificador asociado a una interfaz existente en el almacén de Componentes IU.....	59
Figura A.36. Apariencia del documento “sinónimos.xml” .....	60
Figura A.37. Nuevo sinónimo. ....	61
Figura A.38. Modificación de un sinónimo.....	61
Figura A.39. Eliminación de un sinónimo de una palabra. ....	62
Figura A.40. Nueva palabra en el documento XML.....	62
Figura A.41. Modificación de una palabra en el documento XML. ....	63
Figura A.42. Eliminación de una palabra en el documento XML. ....	64
Figura A.43. Eliminación de una palabra junto con sus sinónimos en el documento XML .....	64
Figura A.44. Documento “BDsComponentes.xml” .....	65

Figura A.45. Modificación de la descripción de la base de datos con id= “1” .....	65
Figura A.46. Inscripción de una nueva base de datos. ....	66
Figura A.47. Función traductorConsulta2. ....	67
Figura A.48. Parte del documento “BDsComponentes.xml” correspondiente a la base de datos LDGP. Marcado el identificador id= “2”. ....	67
Figura A.49. Parte del documento “algoritmoTraductorConsulta2.php3” .....	68
Figura A.50. Parte del documento “algoritmoTraductorConsulta2.php3”. Ruta modificada. ....	68
Figura A.51. Parte del documento “algoritmoTraductorConsulta2.php3” que es necesario cambiar. ....	68
Figura A.52. Parte del documento “algoritmoTraductorConsulta2.php3” que es necesario cambiar. ....	69
Figura A.53. Parte del documento “algoritmoTraductorConsulta2.php3”. Modificando la ruta del fichero sinónimos. ....	69
Figura A.54. Parte del documento “algoritmoTraductorConsulta2.php3”. Modificando el nombre del método. ....	69
Figura A.55. Parte del documento “gestorBD2.php”, Modificando la cabecera del método. ....	70
Figura A.56. Parte del documento “gestorBD2.php”. Modificando el nombre del método. ....	70
Figura A.57. Parte del documento “emisorRespuestas2.php3”. Modificando la cabecera del método. ....	70
Figura A.58. Estructura “if” añadida al fichero “capaPresentacion.php3” .....	71
Figura A.59. Autenticación de usuarios .....	72
Figura A.60. Representa el menú de la aplicación, concretamente las funcionalidades de Sitio Web, las cuales se explican a continuación. ....	73
Figura A.61. Nuevo sitio Web .....	74
Figura A.62. Sitio Web introducido .....	74
Figura A.63. Mensaje de aviso. El sitio web ya existe en la base de datos. ....	75
Figura A.64. Modificar sitio Web. ....	75
Figura A.65. Lista desplegable de Modificar sitio Web. ....	76

Figura A.66. Modificar sitio Web. Datos de sitio Web preparados para modificar.....	76
Figura A.67. Sitio Web Modificado. ....	77
Figura A.68. Mensaje de aviso. Se intenta introducir un sitio Web que ya existe.....	77
Figura A.69. Lista de sitios Web. ....	78
Figura A.70. Confirmación de eliminación de sitio Web.....	78
Figura A.71. Representa el menú de la aplicación, concretamente las funcionalidades de Sub- Páginas Web, las cuales se explican a continuación. ....	79
Figura A.72. Introducir nueva sub-página Web. ....	80
Figura A.73. Subpáginas Web del sitio Web <i>http://www.vc.ehu.es/docarq/</i> . En rojo aparece la subpágina que se acaba de insertar. ....	80
Figura A.74. Mensaje de aviso. La subpágina Web ya existía. ....	81
Figura A.75. Muestra la lista desplegable de subpáginas Web una vez seleccionado un sitio Web determinado. ....	81
Figura A.76. Modificar subpágina Web.....	82
Figura A.77. Mensaje de aviso. La subpágina Web ha sido modificada.....	82
Figura A.78. Mensaje de aviso. La dirección de la subpágina Web que se pretendía modificar ya pertenece a otra subpágina Web. ....	83
Figura A.79. Lista de sub-páginas Web del sitio Web seleccionado en la lista desplegable superior.....	83
Figura A.80. Confirmación de eliminación de una subpágina Web. ....	84
Figura A.81. Representa el menú de la aplicación, concretamente las funcionalidades de Patrimonio, las cuales se explican a continuación. ....	85
Figura A.82. Nuevo elemento patrimonial. Seleccionado el sitio Web y seleccionándose la subpágina Web donde se va insertar el elemento patrimonial. ....	86
Figura A.83. Nuevo elemento patrimonial. Seleccionado el sitio Web y la subpágina Web donde se va insertar el elemento patrimonial. También ha sido seleccionado el tipo de elemento patrimonial.....	87
Figura A.84. Mensaje de aviso. El elemento patrimonial ha sido introducido.....	87

Figura A.85. Modificar elemento patrimonial. Seleccionado el sitio Web y seleccionándose la subpágina Web en la cual se encuentra el elemento patrimonial que se va a modificar. ....	88
Figura A.86. Modificar elemento Patrimonial. Listado de elementos patrimoniales de la subpágina seleccionada. ....	88
Figura A.87. Modificar elemento Patrimonial. Datos del elemento patrimonial seleccionado para ser modificado. ....	89
Figura A.88. Mensaje de aviso. El elemento patrimonial ha sido modificado. ....	89
Figura A.89. Eliminar elemento patrimonial. Seleccionado el sitio Web y seleccionándose la subpágina Web en la cual se encuentra el elemento patrimonial que se va a eliminar. ....	90
Figura A.90. Eliminar elemento Patrimonial. Listado de elementos patrimoniales de la subpágina seleccionada. ....	90
Figura A.91. Confirmación de eliminación de un elemento patrimonial. ....	91
Figura A.92. Representa el menú de la aplicación, concretamente las funcionalidades de Administrador, las cuales se explican a continuación. ....	92
Figura A.93. Nuevo Administrador(o usuario), aparece un formulario a rellenar con los datos de la persona a introducir. ....	93
Figura A.94. Nuevo Administrador(o usuario), lista de administradores y usuarios de la página Web. Remarcado en rojo el nuevo miembro. ....	93
Figura A.95. Mensaje de aviso. Este Nick no está disponible. ....	94
Figura A.96. Mensaje de aviso. Este Nick no está disponible. ....	94
Figura A.97. Modificar Administrador(o usuario). Datos en el formulario. ....	95
Figura A.98. Mensaje de aviso. El administrador ha sido modificado. ....	96
Figura A.99. Administrador(o usuario) modificado, lista de administradores y usuarios de la página Web. Remarcado en rojo el miembro que ha sido modificado. ....	96
Figura A.100. Mensaje de aviso. El administrador ya existe en la base de datos. ....	97
Figura A.101. Modificar contraseña. ....	97
Figura A.102. Administrador(o usuario) modificado, lista de administradores y usuarios de la página Web. Remarcado en rojo el miembro que ha sido modificado. ....	98
Figura A.103. Lista de usuario a los que puede eliminar el administrador Ion. La lista incluye también a Ion ya que Ion se puede eliminar a sí mismo. ....	98

Figura A.104. Confirmación de eliminación de un usuario. ....	99
Figura A.105. Listado de los movimientos que han sido llevados a cabo.....	100
Figura A.106. Lista de administradores. ....	100
Figura A.107. Modificar contraseña administrador. ....	101
Figura A.108. Administrador modificado, lista de administradores y usuarios de la página Web. Remarcado en rojo el miembro que ha sido modificado. ....	101
Figura A.109. Botón “CerrarSesion”. Situado en la parte superior izquierda. ....	102
Figura A.110. Mensaje de aviso .Sesión finalizada. ....	102
Figura A.111. Representa el menú de la aplicación, concretamente la funcionalidad de Sitio Web, la cual se explica a continuación.....	103
Figura A.112. Modificar sitio Web.....	104
Figura A.113. Lista desplegable de Modificar sitio Web.....	104
Figura A.114. Modificar sitio Web. Datos de sitio Web preparados para modificar.....	105
Figura A.115. Sitio Web Modificado.....	105
Figura A.116. Representa el menú de la aplicación, concretamente las funcionalidades de Sub- Páginas Web, las cuales se explican a continuación. ....	106
Figura A.117. Introducir nueva sub-página Web.....	107
Figura A.118. Subpáginas Web del sitio Web <i>http://www.alava.net.default_c.asp</i> .En rojo aparece la subpágina que se acaba de insertar. ....	107
Figura A.119. Mensaje de aviso. La subpágina Web ya existía. ....	108
Figura A.120. Muestra la lista desplegable de subpáginas Web una vez seleccionado un sitio Web determinado. ....	108
Figura A.121. Modificar subpágina Web.....	109
Figura A.122. Mensaje de aviso. La subpágina Web ha sido modificada.....	109
Figura A.123. Mensaje de aviso. La dirección de la subpágina Web que se pretendía modificar ya pertenece a otra subpágina Web.....	110
Figura A.124. Lista de sub-páginas Web del sitio Web seleccionado en la lista desplegable superior.....	110

Figura A.125. Confirmación de eliminación de una subpágina Web. ....	111
Figura A.126. Representa el menú de la aplicación, concretamente las funcionalidades de Patrimonio, las cuales se explican a continuación. ....	112
Figura A.127. Nuevo elemento patrimonial. Seleccionado el sitio Web y seleccionándose la subpágina Web donde se va insertar el elemento patrimonial. ....	113
Figura A.128. Nuevo elemento patrimonial. Seleccionado el sitio Web y la subpágina Web donde se va insertar el elemento patrimonial. También ha sido seleccionado el tipo de elemento patrimonial.....	114
Figura A.129. Mensaje de aviso. El elemento patrimonial ha sido introducido. ....	114
Figura A.130. Modificar elemento patrimonial. Seleccionado el sitio Web y seleccionándose la subpágina Web en la cual se encuentra el elemento patrimonial que se va a modificar. ....	115
Figura A.131. Modificar elemento Patrimonial. Listado de elementos patrimoniales de la subpágina seleccionada. ....	115
Figura A.132. Modificar elemento Patrimonial. Datos del elemento patrimonial seleccionado para ser modificado. ....	116
Figura A.133. Mensaje de aviso. El elemento patrimonial ha sido modificado. ....	116
Figura A.134. Eliminar elemento patrimonial. Seleccionado el sitio Web y seleccionándose la subpágina Web en la cual se encuentra el elemento patrimonial que se va a eliminar. ....	117
Figura A.135. Eliminar elemento Patrimonial. Listado de elementos patrimoniales de la subpágina seleccionada. ....	117
Figura A.136. Confirmación de eliminación de un elemento patrimonial. ....	118
Figura A.137. Representa el menú de la aplicación, concretamente las funcionalidades de Administrador, las cuales se explican a continuación. ....	119
Figura A.138. Listado de los movimientos que han sido llevados a cabo.....	120
Figura A.139. Lista de administradores. ....	120
Figura A.140. Modificar contraseña administrador. ....	120
Figura A.141. Contraseña usuario modificada, lista de administradores y usuarios de la página Web. Remarcado en rojo el miembro que ha sido modificado. ....	121
Figura A.142. Botón “CerrarSesion”. Situado en la parte superior izquierda. ....	122
Figura A.143. Mensaje de aviso .Sesión finalizada. ....	122



## **A. ESTUDIO Y CLASIFICACIÓN DE LOS DATOS**

### **A.1 DESCRIPCIÓN DE LAS ENTIDADES CONTACTADAS**

#### **Arabarri**

- **¿Qué es?**

Somos una sociedad pública, dependiente de la Administración Foral y de los ayuntamientos asociados, encargada de fomentar el conocimiento, la conservación y la recuperación del Patrimonio Cultural Edificado de Álava.

- **¿Qué hace?**

- Unificación de la documentación y catalogación de patrimonio.
- En coordinación con el Centro de Patrimonio Cultural Vasco, elaboramos una base de información permanentemente actualizada sobre los elementos objeto de protección. Además, se gestiona la asimilación de las diversas normativas confluentes (accesibilidad, habitabilidad,...) en los regímenes de protección de los elementos.
- Intervención en el planeamiento urbanístico.
- Elaboración y desarrollo de planes de gestión de conjuntos de interés.
- Gestión de actuaciones de restauración y puesta en uso.
- Asesoramiento a particulares, tramitación de ayudas.
- Gestión de actuaciones de restauración promovidas por Diputación y Ayuntamientos.
- Búsqueda de financiación, patrocinio privado, convenios,...
- Formación de los agentes implicados en la restauración.
- Actividades de difusión del patrimonio y de su recuperación.
- Utilización del Patrimonio como recurso económico.

- **Conclusión**

Por lo que he podido ver se trata de una página que reúne las actuaciones (reparaciones...) que ha ido llevando a cabo a lo largo de su existencia. Esta



página no reúne información que describa el patrimonio, sino las reformas realizadas sobre el propio patrimonio.

En cuanto a la informatización, considero que trabajan contra una base de datos de la cual desconozco sus campos aunque me atrevería a decir que no es una base de datos muy compleja.

### **Archivo histórico de la diócesis de Vitoria-Gasteiz**

- **¿Qué es?**

Organización eclesiástica.

- **¿Qué hace?**

Parece que es una página estática que almacena información de la propia organización. Tiene un apartado de patrimonio, que es meramente explicativo, donde aparece información sobre el catálogo municipal de la diócesis de Vitoria, la fundación Catedral de Santa María, Museo Diocesano de Alto Sacro.

- **Conclusión**

Creo que no tienen ninguna base de datos donde almacenan los datos, o parece por lo menos que esta página no trabaja contra ninguna. De esta organización podríamos reunir algo de información, aunque, como ya he comentado antes, considero que no disponen de base de datos.

### **Archivo municipal de Vitoria-Gasteiz**

- **¿Qué es?**

El Archivo Municipal es un centro abierto al público que contiene información:

- Sobre la actividad del Ayuntamiento de Vitoria-Gasteiz
- Sobre cualquier aspecto relacionado con la ciudad: historia, acontecimientos, evolución, urbanismo, arte, costumbres, personajes... a partir de todo tipo de fuentes documentales que conforman sus distintas secciones:
  - Sección Histórica
  - Sección Administrativa
  - Sección Gráfica
  - Sección de Fotografía
  - Biblioteca de Investigación Histórica y Estudios Locales
  - Biblioteca Técnica

Cada una de ellas recoge, organiza, describe y custodia los documentos que le son propios, controla su difusión, atiende las consultas y préstamos a las oficinas municipales y facilita su consulta por el público interesado dentro de las condiciones de acceso establecidas en cada caso.

- **¿Qué hace?**

Da un servicio de información sobre Vitoria.

- **Conclusión**

Yo he accedido a esta información a través de la página del ayuntamiento, y dicha información es sólo descriptiva, es decir, describe la información, de manera muy genérica, que puedes encontrar en el archivo municipal. Si alguien desea encontrar una información determinada este servicio a través de la web otorgado por el ayuntamiento no le va a aportar casi nada.

### **Diputación Foral de Álava**

- **¿Qué es?**

La página de la Diputación de Álava.

- **¿Qué hace?**

Da un servicio de información sobre Vitoria.

- **Conclusión**

De lo que he podido observar, no sé si me quedará algo, me parece un modelo interesante, con buena organización. En él aparecen una parte importante de los museos, edificios institucionales, albergues, edificios agropecuarios... además del Artium, la catedral de Santa María... Para cada uno de estos aparece una descripción. Obviamente, supongo, que la Diputación contará con un sistema de base de datos que recoja toda la información.

### **EUVE**

- **¿Qué es?**

European Virtual Engineering (EUVE) es un centro tecnológico especializado en soluciones de Realidad Virtual, Televisión, Ingeniería Virtual, Meteorología y Sistemas de Información, Entrenamiento y Apoyo a la Toma de Decisiones.

- **¿Qué hace?**

Tiene como objetivo la generación de conocimiento y su transferencia a las empresas para lograr una mejora en sus productos, servicios e imagen, obteniendo una mayor competitividad mediante el perfeccionamiento y la gestión tecnológica.

- **Conclusión**

No es una institución que trabaje directamente con el patrimonio, sin embargo contiene imágenes virtuales sobre diferentes edificios, ya sean iglesias, catedrales...

### **Fundación Catedral de Santa María de Vitoria**

- **¿Qué es?**

Una fundación que da información sobre la Catedral de San María

- **¿Qué hace?**

Da información de diferente índole sobre la Catedral de San María.

- **Conclusión**

Contiene información variada y bastante completa sobre la catedral de Santa María, no sólo en el aspecto informativo de la misma, sino también en cuanto a los proyectos de restauración, las publicaciones o la propia institución. Me parece bastante claro que los datos que manejan están almacenados en una base de datos.

### **Fundación Euskomedia**

- **¿Qué es?**

Es una organización que contiene imágenes de diversa índole.

- **¿Qué hace?**

Proporciona imágenes de todo tipo.

- **Conclusión**

Puede resultar interesante ya que dispone de un buscador de imágenes en el que podemos encontrar imágenes sobre diferentes tipos de patrimonio.

## **Instituto Alavés de Arqueología**

- **¿Qué es?**

Es un Centro de Investigación Arqueológica dedicado al estudio de nuestro pasado alavés. Comenzó su andadura en 1957 en el seno del desaparecido Consejo de Cultura de la Diputación Foral de Álava, pasando en 1978 a tomar su actual denominación, organizándose dentro del régimen de Asociaciones Culturales sin ánimo de lucro.

- **¿Qué hace?**

Se dedica a todo lo relacionado con la arquitectura en Álava (excavaciones, prospecciones, estudios, publicaciones...).

- **Conclusión**

Contiene información sobre la organización. En cuanto a la información que nos puede interesar, contiene el título, el nombre del autor, la descripción, y en algunas ocasiones la fecha de las publicaciones realizadas hasta la fecha. Aunque la página web parece estática, la organización debe tener una base de datos donde almacene su información.

## **Sociedad de Ciencias Aranzadi**

- **¿Qué es?**

Es una asociación científica sin ánimo de lucro cuyos objetivos son la investigación, divulgación y protección del Patrimonio Cultural y Natural.

- **¿Qué hace?**

Se dedica a la investigación, divulgación y protección del Patrimonio Cultural y Natural.

- **Conclusión**

Parece una página completa y bien estructurada. Se divide en campos como antropología física, arquitectura histórica, arquitectura prehistórica, geología... En cada uno de ellos aparece una breve descripción del campo y otorga la posibilidad de acceder a las noticias, agenda, publicaciones y documentos. En cada uno de estos campos aparece una información con cada publicación, noticia o documento.

En cuanto a la informatización, yo creo que trabajan contra una base de datos donde almacenan toda la información que aparece en la página.

### **Universidad del País Vasco**

- **¿Qué es?**

Es la Universidad del País Vasco (UPV/EHU). Respecto a lo que nos concierne, contiene información abundante y variada sobre el patrimonio de Álava.

- **¿Qué hace?**

Respecto al patrimonio alavés realiza investigaciones sobre el mismo, y otras actividades de otra índole.

- **Conclusión**

Dispone de dos páginas Web bastante completas con información abundante y diversa. Considero que puede ser una institución muy útil para el proyecto.

## A.2 CLASIFICACIÓN DE LOS DATOS

Uno de los apartados más importantes de la aplicación ha sido el estudio realizado sobre los datos disponibles en las diferentes organizaciones. Gracias a este estudio se ha podido llevar a cabo una clasificación coherente de los elementos patrimoniales. Aunque en su momento se elaboró una clasificación de cada una de las organizaciones, en la cual se incluían todas las subpáginas Web de cada una de ellas, de las cuales se presenta una pequeña descripción y se clasificaban los elementos patrimoniales que allí se albergaban, se considera innecesario para el lector mostrar toda esta información, ya que carece de interés y puede resultar hasta pesado.

Por esto, se ha decidido, a modo de ejemplo, mostrar únicamente la clasificación de uno de los sitios Web, Laboratorio de documentación geométrica del patrimonio, con el fin de informar al lector cómo se llevaron a cabo dichas clasificaciones de datos.

### **Laboratorio de documentación geométrica del patrimonio**

**Página principal:** <http://www.vc.ehu.es/docarq/>

#### **Descripción:**

Se trata de la página Web del grupo de investigación en arqueología de la arqueología de la Universidad del País Vasco UPV/EHU.

#### **Sub-páginas:**

- <http://www.vc.ehu.es/docarq/LDA/proyectos/UOlleros/index.htm>

Elemento/ Nombre	Encuadrado	Tipo	Lugar	Periodo	Tipo_datos
Iglesia de Asunción	Arq. Religiosa	Iglesia	Ullibarri de los Olleros	No definido	Descripción

- <http://www.vc.ehu.es/docarq/LDA/proyectos/Gamiz/index.htm>

Elemento/ Nombre	Encuadrado	Tipo	Lugar	Periodo	Tipo_datos
Iglesia de Santa Eulalia	Arq. Religiosa	Iglesia	Gamiz	No definido	Descripción

- <http://www.vc.ehu.es/docarq/LDA/proyectos/Arlucea/Pag2.htm>

Elemento/ Nombre	Encuadrado	Tipo	Lugar	Periodo	Tipo_datos
Iglesia de San Martín	Arq. Religiosa	Iglesia	Arlucea (Bernedo)	No definido	Descripción

- <http://www.vc.ehu.es/docarq/LDA/proyectos/Vinaspre/index.htm>

Elemento/ Nombre	Encuadrado	Tipo	Lugar	Periodo	Tipo_datos
Iglesia de Asunción	Arq. Religiosa	Iglesia	Viñaspre (Lanciego)	No definido	Descripción

- <http://www.vc.ehu.es/docarq/LDA/proyectos/SanVicente/index.htm>

Elemento/ Nombre	Encuadrado	Tipo	Lugar	Periodo	Tipo_datos
Iglesia de San Vicente	Arq. Religiosa	Iglesia	Vitoria-Gasteiz	No definido	Descripción

- <http://www.vc.ehu.es/docarq/LDA/proyectos/Armentia/index.htm>

Elemento/ Nombre	Encuadrado	Tipo	Lugar	Periodo	Tipo_datos
Basílica de San Prudencio de Armentia	Arq. Religiosa	Basílica	Vitoria-Gasteiz	No definido	Descripción

- <http://www.vc.ehu.es/docarq/LDA/proyectos/Ribera/index.htm>

Elemento/ Nombre	Encuadrado	Tipo	Lugar	Periodo	Tipo_datos
Basílica de San Esteban	Arq. Religiosa	Basílica	Ribera de Valderejo	No definido	Descripción

- <http://www.vc.ehu.es/docarq/LDA/proyectos/pdf/Viloria.pdf>

Elemento/ Nombre	Encuadrado	Tipo	Lugar	Periodo	Tipo_datos
Basílica de Santa Eulalia	Arq. Religiosa	Basílica	Villoria	No definido	Descripción

Aún existen muchas más páginas Web, sin embargo no hemos incluido todas, ya que podría resultar pesado para el lector. Consideramos que con la información descrita es suficiente para que el lector se pueda hacer una idea del procedimiento que se ha seguido para clasificar cada una de las páginas Web.

## B. CONCEPTOS IMPORTANTES DE LA APLICACIÓN DE BÚSQUEDA ORIENTADA

A lo largo de este apartado se muestran conceptos importantes de la Aplicación de Búsqueda Orientada, como son el Árbol de Conceptos, los Árboles de Correspondencias, y el Lenguaje Consulta. Para cada uno de ellos se mostrará además su correspondiente DTD.

### B.1 ÁRBOL DE CONCEPTOS

A continuación se puede apreciar tanto el DTD del Árbol de Conceptos (Figura A.1) como el propio Árbol de Conceptos (Figura A.2).

#### B.1.1 DTD ÁRBOL DE CONCEPTOS

```
<!ELEMENT Arbol-conceptos (concepto)>
<!ELEMENT concepto (IU?,((atributo+, isa?) | isa))>
<!ATTLIST concepto nombre CDATA #REQUIRED>
<!ELEMENT atributo (IU, bd+, valor*)>
<!ATTLIST atributo nombre CDATA #REQUIRED>
<!ELEMENT IU EMPTY>
<!ATTLIST IU id CDATA #REQUIRED>
<!ELEMENT bd EMPTY>
<!ATTLIST bd id CDATA #REQUIRED>
<!ELEMENT valor EMPTY>
<!ATTLIST valor val CDATA #REQUIRED>
<!ELEMENT isa (IU?, concepto+)>
```

Figura A.1. DTD Árbol de Conceptos

#### B.1.2 ÁRBOL DE CONCEPTOS

```
<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
<!DOCTYPE Arbol-conceptos SYSTEM "DTDArbolDeConceptos.dtd">
<Arbol-conceptos>
<!-- Inicio Patrimonio -->
<concepto nombre="Patrimonio">
<atributo nombre="Nombre">
<IU id="1" />
<bd id="1" /><bd id="2" />
</atributo>
<atributo nombre="Periodo">
<IU id="3" />
<bd id="1" />
<valor val="Romano"/>
<valor val="SXV"/>
<valor val="SXVI"/>
<valor val="SXVII"/>
<valor val="SXVIII"/>
<valor val="SXIX"/>
<valor val="SXX"/>
```



```

</atributo>
<atributo nombre="Poblacion">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Municipio">
<IU id="2" />
<bd id="1" />
</atributo>
<atributo nombre="Provincia">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Tipo_Datos">
<IU id="2" />
<bd id="1" /> <bd id="2" />
<valor val="descripcion"/>
<valor val="imagenes"/>
<valor val="fotogrametria"/>
<valor val="topografia"/>
</atributo>
<isa>
<!-- Inicio de Excavaciones (isa de Patrimonio) -->
<IU id="Radiol"/>
<concepto nombre="Excavaciones">
<atributo nombre="Nombre">
<IU id="1" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Tipo_Excavacion">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Periodo">
<IU id="3" />
<bd id="1" />
<valor val="Romano"/>
<valor val="SXV"/>
<valor val="SXVI"/>
<valor val="SXVII"/>
<valor val="SXVIII"/>
<valor val="SXIX"/>
<valor val="SXX"/>
</atributo>
<atributo nombre="Poblacion">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Municipio">
<IU id="2" />
<bd id="1" />
</atributo>
<atributo nombre="Provincia">
<IU id="2" />
<bd id="1" /> <bd id="2" />

```

```

</atributo>
<atributo nombre="Tipo_Datos">
<IU id="2" />
<bd id="1" /> <bd id="2" />
<valor val="descripcion"/>
<valor val="imagenes"/>
<valor val="fotogrametria"/>
<valor val="topografia"/>
</atributo>
</concepto>
<!-- Fin Excavaciones (isa de Patrimonio) -->
<!-- Inicio de Edificios (isa de Patrimonio) -->
<concepto nombre="Edificios">
<atributo nombre="Nombre">
<IU id="1" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Tipo_Edificio">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Periodo">
<IU id="3" />
<bd id="1" />
<valor val="Romano"/>
<valor val="SXV"/>
<valor val="SXVI"/>
<valor val="SXVII"/>
<valor val="SXVIII"/>
<valor val="SXIX"/>
<valor val="SXX"/>
</atributo>
<atributo nombre="Poblacion">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Municipio">
<IU id="2" />
<bd id="1" />
</atributo>
<atributo nombre="Provincia">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Tipo_Datos">
<IU id="2" />
<bd id="1" /> <bd id="2" />
<valor val="descripcion"/>
<valor val="imagenes"/>
<valor val="fotogrametria"/>
<valor val="topografia"/>
</atributo>
<isa>
<!-- Inicio de Arquitectura Religiosa (isa de Edificios) -->
<IU id="Radio2"/>

```

```

<concepto nombre="Arquitectura Religiosa">
<atributo nombre="Nombre">
<IU id="1" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Tipo_ArquitecturaReligiosa">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Periodo">
<IU id="3" />
<bd id="1" />
<valor val="Romano"/>
<valor val="SXV"/>
<valor val="SXVI"/>
<valor val="SXVII"/>
<valor val="SXVIII"/>
<valor val="SXIX"/>
<valor val="SXX"/>
</atributo>
<atributo nombre="Poblacion">
<IU id="2" />
<bd id="1" /><bd id="2" />
</atributo>
<atributo nombre="Municipio">
<IU id="2" />
<bd id="1" />
</atributo>
<atributo nombre="Provincia">
<IU id="2" />
<bd id="1" /><bd id="2" />
</atributo>
<atributo nombre="Tipo_Datos">
<IU id="2" />
<bd id="1" /><bd id="2" />
<valor val="descripcion"/>
<valor val="imagenes"/>
<valor val="fotogrametria"/>
<valor val="topografia"/>
</atributo>
</concepto>
<!-- Fin Arquitectura Religiosa (isa de Edificios) -->
<!-- Inicio de Arquitectura Civil (isa de Edificios) -->
<concepto nombre="Arquitectura Civil">
<atributo nombre="Nombre">
<IU id="1" />
<bd id="1" />
</atributo>
<atributo nombre="Tipo_ArquitecturaCivil">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Periodo">
<IU id="3" />
<bd id="1" />

```

```

<valor val="Romano"/>
<valor val="SXV"/>
<valor val="SXVI"/>
<valor val="SXVII"/>
<valor val="SXVIII"/>
<valor val="SXIX"/>
<valor val="SXX"/>
</atributo>
<atributo nombre="Poblacion">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Municipio">
<IU id="2" />
<bd id="1" />
</atributo>
<atributo nombre="Provincia">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Tipo_Datos">
<IU id="2" />
<bd id="1" /> <bd id="2" />
<valor val="descripcion"/>
<valor val="imagenes"/>
<valor val="fotogrametria"/>
<valor val="topografia"/>
</atributo>
</concepto>
<!-- Fin Arquitectura Civil (isa de Edificios) -->
<!-- Inicio de Elementos Menores (isa de Edificios) -->
<concepto nombre="Elementos Menores">
<atributo nombre="Nombre">
<IU id="1" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Tipo_ElementosMenores">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Periodo">
<IU id="3" />
<bd id="1" />
<valor val="Romano"/>
<valor val="SXV"/>
<valor val="SXVI"/>
<valor val="SXVII"/>
<valor val="SXVIII"/>
<valor val="SXIX"/>
<valor val="SXX"/>
</atributo>
<atributo nombre="Poblacion">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>

```

```

<atributo nombre="Municipio">
<IU id="2" />
<bd id="1" />
</atributo>
<atributo nombre="Provincia">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Tipo_Datos">
<IU id="2" />
<bd id="1" /> <bd id="2" />
<valor val="descripcion"/>
<valor val="imagenes"/>
<valor val="fotogrametria"/>
<valor val="topografia"/>
</atributo>
</concepto>
<!-- Fin Elementos Menores (isa de Edificios) -->
</isa>
<!-- Fin de isa de Edificios -->
</concepto>
<!-- Fin de Edificios (isa de Patrimonio) -->
<!-- Inicio de Muebles/Objetos (isa de Patrimonio) -->

<concepto nombre="Objetos">
<atributo nombre="Nombre">
<IU id="1" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Tipo_Muebles/Objetos">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Periodo">
<IU id="3" />
<bd id="1" />
<valor val="Romano"/>
<valor val="SXV"/>
<valor val="SXVI"/>
<valor val="SXVII"/>
<valor val="SXVIII"/>
<valor val="SXIX"/>
<valor val="SXX"/>
</atributo>
<atributo nombre="Poblacion">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Municipio">
<IU id="2" />
<bd id="1" />
</atributo>
<atributo nombre="Provincia">
<IU id="2" />
<bd id="1" /> <bd id="2" />

```

```
</atributo>
<atributo nombre="Tipo_Datos">
<IU id="2" />
<bd id="1" /> <bd id="2" />
<valor val="descripcion"/>
<valor val="imagenes"/>
<valor val="fotogrametria"/>
<valor val="topografia"/>
</atributo>
</concepto>
<!-- Fin de Objetos (isa de Patrimonio) -->

<!-- Fin de isa de Patrimonio -->
</isa>
<!-- Fin de Patrimonio -->
</concepto>
</Arbol-conceptos>
```

Figura A.2. Árbol de Conceptos

## B.2 ÁRBOLES DE CORRESPONDENCIAS

En este anexo se describe el DTD de los Árboles de Correspondencias de nuestra arquitectura y se presentan los documentos XML de los Árboles de Correspondencias de las dos bases de datos integradas en nuestro sistema.

### B.2.1 DTD DE LOS ÁRBOLES DE CORRESPONDENCIAS

El DTD es el mismo para todos los Árboles de Correspondencias del sistema, en este caso dos. En la siguiente figura (figura A.3) se muestra el DTD de los Árboles de Correspondencias del sistema.

```
<!ELEMENT arbol (concepto)>
<!ELEMENT concepto (correspondencia,
((atributo+, isa?) | (isa)))>
<!ATTLIST concepto nombre CDATA #REQUIRED>
<!ELEMENT atributo (correspondencia)>
<!ATTLIST atributo nombre CDATA #REQUIRED>
<!ELEMENT correspondencia (comun?, select?, from?, where)>
<!ELEMENT comun (#PCDATA)>
<!ELEMENT select (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT where (#PCDATA)>
<!ELEMENT isa (concepto+)>
```

Figura A.3. DTD de los Árboles de Correspondencias

### B.2.2 ÁRBOLES DE CORRESPONDENCIAS

En las siguientes figuras (figura A.4 y figura A.5) se muestran los Árboles de Correspondencias de cada uno de los sistemas envoltorio del sistema, el Árbol de Correspondencia asociado a la base de datos que almacena información de las páginas Webs, el Árbol de Correspondencia asociado a la base de datos del Laboratorio de Documentación Geométrica del Patrimonio de la UPV/EHU, respectivamente.

### B.2.2.1 ÁRBOLES DE CORRESPONDENCIAS PÁGINAS WEB.

```
?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE arbol SYSTEM "DTDArbolDeCorrespondencias.dtd">
<arbol>
<!-- Inicio concepto Patrimonio -->
<concepto nombre="Patrimonio">
  <correspondencia>
    <select>DISTINCT spw.Direccion_Web</select>
    <from>subpaganaweb spw, excavaciones ex, arquitecturareligiosa
      ar, arquitecturacivil ac, elementosmenores em, objetos ob
    </from>
    <where>
    </where>
  </correspondencia>
  <atributo nombre="Nombre">
    <correspondencia>
      <where>(spw.Direccion_Web = ex.Direccion_Web AND ex.Nombre
        #op# '%#valor#%' ) OR (spw.Direccion_Web =
        ar.Direccion_Web AND ar.Nombre #op# '%#valor#%' ) OR
        (spw.Direccion_Web = ac.Direccion_Web AND ac.Nombre
        #op# '%#valor#%' ) OR (spw.Direccion_Web =
        em.Direccion_Web AND em.Nombre #op# '%#valor#%' ) OR
        (spw.Direccion_Web = ob.Direccion_Web AND ob.Nombre
        #op# '%#valor#%' )
      </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Periodo">
    <correspondencia>
      <where>(spw.Direccion_Web = ex.Direccion_Web AND ex.Periodo
        #op# '%#valor#%' ) OR (spw.Direccion_Web =
        ar.Direccion_Web AND ar.Periodo #op# '%#valor#%' ) OR
        (spw.Direccion_Web = ac.Direccion_Web AND ac.Periodo
        #op# '%#valor#%' ) OR (spw.Direccion_Web =
        em.Direccion_Web AND em.Periodo #op# '%#valor#%' ) OR
        (spw.Direccion_Web = ob.Direccion_Web AND ob.Periodo
        #op# '%#valor#%' )
      </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Municipio">
    <correspondencia>
      <where>(spw.Direccion_Web = ex.Direccion_Web AND ex.Municipio
        #op# '%#valor#%' ) OR (spw.Direccion_Web =
        ar.Direccion_Web AND ar.Municipio #op# '%#valor#%' )
        OR (spw.Direccion_Web = ac.Direccion_Web AND
        ac.Municipio #op# '%#valor#%' ) OR (spw.Direccion_Web
        = em.Direccion_Web AND em.Municipio #op# '%#valor#%' )
        OR (spw.Direccion_Web = ob.Direccion_Web AND Municipio
        #op# '%#valor#%' )
      </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Poblacion">
```



```

<correspondencia>
  <where>(spw.Direccion_Web = ex.Direccion_Web AND ex.Poblacion
#op# '%#valor#%' ) OR (spw.Direccion_Web =
ar.Direccion_Web AND ar.Poblacion #op# '%#valor#%' )
OR (spw.Direccion_Web = ac.Direccion_Web AND
ac.Poblacion #op# '%#valor#%' ) OR (spw.Direccion_Web
= em.Direccion_Web AND em.Poblacion #op# '%#valor#%' )
OR (spw.Direccion_Web = ob.Direccion_Web AND
ob.Poblacion #op# '%#valor#%' )
  </where>
</correspondencia>
</atributo>
<atributo nombre="Provincia">
  <correspondencia>
    <where>(spw.Direccion_Web = ex.Direccion_Web AND ex.Provincia
#op# '%#valor#%' ) OR (spw.Direccion_Web =
ar.Direccion_Web AND ar.Provincia #op# '%#valor#%' ) OR
(spw.Direccion_Web = ac.Direccion_Web AND ac.Provincia
#op# '%#valor#%' ) OR (spw.Direccion_Web =
em.Direccion_Web AND em.Provincia #op# '%#valor#%' ) OR
(spw.Direccion_Web = ob.Direccion_Web AND ob.Provincia
#op# '%#valor#%' )
    </where>
  </correspondencia>
</atributo>
<atributo nombre="Tipo_Datos">
  <correspondencia>
    <where>(spw.Direccion_Web = ex.Direccion_Web AND
ex.Tipo_datos #op# '%#valor#%' ) OR (spw.Direccion_Web
= ar.Direccion_Web AND ar.Tipo_datos #op# '%#valor#%' )
OR (spw.Direccion_Web = ac.Direccion_Web AND
ac.Tipo_datos #op# '%#valor#%' ) OR (spw.Direccion_Web
= em.Direccion_Web AND em.Tipo_datos #op# '%#valor#%' )
OR (spw.Direccion_Web = ob.Direccion_Web AND
ob.Tipo_datos #op# '%#valor#%' )
    </where>
  </correspondencia>
</atributo>
<isa>
<concepto nombre="Excavaciones">
  <correspondencia>
    <select>DISTINCT ex.Direccion_Web </select>
    <from> Excavaciones ex, subpaginaweb sub
    </from>
    <where>ex.Direccion_Web=sub.Direccion_Web </where>
  </correspondencia>
  <atributo nombre="Nombre">
    <correspondencia>
      <where>ex.Nombre #op# '%#valor#%' </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Tipo_Excavacion">
    <correspondencia>
      <where>ex.Tipo_Excavacion #op# '%#valor#%'</where>
    </correspondencia>
  </atributo>

```

```

</atributo>
<atributo nombre="Periodo">
<correspondencia>
  <where>ex.Periodo #op# '%#valor#%'</where>
</correspondencia>
</atributo>
<atributo nombre="Municipio">
  <correspondencia>
    <where>ex.Municipio #op# '%#valor#%'</where>
  </correspondencia>
</atributo>
<atributo nombre="Poblacion">
  <correspondencia>
    <where>ex.Poblacion #op# '%#valor#%'</where>
  </correspondencia>
</atributo>
<atributo nombre="Provincia">
  <correspondencia>
    <where>ex.Provincia #op# '%#valor#%'</where>
  </correspondencia>
</atributo>
<atributo nombre="Tipo_Datos">
  <correspondencia>
    <where>ex.Tipo_datos #op# '%#valor#%'</where>
  </correspondencia>
</atributo>
</concepto>
<concepto nombre="Edificios">
<correspondencia>
  <select>DISTINCT spw.Direccion_Web</select>
  <from>subpaginaweb spw, arquitecturareligiosa ar,
arquitecturacivil ac, elementosmenores em
  </from>
  <where>
  </where>
</correspondencia>
<atributo nombre="Nombre">
  <correspondencia>
    <where>(spw.Direccion_Web = ar.Direccion_Web AND
ar.Nombre #op# '%#valor#%' ) OR
(spw.Direccion_Web = ac.Direccion_Web AND
ac.Nombre #op# '%#valor#%' ) OR
(spw.Direccion_Web = em.Direccion_Web AND
em.Nombre #op# '%#valor#%' )
    </where>
  </correspondencia>
</atributo>
<atributo nombre="Tipo_Edificio">
  <correspondencia>
    <where>(spw.Direccion_Web = ar.Direccion_Web AND
ar.Tipo_arquitecturaReligiosa #op# '%#valor#%' )
OR (spw.Direccion_Web = ac.Direccion_Web AND
ac.Tipo_arquitecturaCivil #op# '%#valor#%' ) OR
(spw.Direccion_Web = em.Direccion_Web AND
em.Tipo_elementosMenores #op# '%#valor#%' )
  </where>
  </correspondencia>
</atributo>

```

```

    </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Periodo">
    <correspondencia>
      <where>(spw.Direccion_Web = ar.Direccion_Web AND
ar.Periodo #op# '%#valor#%' ) OR
      (spw.Direccion_Web = ac.Direccion_Web AND
ac.Periodo #op# '%#valor#%' ) OR
      (spw.Direccion_Web = em.Direccion_Web AND
em.Periodo #op# '%#valor#%' )
    </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Municipio">
    <correspondencia>
      <where>(spw.Direccion_Web = ar.Direccion_Web AND
ar.Municipio #op# '%#valor#%' ) OR
      (spw.Direccion_Web = ac.Direccion_Web AND
ac.Municipio #op# '%#valor#%' ) OR
      (spw.Direccion_Web = em.Direccion_Web AND
em.Municipio #op# '%#valor#%' )
    </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Poblacion">
    <correspondencia>
      <where>(spw.Direccion_Web = ar.Direccion_Web AND
ar.Poblacion #op# '%#valor#%' ) OR
      (spw.Direccion_Web = ac.Direccion_Web AND
ac.Poblacion #op# '%#valor#%' ) OR
      (spw.Direccion_Web = em.Direccion_Web AND
em.Poblacion #op# '%#valor#%' )
    </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Provincia">
    <correspondencia>
      <where>(spw.Direccion_Web = ar.Direccion_Web AND
ar.Provincia #op# '%#valor#%' ) OR
      (spw.Direccion_Web = ac.Direccion_Web AND
ac.Provincia #op# '%#valor#%' ) OR
      (spw.Direccion_Web = em.Direccion_Web AND
em.Provincia #op# '%#valor#%' )
    </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Tipo_Datos">
    <correspondencia>
      <where>(spw.Direccion_Web = ar.Direccion_Web AND
ar.Tipo_datos #op# '%#valor#%' ) OR
      (spw.Direccion_Web = ac.Direccion_Web AND
ac.Tipo_datos #op# '%#valor#%' ) OR
      (spw.Direccion_Web = em.Direccion_Web AND
em.Tipo_datos #op# '%#valor#%' )
    </where>
    </correspondencia>
  </atributo>

```

```

</where>
</correspondencia>
</atributo>
<isa>
<concepto nombre="Arquitectura Religiosa">
<correspondencia>
<select>DISTINCT ar.Direccion_Web</select>
<from>Arquitecturareligiosa ar, subpaginaweb sub
</from>
<where>ar.Direccion_Web=sub.Direccion_Web </where>
</correspondencia>
<atributo nombre="Nombre">
<correspondencia>
<where>ar.Nombre #op# '%#valor#%' </where>
</correspondencia>
</atributo>
<atributo nombre="Tipo_ArquitecturaReligiosa">
<correspondencia>
<where>ar.Tipo_arquitecturaReligiosa #op#
'%#valor#%'
</where>
</correspondencia>
</atributo>
<atributo nombre="Periodo">
<correspondencia>
where>ar.Periodo #op# '%#valor#%' </where>
</correspondencia>
</atributo>
<atributo nombre="Municipio">
<correspondencia>
<where>ar.Municipio #op# '%#valor#%' </where>
</correspondencia>
</atributo>
<atributo nombre="Poblacion">
<correspondencia>
<where>.Poblacion #op# '%#valor#%' </where>
</correspondencia>
</atributo>
<atributo nombre="Provincia">
<correspondencia>
<where>ar.Provincia #op# '%#valor#%'</where>
</correspondencia>
</atributo>
<atributo nombre="Tipo_Datos">
<correspondencia>
<where>ar.Tipo_datos #op# '%#valor#%'</where>
</correspondencia>
</atributo>
</concepto>
<concepto nombre="Arquitectura Civil">
<correspondencia>
<select>DISTINCT ac.Direccion_Web</select>
<from>Arquitecturacivil ac, subpaginaweb sub </from>
<where>ac.Direccion_Web=sub.Direccion_Web </where>
</correspondencia>

```

```

<atributo nombre="Nombre">
  <correspondencia>
    <where>ac.Nombre #op# '%#valor#%' </where>
  </correspondencia>
</atributo>
<atributo nombre="Tipo_ArquitecturaCivil">
  <correspondencia>
    <where>ac.Tipo_arquitecturaCivil #op# '%#valor#%'
</where>
  </correspondencia>
</atributo>
<atributo nombre="Periodo">
  <correspondencia>
    <where>ac.Periodo #op# '%#valor#%' </where>
  </correspondencia>
</atributo>
<atributo nombre="Municipio">
  <correspondencia>
    <where>ac.Municipio #op# '%#valor#%' </where>
  </correspondencia>
</atributo>
<atributo nombre="Poblacion">
  <correspondencia>
    <where>ac.Poblacion #op# '%#valor#%' </where>
  </correspondencia>
</atributo>
<atributo nombre="Provincia">
  <correspondencia>
    <where>ac.Provincia #op# '%#valor#%' </where>
  </correspondencia>
</atributo>
<atributo nombre="Tipo_Datos">
  <correspondencia>
    <where>ac.Tipo_datos #op# '%#valor#%' </where>
  </correspondencia>
</atributo>
</concepto>
<concepto nombre="Elementos Menores">
  <correspondencia>
    <select>DISTINCT em.Direccion_Web </select>
    <from>elementosmenores em, subpaginaweb sub </from>
    <where>em.Direccion_Web=sub.Direccion_Web </where>
  </correspondencia>
  <atributo nombre="Nombre">
    <correspondencia>
      <where>em.Nombre #op# '%#valor#%' </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Tipo_ElementosMenores">
    <correspondencia>
      <where>em.Tipo_elementosMenores #op# '%#valor#%'
      </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Periodo">

```

```

        <correspondencia>
          <where>em.Periodo #op# '%#valor#%' </where>
        </correspondencia>
      </atributo>
    <atributo nombre="Municipio">
      <correspondencia>
        <where>em.Municipio #op# '%#valor#%' </where>
      </correspondencia>
    </atributo>
    <atributo nombre="Poblacion">
      <correspondencia>
        <where>em.Poblacion #op# '%#valor#%' </where>
      </correspondencia>
    </atributo>
    <atributo nombre="Provincia">
      <correspondencia>
        <where>em.Provincia #op# '%#valor#%' </where>
      </correspondencia>
    </atributo>
    <atributo nombre="Tipo_Datos">
      <correspondencia>
        <where>em.Tipo_datos #op# '%#valor#%' </where>
      </correspondencia>
    </atributo>
  </concepto>
</isa>
</concepto>
<concepto nombre="Objetos">
  <correspondencia>
    <select>DISTINCT ob.Direccion_Web </select>
    <from>Objetos ob, subpaginaweb sub </from>
    <where>ob.Direccion_Web=sub.Direccion_Web </where>
  </correspondencia>
  <atributo nombre="Nombre">
    <correspondencia>
      <where>ob.Nombre #op# '%#valor#%' </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Tipo_Muebles/Objetos">
    <correspondencia>
      <where>ob.Tipo_Objeto #op# '%#valor#%' </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Periodo">
    <correspondencia>
      <where>ob.Periodo #op# '%#valor#%' </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Municipio">
    <correspondencia>
      <where>ob.Municipio #op# '%#valor#%' </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Poblacion">
    <correspondencia>

```

```
<where>ob.Poblacion #op# '%#valor#%' </where>
</correspondencia>
</atributo>
<atributo nombre="Provincia">
  <correspondencia>
    <where>ob.Provincia #op# '%#valor#%' </where>
  </correspondencia>
</atributo>
<atributo nombre="Tipo_Datos">
  <correspondencia>
    <where>ob.Tipo_datos #op# '%#valor#%' </where>
  </correspondencia>
</atributo>
</concepto>
</isa>
</concepto>
</arbol>
```

Figura A.4. Árbol de correspondencias para base de datos Webs

### B.2.2.2 ÁRBOLES DE CORRESPONDENCIAS LDGP.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE arbol SYSTEM "DTDArbolDeCorrespondencias.dtd">
<arbol>
<!-- Inicio concepto Patrimonio -->
<concepto nombre="Patrimonio">
  <correspondencia>
    <comun>
      </comun>
      <select>DISTINCT Proyectos.`Nombre del proyecto`</select>
      <from>Proyectos</from>
      <where></where>
    </correspondencia>
    <atributo nombre="Nombre">
      <correspondencia>
        <where>(Proyectos.`Nombre del Proyecto` #op# '%#valor#%' )
      </where>
      </correspondencia>
    </atributo>
    <atributo nombre="Poblacion">
      <correspondencia>
        <where>(Proyectos.Localidad #op# '%#valor#%' ) </where>
      </correspondencia>
    </atributo>
    <atributo nombre="Provincia">
      <correspondencia>
        <where>(Proyectos.Provincia #op# '%#valor#%' ) </where>
      </correspondencia>
    </atributo>
    <atributo nombre="Tipo_Datos">
      <correspondencia>
        <where>(Proyectos.tecnicas #op# '%#valor#%' ) </where>
      </correspondencia>
    </atributo>
    <isa>
      <concepto nombre="Excavaciones">
        <correspondencia>
          <select>DISTINCT Proyectos.`Nombre del proyecto` </select>
          <from> Proyectos </from>
          <where>(Proyectos.tipo="yacimiento arqueologico") or
          (Proyectos.tipo="cueva") or
          (Proyectos.tipo="paleontologia")
          </where>
        </correspondencia>
        <atributo nombre="Nombre">
          <correspondencia>
            <where>(Proyectos.`Nombre del proyecto` #op# '%#valor#%' ) </where>
          </correspondencia>
        </atributo>
        <atributo nombre="Tipo_Excavacion">
          <correspondencia>
            <where> (Proyectos.tipo #op# '%#valor#%' ) </where>
          </correspondencia>
        </atributo>
      </concepto>
    </isa>
  </concepto>
</arbol>
```



```

<atributo nombre="Poblacion">
  <correspondencia>
    <where>(Proyectos.Localidad #op# '%#valor#%' ) </where>
  </correspondencia>
</atributo>
<atributo nombre="Provincia">
  <correspondencia>
    <where> (Proyectos.Provincia #op# '%#valor#%' ) </where>
  </correspondencia>
</atributo>
<atributo nombre="Tipo_Datos">
  <correspondencia>
    <where> (Proyectos.tecnicas #op# '%#valor#%' ) </where>
  </correspondencia>
</atributo>
</concepto>
<concepto nombre="Edificios">
  <correspondencia>
    <select>DISTINCT Proyectos.`Nombre del proyecto` </select>
    <from>Proyectos</from>
    <where>(Proyectos.tipo="monasterio") or
(Proyectos.tipo="ermita") or
(Proyectos.tipo="fuente") or
(Proyectos.tipo="puente") or
(Proyectos.tipo="torres y castillos") or
(Proyectos.tipo="casco urbano") or
(Proyectos.tipo="muralla") or
(Proyectos.tipo="iglesia") or
(Proyectos.tipo="elementos menores") or
(Proyectos.tipo="palacio") or
(Proyectos.tipo="monasterio")
    </where>
  </correspondencia>
  <atributo nombre="Nombre">
    <correspondencia>
      <where>(Proyectos.`Nombre del proyecto` #op# '%#valor#%' )
      </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Poblacion">
    <correspondencia>
      <where>(Proyectos.Localidad #op# '%#valor#%' ) </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Provincia">
    <correspondencia>
      <where> (Proyectos.Provincia #op# '%#valor#%' ) </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Tipo_Datos">
    <correspondencia>
      <where> (Proyectos.tecnicas #op# '%#valor#%' ) </where>
    </correspondencia>
  </atributo>
</isa>

```

```

<concepto nombre="Arquitectura Religiosa">
  <correspondencia>
    <select>DISTINCT Proyectos.`Nombre del proyecto`
    </select>
    <from>Proyectos</from>
    <where>(Proyectos.tipo="ermita") or
(Proyectos.tipo="iglesia") or
(Proyectos.tipo="monasterio")
    </where>
  </correspondencia>
  <atributo nombre="Nombre">
    <correspondencia>
      <where>(Proyectos.`Nombre del proyecto` #op#
'##valor#%' )
      </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Tipo_ArquitecturaReligiosa">
    <correspondencia>
      <where>(Proyectos.tipo #op# '##valor#%' ) </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Poblacion">
    <correspondencia>
      <where>(Proyectos.Localidad #op# '##valor#%' ) </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Provincia">
    <correspondencia>
      <where>(Proyectos.Provincia #op# '##valor#%' ) </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Tipo_Datos">
    <correspondencia>
      <where>(Proyectos.tecnicas #op# '##valor#%' ) </where>
    </correspondencia>
  </atributo>
</concepto>
<concepto nombre="Arquitectura Civil">
  <correspondencia>
    <select>DISTINCT Proyectos.`Nombre del proyecto`
    </select>
    <from>Proyectos</from>
    <where> (Proyectos.tipo="torres y castillos") or
(Proyectos.tipo="casco urbano") or
(Proyectos.tipo="muralla") or
(Proyectos.tipo="palacio")
    </where>
  </correspondencia>
  <atributo nombre="Nombre">
    <correspondencia>
      <where>(Proyectos.`Nombre del proyecto` #op#
'##valor#%' )
      </where>
    </correspondencia>
  </atributo>

```

```

</atributo>
<atributo nombre="Tipo_ArquitecturaCivil">
<correspondencia>
  <where>(Proyectos.tipo #op# '%#valor#%') </where>
</correspondencia>
</atributo>
<atributo nombre="Poblacion">
  <correspondencia>
    <where>(Proyectos.Localidad #op# '%#valor#%') </where>
  </correspondencia>
</atributo>
<atributo nombre="Provincia">
  <correspondencia>
    <where>(Proyectos.Provincia #op# '%#valor#%')</where>
  </correspondencia>
</atributo>
<atributo nombre="Tipo_Datos">
  <correspondencia>
    <where>(Proyectos.tecnicas #op# '%#valor#%')</where>
  </correspondencia>
</atributo>
</concepto>
<concepto nombre="Elementos Menores">
  <correspondencia>
    <select>DISTINCT Proyectos.`Nombre del proyecto`
    </select>
    <from>Proyectos</from>
    <where>(Proyectos.tipo="fuente") or
(Proyectos.tipo="fuente") or
(Proyectos.tipo="elementos menores")
    </where>
  </correspondencia>
  <atributo nombre="Nombre">
    <correspondencia>
      <where>(Proyectos.`Nombre del proyecto` #op#
'%#valor#%' )
      </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Tipo_ElementosMenores">
    <correspondencia>
      <where>(Proyectos.tipo #op# '%#valor#%') </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Poblacion">
    <correspondencia>
      <where>(Proyectos.Localidad #op# '%#valor#%')</where>
    </correspondencia>
  </atributo>
  <atributo nombre="Provincia">
    <correspondencia>
      <where>(Proyectos.Provincia #op# '%#valor#%')</where>
    </correspondencia>
  </atributo>
  <atributo nombre="Tipo_Datos">

```

```

        <correspondencia>
          <where>(Proyectos.tecnicas #op# '%#valor#%') </where>
        </correspondencia>
      </atributo>
    </concepto>
  </isa>
</concepto>
<concepto nombre="Objetos">
  <correspondencia>
    <select>DISTINCT Proyectos.`Nombre del proyecto` </select>
    <from>Proyectos</from>
    <where>(Proyectos.tipo="imageria")</where>
  </correspondencia>
  <atributo nombre="Nombre">
    <correspondencia>
      <where>(Proyectos.`Nombre del proyecto` #op# '%#valor#%') </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Tipo_Muebles/Objetos">
    <correspondencia>
      <where>(Proyectos.tipo #op# '%#valor#%') </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Poblacion">
    <correspondencia>
      <where>(Proyectos.Localidad #op# '%#valor#%') </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Provincia">
    <correspondencia>
      <where>(Proyectos.Provincia #op# '%#valor#%') </where>
    </correspondencia>
  </atributo>
  <atributo nombre="Tipo_Datos">
    <correspondencia>
      <where>(Proyectos.tecnicas #op# '%#valor#%' ) </where>
    </correspondencia>
  </atributo>
</concepto>
</isa>
</concepto>
</arbol>

```

Figura A.5. Árbol de correspondencias para base de datos LDGP.

### B.3 LENGUAJE CONSULTA

En este apéndice se presenta la especificación formal del Lenguaje de Consulta que hemos definido para comunicar el Mediador con los Sistemas Envoltorios de las bases de datos componentes.

En este Lenguaje de Consulta está definido en XML. A continuación en la figura inferior se muestra el DTD del Lenguaje Consulta.

```
<!ELEMENT consulta (concepto)>
<!ELEMENT concepto (isa | atributo*)>
<!ATTLIST concepto nombre CDATA #REQUIRED>
<!ELEMENT isa (concepto)>
<!ELEMENT atributo (operadorbinario, valor*)>
<!ATTLIST atributo nombre CDATA #REQUIRED>
<!ELEMENT operadorbinario EMPTY>
<!ATTLIST operador-binario op CDATA #REQUIRED>
<!ELEMENT valor EMPTY>
<!ATTLIST valor constante CDATA #REQUIRED>
```

Figura A.6. DTD del Lenguaje Consulta.

## **C. MANUALES DE LA APLICACIÓN**

### **C.1 MANUALES DE MANTENIMIENTO**

Casi tan importante como la aplicación, es entregar al cliente un manual en donde se recojan todas y cada una de las posibles modificaciones que se pueden llevar a cabo en cualquiera de las dos aplicaciones desarrolladas durante el proyecto. Este es precisamente el fin de lo que hemos denominado “Manuales de mantenimiento”. A lo largo de este apartado, se describirán, para cada una de las aplicaciones, todos y cada uno de los posibles cambios que pueden llevarse a cabo, explicando, para cada caso las modificaciones que se deben realizar.

Con el fin de mantener un orden que ayude al lector a comprender mejor el texto, en primer lugar se describirán las posibles modificaciones para la Aplicación de soporte a la base de datos que almacena información sobre el patrimonio, dejando para el final la descripción de los posibles cambio que se pueden llevar a cabo en la Aplicación de búsqueda orientada.

#### **C.1.1 MANUAL DE MANTENIMIENTO-APLICACIÓN DE SOPORTE A LA BASE DE DATOS QUE ALMACENA INFORMACIÓN SOBRE PÁGINAS WEB RELACIONADAS CON EL PATRIMONIO-.**

La aplicación que nos va ayudar a mantener actualizadas las bases de datos del sistema es una aplicación escalable y capaz de adaptarse, en la medida de lo posible, a una serie de cambios que se produzcan en las bases de datos. Con este documento lo que se pretende es instruir al usuario en los cambios que se pueden realizar, y la manera en que éstos deben llevarse a cabo.

Para ello antes de analizar cada uno de los posibles cambios que se pueden realizar, se explicarán aquellos conceptos que se consideran claves para comprender lo que se desarrollará más adelante.

##### **C.1.1.1 BASE DE DATOS CON LAS QUE TRABAJA LA APLICACIÓN**

La aplicación desarrollada actúa sobre dos bases de datos diferentes, la base de datos “sitiosweb” y la base de datos “administradores”. A continuación se explicarán de manera superficial cada una de ellas.

##### **C.1.1.1.2 BASE DE DATOS “SITIOSWEB”**

La base de datos “sitiosweb” almacena todo aquello relativo al patrimonio, es decir, los sitios Webs, las subpáginas Web de cada sitio Web, y los elementos patrimoniales de cada una de las subpáginas Web. Para entendernos mejor definiremos sitio Web como *“Un sitio web (en inglés: website) es un conjunto de páginas web, típicamente comunes a un dominio de Internet o subdominio en la World Wide Web en*

*Internet*” (Wikipedia). Como se puede leer en la definición un sitio Web es un conjunto de páginas Web, y página Web la definimos como “Una página web, también conocida como página de Internet, es una fuente de información adaptada para la World Wide Web (WWW) y accesible mediante un navegador de Internet que normalmente forma parte de un sitio web.” Por lo tanto, un sitio Web tiene un conjunto de páginas Web, y éstas a su vez tienen un conjunto de elementos patrimoniales. En la Figura A.7 podemos observar un esquema visual de lo explicado.

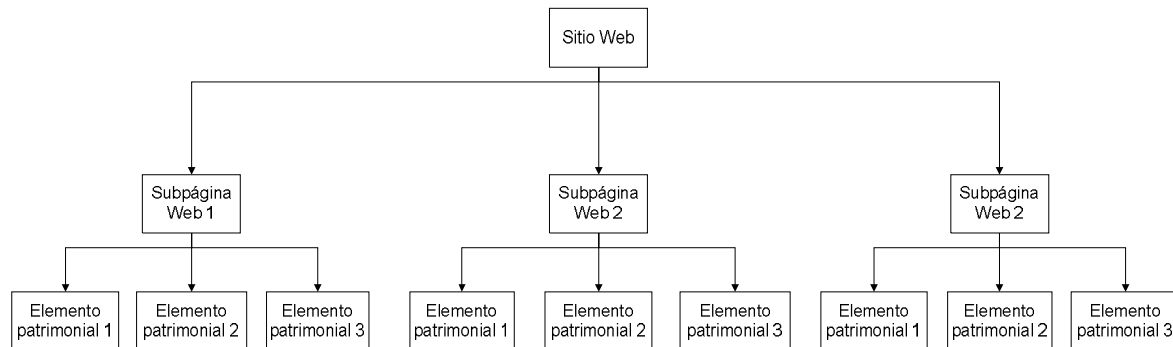


Figura A.7. Esquema de un sitio Web.

Una vez introducidos los conceptos sobre sitio Web y página Web (también denominada subpágina Web), vamos a explicar los elementos de la base de datos.

La base de datos “sitios web” está compuesta por siete tablas, “paginaweb”, “subpaginaweb”, “arquitecturacivil”, “arquitecturareligiosa”, “elementosmenores”, “excavaciones” y “objetos”. Definiremos tabla como Tabla “al tipo de modelado de datos, donde se guardan los datos recogidos por un programa” (wikipedia). Cada una de las tablas tiene un conjunto de campos que la definen, como son por ejemplo para arquitecturacivil: Cod\_patri, Nombre, Periodo...

### C.1.1.1.3 BASE DE DATOS “ADMINISTRADORES”

Se trata de una base de datos que almacena información acerca de los administradores del sistema, y acerca de los movimientos que se producen en el mismo. La base de datos “administradores” está compuesta por dos tablas: “administrador” y “movimientos”.

## C.1.1.2 MODIFICACIONES POSIBLES Y ESCALABILIDAD DE AMBAS BASES DE DATOS

### C.1.1.2.1 BASE DE DATOS “SITIOS WEBS”

Con el fin de aclarar qué modificaciones se pueden llevar a cabo y sobre qué tablas y campos se pueden realizar esas modificaciones, describiremos para cada posible modificación las tablas que se ven afectadas y cuáles son los elementos que deben ser alterados.

#### **C.1.1.2.1.1 CAMBIAR DATOS REFERENTES AL SITIO WEB**

Se refiere a cambiar tanto el nombre de la tabla como sus campos. Esta operación no se podrá llevar a cabo. Independientemente de los cambios que se realicen en la base de datos nunca se podrá modificar ni el nombre de la tabla “paginaweb”, ni el nombre de sus campos “Direccion\_Web”, “Descripción” y “Nombre”. Tampoco se recomienda la eliminación de un campo, ni la inserción de un campo nuevo, ya que supondría un desajuste para la aplicación, lo cual produciría errores importantes.

#### **C.1.1.2.1.2 CAMBIAR DATOS REFERENTES A LA SUBPÁGINA WEB**

Al igual, que en el caso del sitio Web, no se podrán llevar a cabo modificaciones tanto en el nombre de la tabla “subpaginaweb”, como en el nombre de sus campos “Direccion\_Web”, “Descripción” y “Disponible\_en\_Web”... Al igual que en el caso anterior, tampoco es posible la eliminación de alguno de los campos, o la introducción de uno nuevo.

#### **C.1.1.2.1.3 CAMBIAR DATOS REFERENTES AL PATRIMONIO**

En el concepto patrimonio están incluidas las tablas “arquitecturacivil”, “arquitecturareligiosa”, “elementosmenores”, “excavaciones” y “objetos”, cada una con sus respectivos campos. En este caso todas las tablas son perfectamente modificables, al igual que sus campos, excluyendo alguno, que explicaremos más adelante. El sistema lo que lleva a cabo es una búsqueda de las tablas relacionadas con el patrimonio, y en función del tipo de patrimonio que sea, en un principio arquitecturacivil, arquitecturareligiosa, elementosmenores, excavaciones u objetos, nos muestra sus campos. Toda esta información está reunida en un documento XML (datosbd.xml), ubicado en el directorio “AplicacionNormal/PaginasWeb”, en el que se le indica a la aplicación cuáles son las tablas de la base de datos relacionadas con los elemento patrimoniales y los campos de cada tabla.

Por lo tanto las modificaciones que se podrán llevar a cabo son las siguientes:

- **Modificaciones en los nombres de las tablas**

En cuanto al nombre de las tablas todas ellas pueden ser modificadas, es decir, arquitecturacivil puede pasar a llamarse, por ejemplo, elementosciviles. Todas las modificaciones de la base de datos, como en este caso, deberán realizarse directamente sobre ella. Además, de modificar el valor en la base de datos, tal y como se ha comentado anteriormente, existe un documento XML (datosbd.xml), ubicado en el directorio “AplicacionNormal/PaginasWeb”, en el que se le indica a la aplicación cuáles son las tablas de la base de datos relacionadas con los elemento patrimoniales y los campos de cada tabla, que también deberá ser modificado. A modo de ejemplo vamos a mostrar dos partes de dicho documento, en la primera aparecerá la parte inicial del documento y recogerá todo aquello relacionado con la tabla excavaciones, y en la segunda, haciendo hincapié en el ejemplo anterior, podremos observar la información



relacionada con la tabla arquitecturacivil. El estado actual de dicho documento es el siguiente.

```
<?xml version="1.0" encoding="ISO-8859-1" sta
<!DOCTYPE conceptos SYSTEM "DIDdatosbd.dtd">
<!--recordar que los atributos como direccion
<conceptos>
  <concepto nombre="excavaciones">
    <atributo nombre="Nombre">
      <at campo="text"/>
      <sel opcion="si"/>
    </atributo>
    <atributo nombre="Periodo">
      <at campo="text"/>
    </atributo>
    <atributo nombre="Tipo_datos">
      <at campo="text"/>
    </atributo>
    <atributo nombre="Municipio">
      <at campo="text"/>
      <sel opcion="si"/>
    </atributo>
    <atributo nombre="Poblacion">
      <at campo="text"/>
      <sel opcion="si"/>
    </atributo>
    <atributo nombre="Provincia">
      <at campo="text"/>
      <sel opcion="si"/>
    </atributo>
    <atributo nombre="Tipo_excavacion">
      <at campo="text"/>
    </atributo>
    <interno nombre="Direccion_Web"/>
  </concepto>
```

Figura A.8. Parte inicial de datosbd.xml. Muestra la información relacionada con el concepto excavaciones.

```

<concepto nombre="arquitecturacivil">
  <atributo nombre="Nombre">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Periodo">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Tipo_datos">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Municipio">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Poblacion">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Provincia">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Tipo_arquitecturaCivil">
    <at campo="text"/>
  </atributo>
  <interno nombre="Direccion_Web"/>
</concepto>

```

Figura A.9. Parte de datosbd.xml correspondiente a la tabla arquitecturacivil. Muestra la información relacionada con dicho concepto.

Continuando con el ejemplo, vamos a explicar qué pasos habría que seguir para cambiar el nombre de la tabla arquitecturacivil por “elementosciviles”.

El primer paso sería sustituir dicho valor en la base de datos. Este cambio dependerá del sistema gestor de la base de datos en la que esté almacenada la tabla. Por esto, es recomendable ponerse en contacto con algún encargado del mantenimiento de la base de datos, que nos puede indicar los pasos a seguir.

Una vez terminado el paso anterior, habría que realizar el siguiente cambio en el documento “datosbd.xml”. Deberíamos situarnos en la parte del documento donde aparece “<concepto nombre=”arquitecturacivil”>” y sustituir el nombre por el de “elementosciviles”, dando como resultado la siguiente imagen (Figura A.10).

```

<concepto nombre="elementosciviles">
  <atributo nombre="Nombre">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Periodo">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Tipo_datos">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Municipio">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Poblacion">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Provincia">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Tipo_arquitecturaCivil">
    <at campo="text"/>
  </atributo>
  <interno nombre="Direccion_Web"/>
</concepto>

```

Figura A.10. Modificación realizada. Se cambia el nombre de la tabla.

Por último citar que es imprescindible que tanto el nombre que le hemos dado a la tabla en la base de datos como el nombre escrito en el documento XML sean iguales. Para evitar posibles errores es conveniente suprimir los acentos en las palabras.

- **Inserción de una tabla nueva**

El sistema también nos permite introducir nuevas tablas en la base de datos, es decir, a las ya existentes en un principio, “arquitecturacivil”, “arquitecturareligiosa”, “elementosmenores”, “excavaciones” y “objetos”, nos da la posibilidad de añadir otra nueva como por ejemplo, “museos”. Para insertar esta nueva tabla “museos” en el sistema, se deberán llevar a cabo dos procesos.

El primero de ellos será introducir esa tabla con los campos correspondientes en la base de datos. Los campos a introducir (nombre, periodo...) pueden ser los que el usuario cree oportuno, salvo el campo “Direccion\_Web” que debe estar presente siempre, siendo además el campo clave de la tabla, y preferiblemente ocupando el primer lugar de los campos de la misma. Al igual que en el caso de modificación del nombre de una tabla, para mayor seguridad los campos sobre la base de datos deberán ser instruidos por un responsable de la misma.

El segundo proceso que hay que llevar a cabo es plasmar el cambio realizado en la base de datos en el documento XML “datosbd.xml”. Para esto, tendremos que añadir en dicho documento un nuevo elemento concepto con nombre “museos”, y dentro de dicho concepto colocaremos todos los campos (denominados atributos en el documento XML) de dicho concepto, sin olvidar que debe aparecer el campo “Direccion\_Web” como elemento interno dentro del concepto. Para verlo mejor, nos apoyaremos de la siguiente figura.

```

        <atributo nombre="Tipo_arquitecturaCivi
            <at campo="text"/>
        </atributo>
        <interno nombre="Direccion_Web"/>
    </concepto>
    <concepto nombre="museos">
        <atributo nombre="Nombre">
            <at campo="text"/>
        </atributo>
        <atributo nombre="Poblacion">
            <at campo="text"/>
        </atributo>
        <atributo nombre="Provincia">
            <at campo="text"/>
        </atributo>
        <interno nombre="Direccion_Web"/>
    </concepto>

    <concepto nombre="elementosmenores">
        <atributo nombre="Nombre">
            <at campo="text"/>
        </atributo>
        <atributo nombre="Periodo">
            <at campo="text"/>
        </atributo>
    </concepto>

```

Figura A.11. Inserción de un nuevo concepto (tabla) realizada.

Como se aprecia en la Figura A.11, la nueva tabla de la base de datos (concepto para el documento XML) ha sido introducida con los siguientes campos (atributos para el documento XML): Nombre, Población, Provincia, y Direccion\_Web, que es introducido en vez como atributo como “interno”. Es necesario que para cualquier inserción que se lleva a cabo se mantenga este campo de esta forma.

Obviamente los tanto el nombre de la tabla como el de los campos deben coincidir en la base de datos y en el documento XML.

- **Eliminación de una tabla**

El sistema también nos permite eliminar tablas de la base de datos, es decir, nos da la posibilidad de eliminar cualquier tabla de las ya existentes. Si, como ejemplo, queremos eliminar la tabla objeto deberemos, como en los casos anteriores realizar dos procesos.

El primero de ellos será eliminar dicha tabla de la base de datos. Para realizar esta tarea es conveniente ponerse en contacto con un responsable de la misma.

El segundo paso es plasmar esa eliminación en el documento XML, para ello bastará con eliminar el concepto de dicho documento relacionado con la tabla. En el caso del ejemplo, únicamente tendríamos que eliminar la parte seleccionada en la Figura A.12.

```
<atributo nombre="Tipo_elementosMenores">
  <at campo="text"/>
</atributo>
<interno nombre="Direccion_Web"/>
</concepto>
<concepto nombre="objetos">
  <atributo nombre="Nombre">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Periodo">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Tipo_datos">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Municipio">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Poblacion">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Provincia">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Tipo_objeto">
    <at campo="text"/>
  </atributo>
  <interno nombre="Direccion_Web"/>
</concepto>
</conceptos>
```

Figura A.12. Eliminación de concepto (tabla).

- **Modificaciones en los campos de una tabla**

El sistema también se ajusta a posibles cambios sobre campos de una tabla concreta. Supongamos que se quiere modificar uno de los campos de la tabla excavaciones, por ejemplo “Tipo\_Excavacion”, pasándose a llamar únicamente “tipo”. Para llevar a cabo este cambio seguiremos el proceso habitual que hemos seguido en casos anteriores.

En primer lugar, se modificará dicho campo directamente sobre la base de datos. Al igual que en casos anteriores es aconsejable que esta tarea sea llevada a cabo por un encargado de la base de datos.

Una vez llevada a cabo esta modificación, el siguiente paso es reflejarla en el documento XML. Por lo tanto los pasos a seguir son los siguientes: Localizaremos en el documento XML, el concepto “Excavación” que en dicho documento hace referencia a la tabla “Excavación” de la base de datos. Una vez localizado el concepto, buscamos el campo (atributo en el documento) que queremos modificar, en nuestro caso “Tipo\_Excavacion”, y lo único que hacemos es sustituirlo por el nombre que le queramos dar, en este caso “tipo”. La Figura A.13 recoge el cambio que se pretende llevar a cabo.

```

<conceptos>
  <concepto nombre="excavaciones">
    <atributo nombre="Nombre">
      <at campo="text"/>
      <sel opcion="si"/>
    </atributo>
    <atributo nombre="Periodo">
      <at campo="text"/>
    </atributo>
    <atributo nombre="tipo">
      <at campo="text"/>
    </atributo>
    <atributo nombre="Municipio">
      <at campo="text"/>
      <sel opcion="si"/>
    </atributo>
    <atributo nombre="Poblacion">
      <at campo="text"/>
      <sel opcion="si"/>
    </atributo>
    <atributo nombre="Provincia">
      <at campo="text"/>
      <sel opcion="si"/>
    </atributo>
    <atributo nombre="Tipo_excavacion">
      <at campo="text"/>
    </atributo>
    <interno nombre="Direccion_Web"/>
  </concepto>

```

Figura A.13. Cambio del atributo “Tipo\_Excavacion” por “tipo”

- **Inserción de un nuevo campo en una tabla**

Otra de las opciones que nos aporta el sistema es la posibilidad de introducir un nuevo campo en una de las tablas de la base de datos. Para ello, siguiendo el procedimiento de casos anteriores, primero tendremos que introducir dicho campo en la base de datos. Para esto es preferible contar con el encargado del mantenimiento de la misma. En este ejemplo vamos a introducir el campo “País” en la tabla “excavaciones”.

Una vez realizada la modificación en la base de datos, el siguiente paso es actualizar el fichero XML. Para ello localizamos el concepto excavaciones en dicho documento, y añadimos dicho atributo tal y como aparece en la Figura A.14 que se muestra a continuación.

```

<concepto nombre="excavaciones">
  <atributo nombre="Nombre">
    <at campo="text"/>
    <sel opcion="si"/>
  </atributo>
  <atributo nombre="Periodo">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Tipo_datos">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Municipio">
    <at campo="text"/>
    <sel opcion="si"/>
  </atributo>
  <atributo nombre="Poblacion">
    <at campo="text"/>
    <sel opcion="si"/>
  </atributo>
  <atributo nombre="Provincia">
    <at campo="text"/>
    <sel opcion="si"/>
  </atributo>
  <atributo nombre="País">
    <at campo="text"/>
    <sel opcion="si"/>
  </atributo>
  <atributo nombre="Tipo_excavacion">
    <at campo="text"/>
  </atributo>
  <interno nombre="Direccion_Web"/>
</concepto>

```

Figura A.14. Inserción del atributo “País” dentro del concepto “excavaciones”.

Como se aprecia en la figura superior el atributo País ha sido insertado correctamente. Además del nombre, se pueden ver otros sub-atributos del atributo “País” como son <at campo= “text”/> y <sel opcion= “si”/>. Ambos sub-atributos no tienen relación alguna con la información de la base de datos, sin embargo son importantes a la hora de mostrar información en la aplicación que estamos tratando, por lo que serán tratados más adelante.

- **Eliminación de un campo de una tabla**

Otra de las posibilidades que nos permite llevar a cabo el sistema es la eliminación de un campo de una tabla de la base de datos. Por ejemplo la eliminación del campo “Periodo” de la tabla “excavaciones”. Para llevar a cabo esta tarea, necesitaremos completar dos procesos.



El primero de ellos consta de eliminar dicho campo de la base de datos. Para realizar esta tarea es recomendable contar con un responsable del mantenimiento de la base de datos.

El segundo proceso trata de plasmar el cambio realizado en la base de datos, en el documento XML. Para ello localizaremos en dicho documento el concepto relacionado con la tabla “excavaciones”, es decir el concepto “excavaciones”. Una vez localizado nos dispondremos a eliminar el atributo (campo para la base de datos) “Periodo”. Para una correcta eliminación se deberá eliminar aquello marcado en la Figura A.15.

```
<conceptos>
  <concepto nombre="excavaciones">
    <atributo nombre="Nombre">
      <at campo="text"/>
      <sel opcion="si"/>
    </atributo>
    <atributo nombre="Periodo">
      <at campo="text"/>
    </atributo>
    <atributo nombre="Tipo_datos">
      <at campo="text"/>
    </atributo>
    <atributo nombre="Municipio">
      <at campo="text"/>
      <sel opcion="si"/>
    </atributo>
    <atributo nombre="Poblacion">
      <at campo="text"/>
      <sel opcion="si"/>
    </atributo>
    <atributo nombre="Provincia">
      <at campo="text"/>
      <sel opcion="si"/>
    </atributo>
    <atributo nombre="Tipo_excavacion">
      <at campo="text"/>
    </atributo>
    <interno nombre="Direccion_Web"/>
  </concepto>
```

Figura A.15 Eliminación del atributo “Periodo” dentro del concepto “excavaciones”.

#### C.1.1.2.2 BASE DE DATOS “ADMINISTRADORES”

Con el fin de aclarar qué modificaciones se pueden llevar a cabo y sobre qué tablas y campos se pueden realizar esas modificaciones, describiremos para cada

posible modificación las tablas que se ven afectadas y cuáles son los elementos que deben ser alterados.

#### **C.1.1.2.2.1 CAMBIAR DATOS REFERENTES A ADMINISTRADOR**

Se refiere a cambiar tanto el nombre de la tabla como sus campos. Esta operación no se podrá llevar a cabo. Independientemente de los cambios que se realicen en la base de datos nunca se podrá modificar ni el nombre de la tabla “administrador”, ni el nombre de sus campos “Id”, “Nombre” y “Nick”... Tampoco se recomienda la eliminación de un campo, ni la inserción de un campo nuevo, ya que supondría un desajuste para la aplicación, lo cual produciría errores importantes.

#### **C.1.1.2.2.2 CAMBIAR DATOS REFERENTES A MOVIMIENTOS**

Se refiere a cambiar tanto el nombre de la tabla como sus campos. Esta operación no se podrá llevar a cabo. Independientemente de los cambios que se realicen en la base de datos nunca se podrá modificar ni el nombre de la tabla “movimientos”, ni el nombre de sus campos “Codigo\_movimiento”, “Id” y “Accion”... Tampoco se recomienda la eliminación de un campo, ni la inserción de un campo nuevo, ya que supondría un desajuste para la aplicación, lo cual produciría errores importantes.

#### **C.1.1.3 INFORMACIÓN REFERENTE A LAS INTERFACES DE LA APLICACIÓN**

El documento XML además de incluir información importante para mantener una concordancia entre la base de datos y la aplicación, también almacena datos que son relevantes a la hora de mostrar las interfaces de la aplicación al usuario. A lo largo de este apartado se presentan las diferentes interfaces que se basan en la información almacenada en el documento XML para ser construidas.

##### **C.1.1.3.1 NUEVO PATRIMONIO**

En la aplicación existe una opción en el menú principal denominada “Nuevo Patrimonio”. Seleccionando el sitio y la subpágina Web en la que deseamos introducir el nuevo elemento patrimonial, nos aparece la siguiente interfaz (Figura A.16).

NUEVO ELEMENTO PATRIMONIAL

Datos del Elemento Patrimonial

Seleccione el lugar donde se quiere clasificar el contenido:

- excavaciones
- arquitecturareligiosa
- arquitecturacivil
- elementosmenores
- objetos

Nombre:

Periodo:

Tipo\_datos:

Municipio:

Poblacion:

Provincia:

Tipo\_arquitecturaReligiosa:

Figura A.16. Interfaz “Nuevo elemento Patrimonial”.

En la Figura A.17 se puede observar como los atributos que aparecen son exactamente los campos de la tabla arquitecturareligiosa, concepto que ha sido seleccionado a partir del “radiobutton” que se aprecia en la propia imagen. Para comprobar esto observamos el documento XML, en el cual aparece la información asociada a la base de datos.

```

<concepto nombre="arquitecturareligiosa">
  <atributo nombre="Nombre">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Periodo">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Tipo_datos">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Municipio">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Poblacion">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Provincia">
    <at campo="text"/>
  </atributo>
  <atributo nombre="Tipo_arquitecturaReligiosa">
    <at campo="text"/>
  </atributo>
  <interno nombre="Direccion_Web"/>
</concepto>

```

Figura A.17. Información referente al concepto “arquitecturareligiosa” del documento XML.

El único campo de la tabla “arquitecturareligiosa” que no aparece es el campo “Direccion\_Web”, ya que este campo se autoincrementa por sí solo en cada inserción de un elemento patrimonial, por lo que no hay que insertarle ningún valor. Por eso en el documento XML aparece como “interno” en vez como atributo.

Si nos fijamos tanto en la interfaz como en el documento XML observamos que en este último aparece para todos los atributos un sub-atributo <at campo= “text”/>, esto indica que en la interfaz los campos donde se deben rellenar los datos son de tipo texto. Si por algún casual se quiere cambiar el tipo de campo, bastaría con cambiarlo en el documento XML, y asignar para cada atributo el campo correspondiente. Para esto se necesitan nociones básicas de HTML, ya que es necesario conocer los diferentes nombres de los distintos campos que existen para rellenar datos.

### C.1.1.3.2 MODIFICAR PATRIMONIO

Mismo comportamiento que Nuevo patrimonio (Apartado C.1.1.3.1), pero con la siguiente interfaz

Datos del elemento patrimonial	
Nombre:	Santa María
Periodo:	SXX
Tipo_datos:	Descripción
Municipio:	Vitoria-Gasteiz
Poblacion:	Vitoria-Gasteiz
Provincia:	Alava
Tipo_arquitecturaReligiosa:	catedral

Figura A.18. Interfaz “Modificar elemento Patrimonial”.

### C.1.1.3.3 MODIFICAR PATRIMONIO Y ELIMINAR PATRIMONIO

En el menú principal de la aplicación existen dos opciones llamadas “Modificar patrimonio” y “Eliminar patrimonio”. En ambas opciones existe una interfaz que consta de una lista en la que se muestran todos los elementos patrimoniales, de la subpágina Web seleccionada en la lista desplegable anterior, junto con una imagen que hace las funciones de botón, en unos casos para modificar el elemento patrimonial, y en otros casos para eliminarlo, todo en función de la opción seleccionada. A continuación se muestran ambas páginas.

Lista de elementos patrimoniales de <a href="http://www.vc.ehu.es/docarq/LDA/index.htm">http://www.vc.ehu.es/docarq/LDA/index.htm</a>					
Clase	Nombre	Municipio	Poblacion	Provincia	Eliminar
excavaciones	Catedral de Santa María	Vitoria-Gasteiz	Vitoria-Gasteiz	Alava	
excavaciones	Antiguo Instituto	Vitoria-Gasteiz	Vitoria-Gasteiz	Alava	
excavaciones	Santuario de la Encina	Vitoria-Gasteiz	Vitoria-Gasteiz	Alava	
excavaciones	Gracurris	Alfaro	Alfaro	La Rioja	
excavaciones	Castillo de los Monjes	Lumbreras	Lumbreras	La Rioja	
excavaciones	Convento de Valbuena	Logroño	Vitoria	La Rioja	

Figura A.19. Lista de elemento patrimoniales que aparece en el proceso denominado como “Eliminar Patrimonio”.

Lista de elementos patrimoniales de <a href="http://www.vc.ehu.es/docarq/LDA/index.htm">http://www.vc.ehu.es/docarq/LDA/index.htm</a>					
Clase	Nombre	Municipio	Poblacion	Provincia	Modificar
excavaciones	Catedral de Santa María	Vitoria-Gasteiz	Vitoria-Gasteiz	Alava	
excavaciones	Antiguo Instituto	Vitoria-Gasteiz	Vitoria-Gasteiz	Alava	
excavaciones	Santuario de la Encina	Vitoria-Gasteiz	Vitoria-Gasteiz	Alava	
excavaciones	Gracurris	Alfaro	Alfaro	La Rioja	
excavaciones	Castillo de los Monjes	Lumbreras	Lumbreras	La Rioja	
excavaciones	Convento de Valbuena	Logroño	Vitoria	La Rioja	

Figura A.20. Lista de elemento patrimoniales que aparece en el proceso denominado como “Modificar Patrimonio”.

En ambas figuras se puede observar que como cabecera de la lista aparecen una serie de etiquetas, como son “Clase”, “Nombre”, “Municipio”, “Población” y “Provincia”. A continuación se muestra el documento XML al que se ha hecho referencia a lo largo de este apartado.

```

<conceptos>
  <concepto nombre="excavaciones">
    <atributo nombre="Nombre">
      <at campo="text"/>
      <sel opcion="si"/>
    </atributo>
    <atributo nombre="Periodo">
      <at campo="text"/>
    </atributo>
    <atributo nombre="Tipo_datos">
      <at campo="text"/>
    </atributo>
    <atributo nombre="Municipio">
      <at campo="text"/>
      <sel opcion="si"/>
    </atributo>
    <atributo nombre="Poblacion">
      <at campo="text"/>
      <sel opcion="si"/>
    </atributo>
    <atributo nombre="Provincia">
      <at campo="text"/>
      <sel opcion="si"/>
    </atributo>
    <atributo nombre="Tipo_excavacion">
      <at campo="text"/>
    </atributo>
    <interno nombre="Direccion_Web"/>
  </concepto>

```

Figura A.21. Documento XML en el que se muestra el concepto “excavaciones”.

Si analizamos el trozo de documento XML de la Figura A.21 nos damos cuenta que existe un sub-atributo dentro de atributo con la siguiente forma `<sel opción=”si”/>`. Además nos damos cuenta que los atributos que cuentan con ese sub-atributo son exactamente los que aparecen en la cabecera de las listas de las figuras anteriores, además de la etiqueta “Clase” que va aparecer siempre y que nos sirve para diferenciar el elemento patrimonial que estamos tratando.

Para cambiar las etiquetas de las cabeceras que se muestran en las listas bastará con modificar en el documento XML el sub-atributo `<sel opción=”si”/>`, cambiándolo de un atributo a otro.

Importante citar que se han de cambiar sólo dentro del concepto “excavaciones”, ya que el resto asume por defecto los valores aquí escritos (Sólo en el caso de las etiquetas de la cabecera de las listas).

#### C.1.1.4 INFORMACIÓN REFERENTE A LOS PERFILES DE LA APLICACIÓN

En la aplicación existen dos perfiles diferentes: administrador y usuario. El administrador tiene potestad para realizar todas las funcionalidades del sistema, mientras que el usuario tiene funciones limitadas. Una de las limitaciones que se le otorga al usuario es que únicamente tiene potestad para realizar operaciones sobre una serie de páginas que son definidas en un documento XML, denominado “usuarios.xml” situado en el directorio “AplicacionNormal/PaginasWeb”.

Para cada usuario bastará con escribir los sitios Web en los que éste puede realizar algún tipo de operación, para restringir el acceso a operaciones del resto de los sitios Web. En el ejemplo de la imagen siguiente se ve como el usuario “esti” sólo tiene potestad para realizar operaciones sobre el sitio web “http://www.alava.net/default\_c.asp”

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE usuarios SYSTEM "DTDusuarios.dtd">
<usuarios>
  <nick nombre="esti">
    <pagina direccion="http://www.alava.net/default_c.asp"/>
  </nick>
</usuarios>
```

Figura A.22. Documento “usuarios.xml”.

Para añadir usuarios bastará con introducir el nombre de otro usuario y el sitio Web en el que se le permite llevar a cabo operaciones. En la siguiente figura (Figura A.23) se ve un ejemplo de un nuevo usuario.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE usuarios SYSTEM "DTDusuarios.dtd">
<usuarios>
  <nick nombre="esti">
    <pagina direccion="http://www.alava.net/default_c.asp"/>
  </nick>
  <nick nombre="marta">
    <pagina direccion="http://www.todopatrimonio.net"/>
  </nick>
</usuarios>
```

Figura A.23. Documento “usuarios.xml”. Nuevo usuario introducido.

Si queremos que un usuario tenga acceso a más de un sitio Web bastará con añadirle un nuevo atributo página al concepto Nick, como por ejemplo:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE usuarios SYSTEM "DTDusuarios.dtd">
<usuarios>
  <nick nombre="esti">
    <pagina direccion="http://www.alava.net/default_c.asp"/>
  </nick>
  <nick nombre="marta">
    <pagina direccion="http://www.todopatrimonio.net"/>
    <pagina direccion="http://www.arabarri.es"/>
  </nick>
</usuarios>
```

Figura A.24. Documento “usuarios.xml”. Nueva página asociada a usuario.



## **C.1.2 MODIFICACIONES DE LA APLICACIÓN DE BÚSQUEDA ORIENTADA**

Lo que se pretende explicar a lo largo de este apartado, son las posibilidades de cambio que presenta la aplicación de búsqueda orientada desarrollada. La mayoría de las posibles modificaciones de la aplicación están relacionadas con la información asociada al Árbol de Conceptos. Con el fin de facilitar la comprensión del Árbol de Concepto, en forma de resumen, explicaremos las funciones del mismo. En los apartados de “diseño de la aplicación”, e “Implementación de la aplicación” aparece una explicación más exhaustiva.

### **C.1.2.1 ÁRBOL DE CONCEPTOS**

El árbol de Conceptos es el módulo en el que se basa la arquitectura de la aplicación. Se trata de un documento XML, que reúne información sobre los conceptos, relaciones existentes entre dichos conceptos y atributos del sistema, además incluye un identificador para cada uno de los elementos anteriores con el cual se corresponden con una de las interfaces del almacén de componentes IU. En el árbol de Conceptos también aparece para cada atributo uno o varios identificadores que sirven para expresar las bases de datos en las que dichos atributos están presentes.

A continuación, mediante la Figura A.25, se pueden apreciar la información comentada en el párrafo anterior.

```

<!-- Inicio Patrimonio -->
<concepto nombre="Patrimonio">
  <atributo nombre="Nombre">
    <IU id="1" />
    <bd id="1" /><bd id="2" />
  </atributo>
  <atributo nombre="Periodo">
    <IU id="3" />
    <bd id="1" />
    <valor val="Romano"/>
    <valor val="SXV"/>
    <valor val="SXVI"/>
    <valor val="SXVII"/>
    <valor val="SXVIII"/>
    <valor val="SXIX"/>
    <valor val="SXX"/>
  </atributo>
  <atributo nombre="Poblacion">
    <IU id="2" />
    <bd id="1" /> <bd id="2" />
  </atributo>
  <atributo nombre="Municipio">
    <IU id="2" />
    <bd id="1" />
  </atributo>
  <atributo nombre="Provincia">
    <IU id="2" />
    <bd id="1" /> <bd id="2" />
  </atributo>
  <atributo nombre="Tipo_Datos">
    <IU id="2" />
    <bd id="1" /> <bd id="2" />
    <valor val="descripcion"/>
    <valor val="imagenes"/>
    <valor val="fotogrametria"/>
    <valor val="topografia"/>
  </atributo>
</concepto>
<!-- Inicio de Excavaciones (isa de Patrimonio) -->
<IU id="Radiol"/>

```

Figura A.25. Una parte del Árbol de Conceptos.

En la figura superior podemos observar una parte del Árbol de Conceptos, concretamente la parte relativa al concepto “Patrimonio”. En ella aparecen todos los atributos del concepto “Patrimonio” como son “Nombre”, “Población”... (Podemos ver el Árbol de Conceptos completo en el anexo B.1.2). Además redondeado en amarillo, podemos observar como cada atributo, al igual que la relación de Generalización/Especialización (expresada como “isa” en el Árbol de Conceptos), tiene asociado un identificador (expresada como “id” en el Árbol de Conceptos) que se corresponde con una interfaz del almacén de Componentes IU. Tal y como aparece en la figura, redondeado en color negro, para cada atributo se especifica a través del atributo id del concepto “bd” (<bd id=“1”/>) las bases de datos en las cuales aparece dicho

atributo. Obviamente en algún documento deberá asociarse a cada base de datos un identificador, pero eso ya se verá más adelante.

Por último, el árbol de Conceptos nos permite asociar a cada atributo una serie de valores. Por ejemplo en el caso del “Periodo” se le han asociado los valores “Romano”, “Siglo XV”, “Siglo XVI”... Esto nos va a ser útil, para mostrar al usuario en la interfaz correspondiente, una lista desplegable con estos valores, lo cual va a forzar al propio usuario a escoger uno de los valores de la lista.

### **C.1.2.2 MODIFICACIÓN DEL ÁRBOL DE CONCEPTOS**

**Nota:** A lo largo de todas las operaciones que se describen en los siguientes sub apartados, es importante prescindir de los acentos, ya que éstos darán problemas al sistema.

#### **C.1.2.2.1 MODIFICACIÓN DE LA ESTRUCTURA DEL ÁRBOL DE CONCEPTOS (AÑADIR/ELIMINAR CONCEPTOS, ATRIBUTOS, RELACIONES DE GENERALIZACIÓN/ESPECIALIZACIÓN ENTRE CONCEPTOS)**

El sistema es capaz de adaptarse a cualquier Árbol de Conceptos que mantenga la estructura del DTD de dicho árbol (ver anexo B.1.1). Es indispensable conocer el DTD para desarrollar un nuevo Árbol de Conceptos, ya que un Árbol de Conceptos no válido implicará un fallo sistemático en la aplicación. Todos los cambios que conlleven la modificación de la estructura del Árbol de Conceptos supondrán un cambio en los diferentes Árboles de Correspondencia de cada uno de los sistema envoltorio en los cuales se encuentran las bases de datos del sistema. Además de la modificación de la estructura del Árbol existen otras operaciones de cambio que pueden ser llevadas a cabo. Se trata de operaciones sencillas, en las que el encargado de llevarlas a cabo no tiene porqué conocer la estructura del DTD, ya que se trata de cambios intuitivos.

#### **C.1.2.2.2 MODIFICACIÓN DEL NOMBRE DE UN CONCEPTO**

Es común que en una clasificación tan personal como es la del patrimonio histórico, existan conceptos cuyos nombres se adapten más a la clasificación realizada que otros ya existentes. Por esto, es imprescindible facilitar al usuario un posible cambio de este tipo. Para llevarlo a cabo es suficiente con sustituir el nombre actual por el que queramos añadir. A continuación se muestra la Figura A.26, en donde se marca la parte que se debería sustituir por otro nombre. En este caso se cambia el concepto “objetos”.

```

<concepto nombre="Objetos">
<atributo nombre="Nombre">
<IU id="1" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Tipo_Muebles/Objetos">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Periodo">
<IU id="3" />
<bd id="1" />
<valor val="Romano"/>
<valor val="SXV"/>
<valor val="SXVI"/>
<valor val="SXVII"/>
<valor val="SXVIII"/>
<valor val="SXIX"/>
<valor val="SXX"/>
</atributo>
<atributo nombre="Poblacion">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Municipio">
<IU id="2" />
<bd id="1" />
</atributo>

```

Figura A.26. Posible modificación del nombre del concepto “objetos”.

**Importante:** Destacar que todo cambio en el Árbol de Conceptos tiene consecuencias en el sistema, ya que dicho cambio no está reflejado en los Árboles de Correspondencias de los diferentes sistemas envoltorio en las cuales se encuentran las bases de datos. Esto es debido a que cada concepto (representado por su nombre) tiene asociada una serie de información en el Árbol de Correspondencias. Para que este cambio se produjera correctamente a todos sus efectos se debería asociar para el nuevo concepto una serie de información en dicho árbol. Esto sería adecuado que lo llevara a cabo una persona con conocimientos en la materia.

### C.1.2.2.3 MODIFICACIÓN DEL NOMBRE DE UN CONCEPTO

Además de la modificación del nombre de un concepto, es habitual, que los atributos cambien su nombre. Al igual que en el caso de los conceptos bastará con sustituir el nombre de la atributo en el Árbol de Conceptos. Veamos un ejemplo en la siguiente figura (Figura A.27), en donde el atributo “Población”, que está marcado, va a ser sustituido.

```

<concepto nombre="Objetos">
<atributo nombre="Nombre">
<IU id="1" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Tipo_Muebles/Objetos">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Periodo">
<IU id="3" />
<bd id="1" />
<valor val="Romano"/>
<valor val="SXV"/>
<valor val="SXVI"/>
<valor val="SXVII"/>
<valor val="SXVIII"/>
<valor val="SXIX"/>
<valor val="SXX"/>
</atributo>
<atributo nombre="Poblacion">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>

```

Figura A.27. El nombre del atributo “Población” va a ser sustituido.

**Importante:** Destacar que todo cambio en el Árbol de Conceptos tiene consecuencias en el sistema ya que dicho cambio no está reflejado en los Árboles de Correspondencias de los diferentes sistemas envoltorio en las cuales se encuentran las bases de datos. Esto es debido a que cada atributo (representado por su nombre) tiene asociada una serie de información en el Árbol de Correspondencias. Para que este cambio se produjera correctamente a todos sus efectos se debería asociar para el nuevo atributo una serie de información en dicho árbol. Esto sería adecuado que lo llevara a cabo una persona con conocimientos en la materia.

#### C.1.2.2.4 MODIFICACIÓN DE LOS POSIBLES VALORES DE UN ATRIBUTO

Como ya se ha comentado anteriormente, se puede otorgar una serie de valores a un atributo y forzar a que el usuario seleccione uno de ellos. Esto es lo que ocurre actualmente con los atributos “Periodo” y “Tipo\_Datos”.

**Nota:** En el atributo “Periodo” es necesario que los valores mantengan una ordenación en la lista, siendo el periodo más antiguo el primer elemento de la lista y el periodo más reciente el último de la misma.

```

5 <concepto nombre="Patrimonio">
6 <atributo nombre="Nombre">
7 <IU id="1" />
8 <bd id="1" /><bd id="2" />
9 </atributo>
10 <atributo nombre="Periodo">
11 <IU id="3" />
12 <bd id="1" />
13 <valor val="Romano"/>
14 <valor val="SXV"/>
15 <valor val="SXVI"/>
16 <valor val="SXVII"/>
17 <valor val="SXVIII"/>
18 <valor val="SXIX"/>
19 <valor val="SXX"/>
20 </atributo>
21 <atributo nombre="Poblacion">
22 <IU id="2" />
23 <bd id="1" /> <bd id="2" />
24 </atributo>
25 <atributo nombre="Municipio">
26 <IU id="2" />
27 <bd id="1" />
28 </atributo>
29 <atributo nombre="Provincia">
30 <IU id="2" />
31 <bd id="1" /> <bd id="2" />
32 </atributo>
33 <atributo nombre="Tipo_Datos">
34 <IU id="2" />
35 <bd id="1" /> <bd id="2" />
36 <valor val="descripcion"/>
37 <valor val="imagenes"/>
38 <valor val="fotogrametria"/>
39 <valor val="topografia"/>
40 </atributo>

```

Figura A.28. Posibles valores otorgados a los atributos Periodo y Tipo\_Datos.

Estos valores se verán reflejados en la aplicación cuando el usuario elija para restringir la búsqueda uno de estos dos atributos. En el ejemplo que se muestra a continuación, se trata de la interfaz correspondiente al atributo “Periodo” (Figura A.29).

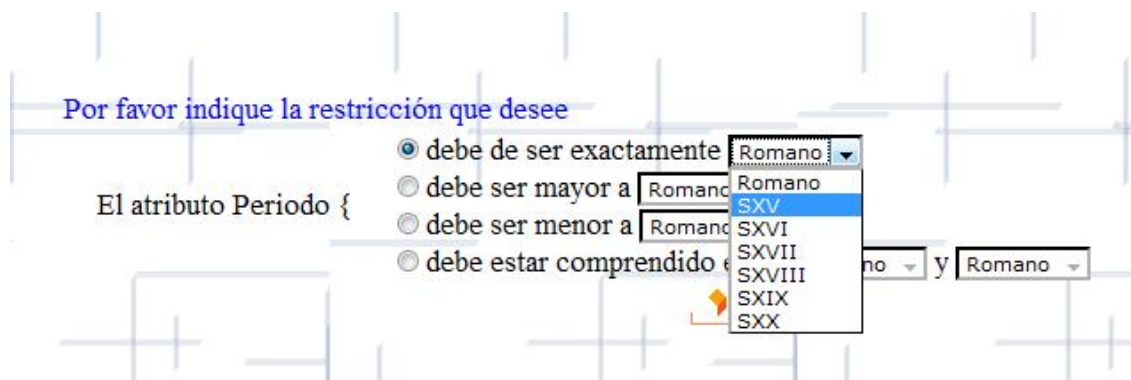


Figura A.29. Resultado de los valores otorgados a un atributo en la aplicación.

Para cambiar los valores asociados a un atributo bastará con modificar el “val”, de la etiqueta valor, de uno de los valores por el valor nuevo. En la Figura A.30 se marca lo que se debería sustituir.

```

<atributo nombre="Periodo">
<IU id="3" />
<bd id="1" />
<valor val="Romano"/>
<valor val="SXV"/>
<valor val="SXVI"/>
<valor val="SXVII"/>
<valor val="SXVIII"/>
<valor val="SXIX"/>
<valor val="SXX"/>
</atributo>
<atributo nombre="Poblacion">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Municipio">
<IU id="2" />
<bd id="1" />
</atributo>

```

Figura A.30. Cambiando uno de los posibles valores del atributo.

Siguiendo el ejemplo, para modificar el valor “Romano” del atributo Periodo bastaría con sustituir el nuevo valor por lo marcado en la Figura A.30.

#### C.1.2.2.5 AÑADIR UN POSIBLE VALOR A UN ATRIBUTO

Además de la modificación, el sistema nos permite insertar un nuevo posible valor para que éste aparezca posteriormente en la aplicación. El procedimiento es sencillo, buscaríamos primero el concepto en el cual se encuentra el atributo en el cual vamos a añadir un posible valor. Por ejemplo añadiríamos un nuevo valor en el atributo “Periodo” del concepto “Excavaciones”. Es habitual que el nuevo valor que se introduce

en un atributo, en este caso “Periodo”, se introduzca para todos los atributos, “Periodo” en el ejemplo, de cada uno de los conceptos del Árbol de Conceptos. Sin embargo, a modo de ejemplo, sólo lo vamos a hacer en dicho atributo en el concepto “Excavaciones”.

```
<atributo nombre="Periodo">
<IU id="3" />
<bd id="1" />
<valor val="Pre-Romano"/>
<valor val="Romano"/>
<valor val="SXV"/>
<valor val="SXVI"/>
<valor val="SXVII"/>
<valor val="SXVIII"/>
<valor val="SXIX"/>
<valor val="SXX"/>
</atributo>
<atributo nombre="Poblacion">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Municipio">
<IU id="2" />
<bd id="1" />
</atributo>
```

Figura A.31. Se añade un posible valor al atributo “Periodo”

En la figura superior podemos observar (está marcado) como se ha introducido un nuevo valor en el atributo “Periodo”. Dicho valor una vez introducido aparecerá sistemáticamente cuando el usuario desee restringir la búsqueda sobre el concepto “Excavaciones” mediante el atributo “Periodo.”

#### C.1.2.2.6 ELIMINAR UN POSIBLE VALOR DE UN ATRIBUTO

Se trata del mismo proceso que el anterior sólo que en sentido inverso. Ahora para eliminar un valor asociado a un atributo bastará con buscar el concepto en el cual se encuentra el atributo (Como en el paso anterior lo más usual es eliminar el valor para todos los atributo con el mismo nombre, es decir todos los atributos “Periodo” del Árbol de Conceptos), y en el atributo que queremos eliminar el valor, eliminamos dicho valor. El borrado del valor debe incluir lo marcado en la Figura A.32.



```

<atributo nombre="Periodo">
<IU id="3" />
<bd id="1" />
<valor val="Romano"/>
<valor val="SXV"/>
<valor val="SXVI"/>
<valor val="SXVII"/>
<valor val="SXVIII"/>
<valor val="SXIX"/>
<valor val="SXX"/>
</atributo>
<atributo nombre="Poblacion">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Municipio">
<IU id="2" />
<bd id="1" />
</atributo>

```

Figura A.32. Eliminar uno de los posibles valores del atributo.

### C.1.2.2.7 MODIFICACIÓN EN LAS BASES DE DATOS EN LAS QUE APARECE UN ATRIBUTO

En el Árbol de Conceptos para cada atributo se le asocia uno o varios identificadores de bases de datos. Estos identificadores expresan las bases de datos en las que dicho atributo está presente. A modo de ejemplo veamos lo siguiente. En el sistema desarrollado, en un principio, existen dos bases de datos. A una de ellas se le asocia el identificador “1”, y a la otra el identificador “2”. Si tomamos, como ejemplo, los atributos “Población” y “Municipio” (Figura A.33) observamos que el atributo “Población” está presente en ambas bases de datos, sin embargo, el atributo “Municipio” únicamente está presente en la segunda.

```

<atributo nombre="Poblacion">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Municipio">
<IU id="2" />
<bd id="1" />
</atributo>
<atributo nombre="Provincia">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>

```

Figura A.33. Bases de datos en las que aparecen los atributos “Población” y “Municipio”.

Para realizar cualquier tipo de modificación sobre las bases de datos en las que están presentes cualquiera de los atributos del Árbol de Conceptos, ya sea modificar un identificador, añadir un identificador de una nueva base de datos, o eliminar uno,

bastará con, cambiar el “id”, añadir `<bd id="id de la nueva base de datos"/>`, o eliminar lo que está subrayado respectivamente. El identificador de la base de datos será explicado más adelante.

#### C.1.2.2.8 MODIFICACIÓN DE LAS INTERFACES DE USUARIO

El Árbol de Conceptos puede tener asociado para cada concepto, atributo, o relación de Generalización/Especialización una interfaz de usuario determinada. Para el caso concreto de nuestra aplicación, salvo en el caso de los conceptos, tanto las relaciones de Generalización/Especialización como los atributos tienen asociado un identificador relacionado con un elemento del almacén de componentes IU. Para modificar un identificador, añadir uno nuevo, o eliminar uno existente, bastará con modificar, eliminar o añadir, dependiendo de lo que se quiera hacer, la información que aparece marcada en la Figura A.34 o en la Figura A.35.

```
<atributo nombre="Municipio">
<IU id="2" />
<bd id="1" />
</atributo>
<atributo nombre="Provincia">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
```

Figura A.34. “Atributo Municipio”. Identificador asociado a una interfaz existente en el almacén de Componentes IU.

```
<isa>
<!-- Inicio de Excavaciones (isa de Patrimonio) -->
<IU id="Radiol"/>
<concepto nombre="Excavaciones">
<atributo nombre="Nombre">
<IU id="1" />
<bd id="1" /> <bd id="2" />
</atributo>
<atributo nombre="Tipo_Excavacion">
<IU id="2" />
<bd id="1" /> <bd id="2" />
</atributo>
```

Figura A.35. “Relación de Generalización/Especialización”. Identificador asociado a una interfaz existente en el almacén de Componentes IU.

Este almacén de componentes se encuentra en el directorio AplicacionGeneral/Aplicación y el nombre del fichero es “funcionesbdComponentes.php3”. En este fichero se almacenan las diferentes interfaces que son mostradas al usuario. En el caso de que existan cambios en la estructura de Árbol de Conceptos, las interfaces pueden no ser útiles, por lo tanto, si no se desea desarrollar unas nuevas, es necesario eliminar los identificadores de las relaciones de

Generalización/Especialización (Relación isa en el Árbol de Conceptos). El sistema es capaz de mostrar, gracias a un esqueleto de frase de Lenguaje Natural Acotado, todos los conceptos del Árbol de Conceptos a pesar de que éstos no tengan asociado una interfaz para ello. Las interfaces de los atributos es necesario mantenerlas, ya que sino éstos no podrán ser mostrados en la aplicación, por lo tanto todo atributo nuevo necesita tener asociado una interfaz. Las interfaces no son específicamente para un atributo concreto sino que cualquier atributo puede ser mostrado por cualquiera de los tres tipos de interfaces que actualmente existen en la aplicación. Por último citar, que salvo el diseño, la información que acompaña a cada interfaz (campos ocultos) debe ser también enviada. Esto es importante tenerlo en cuenta si se desarrollan nuevas interfaces.

### C.1.2.3 MODIFICACIÓN DEL PROCESO DE CONSULTA

Para facilitar la búsqueda de aquello que solicita el usuario, el sistema cuenta con un documento XML para cada sistema envoltorio, en el que aparecen sinónimos de posibles palabras introducidas por el usuario. El documento está situado dentro de cada sistema envoltorio, tiene como nombre “sinónimos.xml” y su apariencia es la que muestra la Figura A.36.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE parecidos SYSTEM "DIDSinonimos.dtd">
<parecidos>
  <atributo nombre="descripcion">
    <sinonimo nombre="escrito"/>
  </atributo>
  <atributo nombre="imagenes">
    <sinonimo nombre="fotos"/>
    <sinonimo nombre="foto"/>
    <sinonimo nombre="imagen"/>
    <sinonimo nombre="fotogrametria"/>
  </atributo>
  <atributo nombre="fotogrametria">
  </atributo>
  <atributo nombre="topografia">
  </atributo>
</parecidos>
```

Figura A.36. Apariencia del documento “sinónimos.xml”

Este documento es muy útil para conseguir una búsqueda más consistente. A continuación se mostrarán las diferentes operaciones que pueden llevarse a cabo en él.

**Nota:** Tanto para este apartado como para los sub-apartados del mismo es importante prescindir de los acentos, ya que éstos darán problemas al sistema.

#### C.1.2.3.1 NUEVO SINÓNIMO DE UNA PALABRA

Para añadir un sinónimo a una palabra ya existente en el documento XML, bastará con introducir lo siguiente: <sinónimo nombre= “sinónimo que se desea

introducir"/>, en la figura siguiente se observa, subrayado, un nuevo sinónimo de la palabra “imágenes”.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE parecidos SYSTEM "DTDSinonimos.dtd">
<parecidos>
  <atributo nombre="descripcion">
    <sinonimo nombre="escrito"/>
  </atributo>
  <atributo nombre="imagenes">
    <sinonimo nombre="fotos"/>
    <sinonimo nombre="foto"/>
    <sinonimo nombre="imagen"/>
    <sinonimo nombre="fotogrametria"/>
    <sinonimo nombre="fotografia"/>
  </atributo>
  <atributo nombre="fotogrametria">
  </atributo>
  <atributo nombre="topografia">
  </atributo>
</parecidos>
```

Figura A.37. Nuevo sinónimo.

### C.1.2.3.2 MODIFICACIÓN DE UN SINÓNIMO DE UNA PALABRA

Para modificar un sinónimo de una de las palabras, basta con identificar dicha palabra y modificar el contenido (donde pone “nombre”) del sinónimo en el cual se quiera desarrollar el cambio. En la Figura A.38 se puede ver marcado aquello que habría que modificar para llevar a cabo el cambio que aquí se pretende.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE parecidos SYSTEM "DTDSinonimos.dtd">
<parecidos>
  <atributo nombre="descripcion">
    <sinonimo nombre="escrito"/>
  </atributo>
  <atributo nombre="imagenes">
    <sinonimo nombre="fotos"/>
    <sinonimo nombre="foto"/>
    <sinonimo nombre="imagen"/>
    <sinonimo nombre="fotogrametria"/>
  </atributo>
  <atributo nombre="fotogrametria">
  </atributo>
  <atributo nombre="topografia">
  </atributo>
</parecidos>
```

Figura A.38. Modificación de un sinónimo.

### C.1.2.3.3 ELIMINACIÓN DE UN SINÓNIMO DE UNA PALABRA

En contraposición al proceso de añadir un nuevo sinónimo a una palabra, en este caso lo único que se deberá realizar es eliminar el sinónimo de la palabra que se desee. Para ello bastará con eliminar lo marcado en la Figura A.39.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE parecidos SYSTEM "DTDSinonimos.dtd">
<parecidos>
  <atributo nombre="descripcion">
    <sinonimo nombre="escrito"/>
  </atributo>
  <atributo nombre="imagenes">
    <sinonimo nombre="fotos"/>
    <sinonimo nombre="foto"/>
    <sinonimo nombre="imagen"/>
    <sinonimo nombre="fotogrametria"/>
    <sinonimo nombre="fotografia"/>
  </atributo>
  <atributo nombre="fotogrametria">
  </atributo>
  <atributo nombre="topografia">
  </atributo>
</parecidos>
```

Figura A.39. Eliminación de un sinónimo de una palabra.

### C.1.2.3.4 NUEVA PALABRA EN EL DOCUMENTO XML

En el caso de que necesitemos añadir una nueva palabra en el documento XML, bastará con añadir en el documento, dentro de las etiquetas <parecidos> y </parecidos>, lo que se muestra subrayado en la Figura A.40.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE parecidos SYSTEM "DTDSinonimos.dtd">
<parecidos>
  <atributo nombre="descripcion">
    <sinonimo nombre="escrito"/>
  </atributo>
  <atributo nombre="imagenes">
    <sinonimo nombre="fotos"/>
    <sinonimo nombre="foto"/>
    <sinonimo nombre="imagen"/>
    <sinonimo nombre="fotogrametria"/>
  </atributo>
  <atributo nombre="fotogrametria">
  </atributo>
  <atributo nombre="topografia">
  </atributo>
</parecidos>
```

Figura A.40. Nueva palabra en el documento XML.

Todos los sinónimos que se quieran incluir en la nueva palabra, han de estar entre las etiquetas marcadas.

#### C.1.2.3.5 MODIFICACIÓN DE UNA PALABRA EN EL DOCUMENTO XML

Modificar una palabra significa modificar su “nombre”. Bastaría con modificar lo que en la Figura A.41 aparece marcado.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE parecidos SYSTEM "DTDSinonimos.dtd">
<parecidos>
  <atributo nombre="descripcion">
    <sinonimo nombre="escrito"/>
  </atributo>
  <atributo nombre="imagenes">
    <sinonimo nombre="fotos"/>
    <sinonimo nombre="foto"/>
    <sinonimo nombre="imagen"/>
    <sinonimo nombre="fotogrametria"/>
  </atributo>
  <atributo nombre="fotogrametria">
  </atributo>
  <atributo nombre="topografia">
  </atributo>
</parecidos>
```

Figura A.41. Modificación de una palabra en el documento XML.

#### C.1.2.3.6 ELIMINACIÓN DE UNA PALABRA DEL DOCUMENTO XML

Para eliminar una palabra del documento XML, vale con eliminar lo que aparece marcado en la Figura A.42. Citar que si una palabra que se quiere eliminar contiene sinónimos, éstos deberán de ser también eliminados.

```

<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE parecidos SYSTEM "DTDSinonimos.dtd">
<parecidos>
  <atributo nombre="descripcion">
    <sinonimo nombre="escrito"/>
  </atributo>
  <atributo nombre="imagenes">
    <sinonimo nombre="fotos"/>
    <sinonimo nombre="foto"/>
    <sinonimo nombre="imagen"/>
    <sinonimo nombre="fotogrametria"/>
  </atributo>
  <atributo nombre="fotogrametria">
  </atributo>
  <atributo nombre="topografia">
  </atributo>
</parecidos>

```

Figura A.42. Eliminación de una palabra en el documento XML.

```

<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE parecidos SYSTEM "DTDSinonimos.dtd">
<parecidos>
  <atributo nombre="descripcion">
    <sinonimo nombre="escrito"/>
  </atributo>
  <atributo nombre="imagenes">
    <sinonimo nombre="fotos"/>
    <sinonimo nombre="foto"/>
    <sinonimo nombre="imagen"/>
    <sinonimo nombre="fotogrametria"/>
  </atributo>
  <atributo nombre="fotogrametria">
  </atributo>
  <atributo nombre="topografia">
  </atributo>
</parecidos>

```

Figura A.43. Eliminación de una palabra junto con sus sinónimos en el documento XML

#### C.1.2.4 MODIFICACIÓN DE LA PRESENTACIÓN DE LOS RESULTADOS

Cuando el sistema muestra los resultados de la consulta al usuario, lo primero que le muestra es una página en la que se describen las bases de datos que han participado en el proceso de consulta. Ésta información es obtenida por el sistema a partir de otro documento XML, ubicado en el directorio AplicacionGeneral/Aplicación y cuyo nombre es “BDsComponentes.xml”. Este documento recoge información como, el nombre de la organización a la que pertenece la base de datos, la dirección de la página Web donde puede encontrarse información sobre esa base de datos, una descripción sobre la información que recoge dicha base de datos y una dirección que es utilizada por el sistema para acceder al módulo traductor de cada uno de los sistemas

envoltorio donde se encuentran las bases de datos. En la Figura A.44 se puede observar la apariencia del documento XML comentado.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE basesdedatos SYSTEM "DTDBDsComponentes.dtd">
<basesdedatos>
  <bd id="1">
    <nombre nombre="Paginas Web"/>
    <direccion paginaWeb=""/>
    <informacion descripcion="Recoge toda la información existente en diferentes sitios Web"/>
    <ruta direccion="../BDWebs/algoritmoTraductorConsulta2.php3"/>
  </bd>
  <bd id="2">
    <nombre nombre="LDGP"/>
    <direccion paginaWeb="http://web.alava.net/PatrimonioHistorico/patrimonio/patrimonihome.htm"/>
    <informacion descripcion="Recoge la información de los diferentes trabajos de investigación llevados a cabo por el Laboratorio de Documentación Geométrica del Patrimonio"/>
    <ruta direccion="../BDLDGP/algoritmoTraductorConsulta2.php3"/>
  </bd>
</basesdedatos>
```

Figura A.44. Documento “BDsComponentes.xml”.

Las únicas modificaciones que se deben realizar en este documento son las relativas al nombre de la base de datos, a la página Web donde existe información acerca de la misma y a la descripción de la base de datos. Para modificar cada uno de estos campos bastará con cambiar la información que aparece dentro de las etiquetas <nombre/>, <dirección/> e <información/> respectivamente. A modo de ejemplo, en la Figura A.45 se indica la información que deberá ser modificada (aparece marcada) para el caso concreto de la descripción de la base de datos con identificador (id) “1”.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE basesdedatos SYSTEM "DTDBDsComponentes.dtd">
<basesdedatos>
  <bd id="1">
    <nombre nombre="Paginas Web"/>
    <direccion paginaWeb=""/>
    <informacion descripcion="Recoge toda la información existente en diferentes sitios Web"/>
    <ruta direccion="../BDWebs/algoritmoTraductorConsulta2.php3"/>
  </bd>
  <bd id="2">
    <nombre nombre="LDGP"/>
    <direccion paginaWeb="http://web.alava.net/PatrimonioHistorico/patrimonio/patrimonihome.htm"/>
    <informacion descripcion="Recoge la información de los diferentes trabajos de investigación llevados a cabo por el Laboratorio de Documentación Geométrica del Patrimonio"/>
    <ruta direccion="../BDLDGP/algoritmoTraductorConsulta2.php3"/>
  </bd>
</basesdedatos>
```

Figura A.45. Modificación de la descripción de la base de datos con id= “1”.

### C.1.2.5 PROCESO DE INSERCIÓN DE UNA NUEVA BASE DE DATOS EN EL SISTEMA

Una de las mayores ventajas de esta aplicación es la escalabilidad que tiene, ya que el sistema nos permite incorporar nuevas bases de datos al mismo sin problema alguno. El sistema actualmente cuenta con dos bases de datos, a modo de ejemplo vamos a suponer a lo largo de este apartado que vamos a introducir una nueva base de datos llamada “catedralDeSantaMariaBD”. El proceso sería el siguiente.



Para explicar el proceso vamos a recorrer cada uno de los módulos en los que se precisa realizar alguna modificación en el orden en el que estos aparecen en el proceso de consulta.

### C.1.2.5.1 MODIFICACIÓN DEL FICHERO “BDsComponentes.xml”

El primer módulo en el que aparece información interesante para el proceso que se está llevando a cabo es el módulo Constructor y Distribuidor de Consultas. Este módulo está ubicado en el directorio AplicacionGeneral/Aplicación y su nombre es “ConstructorYDistribuidosDeConsultas5.php3”. Este módulo utiliza el documento XML “BDsComponentes.xml” para coger la información relativa a las bases de datos del sistema. En este momento sólo aparecen, en dicho documento, las dos fuentes de datos del principio, aquella que almacena información sobre las páginas Web, y la base de datos del Laboratorio de Documentación Geométrica del Patrimonio de la Universidad del País Vasco (UPV/EHU). Por tanto el primer paso es inscribir la nueva base de datos en el documento XML. Siguiendo con el ejemplo en la siguiente imagen se recoge la inscripción de la base de datos “catedralDeSantaMariaBD”.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE basesdedatos SYSTEM "DTDBsComponentes.dtd">
<basesdedatos>
  <bd id="1">
    <nombre nombre="Paginas Web"/>
    <direccion paginaWeb=""/>
    <informacion descripcion="Recoge toda la información existente en diferentes sitios Web"/>
    <ruta direccion="../../../BDWebs/algorithmTraductorConsulta2.php3"/>
  </bd>
  <bd id="2">
    <nombre nombre="LDGP"/>
    <direccion paginaWeb="http://web.alava.net/PatrimonioHistorico/patrimonio/patrimonihome.htm"/>
    <informacion descripcion="Recoge la información de los diferentes trabajos de investigación llevados a cabo por el Laboratorio de Documentación Geométrica del Patrimonio"/>
    <ruta direccion="../../../BDLDGP/algorithmTraductorConsulta2.php3"/>
  </bd>
  <bd id="3">
    <nombre nombre="Catedral de Santa Maria"/>
    <direccion paginaWeb="http://www.catedralvitoria.com/"/>
    <informacion descripcion="Recoge la información de la Catedral de Santa Maria"/>
    <ruta direccion="../../../SantaMaria/algorithmTraductorConsulta2.php3"/>
  </bd>
</basesdedatos>
```

Figura A.46. Inscripción de una nueva base de datos.

Para evitar problemas el “id” de la base de datos (dentro de la etiqueta “bd”) deberá ser distinto al de las bases de datos ya existentes. Además es preferible que siga la secuencia establecida (1, 2, 3, 4...) por lo tanto la nueva base de datos tendrá el identificador id=“3”. En el nombre, la dirección y la descripción escribimos la información referente a la base de datos, y en cuanto a la etiqueta <ruta/> en su atributo “dirección” escribiremos la ruta en donde se encuentra el módulo Traductor (traductor2.php) que, como veremos más adelante, deberá ser copiado de uno de los sistemas envoltorio de las bases de datos existentes, y pegado en el directorio al que ahora hacemos referencia. En este caso hemos supuesto que se encuentra en la ruta “./SantaMaria/algorithmTraductorConsulta2.php3”.

Una vez inscrita la base de datos en el documento “BDsComponentes.xml”, podemos dar por cerrado el módulo Constructor y Distribuidor de Consultas.

### C.1.2.5.2 CREACIÓN DIRECTORIO COMO “SISTEMA ENVOLTORIO”

El segundo paso es el más complejo ya que consta de varias partes. La primera de ellas, es crear un directorio con el nombre que se le quiera dar en el servidor donde se va a ubicar lo correspondiente a nuestra aplicación. Esta carpeta será el “sistema envoltorio” de la nueva base de datos. Dentro de dicho directorio deberemos colocar el módulo Traductor, y el árbol de Correspondencias, además de otros ficheros .php3. Para conseguir el módulo Traductor, copiamos del sistema envoltorio de la Base de datos del Laboratorio de Documentación Geométrica del Patrimonio de la Universidad del País Vasco (UPV/EHU) el fichero “algoritmoTraductorConsulta2.php3” ubicado en el directorio “AplicacionGeneral/BDLDGP”, y lo pegamos dentro del “sistema envoltorio” (el directorio que hemos creado) de la nueva base de datos del sistema. Repetimos el proceso con los siguientes ficheros, todos ellos ubicados en el mismo directorio que el “algoritmoTraductorConsulta2.php3”: “sinónimos.xml”, “DTDSinonimos.xml”, “emisorRespuestas2.php3” y “gestorBD2.php3”.

### C.1.2.5.3 MODIFICACIÓN DEL FICHERO “algoritmoTraductorConsulta2.php3”

Una vez terminado esto pasamos a la segunda parte de este segundo paso. En la segunda línea del “algoritmoTraductorConsulta2.php3”, dentro del “sistema envoltorio” de la nueva base de datos, debido a problemas que tiene nuestra herramienta (php) con los nombres, debemos realizar algún pequeño cambio.

```
1 <?php
2 function traductorConsulta2($contenido,$pi)
3 {
```

Figura A.47. Función traductorConsulta2.

En la Figura A.47 se puede ver como el nombre de la función es “traductorConsulta2”, siendo ese “2” el identificador de la base de datos que aparece en el documento XML “BDsComponentes.xml” comentado anteriormente. En la Figura A.48 comprobamos esta información.

```
<bd id="2">
  <nombre nombre="LDGP"/>
  <direccion paginaWeb="http://web.alava.net/PatrimonioHistorico/
  <informacion descripcion="Recoge la información de los diferentes
  Documentación Geométrica del Patrimonio"/>
  <ruta direccion="../BDLDGP/algoritmoTraductorConsulta2.php3"/>
</bd>
```

Figura A.48. Parte del documento “BDsComponentes.xml” correspondiente a la base de datos LDGP. Marcado el identificador id= “2”.

Por lo tanto la función traductorConsulta del fichero “algoritmoTraductorConsulta2.php3” del nuevo “sistema envoltorio” deberá ser traductorConsulta3, tal y como indica su “id” en el documento “BDsComponentes.xml”.

Siguiendo con los cambios dentro del documento “algoritmoTraductorConsulta2.php3” en la línea 18 nos encontramos tal y como muestra la Figura A.49 una “ruta” utilizada para abrir el Árbol de Correspondencias del “sistema envoltorio” de la nueva base de datos.

```
17     $consultaXML='consulta.xml';  
18     $correspondencia='../BDLDGP/ArbolDeCorrespondenciasLDGP.xml';
```

Figura A.49. Parte del documento “algoritmoTraductorConsulta2.php3”.

Dicha “ruta” deberá ser modificada por aquella en donde se encuentre el Árbol de Correspondencias que debe ser construido para transformar el Lenguaje de Consulta en el lenguaje propio de la base de datos. En nuestro caso, para el ejemplo que venimos siguiendo el resultado sería el que se muestra en la Figura A.50.

```
17     $consultaXML='consulta.xml';  
18     $correspondencia='../SantaMaria/ArbolDeCorrespondenciasLDGP.xml';
```

Figura A.50. Parte del documento “algoritmoTraductorConsulta2.php3”. Ruta modificada.

Debido al problema de nombres comentado anteriormente, es necesario cambiar tanto en la línea 39, como en las líneas 373 y 388, el nombre de la función, cambiando como en el caso de la función traductorConsulta3, el número que en esta se indica por el identificador “id” que aparece en el documento “BDsComponentes.xml”. En las siguientes figuras, figura A.51 y figura A.52 se muestran ambos cambios respectivamente.

```
$elementoActual=BuscarConceptoActual2($elementoActual);  $elementoActual=BuscarConceptoActual3($elementoActual);
```

Figura A.51. Parte del documento “algoritmoTraductorConsulta2.php3” que es necesario cambiar.

```

Function BuscarConceptoActual2($elementoActual)
{
    foreach ($elementoActual->children() as $hijo)
    {
        if($hijo->getName()=="isa")
        {
            $isa=$elementoActual->isa;
        }
        else
        {
            return $elementoActual;
        }
    }
    $elementoActual=$isa->concepto;
    return BuscarConceptoActual2($elementoActual);
}

Function BuscarConceptoActual3($elementoActual)
{
    foreach ($elementoActual->children() as $hijo)
    {
        if($hijo->getName()=="isa")
        {
            $isa=$elementoActual->isa;
        }
        else
        {
            return $elementoActual;
        }
    }
    $elementoActual=$isa->concepto;
    return BuscarConceptoActual3($elementoActual);
}

```

Figura A.52. Parte del documento “algoritmoTraductorConsulta2.php3” que es necesario cambiar.

Continuando con las modificaciones, también es necesario cambiar la “ruta” del fichero “sinónimos.xml”, línea 107. Para ello seguiremos el mismo proceso que para caso anteriores, escribiendo la ruta exacta en donde se encuentra dicho documento. En la siguiente Figura (Figura A.53) se muestra el cambio realizado para nuestro ejemplo.

```

$arraydesinonimos[0]=$valordelAtributo;
$sinonimos='../SantaMaria/sinonimos.xml';
$whereSinonimosCompleto="";

```

Figura A.53. Parte del documento “algoritmoTraductorConsulta2.php3”. Modificando la ruta del fichero sinónimos.

Por último en la línea 355 debemos cambiar el nombre del método cambiando en “2” actual por el número de identificación de la base de datos inscrita en el documento XML “BDsComponentes.xml”, en el caso del ejemplo “3”.

```

355          return gestorConsulta3($consultaTotal, $pi);
---
```

Figura A.54. Parte del documento “algoritmoTraductorConsulta2.php3”. Modificando el nombre del método.

#### C.1.2.5.4 MODIFICACIÓN DEL FICHERO “gestorBD2.php3”

Como consecuencia de la modificación en la llamada del método, dentro del fichero “gestorBD2.php3”, en la tercera línea realizaremos el mismo cambio. Éste se puede ver en la Figura A.55.

```

-----
function gestorConsulta3($consultaTotal, $pi)
{

```

Figura A.55. Parte del documento “gestorBD2.php”, Modificando la cabecera del método.

En el fichero “gestorBD2.php3” se deben incluir los ficheros que hagan falta para comunicarse con la nueva base de datos.

Además en la línea del propio fichero se deberá sustituir el nombre del método cambiando el número del mismo por el por el número de identificación de la base de datos inscrita en el documento XML “BDsComponentes.xml”, en el caso del ejemplo “3”.

```
$resultado=$conexionNicio->query($consulta);  
return emisionRespuesta3($resultado,$numFilas,$pi);
```

Figura A.56. Parte del documento “gestorBD2.php”. Modificando el nombre del método.

#### C.1.2.5.5 MODIFICACIÓN DEL FICHERO “emisorRespuestas2.php3”

Como consecuencia del cambio en el nombre en la llamada del método, dentro del fichero “emisorRespuestas2.php3”, en la segunda línea realizaremos el mismo cambio. Éste se puede ver en la Figura A.57.

```
1 <?php  
2 function emisionRespuesta3($resultado,$numFilas,$pi)  
3 {
```

Figura A.57. Parte del documento “emisorRespuestas2.php3”. Modificando la cabecera del método.

#### C.1.2.5.6 ÁRBOL DE CORRESPONDENCIAS

El Árbol de Correspondencias deberá ser elaborado por algún individuo con conocimientos necesarios para llevar a cabo el proceso. Tanto en el apartado de diseño de la aplicación, como en el apartado de implementación de la misma, viene información acerca de cómo se ha de desarrollar un Árbol de Correspondencias. Éste tiene que albergar información útil para conseguir la transformación de Lenguaje Consulta al lenguaje propio de la nueva base de datos del sistema.

#### C.1.2.5.7 MODIFICACIÓN DEL FICHERO “capaPresentacion.php3”

En este fichero se deberá crear un método que muestre los resultados obtenidos. Obviamente para cada base de datos la información que se muestra es diferente ya que los datos conseguidos en la consulta no son los mismos para todas las bases de datos. En el fichero lo único que se ha de añadir es una estructura “if”, en la cual mediante al

atributo “id” se le asocia el número correspondiente al identificador “id” asignado a la base de datos en el documento XML “BDsComponentes.xml”.

En la siguiente figura, figura A.58, aparece la estructura “if” añadida.

```
//else if($_GET["id"]=="2" && isset($_GET["resu"]))
if($_id=="3")
{
    if(isset($_GET["resu"]))
    {
        $r=$_GET["resu"];
    }
    else if(isset($_POST["resu".$ese]))
    {
        $r=$_POST["resu".$ese];
    }
    $tmpo = stripslashes($r);
    $tmpao = urldecode($tmpo);
}
```

Figura A.58. Estructura “if” añadida al fichero “capaPresentacion.php3”.

En la figura se parecía claramente con el identificador es igual a “3”, número asociado al identificador de la base de datos en el documento “BDsComponentes.xml”. Únicamente ahora sólo falta desarrollar una página para mostrar los resultados. Obviamente para esta tarea es necesario alguien con conocimientos sobre la herramienta utilizada.

## C.2 MANUAL DE USUARIO DE LA APLICACIÓN QUE GESTIONA LA BASE DE DATOS “SITIOS WEB” Y “ADMINISTRADORES”

A lo largo de este manual de usuario se irán explicando cada una de las funcionalidades de la aplicación correspondiente al soporte de la base de datos que almacena información sobre las páginas Webs de las diferentes instituciones relacionadas con el patrimonio, con el fin de que el usuario pueda aclarar cualquier tipo de duda que le pueda surgir durante el uso de la misma. La explicación se realizará de manera ordenada, comenzando con la página de inicio y continuando con cada una de las funcionalidades mostradas en el menú de la aplicación en el orden en que estas aparecen. La aplicación está dividida en dos claros perfiles: administrador y usuario. Para que el lector no confunda las funcionalidades de cada perfil, estos serán diferenciados claramente desde el principio.

El perfil “administrador” será el primero en explicarse, una vez se haya finalizado con este comenzaremos con el perfil “usuario”. Antes de explicar las funcionalidades a las que tienen acceso cada uno de los perfiles, se explicará la pantalla de autenticación de usuarios.

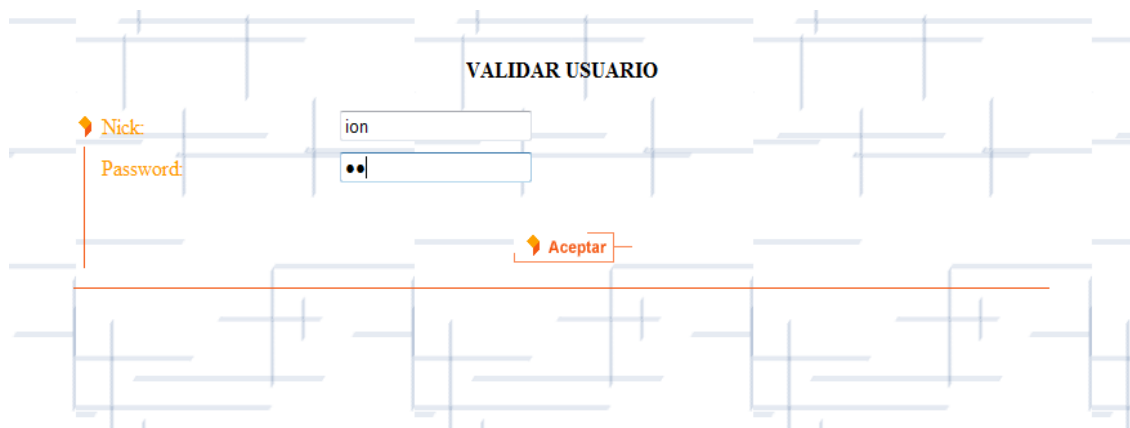


Figura A.59. Autenticación de usuarios

El usuario, y cuando se refiere a usuario se refiere a cualquier individuo que intente acceder al sistema ya sea su perfil “administrador” o “usuario”, tal y como se aprecia en la Figura A.59 introduce su Nick (no su nombre) y su contraseña (password). El sistema valida que el usuario y la contraseña son correctas y además coinciden, y en función del perfil que tenga dicho usuario le mostrará una pantalla u otra. Citar que la contraseña se guarda encriptada en la base de datos, por lo que nadie que acceda a dicha base de datos podrá conocer las contraseñas de los demás.

## C.2.1 PERFIL ADMINISTRADOR

Se trata del perfil con más poder en la aplicación, aquella persona a la que se le otorgue el perfil de administrador puede realizar cualquiera de las funcionalidades que presenta la aplicación, sin restricción alguna. A continuación se irán describiendo cada una de ellas en el orden exacto en el que aparecen en el menú (parte izquierda de la pantalla). Cuando el administrador accede a la aplicación la primera pantalla que se encuentra es una pantalla de bienvenida, en la que en el centro de la misma se encuentra un mensaje de bienvenida y en la parte izquierda de dicha pantalla se sitúa el menú que se mantendrá en dicha parte izquierda a lo largo de toda la aplicación.

Las tres primeras funcionalidades que aparecen en la aplicación son las relacionadas con el sitio Web. Antes de comenzar a explicar cada una de ellas, nos introduciremos en el concepto que le hemos dado a “sitio Web”. Definimos sitio Web como la ubicación en la red donde aparecen todas y cada una de las páginas de dicho sitio Web, es decir, la página principal de una página Web. Tomando como ejemplo, la página del Laboratorio de Documentación Geométrica del Patrimonio (UPV-EHU), la dirección del sitio Web sería <http://www.vc.ehu.es/docarq/> que es la página principal, cada una del resto de páginas Web de dicho sitio Web, serán clasificadas como sub-páginas Web.

Una vez citado esto se explicarán las funcionalidades referentes al sitio Web: Nuevo sitio Web, Modificar sitio Web y Eliminar sitio Web.

### C.2.1.1 FUNCIONALIDADES SITIO WEB



Figura A.60. Representa el menú de la aplicación, concretamente las funcionalidades de Sitio Web, las cuales se explican a continuación.



**Nuevo sitio Web:** Dicha funcionalidad nos permite introducir un sitio Web en nuestra base de datos. Para ello, como se observa en la Figura A.61 se deben rellenar campos tales “Dirección Web”, “Nombre de la Organización”, “Descripción”.

**NUEVO SITIO WEB**

**Datos del Sitio Web**

Dirección Web:

Nombre de la organización:

Descripción:

Figura A.61. Nuevo sitio Web

Una vez rellenados los campos y pulsando el botón introducir, el sistema incluirá dicho sitio Web en la base de datos, y mostrará al usuario la siguiente pantalla en donde se le indica todos y cada uno de los sitiosWeb y en rojo el sitio Web que acaba de introducir.

Lista sitio Web	
Dirección Web	Nombre
http://www.alava.net/default_c.asp	Diputacion Foral de Alava
http://www.catedralvitoria.com/	Fundación de la Catedral de Santa María.
http://www.vc.ehu.es/docarq/	Laboratorio de documentación geométrica del patrimonio
http://www.vc.ehu.es/docarq/LDA/index.htm	Laboratorio de documentación geométrica del patrimonio
www.parecequevabien.com	parece que va bien

Figura A.62. Sitio Web introducido.

Si, por el contrario, se intenta introducir un sitio Web ya existe el sistema no lo permitirá ya que conoce que existe un sitio Web ya introducido en la base de datos. El sistema se lo mostrará al usuario mediante un mensaje por pantalla. El mensaje será el que se muestra en la figura A.63.

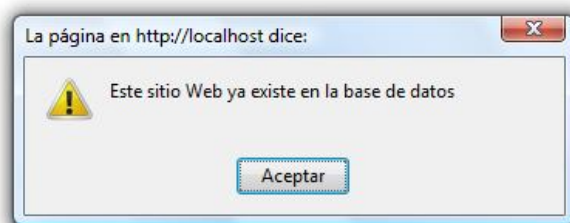


Figura A.63. Mensaje de aviso. El sitio web ya existe en la base de datos.

**Modificar sitio Web:** Modificar sitio Web nos permite modificar un sitio Web existente en nuestro sistema. Esta funcionalidad consta de dos partes. La primera de ellas es una lista desplegable en donde aparecen todos y cada uno de los sitiosWeb existentes en el sistema. Siempre que se introduzca un nuevo sitio Web en el sistema aparecerá en dicha lista. De dicha lista desplegable se seleccionará aquel sitio Web del que queramos modificar sus datos. Para que esto que se ha explicado quede visualmente explicado a continuación se muestran dos figuras, una que recoge las dos partes de la funcionalidad que se está explicando y otra que recoge únicamente la lista desplegable.

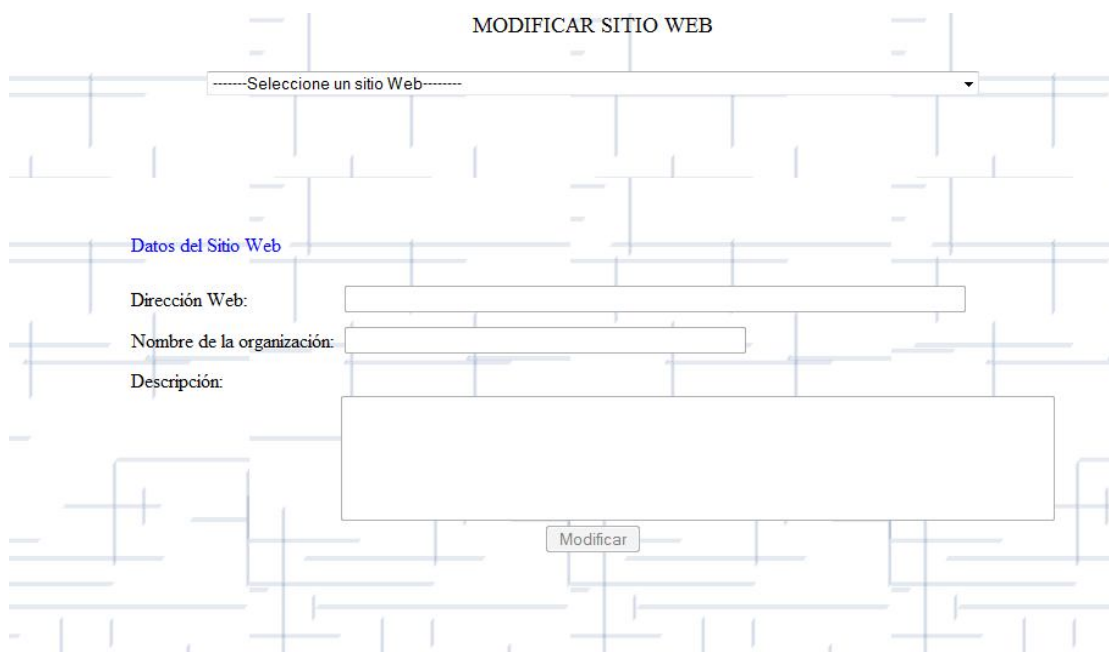


Figura A.64. Modificar sitio Web.

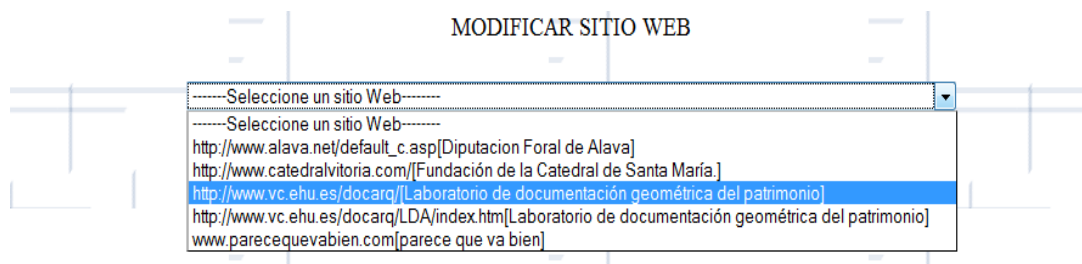


Figura A.65. Lista desplegable de Modificar sitio Web.

Una vez que el “administrador” seleccione un sitio Web automáticamente el sistema cargara sus datos en los campos del formulario que se muestran en la Figura A.66.



Figura A.66. Modificar sitio Web. Datos de sitio Web preparados para modificar.

Una vez que los datos están cargados bastará con modificar aquellos que se quieran modificar. Una vez modificados pulsamos el botón modificar y el sistema nos actualizará los datos en la base de datos. Para asegurarnos de que realmente los ha modificado el sistema nos mostrará el siguiente mensaje.

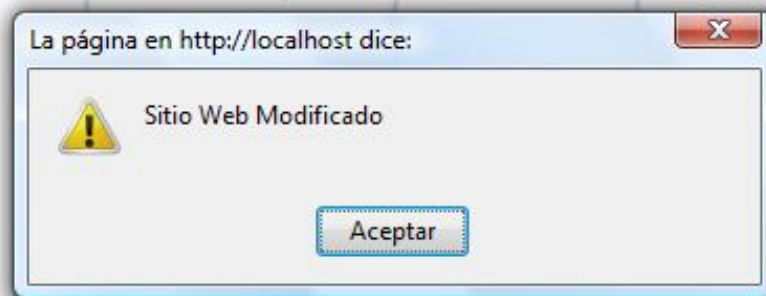


Figura A.67. Sitio Web Modificado.

Posteriormente nos volverá a mostrar la página donde se modifica el sitio Web con la nueva información que ha sido actualizada.

Si el administrador, por equivocación, intenta cambiar la dirección Web del sitio Web por otro la dirección de otro sitio Web ya existente en la base de datos el sistema no se lo permitirá mostrándole el siguiente mensaje de aviso.

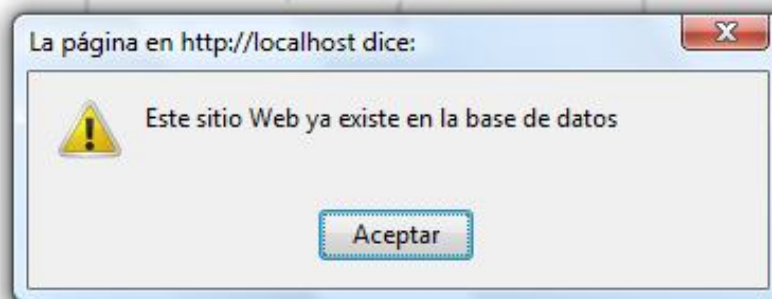


Figura A.68. Mensaje de aviso. Se intenta introducir un sitio Web que ya existe.

**Eliminar sitio Web:** Este sub-apartado de sitio Web nos permite eliminar un sitio Web existente en nuestro sistema. Para esto el sistema muestra en una lista todos y cada uno de los sitios Web que aparecen en la base de datos del sistema. Para que no haya lugar a una posible confusión, el sistema nos muestra tanto la dirección Web del sitio Web como el nombre de la organización a la que la página pertenece. Además de estos datos nos muestra un icono, una especie de x, que nos permite eliminar el sitio Web al que hace referencia. La Figura A.69 muestra perfectamente lo que se ha explicado.

Lista de Sitios Web		
Dirección Web	Nombre	Eliminar
http://www.alava.net/default_c.asp	Diputacion Foral de Alava	<input type="checkbox"/>
http://www.catedralvitoria.com/	Fundación de la Catedral de Santa María.	<input type="checkbox"/>
http://www.vc.ehu.es/docarq/	Laboratorio de documentación geométrica del patrimonio	<input type="checkbox"/>
http://www.vc.ehu.es/docarq/LDA/index.htm	Laboratorio de documentación geométrica del patrimonio	<input type="checkbox"/>
www.parecequevabien.com	parece que va bien	<input type="checkbox"/>

Figura A.69. Lista de sitios Web.

Para eliminar un sitio Web bastará con pulsar la el botón de la columna eliminar que corresponda con el sitio Web que se desea eliminar. Con el fin de evitar que por un error de selección a la hora de eliminar el administrador elimine un sitio Web que no quería eliminar, el sistema obligará al administrador a confirmar la eliminación del elemento que pretende eliminar mediante el siguiente mensaje.

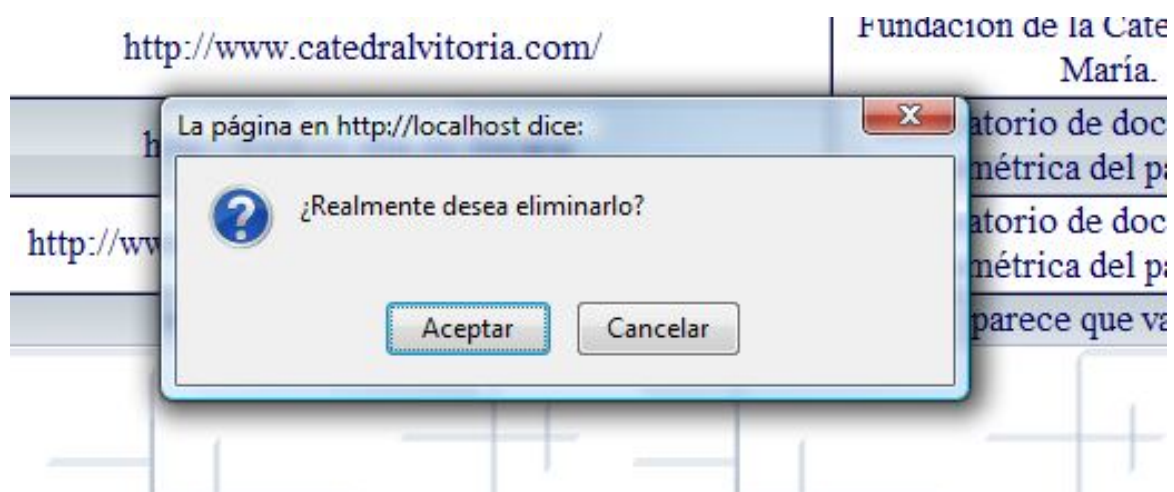


Figura A.70. Confirmación de eliminación de sitio Web.

Una vez aceptado el cuadro de la imagen superior, el sitio Web quedará completamente eliminado, y arrastrará a su eliminación todas las sub-páginas que contenga su sitio Web, así como los elementos patrimoniales que contengan dichas sub-páginas. Se trata de una eliminación en cascada.

Una vez que se elimine el sitio Web el sistema mostrará al administrador la misma lista de la Figura A.69, obviamente, sin el sitio Web eliminado.

Las siguientes funcionalidades que se van explicar son las referentes a las sub-páginas Web. Recordemos que un sitio Web consta de una o muchas sub-páginas Web.

### **C.2.1.2 FUNCIONALIDADES SUB-PÁGINAS WEB**

Las funcionalidades de esta parte de la aplicación son: Nueva sub-páginasWeb, Modificar sub-páginas Web y Eliminar sub-páginas Web.

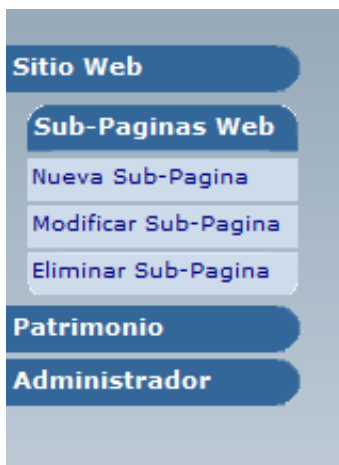


Figura A.71. Representa el menú de la aplicación, concretamente las funcionalidades de Sub-Páginas Web, las cuales se explican a continuación.

**Nueva sub-páginas Web:** Esta funcionalidad nos permite añadir una sub-páginas Web a un sitio Web existente. Para ello deberemos seleccionar en una lista desplegable, en donde aparecen todos y cada uno de los sitiosWeb existentes en el sistema, el sitio Web en el que queremos introducir nuestra subpágina Web.

NUEVA SUBPAGINA WEB

Indique el sitio Web en el que quiera insertar una subpágina

Datos de la Página Web

Dirección de la Sub-Página Web:

Descripción:

Disponible sobre este tema en la Web:

Figura A.72. Introducir nueva sub-página Web.

Tal y como se puede observar en la Figura A.72 el administrador ha seleccionado un sitio Web, concretamente en el que desea introducir una nueva sub-página Web, y ahora bastará con introducir los datos de dicha subpágina Web y dicha subpágina quedará registrada.

Para hacérselo saber al administrador, el sistema lista por pantalla todas las subpáginas del sitio Web en el que se ha insertado la nueva subpágina, subrayando esta con color rojo, tal y como aparece en la Figura A.73.

Lista de Subpáginas del sitio Web <a href="http://www.vc.ehu.es/docarq/">http://www.vc.ehu.es/docarq/</a>	
Dirección Web	Descripción
<a href="http://www.vc.ehu.es/docarq/Proyectos">http://www.vc.ehu.es/docarq/Proyectos</a>	Sub-página donde se encuentran todos los proyectos del laboratorio.

Figura A.73. Subpáginas Web del sitio Web <http://www.vc.ehu.es/docarq/>. En rojo aparece la subpágina que se acaba de insertar.

Si el administrador intenta introducir una subpágina Web que ya existía en el sitio Web seleccionado, el sistema no se lo permitirá y le mostrará el siguiente mensaje de aviso.

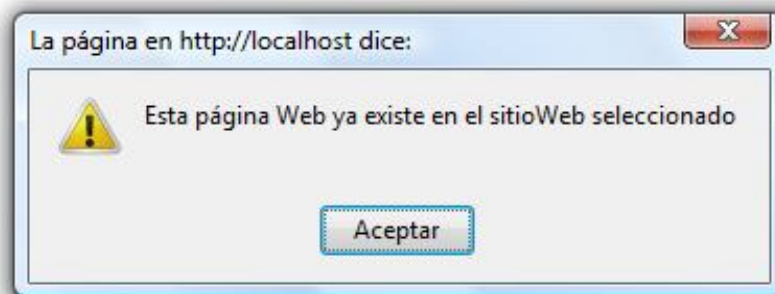


Figura A.74. Mensaje de aviso. La subpágina Web ya existía.

**Modificar sub-páginasWeb:** Como su propio nombre indica esta funcionalidad nos permite modificar una sub-página Web de un sitio Web existente. Para ello deberemos seleccionar, como en el caso anterior, en una lista desplegable, en donde aparecen todos y cada uno de los sitiosWeb existentes en el sistema, el sitio Web del que queremos modificar nuestra subpágina Web. Una vez seleccionado de la primera lista desplegable el sitio Web del que queremos modificar la subpágina Web, se debe seleccionar en una nueva lista desplegable la subpágina Web que va a ser modificada. Obviamente en esta segunda lista desplegable únicamente contendrá aquellas subpáginas Web del sitio Web seleccionado en la primera lista desplegable.

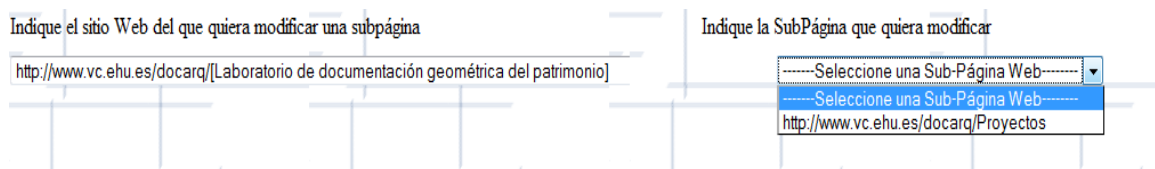


Figura A.75. Muestra la lista desplegable de subpáginas Web una vez seleccionado un sitio Web determinado.

Como se puede apreciar en la Figura A.75 la lista desplegable de sub-páginas Web muestra sólo aquellas que pertenecen al sitio Web seleccionado en la primera lista desplegable.

Una vez que se ha seleccionado la subpágina Web los campos del formulario se rellenarán de forma automática con los datos de dicha subpágina.



Indique el sitio Web del que quiera modificar una subpágina  
http://www.vc.ehu.es/docarq/[Laboratorio de documentación geométrica del patrimonio]

Indique la SubPágina que quiera modificar  
http://www.vc.ehu.es/docarq/Proyectos

**Datos de la Sub-Página Web**

Dirección Web: http://www.vc.ehu.es/docarq/Proyectos

Descripción: Sub-página donde se encuentran todos los proyectos del laboratorio.

Disponible en la Web: Proyectos llevados a cabo por el laboratorio.

Disponible en la organización: Documentación de todos los proyectos que se han desarrollado en los últimos 20 años

Modificar

Figura A.76. Modificar subpágina Web.

El administrador modifica los datos que desee sobre la subpágina Web y pulsará el botón “modificar”. Una vez que haga esto los datos quedarán registrados en la base de datos. Para mostrar al administrador que la tarea ha sido realizada le mostrará un mensaje como el siguiente.

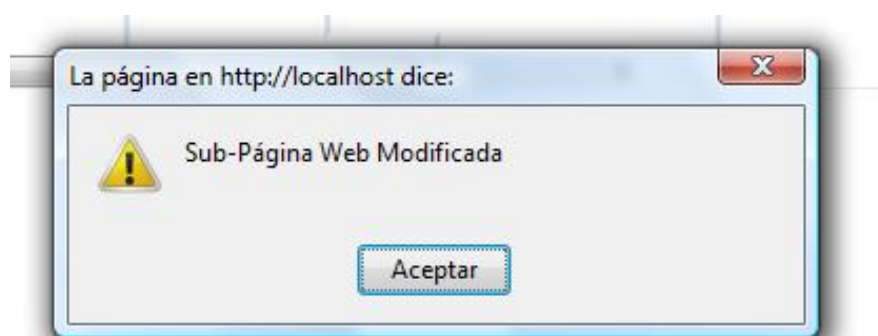


Figura A.77. Mensaje de aviso. La subpágina Web ha sido modificada.

Si el administrador intenta modificar la dirección de la subpágina Web asignándole la dirección de otra subpágina Web que ya existe en dicho sitio Web, el sistema no se lo permitirá y le mostrará un mensaje como el siguiente.

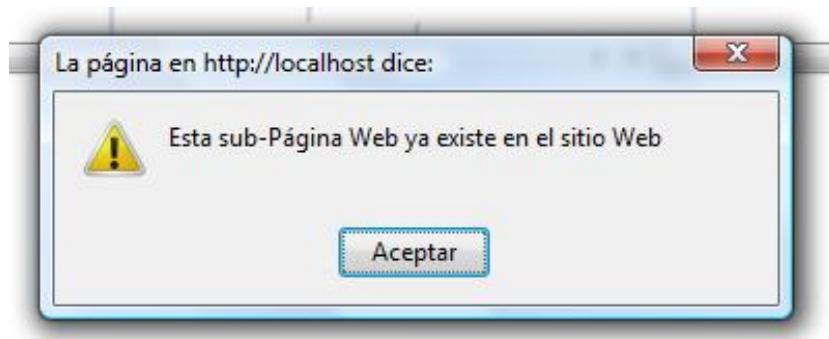



Figura A.78. Mensaje de aviso. La dirección de la subpágina Web que se pretendía modificar ya pertenece a otra subpágina Web.

**Eliminar sub-páginas Web:** Como su propio nombre indica esta funcionalidad nos permite eliminar una sub-página Web de un sitio Web existente. Para ello deberemos seleccionar, como en los casos anteriores, en una lista desplegable, en donde aparecen todos y cada uno de los sitios Web existentes en el sistema, el sitio Web del que queremos eliminar nuestra subpágina Web. Una vez seleccionado de la primera lista el sitio Web del que vamos a eliminar una subpágina, el sistema listará todas las subpáginas Web de dicho sitio Web junto con una imagen , que permitirá al administrador eliminar la subpágina que desee.

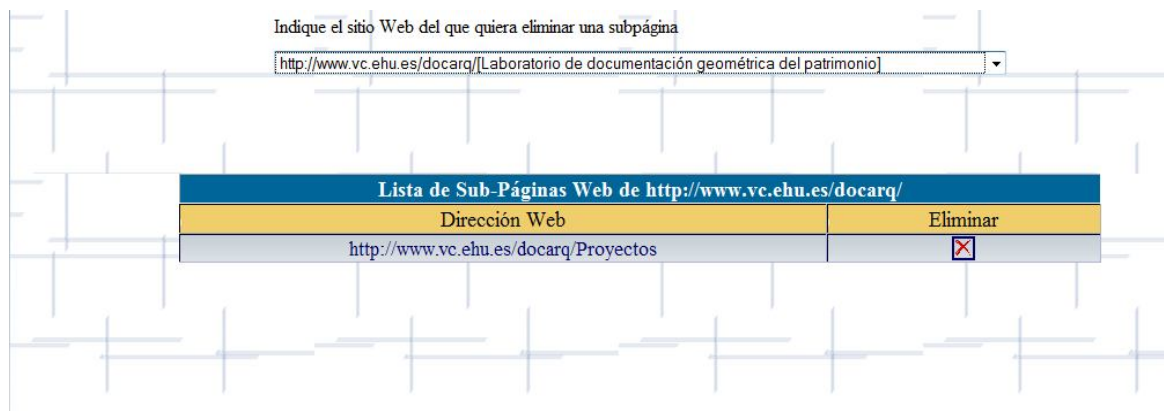


Figura A.79. Lista de sub-páginas Web del sitio Web seleccionado en la lista desplegable superior.

Con el fin de evitar que por un error de selección a la hora de eliminar el administrador elimine una subpágina Web que no quería eliminar, el sistema obligará al administrador a confirmar la eliminación del elemento que pretende eliminar mediante el siguiente mensaje.

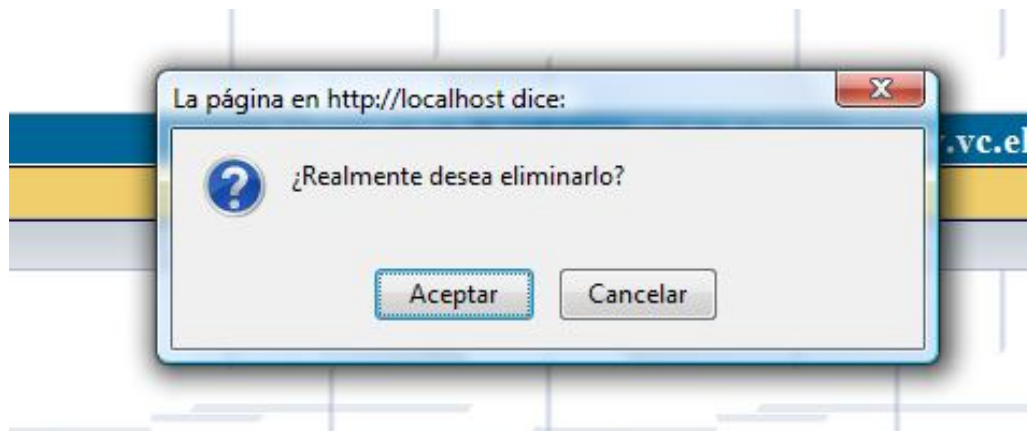


Figura A.80. Confirmación de eliminación de una subpágina Web.

**Una vez aceptado el cuadro de la imagen superior, la subpágina Web quedará completamente eliminada, y arrastrará a su eliminación todos los elementos patrimoniales que contengan dicha sub-página Web. Se trata de una eliminación en cascada.**

Una vez que se elimine el subpágina Web, el sistema mostrará al administrador la misma lista de la Figura A.79, obviamente, sin la subpágina Web eliminada.

Las siguientes funcionalidades que se van explicar son las referentes a las sub-páginas Web. Recordemos que un sitio Web consta de una o muchas sub-páginas Web.

### C.2.1.3 FUNCIONALIDADES PATRIMONIO

Las funcionalidades de esta parte de la aplicación son: Nuevo Patrimonio, Modificar Patrimonio y Eliminar Patrimonio.



Figura A.81. Representa el menú de la aplicación, concretamente las funcionalidades de Patrimonio, las cuales se explican a continuación.

**Nuevo Patrimonio:** Esta funcionalidad nos permite añadir un nuevo elemento patrimonial a una subpágina Web existente. Para ello deberemos seleccionar en una lista desplegable, en donde aparecen todos y cada uno de los sitiosWeb existentes en el sistema, el sitio Web en el que se encuentra la subpágina Web en donde queremos introducir el elemento patrimonial. Una vez seleccionado el sitio Web, el administrador debe seleccionar en otra lista desplegable la subpágina Web en la cual quiere introducir el elemento patrimonial. Obviamente en esta segunda lista desplegable sólo aparecen las subpáginas Web que existen dentro del sitio Web seleccionado. En la Figura A.82 se puede observar claramente como existen dos listas desplegables, en las cuales se selecciona el sitio Web y la subpágina donde se insertará el elemento patrimonial.

Indique el sitio Web del que quiera añadir un elemento patrimonial

Indique la SubPágina en la que quiera insertar un elemento patrimonial

-----Seleccione una Sub-Página Web-----

-----Seleccione una Sub-Página Web-----

NUEVO ELEMENTO PATRIMONIAL

Datos del Elemento Patrimonial

Seleccione el lugar donde se quiere clasificar el contenido:

- excavaciones
- arquitectura religiosa
- arquitectura civil
- elementos menores
- objetos

Figura A.82. Nuevo elemento patrimonial. Seleccionado el sitio Web y seleccionándose la subpágina Web donde se va insertar el elemento patrimonial.

Una vez seleccionado tanto el sitio Web como la subpágina Web, seleccionamos el tipo de elemento patrimonial que se va a insertar. Cabe destacar que esta clasificación que aparece en la Figura A.82 **puede ser modificable en función de la base de datos**, pero esto ya ha sido comentado en los Manuales de mantenimiento (C.1.1). En función del elemento patrimonial que se escoja aparecerán una serie de atributos u otros, lógicamente en función de los atributos que dicho elemento patrimonial seleccionado disponga en la base de datos.

Indique el sitio Web del que quiera añadir un elemento patrimonial  
 http://www.vc.ehu.es/docarq/[Laboratorio de documentación geométrica del patrimonio]

Indique la SubPágina en la que quiera insertar un elemento patrimonial  
 http://www.vc.ehu.es/docarq/Proyectos

NUEVO ELEMENTO PATRIMONIAL

[Datos del Elemento Patrimonial](#)

Seleccione el lugar donde se quiere clasificar el contenido:

- excavaciones
- arquitecturareligiosa
- arquitecturacivil
- elementosmenores
- objetos

Nombre:

Periodo:

Tipo\_datos:

Municipio:

Poblacion:

Provincia:

Tipo\_arquitecturaReligiosa:

Figura A.83. Nuevo elemento patrimonial. Seleccionado el sitio Web y la subpágina Web donde se va insertar el elemento patrimonial. También ha sido seleccionado el tipo de elemento patrimonial

Una vez que se rellenen los campos con la información relativa al elemento patrimonial, bastará con pulsar el botón “introducir” y el elemento patrimonial será introducido. Para mostrar al administrador que la tarea ha sido realizada le mostrará un mensaje como el siguiente.

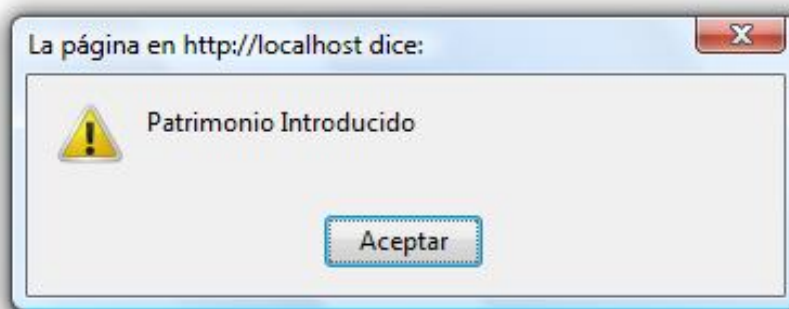


Figura A.84. Mensaje de aviso. El elemento patrimonial ha sido introducido.

**Modificar Patrimonio:** Esta funcionalidad nos permite modificar un elemento patrimonial de una subpágina Web existente. Para ello deberemos seleccionar en una lista desplegable, en donde aparecen todos y cada uno de los sitiosWeb existentes en el

sistema, el sitio Web en el que se encuentra la subpágina Web en donde queremos introducir el elemento patrimonial. Una vez seleccionado el sitio Web, el administrador debe seleccionar en otra lista desplegable la subpágina Web en la cual se encuentra el elemento patrimonial que quiere modificar. Obviamente en esta segunda lista desplegable sólo aparecen las subpáginas Web que existen dentro del sitio Web seleccionado. En la Figura A.85 se puede observar claramente como existen dos listas desplegables, en las cuales se selecciona el sitio Web y la subpágina donde se encuentra el elemento patrimonial que se desea modificar.

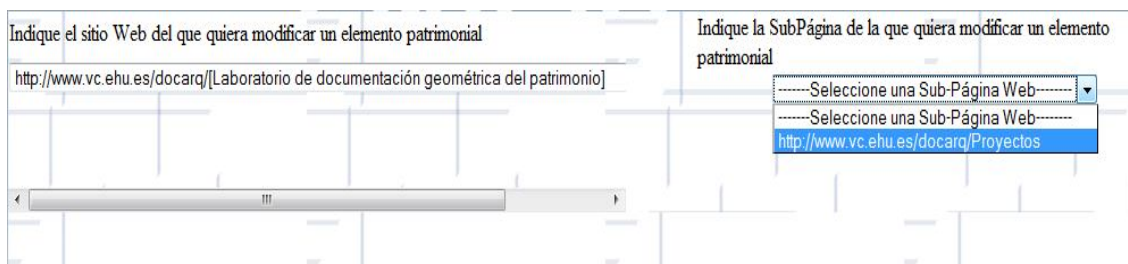



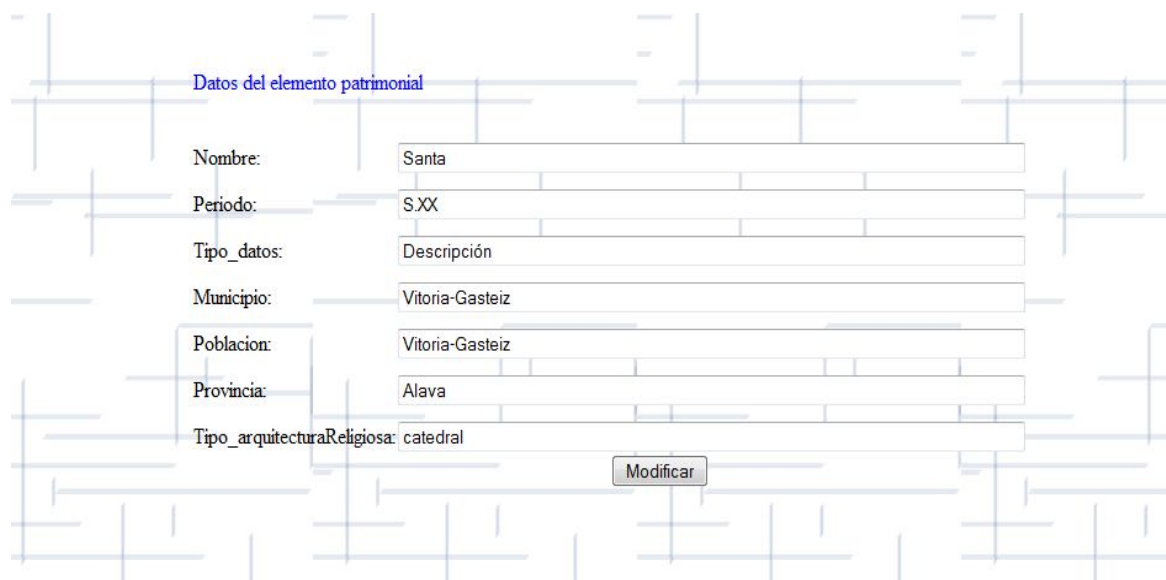
Figura A.85. Modificar elemento patrimonial. Seleccionado el sitio Web y seleccionándose la subpágina Web en la cual se encuentra el elemento patrimonial que se va a modificar.

Una vez seleccionado tanto el sitio Web como la subpágina Web en donde se encuentra el elemento patrimonial a modificar, el sistema nos mostrará una lista con los elementos patrimoniales de la subpágina seleccionada.



Figura A.86. Modificar elemento Patrimonial. Listado de elementos patrimoniales de la subpágina seleccionada.

Junto a cada elemento patrimonial nos encontramos el siguiente símbolo . Pulsando dicho símbolo correspondiente al elemento patrimonial que se quiere modificar nos encontremos la siguiente pantalla.



Datos del elemento patrimonial

Nombre: Santa

Periodo: S.XX

Tipo\_datos: Descripción

Municipio: Vitoria-Gasteiz

Poblacion: Vitoria-Gasteiz

Provincia: Alava

Tipo\_arquitecturaReligiosa: catedral

Modificar

Figura A.87. Modificar elemento Patrimonial. Datos del elemento patrimonial seleccionado para ser modificado.

Una vez que el administrador realice los cambios que crea necesario, pulsará el botón “modificar”, y el sistema modificará dichos datos.

Para mostrar al administrador que la tarea ha sido realizada le mostrará un mensaje como el siguiente.

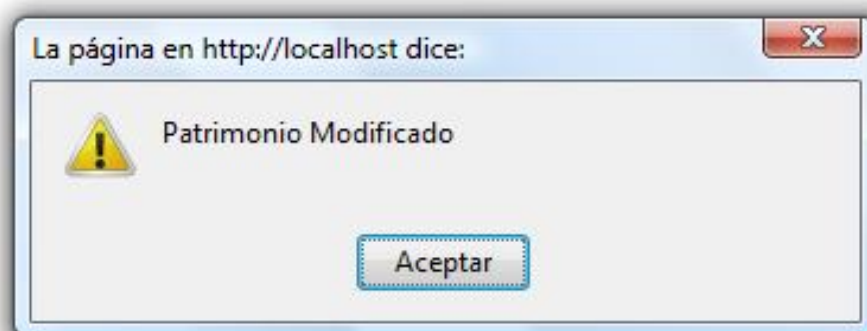


Figura A.88. Mensaje de aviso. El elemento patrimonial ha sido modificado.



**Eliminar Patrimonio:** Esta funcionalidad nos permite eliminar un elemento patrimonial de una subpágina Web existente. Para ello deberemos seleccionar en una lista desplegable, en donde aparecen todos y cada uno de los sitios Web existentes en el sistema, el sitio Web en el que se encuentra la subpágina Web en donde queremos eliminar el elemento patrimonial. Una vez seleccionado el sitio Web, el administrador debe seleccionar en otra lista desplegable la subpágina Web en la cual se encuentra el elemento patrimonial que quiere eliminar. Obviamente en esta segunda lista desplegable sólo aparecen las subpáginas Web que existen dentro del sitio Web seleccionado. En la Figura A.89 se puede observar claramente como existen dos listas desplegables, en las cuales se selecciona el sitio Web y la subpágina donde se encuentra el elemento patrimonial que se desea eliminar.

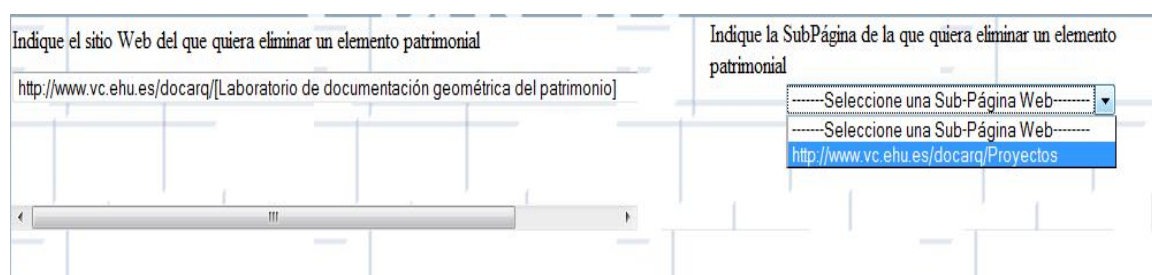


Figura A.89. Eliminar elemento patrimonial. Seleccionado el sitio Web y seleccionándose la subpágina Web en la cual se encuentra el elemento patrimonial que se va a eliminar.

Una vez que es seleccionado tanto el sitio Web como la subpágina Web en donde se encuentra el elemento patrimonial que se desea eliminar, el sistema nos muestra una lista con todos los elementos patrimoniales de la subpágina Web seleccionada en la lista desplegable.

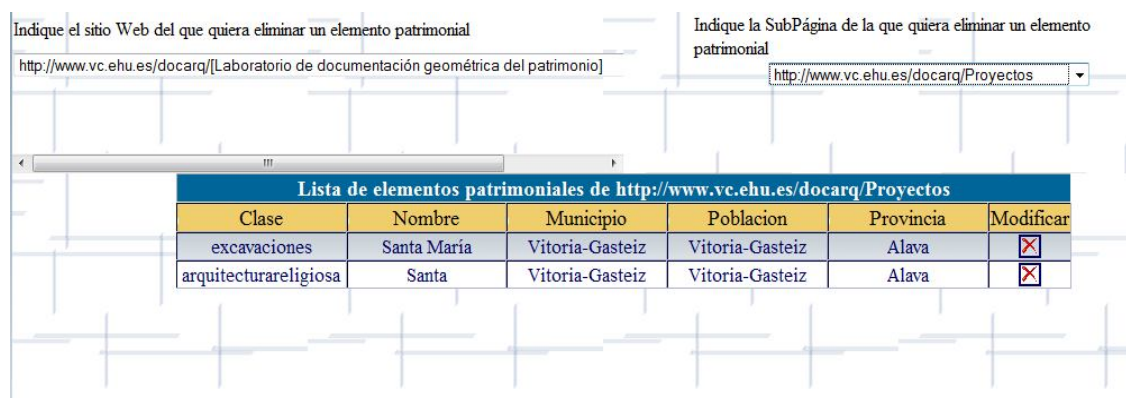



Figura A.90. Eliminar elemento Patrimonial. Listado de elementos patrimoniales de la subpágina seleccionada.

Junto a cada elemento patrimonial nos encontramos el siguiente símbolo . Pulsando dicho símbolo correspondiente al elemento patrimonial que se quiere eliminar podremos eliminar dicho elemento patrimonial.

Con el fin de evitar que por un error de selección a la hora de eliminar el administrador elimine una subpágina Web que no quería eliminar, el sistema obligará al administrador a confirmar la eliminación del elemento que pretende eliminar mediante el siguiente mensaje.

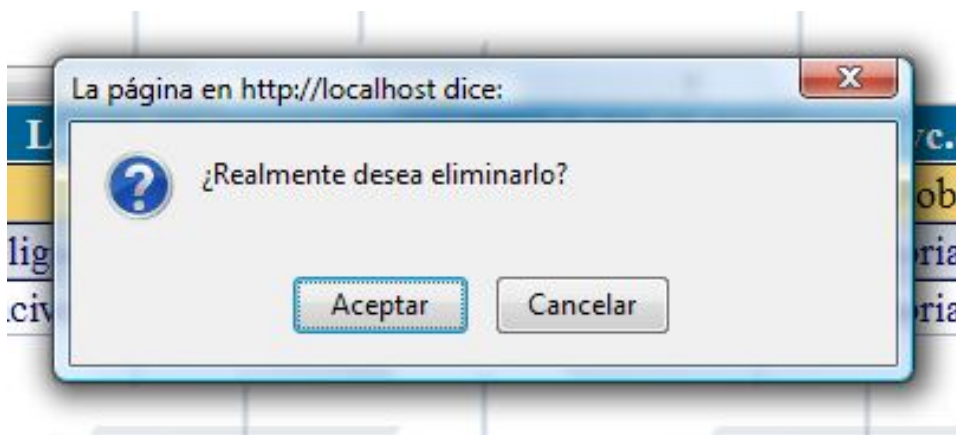


Figura A.91. Confirmación de eliminación de un elemento patrimonial.

Una vez aceptado el cuadro de la imagen superior, el elemento patrimonial quedará completamente eliminado.

Una vez que se elimine el elemento patrimonial, el sistema mostrará al administrador la misma lista de la Figura A.90, obviamente, sin el elemento patrimonial eliminado.

#### C.2.1.4 FUNCIONALIDADES ADMINISTRADOR

Las siguientes funcionalidades que se van explicar son las referentes a los administradores de la aplicación que se ha desarrollado a lo largo de este proyecto. Con el fin de que el administrador conozca hasta donde llega su poder en lo que se refiere al manejo de usuario y del resto de los administradores, se van a explicar detalladamente cada una de las funcionalidades.



Figura A.92. Representa el menú de la aplicación, concretamente las funcionalidades de Administrador, las cuales se explican a continuación.

**Nuevo Administrador:** Un administrador tiene completa potestad para añadir tanto nuevos administradores como nuevos usuarios. Es más, los administradores son los únicos que pueden realizar esta tarea ya que para los usuarios este tipo de tareas están restringidas.

Se trata del mismo proceso tanto para introducir un administrador como para introducir un usuario, el formulario es el mismo, tal y como se puede observar en la Figura A.93.

Datos del Administrador

Nombre:

Nick:

Password:

Confirmar Password:

Perfil: 

- Administrador
- Administrador
- Usuario

Telefono:

Correo:

Figura A.93. Nuevo Administrador(o usuario), aparece un formulario a rellenar con los datos de la persona a introducir.

El administrador debe rellenar los campos y seleccionar en la lista desplegable el perfil que desea asignar al nuevo componente de la página Web. Una vez rellenado los campos y seleccionado el perfil correspondiente, pulsando introducir, el nuevo miembro quedará registrado en la base de datos.

Para que el administrador se asegure que el nuevo miembro ha sido introducido correctamente, el sistema le mostrará una lista con todos los usuarios y administradores existente remarcando en color rojo el nuevo miembro introducido tal y como se aprecia en la siguiente figura.

Lista de Administradores		
Nombre	Nick	Rol
Estitxu	esti	Usuario
falete	faletes	Usuario
Ion Marín	ion	Administrador
pepepepe	pepe	Administrador
Pedro	peter	Usuario

Figura A.94. Nuevo Administrador(o usuario), lista de administradores y usuarios de la página Web. Remarcado en rojo el nuevo miembro.

Si el administrador se confunde e intenta introducir un usuario o un administrador, con el Nick de otro usuario que ya esté registrado, el sistema no se lo permitirá y se lo hará saber mediante el siguiente mensaje.

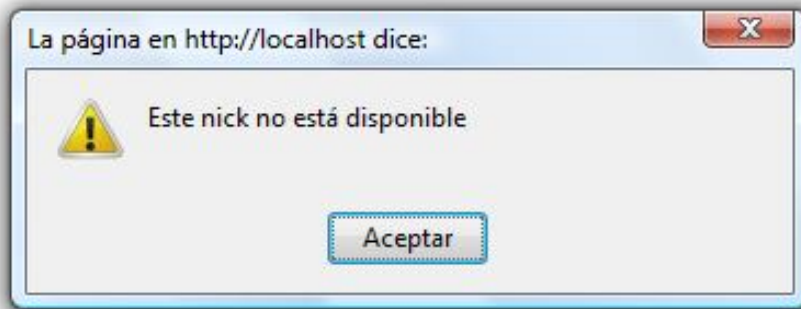


Figura A.95. Mensaje de aviso. Este Nick no está disponible.

**Modificar Administrador:** Un administrador tiene completa potestad para añadir tanto nuevos administradores como nuevos usuarios, sin embargo a la hora de modificar es diferente. Un administrador sólo podrá modificar sus datos y el de los usuarios, es decir, no podrá modificar los datos de los demás administradores.

Para esto el sistema le muestra una página que consta de dos partes. La primera de ellas es una lista desplegable en la cual el administrador debe seleccionar el usuario al que quiere modificar los datos, o seleccionarse a sí mismo en el caso de que vaya a modificar sus propios datos.

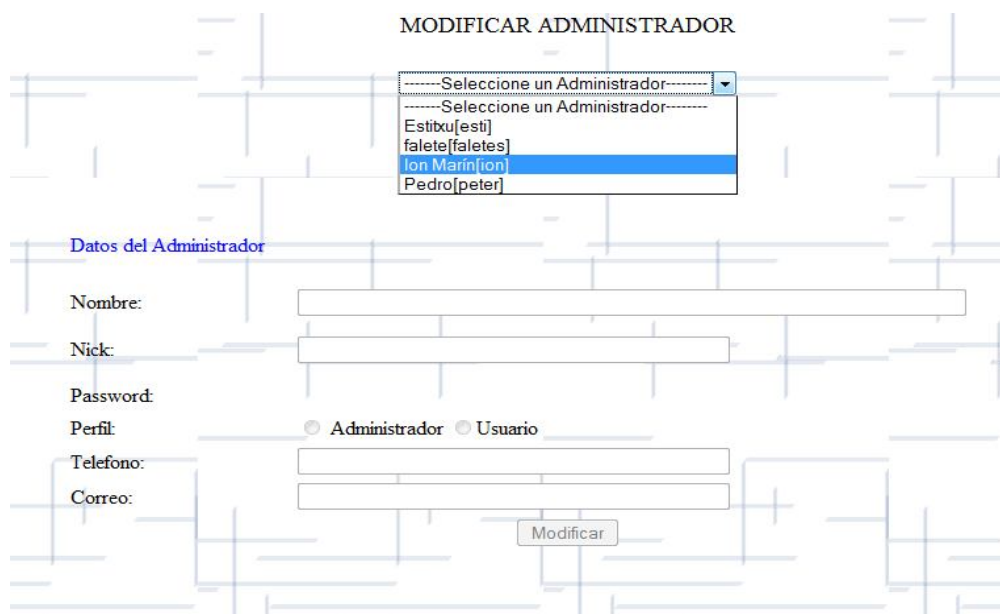


Figura A.96. Mensaje de aviso. Este Nick no está disponible.

Una vez que el administrador selecciona la persona cuyos datos va a modificar, éstos aparecerán en el formulario, tal y como muestra la Figura A.97.

The image shows a web form titled "MODIFICAR ADMINISTRADOR". At the top, there is a dropdown menu with "Ion Marín[ion]" selected. Below this is a section header "Datos del Administrador". The form contains several input fields: "Nombre:" with "Ion Marín", "Nick:" with "ion", "Password:" with a link "Modificar contraseña", "Perfil:" with radio buttons for "Administrador" (selected) and "Usuario", "Telefono:" with "680400829", and "Correo:" with "imarin005@ikasle.ehu.es". A "Modificar" button is located at the bottom right of the form.

Figura A.97. Modificar Administrador(o usuario). Datos en el formulario.

Como vemos en la Figura A.97, se pueden modificar datos tales como Nick nombre teléfono o correo, pero además nos permite cambiar a un usuario su perfil a administrador y viceversa. Además de esto se aprecia que el sistema nos da la posibilidad de cambiar la contraseña tanto de un usuario como la nuestra propia. Si elegimos modificar los datos excepto la contraseña bastará con, rellenar los campos con los datos deseados, pulsar el botón modificar, y los nuevos datos serán registrados. El sistema para avisar al administrador que la tarea se ha realizado correctamente le mostrará el siguiente mensaje.

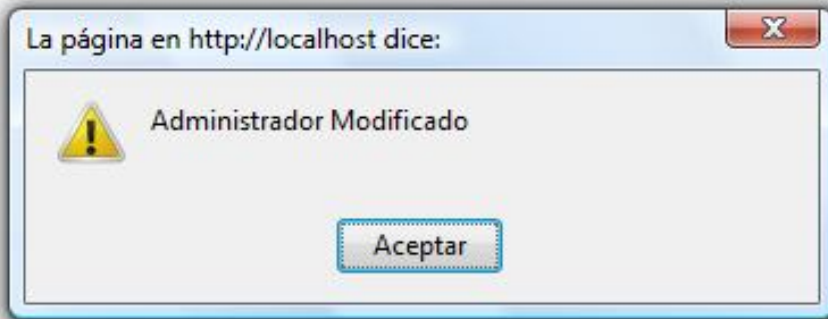


Figura A.98. Mensaje de aviso. El administrador ha sido modificado.

Además el sistema mostrará una lista con todos los usuarios y administradores existente remarcando en color rojo el nuevo miembro introducido con sus nuevos datos tal y como se aprecia en la siguiente figura.

Lista de Administradores		
Nombre	Nick	Rol
Estitxu	esti	Usuario
falete	faletes	Usuario
Ion Marín	ion	Administrador
pepepepe	pepe	Administrador
Pedro	peter	Usuario
ola	ola	Administrador

Figura A.99. Administrador(o usuario) modificado, lista de administradores y usuarios de la página Web. Remarcado en rojo el miembro que ha sido modificado.

Si el administrador se confunde e intenta modificar un usuario, o a sí mismo con el Nick de otro usuario que ya esté registrado, el sistema no se lo permitirá y se lo hará saber mediante el siguiente mensaje.

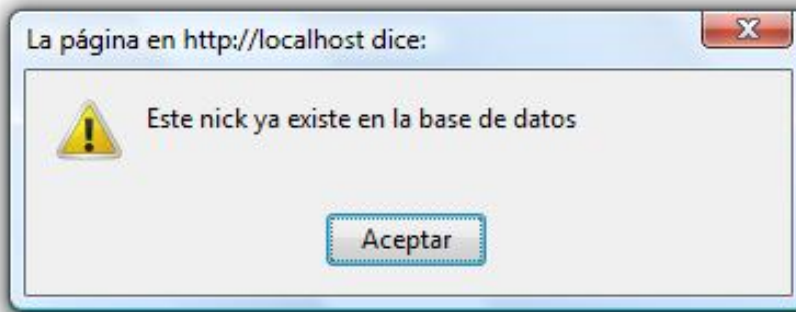


Figura A.100. Mensaje de aviso. El administrador ya existe en la base de datos.

Si por el contrario el administrador desea cambiar la contraseña de alguno de los usuarios, o la suya propia seleccionará en link “Modificar contraseña”, y le aparecerá la siguiente pantalla.

Una interfaz web para modificar un administrador. El título es "MODIFICAR ADMINISTRADOR". Debajo hay un menú desplegable con "Ion Marín[ion]". A la izquierda, el texto "Datos del Administrador" está subrayado. Hay tres campos de entrada: "Nick:" con el valor "ion", "Nueva Contraseña:" y "Confirmar Contraseña:". Debajo de los campos hay un botón "Modificar".

Figura A.101. Modificar contraseña.

Para modificar la contraseña bastará con completar los campos nueva contraseña y confirmar contraseña de forma correcta y pulsar el botón modificar.



El sistema para avisar al administrador que la tarea se ha realizado correctamente mostrará una lista con todos los usuarios y administradores existente remarcando en color rojo el miembro del cual ha sido modificada su contraseña tal y como se aprecia en la siguiente figura.

Lista de Administradores		
Nombre	Nick	Rol
Estitxu	esti	Usuario
falete	faletes	Usuario
Ion Marín	ion	Administrador
pepepepe	pepe	Administrador
Pedro	peter	Usuario
ola	ola	Administrador


Figura A.102. Administrador(o usuario) modificado, lista de administradores y usuarios de la página Web. Remarcado en rojo el miembro que ha sido modificado.

**Eliminar Administrador:** Al igual que sucede en la funcionalidad de modificar, el administrador sólo podrá eliminar así mismo y a los usuarios, es decir, no podrá eliminar a ninguno de los demás administradores.

Para eliminarse a sí mismo o a un usuario el sistema le muestra al administrador una lista con todos los usuarios y con el propio administrador como la que se muestra a continuación.

Lista de Administradores		
Nick	Nombre	Eliminar
esti	Estitxu	<input type="checkbox"/>
faletes	falete	<input type="checkbox"/>
ion	Ion Marín	<input type="checkbox"/>
peter	Pedro	<input type="checkbox"/>

Figura A.103. Lista de usuario a los que puede eliminar el administrador Ion. La lista incluye también a Ion ya que Ion se puede eliminar a sí mismo.

Pulsando en el símbolo  que corresponde al usuario que se pretende eliminar la eliminación se llevará a cabo.

Con el fin de evitar que por un error de selección a la hora de eliminar, el administrador elimine un usuario que no quería eliminar (o incluso a sí mismo), el sistema obligará al administrador a confirmar la eliminación del usuario que pretende eliminar mediante el siguiente mensaje.

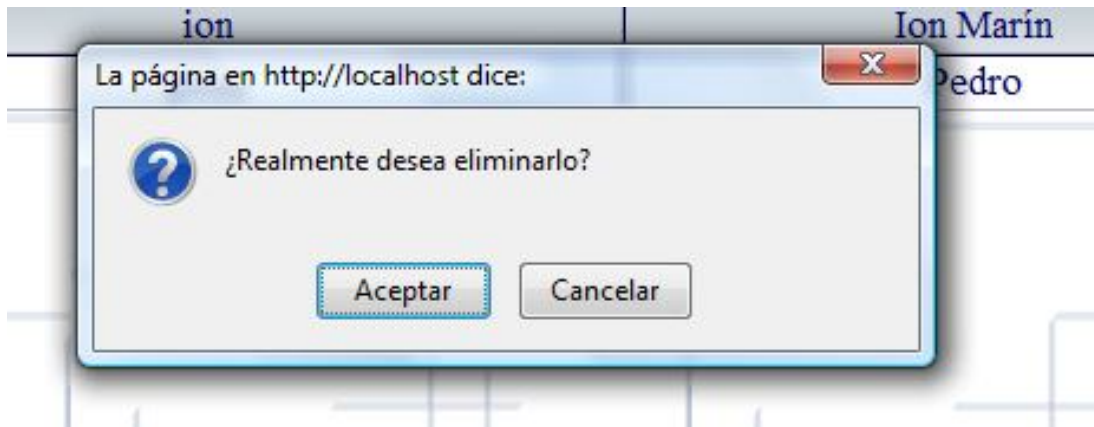


Figura A.104. Confirmación de eliminación de un usuario.

Una vez aceptado el cuadro de la imagen superior, el usuario quedará completamente eliminado.

Una vez que se elimine el usuario, el sistema mostrará al administrador la misma lista de la Figura A.103, obviamente, sin el usuario eliminado.

**Ver Movimientos:** Con el fin de que los administradores de la aplicación puedan controlar en cada momento qué movimientos son realizados, quién los realiza y el momento exacto en el que son llevados a cabo, el sistema le muestra la siguiente tabla ordenada cronológicamente.

Lista de Movimientos		
Responsable	Acción	Fecha
ion	Eliminar patrimonio: san marcos de la subpágina <a href="http://www.vc.ehu.es/docarq/Proyectos">http://www.vc.ehu.es/docarq/Proyectos</a>	2009-01-13 15:10:44
ion	Insertar Administrador:peter	2009-01-13 15:33:44
ion	Insertar Administrador:ola	2009-01-13 15:38:46
ion	Modificar Administrador:ion	2009-01-13 15:50:41
ion	Modificar contraseña Administrador:ion	2009-01-13 15:59:46
Ver anteriores		Ver siguientes

Figura A.105. Listado de los movimientos que han sido llevados a cabo.

**Ver Administradores:** Con el fin de que los administradores de la aplicación puedan conocer el personal que mantiene actualizada la aplicación, el sistema lista todos los usuario y los administradores de la aplicación tal y como aparece en la siguiente figura.

Lista de Administradores		
Nombre	Nick	Rol
Estitxu	esti	Usuario
falete	faletes	Usuario
Ion Marín	ion	Administrador
pepepepe	pepe	Administrador
Pedro	peter	Usuario

Figura A.106. Lista de administradores.

**Modificar contraseña:** Permite al administrador modificar su contraseña. Para modificar la contraseña bastará con completar los campos nueva contraseña y confirmar contraseña de forma correcta y pulsar el botón modificar, tal y como se muestra en la siguiente figura.

Datos del Administrador

Nick:

Nueva Contraseña:

Confirmar Contraseña:

Figura A.107. Modificar contraseña administrador.

El sistema para avisar al administrador que la tarea se ha realizado correctamente mostrará una lista con todos los usuarios y administradores existente remarcando en color rojo el miembro del cual ha sido modificada su contraseña tal y como se aprecia en la siguiente figura.

Lista de Administradores		
Nombre	Nick	Rol
Estitxu	esti	Usuario
falete	faletes	Usuario
Ion Marín	ion	Administrador
pepepepe	pepe	Administrador
Pedro	peter	Usuario
ola	ola	Administrador

Figura A.108. Administrador modificado, lista de administradores y usuarios de la página Web. Remarcado en rojo el miembro que ha sido modificado.

### C.2.1.5 CERRAR SESIÓN

Por último, cuando el administrador quiera abandonar la aplicación bastará con pulsar el botón situado en la parte superior izquierda “cerrar sesión”.

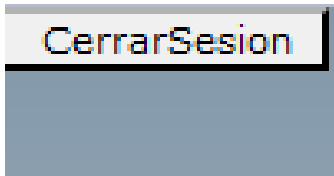


Figura A.109. Botón “CerrarSesion”. Situado en la parte superior izquierda.

Una vez pulsado dicho botón el sistema le informará al administrador del cierre de la sesión con el siguiente mensaje.

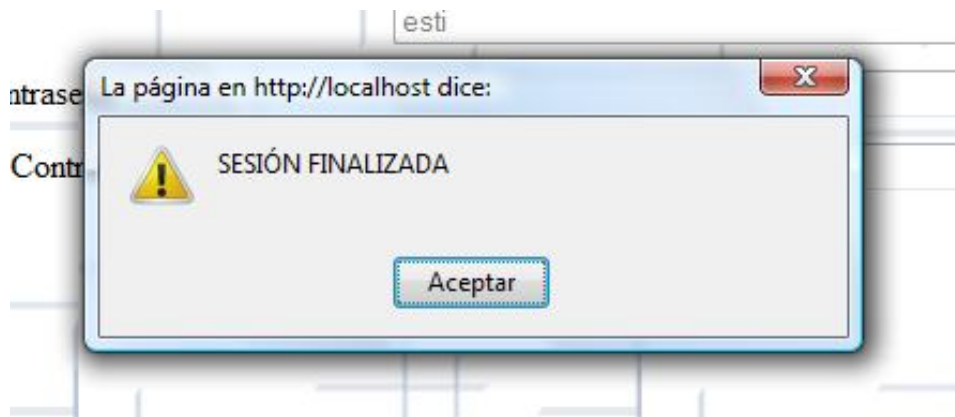


Figura A.110. Mensaje de aviso .Sesión finalizada.

## C.2.2 PERFIL USUARIO

Se trata del perfil con capacidad de participación limitada en la aplicación. Todos los usuarios tendrán el mismo poder de participación en cuanto a funcionalidades. Sin embargo, sólo podrán aplicar esas funcionalidades a los sitios Web en los que tengan permisos para actuar. A continuación se irán describiendo cada una de las funcionalidades en el orden exacto en el que aparecen en el menú (parte izquierda de la pantalla). Cuando el usuario accede a la aplicación la primera pantalla que se encuentra es una pantalla de bienvenida, en la que en el centro de la misma se encuentra un mensaje de bienvenida y en la parte izquierda de dicha pantalla se sitúa el menú que se mantendrá en dicha parte izquierda a lo largo de toda la aplicación.

### C.2.2.1 FUNCIONALIDADES SITIO WEB

Las tres primeras funcionalidades que aparecen en la aplicación son las relacionadas con el sitio Web. Antes de comenzar a explicar cada una de ellas, nos introduciremos en el concepto que le hemos dado a “sitio Web”. Definimos sitio Web como la ubicación en la red donde aparecen todas y cada una de las páginas de dicho sitio Web, es decir, la página principal de una página Web. Tomando como ejemplo, la página del Laboratorio de Documentación Geométrica del Patrimonio (UPV-EHU), la dirección del sitio Web sería <http://www.vc.ehu.es/docarq/> que es la página principal, cada una del resto de páginas Web de dicho sitio Web, serán clasificadas como sub-páginas Web.

Una vez citado esto se explicarán las funcionalidades referentes al sitio Web:  
Nuevo Modificar sitio Web.



Figura A.111. Representa el menú de la aplicación, concretamente la funcionalidad de Sitio Web, la cual se explica a continuación.

**Modificar sitio Web:** Modificar sitio Web nos permite modificar un sitio Web existente en nuestro sistema. Esta funcionalidad consta de dos partes. La primera de ellas es una lista desplegable en donde aparecen todos y cada uno de los sitiosWeb existentes en el sistema **a los que el usuario tiene permiso para acceder**. Siempre que al usuario se le otorgue privilegio sobre un nuevo sitio Web aparecerá en dicha lista. De dicha lista desplegable se seleccionará aquel sitio Web del que queramos modificar sus datos. Para que esto que se ha explicado quede visualmente explicado a continuación se muestran dos figuras, una que recoge las dos partes de la funcionalidad que se está explicando y otra que recoge únicamente la lista desplegable.

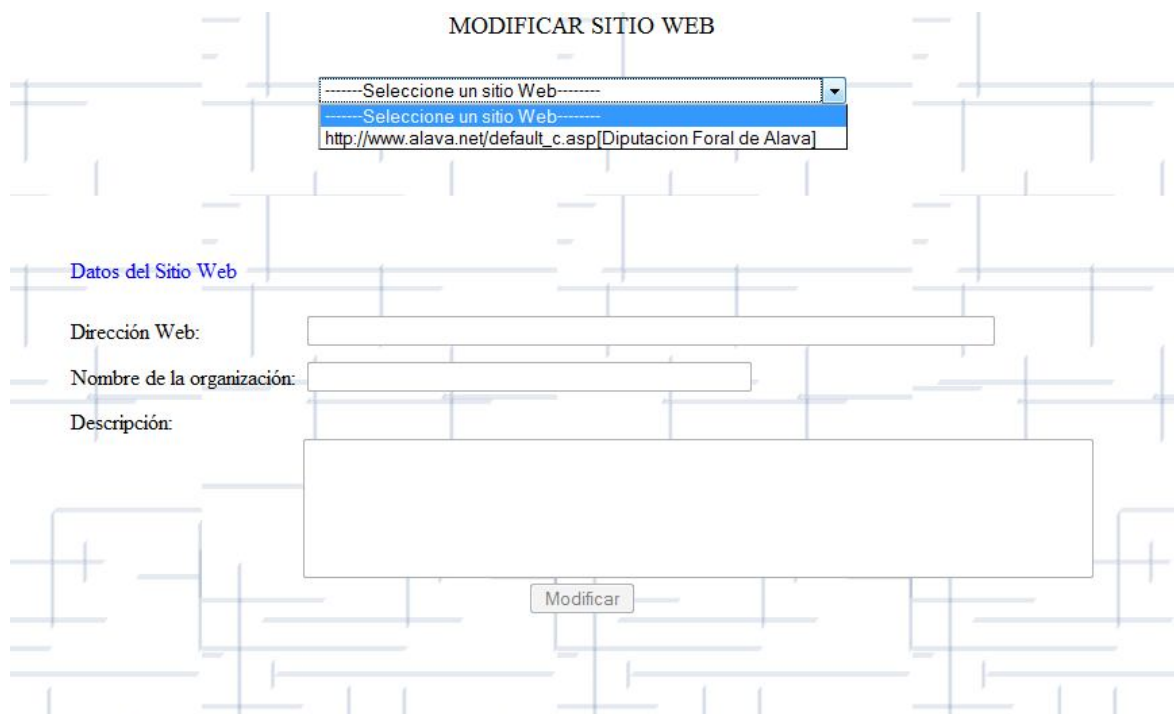


Figura A.112. Modificar sitio Web.

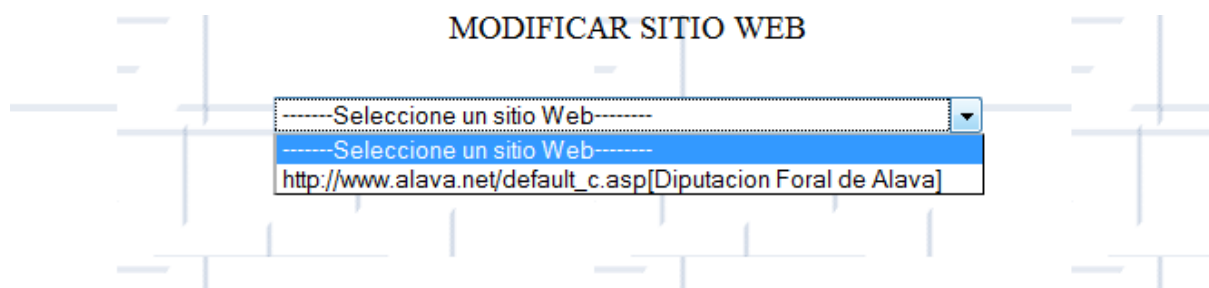


Figura A.113. Lista desplegable de Modificar sitio Web.

Una vez que el “usuario” seleccione un sitio Web automáticamente el sistema cargará sus datos en los campos del formulario que se muestran en la Figura A.114.

MODIFICAR SITIO WEB

http://www.alava.net/default\_c.asp[Diputacion Foral de Alava] ▾

**Datos del Sitio Web**

Dirección Web:

Nombre de la organización:

Descripción:

Figura A.114. Modificar sitio Web. Datos de sitio Web preparados para modificar.

Una vez que los datos están cargados bastará con modificar aquellos que se quieran modificar. Una vez modificados pulsamos el botón modificar y el sistema nos actualizará los datos en la base de datos. Para asegurarnos de que realmente los ha modificado el sistema nos mostrará el siguiente mensaje.

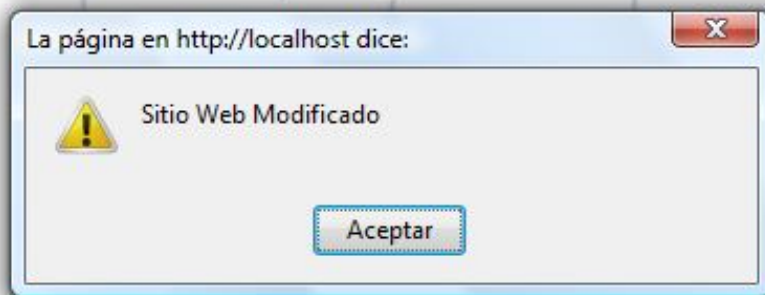


Figura A.115. Sitio Web Modificado.

Posteriormente nos volverá a mostrar la página donde se modifica el sitio Web con la nueva información que ha sido actualizada.

Para prevenir que el usuario cambie la dirección Web del sitio Web, se bloqueará ese campo



### C.2.2.2 FUNCIONALIDADES SUB-PÁGINAS WEB

Las siguientes funcionalidades que se van explicar son las referentes a las sub-páginas Web. Recordemos que un sitio Web consta de una o muchas sub-páginas Web. Las funcionalidades de esta parte de la aplicación son: Nueva sub-páginasWeb, Modificar sub-páginasWeb y Eliminar sub-páginasWeb.

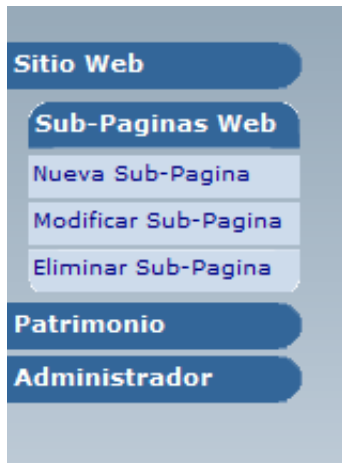


Figura A.116. Representa el menú de la aplicación, concretamente las funcionalidades de Sub-Páginas Web, las cuales se explican a continuación.

**Nueva sub-páginasWeb:** Esta funcionalidad nos permite añadir una sub-páginas Web a un sitio Web existente. Para ello deberemos seleccionar en una lista desplegable, en donde aparecen todos y cada uno de los sitiosWeb existentes en el sistema a los que el usuario tiene permiso de acceso, el sitio Web en el que queremos introducir nuestra subpágina Web.

NUEVA SUBPAGINA WEB

Indique el sitio Web en el que quiera insertar una subpágina

-----Seleccione un sitio Web-----  
 -----Seleccione un sitio Web-----  
[http://www.alava.net/default\\_c.asp](http://www.alava.net/default_c.asp)[Diputacion Foral de Alava]

**Datos de la Página Web**

Dirección de la Sub-Página Web:

Descripción:

Disponible sobre este tema en la Web:

Figura A.117. Introducir nueva sub-página Web.

Tal y como se puede observar en la Figura A.117 el usuario ha seleccionado un sitio Web, concretamente en el que desea introducir una nueva sub-página Web, y ahora bastará con introducir los datos de dicha subpágina Web y dicha subpágina quedará registrada.

Para hacérselo saber al usuario, el sistema lista por pantalla todas las subpáginas del sitio Web en el que se ha insertado la nueva subpágina, subrayando esta con color rojo, tal y como aparece en la Figura A.118.

Lista de Subpáginas del sitio Web <a href="http://www.alava.net/default_c.asp">http://www.alava.net/default_c.asp</a>	
Dirección Web	Descripción
<a href="http://web.alava.net/PatrimonioHistorico/patrimonio/patrimoniohome.htm">http://web.alava.net/PatrimonioHistorico/patrimonio/patrimoniohome.htm</a>	Patrimonio Histórico Arquitectónico de la Diputación Foral de Álava
<a href="http://www.alava.net/default_c.asp/pagina2.php">http://www.alava.net/default_c.asp/pagina2.php</a>	Reúne información diversa sobre iglesias.

Figura A.118. Subpáginas Web del sitio Web [http://www.alava.net/default\\_c.asp](http://www.alava.net/default_c.asp) .En rojo aparece la subpágina que se acaba de insertar.

Si el usuario intenta introducir una subpágina Web que ya existía en el sitio Web seleccionado, el sistema no se lo permitirá y le mostrará el siguiente mensaje de aviso.

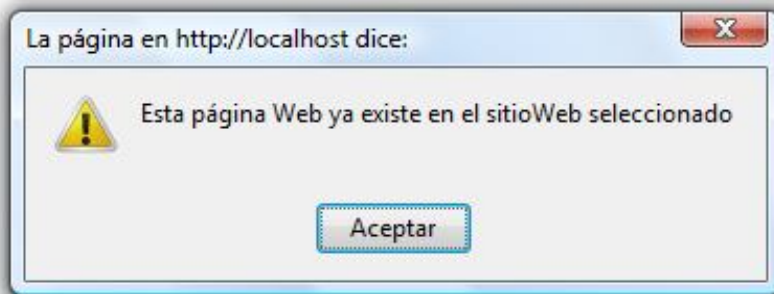


Figura A.119. Mensaje de aviso. La subpágina Web ya existía.

**Modificar sub-páginas Web:** Como su propio nombre indica esta funcionalidad nos permite modificar una sub-página Web de un sitio Web existente. Para ello deberemos seleccionar, como en el caso anterior, en una lista desplegable, en donde aparecen todos y cada uno de los sitios Web existentes en el sistema a los que el usuario tiene permiso de acceso, el sitio Web del que queremos modificar nuestra subpágina Web. Una vez seleccionado de la primera lista desplegable el sitio Web del que queremos modificar la subpágina Web, se debe seleccionar en una nueva lista desplegable la subpágina Web que va a ser modificada. Obviamente en esta segunda lista desplegable únicamente contendrá aquellas subpáginas Web del sitio Web seleccionado en la primera lista desplegable.



Figura A.120. Muestra la lista desplegable de subpáginas Web una vez seleccionado un sitio Web determinado.

Como se puede apreciar en la Figura A.120 la lista desplegable de sub-páginas Web muestra sólo aquellas que pertenecen al sitio Web seleccionado en la primera lista desplegable.

Una vez que se ha seleccionado la subpágina Web los campos del formulario se rellenarán de forma automática con los datos de dicha subpágina.

Indique el sitio Web del que quiera modificar una subpágina

Indique la SubPágina que quiera modificar

Datos de la Sub-Página Web

Dirección Web:

Descripción:

Disponible en la Web:

Disponible en la organización:

Figura A.121. Modificar subpágina Web.

El usuario modifica los datos que desee sobre la subpágina Web y pulsará el botón “modificar”. Una vez que haga esto los datos quedarán registrados en la base de datos. Para mostrar al usuario que la tarea ha sido realizada le mostrará un mensaje como el siguiente.

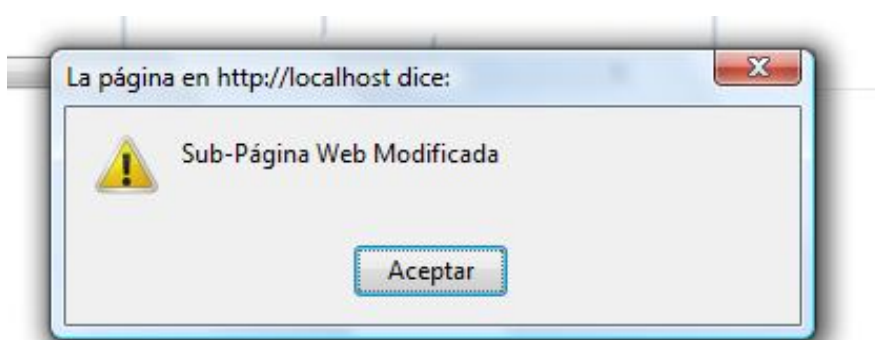


Figura A.122. Mensaje de aviso. La subpágina Web ha sido modificada.

Si el usuario intenta modificar la dirección de la subpágina Web asignándole la dirección de otra subpágina Web que ya existe en dicho sitio Web, el sistema no se lo permitirá y le mostrará un mensaje como el siguiente.

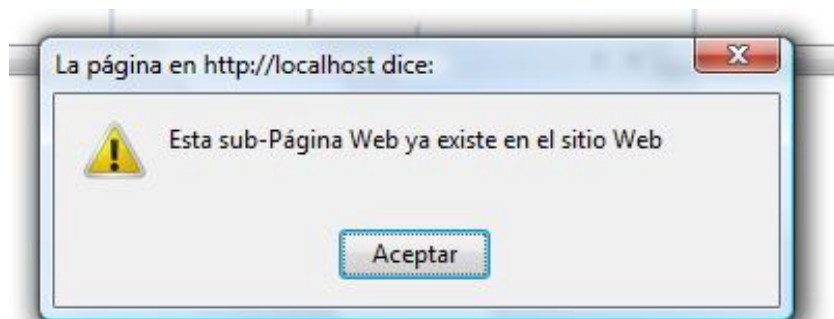


Figura A.123. Mensaje de aviso. La dirección de la subpágina Web que se pretendía modificar ya pertenece a otra subpágina Web.


**Eliminar sub-páginas Web:** Como su propio nombre indica esta funcionalidad nos permite eliminar una sub-página Web de un sitio Web existente. Para ello deberemos seleccionar, como en los casos anteriores, en una lista desplegable, en donde aparecen todos y cada uno de los sitios Web existentes en el sistema a los que el usuario tiene permiso de acceso, el sitio Web del que queremos eliminar nuestra subpágina Web. Una vez seleccionado de la primera lista el sitio Web del que vamos a eliminar una subpágina, el sistema listará todas las subpáginas Web de dicho sitio Web junto con una imagen , que permitirá al usuario eliminar la subpágina que desee.



Figura A.124. Lista de sub-páginas Web del sitio Web seleccionado en la lista desplegable superior.

Con el fin de evitar que por un error de selección a la hora de eliminar el usuario elimine una subpágina Web que no quería eliminar, el sistema obligará al usuario a confirmar la eliminación del elemento que pretende eliminar mediante el siguiente mensaje.

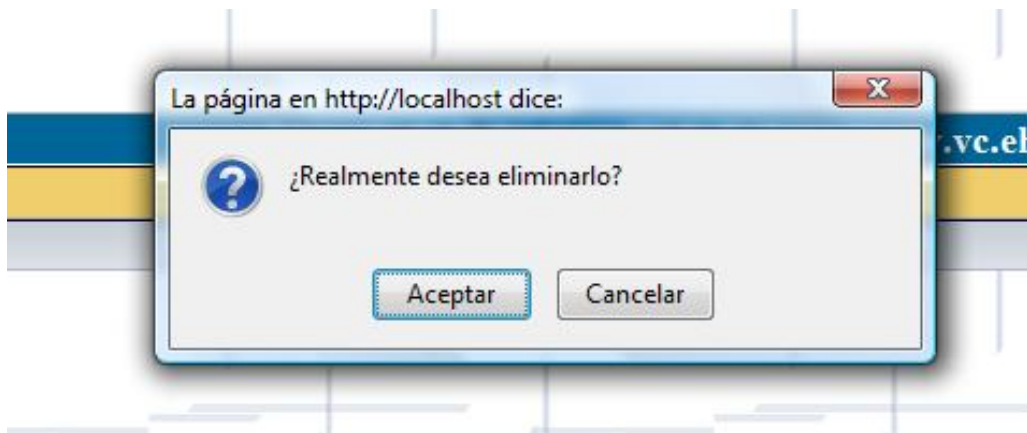


Figura A.125. Confirmación de eliminación de una subpágina Web.

Una vez aceptado el cuadro de la imagen superior, la subpágina Web quedará completamente eliminada, y arrastrará a su eliminación todos los elementos patrimoniales que contengan dicha sub-página Web. Se trata de una eliminación en cascada.

Una vez que se elimine el subpágina Web, el sistema mostrará al usuario la misma lista de la Figura A.124, obviamente, sin la subpágina Web eliminada.

### C.2.2.3 FUNCIONALIDADES PATRIMONIO

Las siguientes funcionalidades que se van explicar son las referentes a las sub-páginas Web. Recordemos que un sitio Web consta de una o muchas sub-páginas Web. Las funcionalidades de esta parte de la aplicación son: Nuevo Patrimonio, Modificar Patrimonio y Eliminar Patrimonio.



Figura A.126. Representa el menú de la aplicación, concretamente las funcionalidades de Patrimonio, las cuales se explican a continuación.

**Nuevo Patrimonio:** Esta funcionalidad nos permite añadir un nuevo elemento patrimonial a una subpágina Web existente. Para ello deberemos seleccionar en una lista desplegable, en donde aparecen todos y cada uno de los sitios Web existentes en el sistema a los que el usuario tiene permiso de acceso, el sitio Web en el que se encuentra la subpágina Web en donde queremos introducir el elemento patrimonial. Una vez seleccionado el sitio Web, el administrador debe seleccionar en otra lista desplegable la subpágina Web en la cual quiere introducir el elemento patrimonial. Obviamente en esta segunda lista desplegable sólo aparecen las subpáginas Web que existen dentro del sitio Web seleccionado. En la Figura A.127 se puede observar claramente como existen dos listas desplegables, en las cuales se selecciona el sitio Web y la subpágina donde se insertará el elemento patrimonial.

Indique el sitio Web del que quiera añadir un elemento patrimonial

Indique la SubPágina en la que quiera insertar un elemento patrimonial

-----Seleccione una Sub-Página Web-----

-----Seleccione una Sub-Página Web-----

NUEVO ELEMENTO PATRIMONIAL

Datos del Elemento Patrimonial

Seleccione el lugar donde se quiere clasificar el contenido:

- excavaciones
- arquitectura religiosa
- arquitectura civil
- elementos menores
- objetos

Figura A.127. Nuevo elemento patrimonial. Seleccionado el sitio Web y seleccionándose la subpágina Web donde se va insertar el elemento patrimonial.

Una vez seleccionado tanto el sitio Web como la subpágina Web, seleccionamos el tipo de elemento patrimonial que se va a insertar. Cabe destacar que esta clasificación que aparece en la Figura A.127, **puede ser modificable en función de la base de datos**, pero esto ya ha sido comentado en los Manuales de mantenimiento (C.1.1). En función del elemento patrimonial que se escoja aparecerán una serie de atributos u otros, lógicamente en función de los atributos que dicho elemento patrimonial seleccionado disponga en la base de datos.



Indique el sitio Web del que quiera añadir un elemento patrimonial

Indique la SubPágina en la que quiera insertar un elemento patrimonial

Datos del Elemento Patrimonial

Seleccione el lugar donde se quiere clasificar el contenido:

- excavaciones
- arquitecturareligiosa
- arquitecturacivil
- elementosmenores
- objetos

Nombre:

Periodo:

Tipo\_datos:

Municipio:

Poblacion:

Provincia:

Tipo\_arquitecturaReligiosa:

Figura A.128. Nuevo elemento patrimonial. Seleccionado el sitio Web y la subpágina Web donde se va insertar el elemento patrimonial. También ha sido seleccionado el tipo de elemento patrimonial

Una vez que se rellenen los campos con la información relativa al elemento patrimonial, bastará con pulsar el botón “introducir” y el elemento patrimonial será introducido. Para mostrar al usuario que la tarea ha sido realizada le mostrará un mensaje como el siguiente.

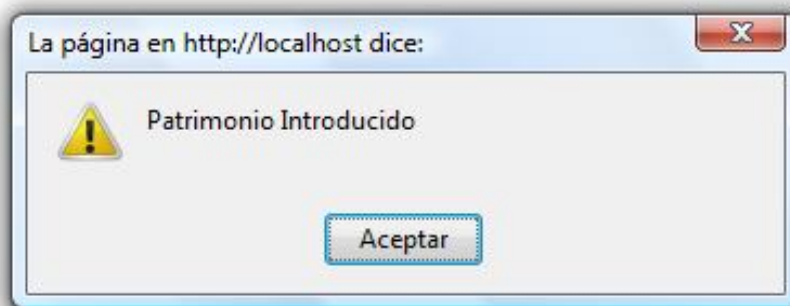


Figura A.129. Mensaje de aviso. El elemento patrimonial ha sido introducido.

**Modificar Patrimonio:** Esta funcionalidad nos permite modificar un elemento patrimonial de una subpágina Web existente. Para ello deberemos seleccionar en una lista desplegable, en donde aparecen todos y cada uno de los sitios Web existentes en el sistema a los que el usuario tiene permiso de acceso, el sitio Web en el que se encuentra la subpágina Web en donde queremos introducir el elemento patrimonial. Una vez

seleccionado el sitio Web, el administrador debe seleccionar en otra lista desplegable la subpágina Web en la cual se encuentra el elemento patrimonial que quiere modificar. Obviamente en esta segunda lista desplegable sólo aparecen las subpáginas Web que existen dentro del sitio Web seleccionado. En la Figura A.130 se puede observar claramente como existen dos listas desplegables, en las cuales se selecciona el sitio Web y la subpágina donde se encuentra el elemento patrimonial que se desea modificar.

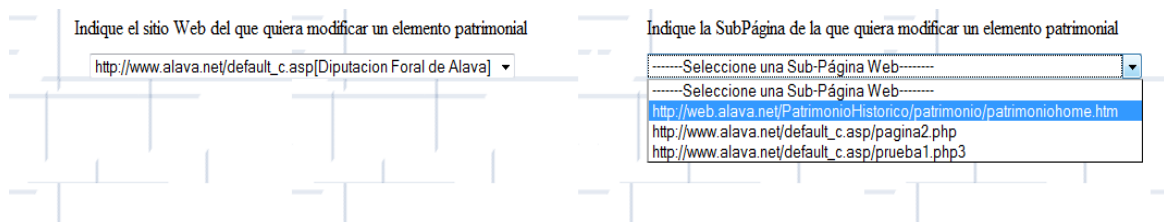


Figura A.130. Modificar elemento patrimonial. Seleccionado el sitio Web y seleccionándose la subpágina Web en la cual se encuentra el elemento patrimonial que se va a modificar.

Una vez seleccionado tanto el sitio Web como la subpágina Web en donde se encuentra el elemento patrimonial a modificar, el sistema nos mostrará una lista con los elementos patrimoniales de la subpágina seleccionada.

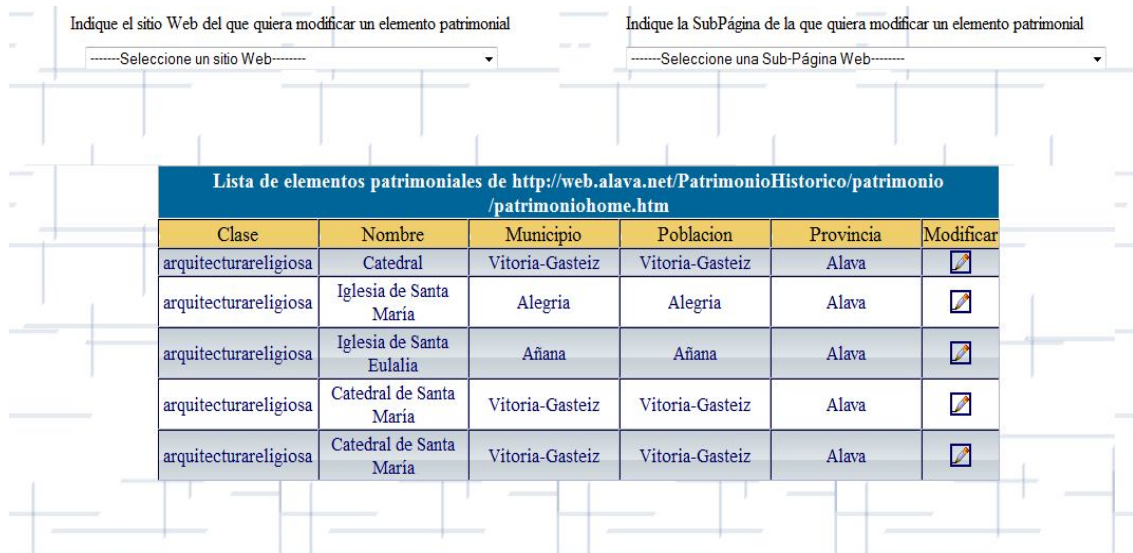



Figura A.131. Modificar elemento Patrimonial. Listado de elementos patrimoniales de la subpágina seleccionada.

Junto a cada elemento patrimonial nos encontramos el siguiente símbolo . Pulsando dicho símbolo correspondiente al elemento patrimonial que se quiere modificar nos encontremos la siguiente pantalla.

Datos del elemento patrimonial

Nombre:	Iglesia de Santa María
Periodo:	S.XVI
Tipo_datos:	Descripción
Municipio:	Alegria
Poblacion:	Alegria
Provincia:	Alava
Tipo_arquitecturaReligiosa:	Iglesia

Modificar

Figura A.132. Modificar elemento Patrimonial. Datos del elemento patrimonial seleccionado para ser modificado.

Una vez que el usuario realice los cambios que crea necesario, pulsará el botón “modificar”, y el sistema modificará dichos datos.

Para mostrar al usuario que la tarea ha sido realizada le mostrará un mensaje como el siguiente.

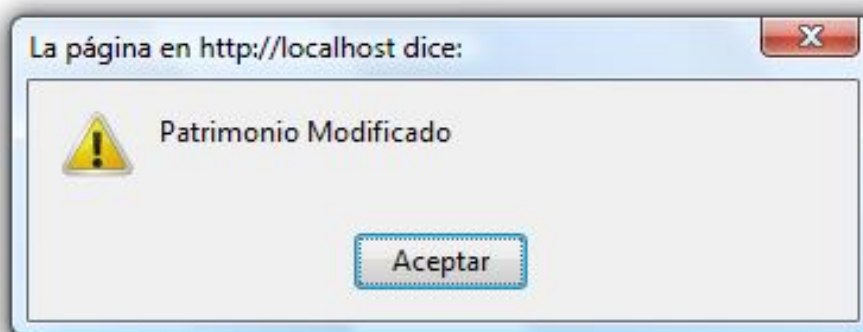


Figura A.133. Mensaje de aviso. El elemento patrimonial ha sido modificado.

**Eliminar Patrimonio:** Esta funcionalidad nos permite eliminar un elemento patrimonial de una subpágina Web existente. Para ello deberemos seleccionar en una lista desplegable, en donde aparecen todos y cada uno de los sitios Web existentes en el sistema a los que el usuario tiene permiso de acceso, el sitio Web en el que se encuentra la subpágina Web en donde queremos eliminar el elemento patrimonial. Una vez

seleccionado el sitio Web, el administrador debe seleccionar en otra lista desplegable la subpágina Web en la cual se encuentra el elemento patrimonial que quiere eliminar. Obviamente en esta segunda lista desplegable sólo aparecen las subpáginas Web que existen dentro del sitio Web seleccionado. En la Figura A.134 se puede observar claramente como existen dos listas desplegables, en las cuales se selecciona el sitio Web y la subpágina donde se encuentra el elemento patrimonial que se desea eliminar.

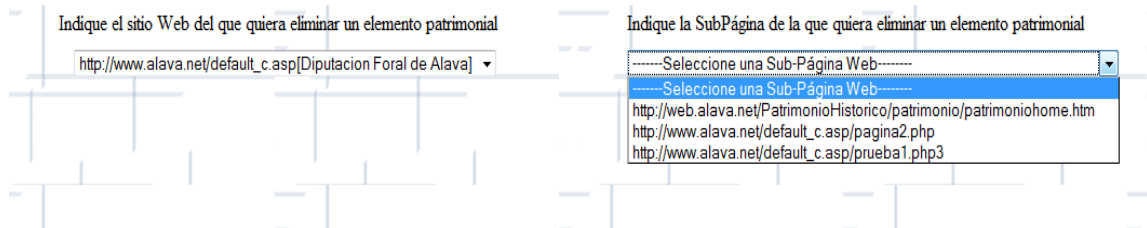


Figura A.134. Eliminar elemento patrimonial. Seleccionado el sitio Web y seleccionándose la subpágina Web en la cual se encuentra el elemento patrimonial que se va a eliminar.

Una vez que es seleccionado tanto el sitio Web como la subpágina Web en donde se encuentra el elemento patrimonial que se desea eliminar, el sistema nos muestra una lista con todos los elementos patrimoniales de la subpágina Web seleccionada en la lista desplegable.

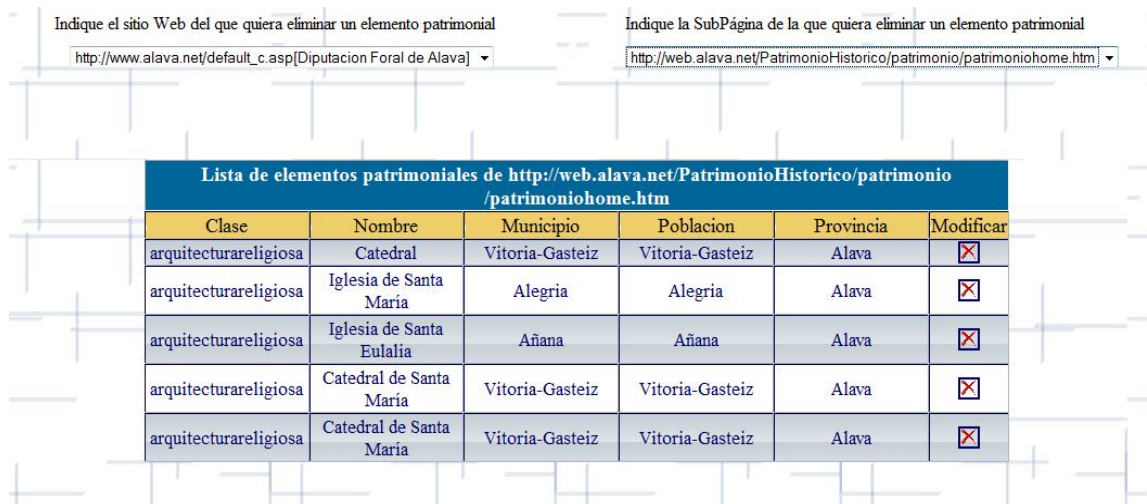


Figura A.135. Eliminar elemento Patrimonial. Listado de elementos patrimoniales de la subpágina seleccionada.

Junto a cada elemento patrimonial nos encontramos el siguiente símbolo . Pulsando dicho símbolo correspondiente al elemento patrimonial que se quiere eliminar podremos eliminar dicho elemento patrimonial.

Con el fin de evitar que por un error de selección a la hora de eliminar el usuario elimine una subpágina Web que no quería eliminar, el sistema obligará al usuario a confirmar la eliminación del elemento que pretende eliminar mediante el siguiente mensaje.

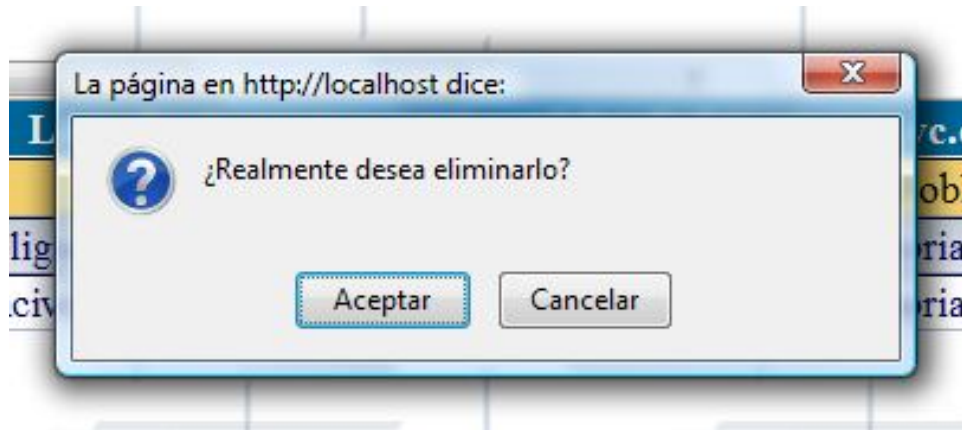


Figura A.136. Confirmación de eliminación de un elemento patrimonial.

Una vez aceptado el cuadro de la imagen superior, el elemento patrimonial quedará completamente eliminado.

Una vez que se elimine el elemento patrimonial, el sistema mostrará al usuario la misma lista de la Figura A.135, obviamente, sin el elemento patrimonial eliminado.

#### C.2.2.4 FUNCIONALIDADES ADMINISTRADOR

Las siguientes funcionalidades que se van explicar son las referentes a los usuarios de la aplicación que se ha desarrollado a lo largo de este proyecto. Con el fin de que dicho usuario conozca su potestad a la hora de llevar a cabo diversas funcionalidades relacionadas con las tareas del administrador, se van a explicar detalladamente cada una de las funcionalidades.



Figura A.137. Representa el menú de la aplicación, concretamente las funcionalidades de Administrador, las cuales se explican a continuación.

**Ver Movimientos:** Con el fin de que también los usuarios de la aplicación puedan observar en cada momento qué movimientos son realizados, quién los realiza y el momento exacto en el que son llevados a cabo, el sistema le muestra la siguiente tabla ordenada cronológicamente.

Lista de Movimientos		
Responsable	Acción	Fecha
ion	Eliminar patrimonio: san marcos de la subpágina http://www.vc.ehu.es/docarq/Proyectos	2009-01-13 15:10:44
ion	Insertar Administrador:peter	2009-01-13 15:33:44
ion	Insertar Administrador:ola	2009-01-13 15:38:46
ion	Modificar Administrador:ion	2009-01-13 15:50:41
ion	Modificar contraseña Administrador:ion	2009-01-13 15:59:46
Ver anteriores		Ver siguientes

Figura A.138. Listado de los movimientos que han sido llevados a cabo.

**Ver Administradores:** Con el fin de que también los usuarios de la aplicación puedan conocer el personal que mantiene actualizada la aplicación, el sistema lista todos los usuario y los administradores de la aplicación tal y como aparece en la siguiente figura.

Lista de Administradores		
Nombre	Nick	Rol
Estitxu	esti	Usuario
falete	faletes	Usuario
Ion Marín	ion	Administrador
pepepepe	pepe	Administrador
Pedro	peter	Usuario

Figura A.139. Lista de administradores.

**Modificar contraseña:** Permite al usuario modificar su contraseña. Para modificar la contraseña bastará con completar los campos nueva contraseña y confirmar contraseña de forma correcta y pulsar el botón modificar, tal y como se muestra en la siguiente figura.

Datos del Administrador

Nick:

Nueva Contraseña:

Confirmar Contraseña:

Figura A.140. Modificar contraseña administrador.

El sistema para avisar al usuario que la tarea se ha realizado correctamente mostrará una lista con todos los usuarios y administradores existente remarcando en color rojo el miembro del cual ha sido modificada su contraseña tal y como se aprecia en la siguiente figura.

Lista de Administradores		
Nombre	Nick	Rol
Estitxu	esti	Usuario
falete	faletes	Usuario
Ion Marín	ion	Administrador
pepepepe	pepe	Administrador
Pedro	peter	Usuario
ola	ola	Administrador

Figura A.141. Contraseña usuario modificada, lista de administradores y usuarios de la página Web. Remarcado en rojo el miembro que ha sido modificado.



### C.2.2.5 CERRAR SESIÓN

Por último, cuando el usuario quiera abandonar la aplicación bastará con pulsar el botón situado en la parte superior izquierda “cerrar sesión”.

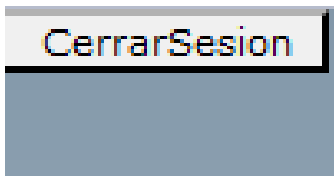


Figura A.142. Botón “CerrarSesion”. Situado en la parte superior izquierda.

Una vez pulsado dicho botón el sistema le informará al usuario del cierre de la sesión con el siguiente mensaje.

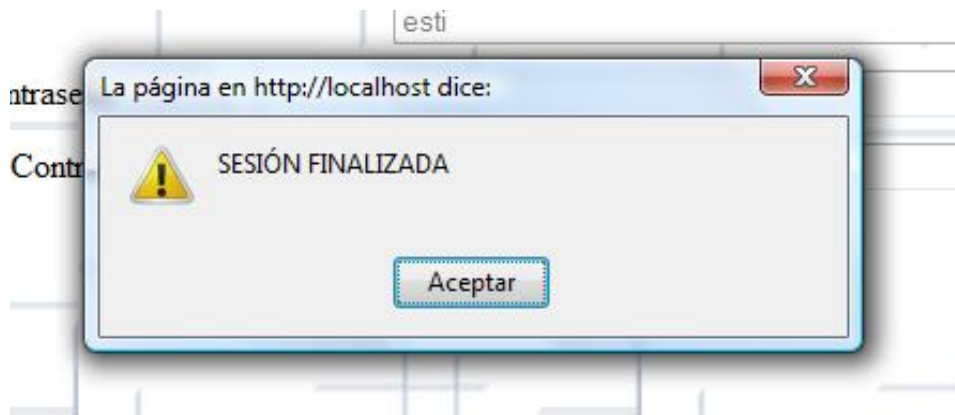


Figura A.143. Mensaje de aviso .Sesión finalizada.

## D. ACTAS DE REUNIÓN

### D.1. ACTAS DE REUNIÓN CLIENTE

#### D.1.1 Acta del martes 2 de septiembre de 2008

Universidad del País Vasco/  
Euskal Herriko Unibertsitatea  
(UPV/EHU)



### ACTA DE REUNIÓN

<b>TIPO DE REUNIÓN:</b>	Ordinaria, de seguimiento
<b>FECHA:</b>	Martes, 2 de septiembre de 2008.
<b>HORA:</b>	16:00 horas.
<b>LUGAR:</b>	Despacho de José Manuel Valle situado en el aulario de Vitoria-Gasteiz, perteneciente a la Universidad del País Vasco (UPV/EHU).
<b>ASISTENTES:</b>	Ion Marín y José Manuel Valle
<b>CONVOCADA POR:</b>	Ion Marín.

#### ORDEN DEL DÍA

1. Situación actual de propuesta del proyecto.
2. Situación actual del pre-proyecto.
3. Puesta a punto del comienzo de la parte de investigación del proyecto.
4. Dudas y preguntas sobre el Proyecto de Fin de Carrera.

#### CONCLUSIONES

##### **1. Situación actual de propuesta del proyecto.**

José Manuel, encargado de las relaciones con el cliente y promotor de la idea del proyecto, se interesa por la situación de la propuesta del proyecto. Dicha

propuesta fue realizada por Ion durante el mes de Julio, y fue corregida/mejorada por el propio Jose Manuel en dos ocasiones, dando como resultado el documento de propuesta de proyecto que fue enviado al tutor del proyecto, José Miguel Blanco, con el fin de que éste lo valorase en función de los contenidos que se van a desarrollar.

José Miguel dio el visto bueno, por lo que en principio el proyecto se podía poner en marcha.

## **2. Situación actual del pre-proyecto.**

Además de la propuesta José Manuel mandó a Ion realizar un pequeño cronograma en el que se indicaran de manera detallada las fases en el que subdividía el proyecto. Ion lo elaboró y lo envió a José Miguel. El cronograma sufrió varias modificaciones, debidas todas ellas a sugerencias del propio José Miguel. Éste además propuso a Ion comenzar con el Documento de Objetivos del Proyecto (D.O.P).

Ion pone al día a José Manuel sobre la situación en la que se encuentra su D.O.P, el cual está bastante avanzado. Sin embargo Ion considera que necesita mayor conocimiento de lo que se quiere desarrollar para completar el D.O.P. de manera satisfactoria.

## **3. Puesta a punto del comienzo de la parte de investigación del proyecto.**

José Manuel enseña a Ion, una serie de páginas Web como son mundoviaje.com, edreams.com, atrapalo.com ... y sugiere a Ion que la solución para este proyecto puede ser similar a la que presentan esas páginas.

Ion, de acuerdo con José Manuel, sugiere que eso se trata de Bases de Datos Distribuidas, por lo que durante las dos semanas en las que José Manuel y Álvaro van a estar ausentes, se compromete a informarse todo lo posible sobre este tipo de bases de datos.

## **4. Dudas y preguntas sobre el Proyecto de Fin de Carrera.**

Ninguna que destacar, salvo que José Manuel y Álvaro van ausentarse durante dos semanas por motivos laborales.

## D.1.2 Acta del lunes 22 de septiembre de 2008

Universidad del País Vasco/  
Euskal Herriko Unibertsitatea  
(UPV/EHU)



### ACTA DE REUNIÓN

- TIPO DE REUNIÓN:** Ordinaria, de seguimiento
- FECHA:** Lunes, 22 de septiembre de 2008.
- HORA:** 10:00 horas.
- LUGAR:** Edificio de investigación junto al aulario de Vitoria-Gasteiz, perteneciente a la Univeridad del País Vasco (UPV/EHU).
- ASISTENTES:** Ion Marín y Álvaro
- CONVOCADA POR:** Ion Marín.

### ORDEN DEL DÍA

1. Puesta al día de la situación del proyecto.
2. Revisión del D.O.P.
3. Existencia de organizaciones interesados en unirse al proyecto.
4. Dudas y preguntas sobre el Proyecto de Fin de Carrera.

### CONCLUSIONES

- 1. Puesta al día de la situación del proyecto.**

Cada parte expone las noticias nuevas que se han producido desde el último contacto entre ambas.

Álvaro comenta el posible interés de varias organizaciones en unirse al proyecto. Por otra parte, Ion explica los puntos más relevantes tratados en la reunión mantenida con José Miguel, tutor del proyecto, el 12-09-2008.

## **2. Revisión del D.O.P.**

Tanto Álvaro como José Manuel estuvieron revisando el D.O.P. realizado por Ion la semana previa a la reunión. Álvaro sugerirá alguna pequeña mejora, más por la forma en que éste está redactado que por su contenido. Sin embargo, también cree que la planificación, en cuanto a horas se refiere, no es adecuada, tachándola de optimista.

## **3. Existencia de organizaciones interesadas en unirse al proyecto.**

La semana previa a la reunión, Álvaro y José Manuel redactaron un escrito en el cual se explicaba en qué consistía el proyecto y porqué era importante que se llevara a cabo. Este escrito se envía a diversas organizaciones que tratan el patrimonio en Álava con el fin de que, siempre y cuando estén interesadas, puedan unirse al proyecto.

Álvaro me entrega una lista con las organizaciones a las que se les ha enviado la propuesta, y me comenta que en los próximos días mantendrá conversaciones con la inmensa mayoría, con el fin de que el proceso se acelere. Por otra parte, me pregunta qué es lo que necesitamos saber sobre el modo de almacenamiento de los datos de las diferentes organizaciones, para conseguir esa información a través de las propias organizaciones. Para asegurarnos de hacer las cosas bien desde el principio, coincidimos en que lo mejor es que se realice una lista con las cosas que necesitamos saber. De dicha lista me encargo yo, aunque aclaro en la reunión, que me ayudará del tutor del proyecto.

## **4. Dudas y preguntas sobre el Proyecto de Fin de Carrera.**

Ninguna importante a destacar.

### D.1.3 Acta del viernes 3 de octubre de 2008

Universidad del País Vasco/  
Euskal Herriko Unibertsitatea  
(UPV/EHU)



## ACTA DE REUNIÓN

- TIPO DE REUNIÓN:** Ordinaria, de seguimiento
- FECHA:** Viernes, 3 de octubre de 2008.
- HORA:** 11:00 horas.
- LUGAR:** Edificio de investigación junto al aula de Vitoria-Gasteiz, perteneciente a la Universidad del País Vasco (UPV/EHU).
- ASISTENTES:** Ion Marín, José Manuel Valle y Álvaro Rodríguez
- CONVOCADA POR:** Ion Marín.

### ORDEN DEL DÍA

1. Situación del proyecto a fecha de hoy.
2. Situación de las entrevistas con las organizaciones presuntamente interesadas en el proyecto.
3. Intercambio de ideas sobre la información necesaria.
4. Pasos a seguir en los próximos días.
5. Dudas y preguntas sobre el Proyecto de Fin de Carrera.

### CONCLUSIONES

#### **1. Situación del proyecto a fecha de hoy.**

El alumno, Ion Marín explica los progresos que ha tenido en su investigación sobre una arquitectura que solucione el problema que presenta el proyecto. Comenta a José Manuel y a Álvaro que ha encontrado una posible solución, una *Arquitectura para Federación de Bases de Datos Documentales basada en*

*Ontologías*, pero que para poder llevarla a cabo necesita total libertad para consultar las fuentes de datos de todas y cada una de las organizaciones que están interesadas en este proyecto.

Con el fin de que tanto Álvaro como José Manuel, quien abandona la reunión antes de tiempo, puedan hacerse una idea de cómo es esa posible solución, Ion explica cuál sería el procedimiento a seguir e introduce conceptos como árbol de conceptos o árbol de correspondencias.

## **2. Situación de las entrevistas con las organizaciones presuntamente interesadas en el proyecto.**

Durante la semana, mientras Ion intentaba encontrar una arquitectura que solucionara el problema que se presenta en el proyecto, Álvaro ha estado manteniendo reuniones con diferentes organizaciones con el fin de que éstas le proporcionaran información acerca de sus fuentes de datos. A medida que Álvaro ha ido recapitulando información, se la ha ido enviando a Ion, por lo que éste último ya disponía de algunos datos antes de la reunión.

Sin embargo, la reunión se centra en la información recogida sobre la biblioteca virtual de la universidad del País Vasco. Álvaro cree que puede ser un modelo a seguir, sin embargo Ion le sugiere que no se fije en dicho modelo, ya que no tiene nada que ver con lo que el presente proyecto requiere. Esto es debido, a que el servicio web de la biblioteca se trata de una base de datos centralizada o bien una base de datos distribuida, pero no tiene nada que ver con una integración de diferentes fuentes de datos.

## **3. Intercambio de ideas sobre la información necesaria.**

Ion sugiere a Álvaro que se centren únicamente en dos fuentes de datos, a las cuales, además, tengan total libertad de acceso. Álvaro comenta que durante la semana siguiente (6-10-2008/13-10-2008) va a intentar conseguir esa información, y sobre todo esa posible libertad de acceso.

## **4. Pasos a seguir en los próximos días.**

El alumno, Ion, comenta que de manera urgente se reunirá con José Miguel, tutor del proyecto, para preguntarle sus opiniones, primero, acerca de la posible arquitectura a desarrollar como solución a nuestro proyecto, y segundo, para conocer que alternativas existen si es imposible conseguir que una organización nos proporcione acceso libre a su estructura de datos y a sus propios datos.

Álvaro por su parte seguirá recapitulando información sobre los datos de las diversas organizaciones, y además mantendrá alguna entrevista con nuevos clientes interesados.

**5. Dudas y preguntas sobre el Proyecto de Fin de Carrera.**

Nada que destacar.



## D.1.4 Acta del jueves 9 de octubre de 2008

Universidad del País Vasco/  
Euskal Herriko Unibertsitatea  
(UPV/EHU)



### ACTA DE REUNIÓN

- TIPO DE REUNIÓN:** Ordinaria, de seguimiento
- FECHA:** Jueves, 9 de octubre de 2008.
- HORA:** 9:00 horas.
- LUGAR:** Edificio de investigación junto al aula de Vitoria-Gasteiz, perteneciente a la Universidad del País Vasco (UPV/EHU).
- ASISTENTES:** Ion Marín y Álvaro Rodríguez
- CONVOCADA POR:** Ion Marín.

### ORDEN DEL DÍA

1. Situación de las entrevistas con las organizaciones presuntamente interesadas en el proyecto.
2. Descripción de los puntos importantes de la reunión mantenida el pasado martes 7 de octubre con José Miguel Blanco.
3. Pasos a seguir en los próximos días.
4. Dudas y preguntas sobre el Proyecto de Fin de Carrera.

### CONCLUSIONES

- 1. Situación de las entrevistas con las organizaciones presuntamente interesadas en el proyecto.**

Álvaro describe a Ion la situación actual de las diferentes entrevistas, la cual apenas ha variado respecto a la semana pasada. Sin embargo, Álvaro puntualiza

que mañana tienen una reunión con la fundación de la catedral de Santa María, y que le comentarán a Ion los puntos más relevantes de la misma en los próximos días.

## **2. Descripción de los puntos importantes de la reunión mantenida el pasado martes 7 de octubre con José Miguel Blanco.**

Ion describe a Álvaro de manera superficial la reunión mantenida con José Miguel el pasado martes. Sin embargo, explica a Álvaro las conclusiones obtenidas de manera detallada. Ion propone a Álvaro comenzar con la parte práctica del proyecto sin tener en cuenta las bases de datos de las organizaciones, y basándose en un principio en el contenido existente en las páginas Web de las mismas.

Esto es debido a que tanto Ion, como el tutor del proyecto, José Miguel, consideran arriesgado esperar a que las diferentes organizaciones cedan un acceso libre a sus bases de datos, ya que esto, bien no podría llegar nunca o bien tardaría mucho tiempo e implicaría que el proyecto se alargase una eternidad.

## **3. Pasos a seguir en los próximos días.**

Ion y Álvaro acuerdan que lo ideal es empezar ya a realizar un estudio profundo de los datos disponibles para poder conocer los servicios que va a prestar nuestra aplicación. Por lo que se citan el lunes a la mañana para comenzar con el estudio de los datos de cada una de las organizaciones.

Por otra parte Álvaro y José Manuel mantendrán mañana viernes 10 de octubre una reunión con los encargados de la fundación de Santa María..

## **4. Dudas y preguntas sobre el Proyecto de Fin de Carrera.**

Nada que destacar.

## D.1.5 Acta del miércoles 12 de noviembre de 2008

Universidad del País Vasco/  
Euskal Herriko Unibertsitatea  
(UPV/EHU)



### ACTA DE REUNIÓN

- TIPO DE REUNIÓN:** Ordinaria, de seguimiento
- FECHA:** Miércoles, 12 de noviembre de 2008.
- HORA:** 10:00 horas.
- LUGAR:** Despacho de José Manuel Valle situado en el aula de Vitoria-Gasteiz, perteneciente a la Universidad del País Vasco (UPV/EHU).
- ASISTENTES:** Ion Marín, Álvaro Rodríguez y José Manuel Valle
- CONVOCADA POR:** Ion Marín.

### ORDEN DEL DÍA

1. Puesta al día de la situación del proyecto a 12-11-2008.
2. Revisión de lo llevado a cabo hasta el momento.
3. Futuro del proyecto.
4. Dudas y preguntas sobre el Proyecto de Fin de Carrera.

### CONCLUSIONES

1. **Puesta al día de la situación del proyecto a 12-11-2008.**

Ion Marín explica a Álvaro y José Manuel la situación actual del proyecto, en el que a su parecer se han abarcado ya las fases de análisis y diseño del mismo, por lo que considera que el siguiente paso a realizar es su implementación.

## **2. Revisión de lo llevado a cabo hasta el momento.**

Durante la reunión Ion explica a José Manuel y Álvaro el diseño elaborado, mostrando su estructura, su descripción, y argumentando con una serie de ventajas e inconvenientes su puesta en marcha. Ion sugiere a los clientes que realicen alguna sugerencia de mejora o de modificación, ya que es el momento oportuno para ello, ya que posteriormente una modificación, contando que se pueda llevar a cabo, costaría, en términos de tiempo, mucho más del doble.

## **3. Futuro del proyecto.**

La parte “cliente” parece que queda convencida con el diseño propuesto por el desarrollador del proyecto, por lo que en las próximas fechas Ion desarrollará la parte experimental del proyecto.

## **4. Dudas y preguntas sobre el Proyecto de Fin de Carrera.**

Nada que destacar.

## D.1.6 Acta del martes 8 de diciembre de 2008

Universidad del País Vasco/  
Euskal Herriko Unibertsitatea  
(UPV/EHU)



### ACTA DE REUNIÓN

- TIPO DE REUNIÓN:** Ordinaria, de seguimiento
- FECHA:** Martes, 8 de diciembre de 2008.
- HORA:** 10:00 horas.
- LUGAR:** Edificio de investigación junto al aula de Vitoria-Gasteiz, perteneciente a la Universidad del País Vasco (UPV/EHU).
- ASISTENTES:** Ion Marín y Álvaro Rodríguez
- CONVOCADA POR:** Álvaro Rodríguez.

### ORDEN DEL DÍA

1. Situación actual del proyecto.
2. Posibles mejoras del proyecto.
3. Previsión de finalización del proyecto.
4. Dudas y preguntas sobre el Proyecto de Fin de Carrera.

### CONCLUSIONES

#### 1. Situación actual del proyecto.

Después de un mes aproximadamente sin estar en contacto, Álvaro se interesa por la situación actual del proyecto. Álvaro, conociendo que Ion lleva inmerso durante un mes en el desarrollo de la aplicación práctica, pide a éste que le enseñe lo que ha realizado, con el fin de conocer la situación de la aplicación, y

poder comprobar si la idea que tenía el cliente sobre lo que quería al comienzo del proyecto se asemeja al proyecto real.

## **2. Posibles mejoras del proyecto.**

Álvaro tras ojear con detenimiento la aplicación (Ion le presenta una visita guiada por la misma), se muestra, a priori, conforme con el resultado que hasta el momento se había conseguido. Sin embargo, propone a Ion una serie de posibles mejoras en el aspecto de la aplicación. Las sugerencias no son de vital importancia, por lo que el desarrollador del proyecto considera óptimas y factibles de llevar a cabo.

## **3. Previsión de finalización del proyecto.**

Debido a que el proyecto lleva prácticamente tres meses de desarrollo diario, Álvaro y Ion coinciden en que en el periodo de un mes es el momento de darlo por finiquitado, ya que a falta de terminar la aplicación práctica y un informe que recoja el análisis de los resultados obtenidos, la evaluación del proyecto es satisfactoria.

## **4. Dudas y preguntas sobre el Proyecto de Fin de Carrera.**

Ninguna que destacar.

## D.1.7 Acta del lunes 22 de diciembre de 2008

Universidad del País Vasco/  
Euskal Herriko Unibertsitatea  
(UPV/EHU)



- TIPO DE REUNIÓN:** Ordinaria, de seguimiento
- FECHA:** Lunes, 22 de diciembre de 2008.
- HORA:** 10:30 horas.
- LUGAR:** Edificio de investigación junto al aula de Vitoria-Gasteiz, perteneciente a la Universidad del País Vasco (UPV/EHU).
- ASISTENTES:** Ion Marín, José Manuel Valle y Álvaro Rodríguez
- CONVOCADA POR:** Ion Marín.

### ORDEN DEL DÍA

1. Situación del proyecto a fecha de hoy.
2. Sugerencias de mejora.
3. Temas formales.
4. Dudas y preguntas sobre el Proyecto de Fin de Carrera.

### CONCLUSIONES

#### **1. Situación del proyecto a fecha de hoy.**

Debido a la ausencia de José Manuel en la reunión del pasado 8 de diciembre de 2008 y a la continuación de desarrollo del proyecto por parte de Ion Marín, éste explica el funcionamiento completo del proyecto tanto al propio José Manuel como a Álvaro. Aunque aún quedan cosas por terminar y aspectos por mejorar, la parte práctica está casi terminada.

## **2. Sugerencias de mejora.**

A pesar de que José Manuel queda bastante satisfecho con el proyecto que se ha llevado a cabo, sugiere una serie de mejoras sobre las aplicaciones, en las que destacan dos por encima del resto: Dinamismo de la aplicación con vistas a modificaciones en la base de datos, y restricción de acceso para cada usuario. Las sugerencias serán analizadas por Ion, y llevadas a cabo en función del tiempo que cueste implantarlas y las modificaciones que supondrán en la propia aplicación.

## **3. Temas formales.**

Se comentan temas formales acerca del proyecto.

## **4. Dudas y preguntas sobre el Proyecto de Fin de Carrera.**

Nada que destacar.



## D.1.8 Acta del miércoles 15 de diciembre de 2009

Universidad del País Vasco/  
Euskal Herriko Unibertsitatea  
(UPV/EHU)



### ACTA DE REUNIÓN

- TIPO DE REUNIÓN:** Ordinaria, de seguimiento
- FECHA:** Miércoles, 15 de enero de 2009.
- HORA:** 12:00 horas.
- LUGAR:** Edificio de investigación junto al aula de Vitoria-Gasteiz, perteneciente a la Universidad del País Vasco (UPV/EHU).
- ASISTENTES:** Ion Marín, Álvaro Rodríguez y José Manuel Valle.
- CONVOCADA POR:** Ion Marín.

### ORDEN DEL DÍA

1. Situación actual del proyecto.
2. Reunión mantenida con José Miguel Blanco, tutor del proyecto, el día 14-1-2009.
3. Pasos a seguir en los próximos días.
4. Instalación del proyecto en el servidor de la universidad.
5. Dudas y preguntas sobre el Proyecto de Fin de Carrera.

### CONCLUSIONES

#### **1. Situación actual del proyecto.**

Ion describe tanto a Álvaro como a José Manuel, la situación actual del proyecto. Éste está prácticamente acabado, a falta únicamente del tratamiento de

un atributo de una de las aplicaciones. Además Ion enseña las aplicaciones a José Manuel en las que ya aparecen las mejoras que éste había pedido en la reunión anterior.

## **2. Reunión mantenida con José Miguel Blanco, tutor del proyecto, el día 14-1-2009.**

El alumno explica de manera superficial la reunión mantenida con José Miguel el 14 de enero de 2009. José Manuel se interesa por la opinión transmitida a Ion por parte de José Miguel acerca de la opinión personal de éste último sobre la aplicación. Ion le trasmite la opinión de José Miguel así como una serie de detalles que carecen de la importancia necesaria como para ser aquí descritos.

## **3. Pasos a seguir en los próximos días.**

El primer paso es terminar completamente la aplicación ya que aún falta el tratamiento de un atributo para finalizar la misma. Una vez terminado esto, es necesario continuar redactando la parte correspondiente de la memoria referente a la implementación de la aplicación.

Para terminar, es necesario la elaboración de los manuales de mantenimiento, así como un apartado de relaciones con el cliente, sin olvidarnos del apartado correspondiente a la gestión de cambios del proyecto (seguimiento del proyecto). La última fase será las conclusiones y la puesta a punto de la memoria.

En cuanto a la parte práctica, es necesario su instalación en el servidor de la Universidad del País Vasco.

## **4. Instalación del proyecto en el servidor de la universidad.**

Cliente y alumno comentan los pasos que se deben llevar a cabo para conseguir instalar la aplicación en el servidor. Ion pide para ello a la universidad un espacio en la base de datos en donde pueda ubicar las tres bases de datos que se han desarrollado en el proyecto.

## **5. Dudas y preguntas sobre el Proyecto de Fin de Carrera.**

Nada que destacar.

## D.2. ACTAS DE REUNIÓN TUTOR

### D.2.1 Acta del viernes 12 de septiembre de 2008

Universidad del País Vasco/  
Euskal Herriko Unibertsitatea  
(UPV/EHU)



#### ACTA DE REUNIÓN

- TIPO DE REUNIÓN:** Ordinaria, de seguimiento
- FECHA:** Viernes, 12 de septiembre de 2008.
- HORA:** 11:00 horas.
- LUGAR:** Despacho de José Miguel Blanco, situado en la facultad de Informática de San Sebastián-Donostia, perteneciente a la Universidad del País Vasco (UPV/EHU).
- ASISTENTES:** Ion Marín y José Miguel Blanco
- CONVOCADA POR:** Ion Marín.

#### ORDEN DEL DÍA

1. Toma de contacto entre alumno y profesor en relación al desarrollo del proyecto.
2. Revisión de lo llevado a cabo sobre el proyecto hasta el día de hoy.
3. Definición de la dinámica de reuniones entre alumno y profesor.
4. Intercambio de ideas acerca de los pasos a seguir para realizar la parte experimental del proyecto.
5. Dudas y preguntas sobre el Proyecto de Fin de Carrera.

## CONCLUSIONES

### **1. Toma de contacto entre alumno y profesor en relación al desarrollo del proyecto.**

—

### **2. Revisión de lo llevado a cabo sobre el proyecto hasta el día de hoy.**

José Miguel ha realizado alguna observación acerca del trabajo realizado hasta la fecha. La observación más importante ha sido una mezcla de conceptos entorno al análisis de las soluciones propuestas. Para este análisis, José Miguel me ha explicado que es necesario realizar un estudio previo sobre los servicios que prestan actualmente las instituciones u organizaciones en lo referente al patrimonio digital. No podemos diseñar soluciones, si primero no sabemos lo que podemos o no podemos ofertar.

### **3. Definición de la dinámica de reuniones entre alumno y profesor.**

La dinámica de las reuniones se basará en las necesidades del alumno. En función de los problemas o dudas que vayan surgiendo durante el desarrollo del proyecto, se pueden producir más o menos reuniones.

### **4. Intercambio de ideas acerca de los pasos a seguir para realizar la parte experimental del proyecto.**

Este punto era importante ya que, en principio, el alumno tenía una idea sobre cómo se podía realizar esta parte experimental, sin embargo los argumentos del tutor del proyecto, han implicado un cambio de opinión en el alumno. El profesor ha explicado que en un “proyecto real” la solución propuesta por el alumno no tendría futuro, ya que ésta implicaría un cambio en las bases de datos de las organizaciones, las cuales, dudo mucho lo apoyen. Debido a esto, en principio y a día de hoy, la solución más acertada es una integración de bases de datos heterogéneas. Para poder llevar a cabo esto, en primer lugar es importante intentar recoger la máxima información posible acerca de los servicios y funcionalidades que ofrecen cada una de las organizaciones, y, en el caso de que fuera posible, realizar un estudio sobre las estructuras de las bases de datos de dichas organizaciones.

### **5. Dudas y preguntas sobre el Proyecto de Fin de Carrera.**

Se ha enfocado el tema de la posibilidad del compaginar el trabajo con el PFC. De momento sin más consecuencias.

## D.2.2 Acta del martes 7 de octubre de 2008

Universidad del País Vasco/  
Euskal Herriko Unibertsitatea  
(UPV/EHU)



### ACTA DE REUNIÓN

**TIPO DE REUNIÓN:** Ordinaria, de seguimiento

**FECHA:** Martes, 7 de octubre de 2008.

**HORA:** 11:00 horas.

**LUGAR:** Despacho de José Miguel Blanco, situado en la facultad de Informática de San Sebastián-Donostia, perteneciente a la Universidad del País Vasco (UPV/EHU).

**ASISTENTES:** Ion Marín y José Miguel Blanco

**CONVOCADA POR:** Ion Marín.

#### ORDEN DEL DÍA

1. Situación del proyecto a fecha de hoy.
2. Análisis superficial de una solución a un problema similar al nuestro, que nos puede servir como orientación para la solución de nuestro proyecto.
3. En el supuesto de que éste primero no sirviera, enumerar alternativas.
4. Dudas y preguntas sobre el Proyecto de Fin de Carrera.

#### CONCLUSIONES

##### **1. Situación del proyecto a fecha de hoy.**

El alumno, Ion Marín, ha informado al tutor, José Miguel Blanco, sobre la situación actual del proyecto. Le ha comentado aspectos como, la situación actual de las fuentes de datos, que todavía no están disponibles, los plazos del proyecto y las ideas que tenía el propio alumno para llevar a cabo una solución al proyecto.

**2. Análisis superficial de una solución a un problema similar al nuestro, que nos puede servir como orientación para la solución de nuestro proyecto.**

La semana previa a la reunión el alumno había estado investigando sobre algunas arquitecturas que podrían servir para solucionar nuestro problema. De entre las arquitecturas investigadas hubo una que le pareció más interesante, la denominada *Arquitectura para Federación de Bases de Datos Documentales basada en Ontologías*.

Debido al interés del alumno en este tipo de arquitectura consiguió en la Web una Tesis Doctoral sobre esta arquitectura, la cual estuvo leyendo durante cinco días intentando conseguir obtener una solución que sirviera para el proyecto.

El alumno se había dado cuenta durante el análisis de esta propuesta que sin un conocimiento profundo de todas las fuentes de datos interesadas en el proyecto, el cual era prácticamente imposible conseguir, la elaboración de una arquitectura como esta se convertía en una utopía. José Miguel no hizo más que confirmar lo que el alumno temía.

Aunque esta arquitectura era prácticamente imposible de llevar a cabo, si que sirvió para buscar soluciones alternativas.

**3. En el supuesto de que éste primero no sirviera, enumerar alternativas.**

La primera propuesta había sido anulada debido a la imposibilidad de tener acceso libre a las fuentes de datos de aquellos interesados en este proyecto. José Miguel intentó orientar el proyecto en función de la primera propuesta, es decir, propuso una alternativa similar a la propuesta rechazada pero con muchas posibilidades de llevarse a cabo de manera satisfactoria. La alternativa consistiría en mantener la parte inicial de la propuesta rechazada, es decir, realizar un análisis de los datos, elaborar un árbol de conceptos en función de los servicios que se quieran ofrecer, y llevar a cabo un sitio web en el que además de una parte estática apareciera un buscador que nos ofreciera, de manera menos óptima que la propuesta rechazada, los sitios Web en donde apareciera la información requerida por el usuario. Además, se podría incluir una breve descripción de los sitios Web lo que ayudaría al usuario a conocer de antemano la información que se iba a encontrar en cada uno de ellos. Además de esto se habló también de ampliar el proyecto, elaborando, por ejemplo, un programa que consiguiera adaptar a su árbol de conceptos un nuevo sitio Web.

El alumno, después del intercambio de ideas con el tutor, obtuvo una posible alternativa para solucionar el problema que presenta este proyecto, sin embargo, esta propuesta puede ser modificada parcialmente, en función de los intereses del cliente, y de las posibilidades de mejora que pueda encontrarse el propio alumno.

#### **4. Dudas y preguntas sobre el Proyecto de Fin de Carrera.**

Nada destacable.

### D.2.3 Acta del miércoles 14 de enero de 2009

Universidad del País Vasco/  
Euskal Herriko Unibertsitatea  
(UPV/EHU)



## ACTA DE REUNIÓN

- TIPO DE REUNIÓN:** Ordinaria, de seguimiento
- FECHA:** Miércoles, 14 de enero de 2009.
- HORA:** 10:00 horas.
- LUGAR:** Despacho de José Miguel Blanco, situado en la facultad de Informática de San Sebastián-Donostia, perteneciente a la Universidad del País Vasco (UPV/EHU).
- ASISTENTES:** Ion Marín y José Miguel Blanco
- CONVOCADA POR:** Ion Marín.

### ORDEN DEL DÍA

1. Puesta al día en la situación actual del proyecto.
2. Explicación por parte de Ion sobre lo que ha desarrollado como proyecto.
3. Opinión de José Miguel al respecto del proyecto. Posibles mejoras, cosas que cambiar...
4. Análisis de los pasos que faltan para terminar el proyecto
5. Temas formales. Registro del proyecto.
6. Dudas y preguntas sobre el Proyecto de Fin de Carrera.



## CONCLUSIONES

### **1. Puesta al día en la situación actual del proyecto.**

El alumno, Ion Marín, ha informado al tutor, José Miguel Blanco, sobre la situación actual del proyecto. El estado actual de las aplicaciones, relaciones con el cliente...

### **2. Explicación por parte de Ion sobre lo que ha desarrollado como proyecto.**

Tras la puesta al día de lo que al proyecto se refiere, Ion ha explicado a José Miguel lo que se ha desarrollado como parte experimental del proyecto de fin de carrera. Ha explicado de manera superficial cada una de las dos aplicaciones. Tutor y alumno han mantenido una conversación acerca de las opiniones del cliente sobre las aplicaciones desarrolladas y sobre si éstas han sido ya probadas por los clientes. También se ha comentado su situación, es decir, si están ya acabadas o lo que falta para ello.

### **3. Opinión de José Miguel al respecto del proyecto. Posibles mejoras, cosas que cambiar...**

En cuanto a la parte práctica del proyecto, José Miguel se ha mantenido al margen, ya que considera que ya existe un cliente quien es quien debe valorar si dicha parte práctica es adecuada o no. Respecto a la gestión del proyecto José Miguel ha recordado a Ion que falta el análisis de requerimientos, parte importante del mismo, y el cual es necesario tenerlo en cuanto antes. Ion se ha comprometido a entregárselo en cuanto pueda, ya que dicho documento, tal y como acordó Ion con el cliente, iba a ser redactado por el propio cliente ya que se considera que el cliente es quien debe aclarar esos requisitos. Aunque, si bien es cierto la relación continua mantenida entre el cliente y el alumno deja bastante claro que los requisitos han sido claros desde el primer momento, es mucho más idóneo que esto quede documentado.

### **4. Análisis de los pasos que faltan para terminar el proyecto**

El primer paso es terminar completamente la parte práctica. A lo que le sigue redactar los manuales de usuario de la aplicación, junto con los manuales de mantenimiento de la misma. Una vez terminado esto, se debe continuar con explicación de la aplicación que se ha desarrollado. Además pueden existir diversos cambios tanto en el diseño como en la planificación que deben ser documentados. Por otro lado, se considera interesante redactar un apartado en el que se documente las relaciones con el cliente. La situación inicial (demanda por parte del cliente), la gestión de las relaciones con el mismo, y un cierre que incluya una pequeña valoración del resultado por parte del cliente. También falta

la fase de pruebas, la cual deberá de ser además documentada. Por último es necesaria una conclusión del proyecto.

**5. Temas formales. Registro del proyecto.**

El proyecto queda registrado a falta de otorgarle un título del cual se encargará Ion.

**6. Dudas y preguntas sobre el Proyecto de Fin de Carrera.**

Ion le ha hecho saber a su tutor sus problemas con los dtd. Sin nada más que destacar.



**LABORATORIO DE DOCUMENTACIÓN GEOMÉTRICA DEL PATRIMONIO**

Grupo de Investigación en Patrimonio Construido -GPAC- (UPV-EHU)

Aulario de las Nieves, edificio de Institutos Universitarios  
C/ Nieves Cano 33, 01006 Vitoria-Gasteiz (España-Spain).  
Tfno: +34 945 013222 / 013264  
e-mail: [ldgp@ehu.es](mailto:ldgp@ehu.es) web: <http://www.ldgp.es>



UPV EHU