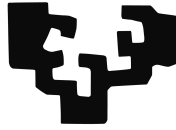


eman ta zabal zazu



Universidad  
del País Vasco

Euskal Herriko  
Unibertsitatea

Grado en Ingeniería Informática  
Ingeniería del Software

Proyecto de Fin de Grado

---

**EvDiscover**  
**Sistematización del soporte a la gestión de  
estudios clínicos**

---

Autor

*Ioritz Cabero Llama*

Directores

*Dr. Juan Manuel Pikatza Atxa*

*Dr. Eduardo Gonzalez Perez-Yarza*

informatika  
fakultatea



facultad de  
informática

2017



---

## **Agradecimientos**

---

Antes de comenzar, me gustaría agradecer en estas líneas a todas aquellas personas que durante los años de carrera me han apoyado.

En primer lugar, a mi familia sin la cual no podría haber llegado donde estoy. Ellos han confiado en mí desde el principio y han estado ahí siempre infundiéndome ánimos.

A mis compañeros, por su continuo apoyo y ayuda cuando lo he necesitado. Gracias a ellos he crecido tanto profesional como personalmente, y he aprendido mucho acerca del trabajo en grupo.

A mi tutor, Juan Manuel Pikatza Atxa, por tener la paciencia necesaria para trabajar junto a mí y dirigirme en este proyecto, guiándome en los momentos en que me ha hecho falta, y ayudándome con los problemas que han podido surgir en el camino.

Por último, y no menos importante, quiero agradecer a todo el profesorado los conocimientos transmitidos. En concreto a José Miguel Blanco y José Ángel Vadillo, los cuales han sido un referente para mí.

---

## Resumen

---

Este documento contiene la memoria del Trabajo de Fin de Grado que trata sobre la aplicación EvDiscover desarrollada por Ioritz Cabero Llama y tiene como objetivo obtener la titulación del Grado de Ingeniería Informática en la Facultad de Informática de Donostia / San Sebastián, en la UPV/EHU. El contenido de la misma está basado en la norma *AENOR UNE 157801:2007*.

El objetivo de este proyecto es obtener un elemento software en forma de aplicación web que ayude al complejo problema que supone la gestión de los estudios clínicos. Partiendo de la hipótesis de que los sistemas *BPM (Business Process Management)* son adecuados para este tipo de problemas, en este documento se describe la solución planteada siguiendo la metodología *OpenUp*, junto a un modelo de desarrollo basado en prototipos.

El resultado de este proyecto es una aplicación web para ayudar en la gestión de los estudios clínicos. Para ello, se presenta una solución basada en dos componentes distintos, una para el proceso de gestión de un estudio clínico basada en un sistema *BPM*, y otra dedicada a la obtención de datos mediante cuestionarios. Dada la inmensa complejidad de este dominio, no ha sido posible plantear una solución global a todos los estudios clínicos. No obstante, y gracias a las características de los sistemas *BPM*, se ha desarrollado una base que puede ser optimizada en el futuro, de cara a obtener resultados cada vez más eficientes y eficaces mediante la mejora continua de los procesos.

**Palabras clave:** Estudios Clínicos, Gestión, BPM.



---

# Índice general

---

<b>Resumen</b>	<b>I</b>
<b>Índice general</b>	<b>III</b>
<b>Índice de figuras</b>	<b>IX</b>
<b>Índice de tablas</b>	<b>XI</b>
<b>I Presentación al cliente</b>	<b>1</b>
<b>1. Introducción</b>	<b>5</b>
<b>2. Objeto</b>	<b>9</b>
2.1. Finalidad . . . . .	9
2.2. Objetivos . . . . .	10
<b>3. Antecedentes</b>	<b>11</b>
3.1. Ensayos clínicos . . . . .	11
3.1.1. Definición . . . . .	11
3.1.2. Clasificación . . . . .	12
3.1.3. Tipos de estudios . . . . .	16
3.2. Soluciones a la gestión de estudios clínicos . . . . .	16
	<b>III</b>

3.2.1. Utilización de sistemas BPM en la gestión de estudios clínicos . . .	17
3.3. Proveedores de las tecnologías BPM . . . . .	17
3.4. BPMN . . . . .	17
<b>4. Descripción de la situación actual</b>	<b>21</b>
4.1. Soluciones previas . . . . .	21
4.2. Puesta en marcha de los sistemas BPM . . . . .	21
4.3. Formación previa . . . . .	22
<b>5. Normas y referencias</b>	<b>23</b>
5.1. Disposiciones y normas aplicadas . . . . .	23
5.1.1. Norma UNE 157801:2007 . . . . .	23
5.1.2. LOPD: Ley Orgánica de Protección de Datos . . . . .	24
5.2. Bibliografía . . . . .	25
5.3. Métodos, Herramientas, Modelos, Métricas y Prototipos . . . . .	26
5.3.1. Modelo de desarrollo basado en prototipos . . . . .	26
5.3.2. Metodología de desarrollo OpenUp . . . . .	27
5.3.3. Herramientas y librerías . . . . .	28
<b>6. Definiciones y abreviaturas</b>	<b>31</b>
<b>7. Requisitos iniciales</b>	<b>35</b>
7.1. Requisitos funcionales . . . . .	35
7.2. Requisitos no funcionales . . . . .	36
<b>8. Alcance</b>	<b>39</b>
8.1. Alcance del proyecto . . . . .	39
8.1.1. Exclusiones . . . . .	40

---

<b>9. Hipótesis y restricciones</b>	<b>43</b>
9.1. Hipótesis	43
9.2. Restricciones	43
<b>10. Estudio de alternativas y viabilidad</b>	<b>45</b>
10.1. Suites BPM	45
10.1.1. IBM BPM	45
10.1.2. Bizagi Studio	46
10.1.3. Selección de la suite BPM	46
10.2. Cumplimentación de cuestionarios	48
10.2.1. Framework de desarrollo	48
10.2.2. Base de datos	48
10.3. Estudio de viabilidad	50
<b>11. Descripción del sistema propuesto</b>	<b>51</b>
11.1. Características de la solución	51
11.1.1. Restricciones	51
11.1.2. Qué hace la solución	52
11.2. EvDiscover-Bizagi	54
11.2.1. Arquitectura	54
11.2.2. Análisis	56
11.2.3. Diseño e implementación	61
11.2.4. Proceso Implementado	67
11.3. EvDiscover-Forms	77
11.3.1. Arquitectura	77
11.3.2. Tecnologías utilizadas	78
11.3.3. Análisis	79



11.3.4. Diseño . . . . .	82
11.3.5. Implementación . . . . .	88
11.4. Integración . . . . .	91
11.5. Ventajas y desventajas . . . . .	93
11.5.1. Ventajas . . . . .	94
11.5.2. Desventajas . . . . .	95
11.6. Conclusiones . . . . .	95
11.6.1. Líneas futuras . . . . .	96
<b>12. Análisis de riesgos</b>	<b>99</b>
<b>13. Organizaciones y gestión del proyecto</b>	<b>101</b>
<b>14. Planificación temporal</b>	<b>103</b>
<b>15. Presupuesto</b>	<b>105</b>
<b>II Elaboración del proyecto</b>	<b>107</b>
<b>1. Plan inicial del proyecto</b>	<b>109</b>
1.1. Alcance . . . . .	109
1.1.1. Alcance mínimo . . . . .	109
1.1.2. Ampliación del alcance . . . . .	111
1.2. WBS/EDT . . . . .	111
1.3. Planificación temporal . . . . .	112
1.3.1. Entregables . . . . .	112
1.3.2. Hitos . . . . .	112
1.3.3. Diagrama de Gantt . . . . .	112

---

1.3.4. Estimación inicial . . . . .	113
1.4. Plan de comunicación y seguimiento . . . . .	114
1.5. Plan para la gestión de la calidad . . . . .	114
1.6. Selección de la suite BPM . . . . .	114
1.7. Sistema de información . . . . .	114
1.8. Gestión de los riesgos . . . . .	115
<b>2. Seguimiento y control</b>	<b>117</b>
2.1. Gestión del alcance . . . . .	117
2.2. Objetivos por iteración . . . . .	118
2.3. Desviaciones . . . . .	119
2.3.1. Desviaciones en los plazos . . . . .	119
2.3.2. Desviaciones en las estimaciones . . . . .	119
<b>Anexos</b>	<b>125</b>
<b>A. Índice de la web del proyecto</b>	<b>125</b>
<b>B. Actas</b>	<b>127</b>



---

## Índice de figuras

---

3.1. Cuadrante mágico sobre suites BPM 2016 . . . . .	18
5.1. Desarrollo basado en prototipos . . . . .	26
5.2. Ciclo de vida de <i>OpenUp</i> . . . . .	28
10.1. Cuadrante mágico de Gartner sobre bases de datos operacionales. . . . .	50
11.1. Diagrama de componentes del sistema software . . . . .	52
11.2. Arquitectura de Bizagi . . . . .	54
11.3. Diagrama de casos de uso de EvDiscover-Bizagi. . . . .	57
11.4. Asistente de procesos de Bizagi Studio. . . . .	62
11.5. Modelo de datos del proceso. . . . .	64
11.6. Editor de formularios de Bizagi Studio. . . . .	65
11.7. Formulario de definición de una regla en Bizagi. . . . .	66
11.8. Proceso general para el control del estudio. . . . .	68
11.9. Fase de inicio del proceso. . . . .	69
11.10 Subproceso <i>Creación Inicial</i> . . . . .	70
11.11 Subproceso <i>Creación de grupos</i> . . . . .	71
11.12 Fase de evolución del proceso. . . . .	72
11.13 Subproceso <i>Tareas administrativas</i> . . . . .	73

11.14	Subproceso <i>Almacenar médicos API</i> . . . . .	74
11.15	Subproceso <i>Desarrollo del estudio</i> . . . . .	75
11.16	Fase de finalización del estudio . . . . .	77
11.17	Patrón <i>MVC</i> con <i>Express, Jade y Mongoose</i> . . . . .	77
11.18	Ejemplo de documento almacenado por <i>mongoDB</i> . . . . .	78
11.19	Casos de uso de <i>EvDiscover-Forms</i> . . . . .	80
11.20	Modelo de datos de <i>EvDiscover-Forms</i> . . . . .	83
11.21	Diagrama de secuencia de iniciar sesión. . . . .	84
11.22	Interfaz del caso de uso iniciar sesión. . . . .	84
11.23	Diagrama de secuencia de cumplimentar formularios. . . . .	85
11.24	Interfaz del caso de uso cumplimentar formularios. . . . .	86
11.25	Interfaz de ejemplo de uno de los formularios implementados. . . . .	86
11.26	Diagrama de secuencia de extraer datos. . . . .	87
11.27	Interfaz de usuario para la extracción de datos. . . . .	88
11.28	Ejemplo de un modelo de <i>mongoose</i> . . . . .	89
11.29	Ejemplo de conversión <i>Jade</i> a <i>HTML</i> . . . . .	90
11.30	Ventana de configuración de datos de entrada del conector. . . . .	92
11.31	Diagrama de secuencia de alta de estudio en la <i>API</i> . . . . .	92
11.32	Diagrama de secuencia de alta de usuario. . . . .	93
11.33	Diagrama de secuencia de generación de solicitud de formulario . . . . .	93
11.34	Diagrama de secuencia de comprobación de formularios pendientes . . . . .	94
14.1.	Diagrama de gantt con las fechas estimadas para el desarrollo final . . . . .	103
1.1.	WBS/EDT del proyecto . . . . .	111
1.2.	Diagrama Gantt inicial del proyecto . . . . .	113
2.1.	Diagrama Gantt final del proyecto . . . . .	120

---

## Índice de tablas

---

3.1. Clasificación de los distintos ensayos clínicos.[Wikipedia, 2017] . . . . .	13
8.1. Artefactos de la metodología <i>OpenUp</i> a presentar . . . . .	40
10.1. Comparativa de suites BPM . . . . .	47
10.2. Comparativa de Frameworks Web . . . . .	49
11.1. Tipos de usuario utilizables en cada sistema . . . . .	53
11.2. Tabla de direcciones de la <i>API</i> de EvDiscover-Forms. . . . .	91
12.1. Riesgo de dependencia tecnológica. . . . .	99
12.2. Riesgo de actualizaciones del software de desarrollo. . . . .	99
12.3. Riesgo de duración del proyecto. . . . .	100
12.4. Riesgo de caídas del servidor. . . . .	100
12.5. Riesgo por incumplimiento de legislación. . . . .	100
14.1. Estimación temporal de las siguientes fases del proyecto siguiendo la metodología <i>OpenUP</i> . . . . .	104
15.1. Presupuesto para el personal . . . . .	105
15.2. Presupuesto para material . . . . .	106
1.1. Tabla de estimaciones . . . . .	113

1.2. Riesgo derivado del Aprendizaje Tecnológico. . . . .	116
1.3. Riesgo derivado de la pérdida de datos . . . . .	116
1.4. Tabla de riesgos derivada de requisitos y plazos. . . . .	116
2.1. Objetivos por iteración . . . . .	118
2.2. Tabla de dedicaciones(en horas) . . . . .	121

# **PARTE I**

## **Presentación al cliente**





**Cliente:**

Nombre: Juan Manuel Pikatza Atxa

C.I.F:

Dirección:

Teléfono:

**Proveedor:**

Nombre: Ioritz Cabero Llama

Titulación: Graduado en Ingeniería informática

C.I.F:

Dirección:

Teléfono:

**Resumen:**

El objetivo de este proyecto es mejorar la gestión de los estudios clínicos obteniendo un prototipo de aplicación web que facilite dicha gestión.

**Duración:**

9 meses

**Presupuesto:**

48001€



## Introducción

---

En este capítulo se realiza una descripción breve del objetivo de este documento y su estructura.

La estructura de este documento está basada en la norma *AENOR UNE 157801:2007*, la cual establece los criterios generales para el desarrollo de proyectos de sistemas de información. El objetivo del mismo es presentar una solución al problema de la gestión de los estudios clínicos.

La estructuración de la memoria viene definida en la norma mencionada, y es la siguiente:

- **Parte I: Información a presentar al cliente**

Se presenta la solución al cliente, conteniendo la información relevante para su evaluación por parte del cliente.

- **Introducción:** Explicación del objetivo, contenido y estructura del documento, haciendo referencia a la norma *UNE 157801:2007* y su aplicación.
- **Objeto:** Objetivo final del proyecto, y la finalidad que justifica su ejecución
- **Antecedentes:** Elementos significativos que tienen influencia en el proyecto.
- **Descripción de la situación actual:** Punto de partida del proyecto junto con sus condicionantes.
- **Normas y referencias:** Recopilación de las normas y referencias utilizadas en el proyecto. Además, se incluyen los métodos de desarrollo utilizados y las herramientas empleadas.

- **Definiciones y abreviaturas:** Para facilitar la comprensión del texto mediante la definición de la terminología empleada.
  - **Requisitos iniciales:** Características funcionales y no funcionales a cumplir en los entregables, o en el desarrollo del mismo.
  - **Alcance:** Alcance final del proyecto con todo lo que incluye y no incluye.
  - **Hipótesis y restricciones:** Hipótesis de partida del proyecto y restricciones impuestas por parte del cliente.
  - **Estudio de alternativas y viabilidad:** Alternativas de solución tenidas en cuenta, y decisiones de las tecnologías a utilizar.
  - **Descripción de la solución propuesta:** Descripción de la propuesta, y las características de la misma.
  - **Análisis de riesgos:** Enumeración de los riesgos del proyecto, impacto de los mismos, y plan de mitigación.
  - **Organización y gestión del proyecto:** Sistemática a utilizar en las futuras fases de ejecución del proyecto.
  - **Planificación temporal:** Planificación de las siguientes fases del proyecto.
  - **Presupuesto:** Coste total del proyecto en base a la planificación, y la organización planteada.
- **Parte II: Información producida durante la elaboración del proyecto**

Este apartado no está incluido dentro de la norma, pero al tratarse de un proyecto de fin de grado de la Facultad de Informática, se requiere como necesario para la evaluación del mismo.

Incluye los siguientes capítulos.

- **Plan inicial del proyecto:** Presentación de la planificación inicial desarrollada para la elaboración de este proyecto.
- **Seguimiento y control:** Definición del alcance final obtenido, una descripción de los objetivos establecidos en las distintas fases del proyecto, y las desviaciones en cuanto a horas y fechas respecto a la planificación inicial.

En los anexos de esta memoria podrá encontrarse el índice con los contenidos del sitio web del proyecto (ver [anexo A](#)), así como las actas de las reuniones relativas a la gestión del proyecto (ver [anexo B](#)).

En el sitio web que se ha creado para este proyecto se pueden encontrar todos los detalles técnicos mencionados en la memoria, así como los documentos de gestión utilizados. Dicha página web será accesible a través del siguiente enlace disponible desde el 01/08/2017 hasta el 26/09/2017, ambos incluidos.

<https://evdiscover-9fc8c.firebaseio.com/>

Para tener acceso a los contenidos de la web del proyecto fuera de ese plazo contactar con el autor en la dirección de correo electrónico:

[icabero002@ikasle.ehu.es](mailto:icabero002@ikasle.ehu.es)



## CAPÍTULO 2

---

### Objeto

---

En este capítulo se realizará la descripción del objetivo y de la finalidad del proyecto.

#### 2.1. Finalidad

Los ensayos o estudios clínicos son procesos complejos que constituyen la fuente principal de consecución de evidencia científica sobre la validez de tratamientos y tecnologías. Estos ensayos o estudios involucran a múltiples usuarios, por lo que mantener el control de los mismos es de vital importancia. El problema al que se pretende dar solución mediante este proyecto, es la automatización (en la medida de lo posible) del proceso de gestión de un ensayo clínico.

La finalidad de este proyecto es dar una solución inicial en forma de prototipo al problema de automatizar la gestión de los estudios clínicos. Para ello, tendremos en cuenta los distintos interesados que formarán parte de este ecosistema. La idea general es la de aliviar la pesadez del trabajo manual que se realiza hasta ahora, tanto en el proceso de adquisición de candidatos, como en la recolección de resultados, y control de actividad de los participantes.



## 2.2. Objetivos

Tal y como se describirá más adelante existen muchos tipos distintos de ensayo clínico. Debido a eso, será imposible realizar una aplicación genérica que permita la gestión de todos ellos. Para ello, se ha decidido centrar los objetivos de este proyecto en dar una solución inicial a ensayos clínicos de tipo prospectivo en forma de aplicación web. No obstante, por el tipo de estudios en los que se centrara esta primera solución, es importante definir unos objetivos específicos para la misma:

- Ayudar en la gestión de todo el proceso relativo al ensayo clínico.
  - Diseñar el estudio y obtener los permisos del Comité Ético de Investigación Clínica (CEIC) del territorio en el que se va a realizar.
  - Obtención de candidatos y formación de grupos.
  - Control de la actividad realizada por cada uno de los participantes durante el desarrollo del mismo.
  - Control de incidencias y facilidad en la comunicación con los distintos participantes.
  - Aseguramiento del rigor del estudio.
- Obtener los resultados del estudio clínico en un formato fácilmente analizable.
  - Datos importables por herramientas de carácter estadístico.

Aunque se desarrollara una aplicación muy centrada en un tipo muy concreto de ensayo, el proyecto pretende establecer una base sobre la que poder ir evolucionando y mejorando los procesos de gestión de un estudio clínico. Estas mejoras podrán ser realizadas por otros equipos de trabajo a partir de los resultados y conclusiones obtenidas en este proyecto.

### Antecedentes

---

En este capítulo se recogen los elementos significativos del pasado que han tenido influencia en el proyecto.

#### 3.1. Ensayos clínicos

##### 3.1.1. Definición

Cuando tratamos de definir que es un “ensayo clínico”, nos encontramos ante varias definiciones para un mismo término. En este documento utilizaremos las publicadas por José Antonio Durán Quintana [[Durán Quintana, 2004](#)].

- *«El estudio experimental de la eficacia y tolerabilidad de los fármacos en pacientes»* (Definición epidemiológica).
- *«Toda investigación efectuada en seres humanos para determinar o confirmar los efectos clínicos, farmacológicos y/o demás efectos farmacodinámicos, y/o detectar las reacciones adversas, y/o estudiar la absorción, distribución, metabolismo y excreción de uno o varios medicamentos en investigación con el fin de determinar su seguridad y/o su eficacia»* (Artículo 2 del Real Decreto 223/2004).

Dicho en una forma más sencilla, podríamos definir el término como «*Estudios analíticos relacionados con medicamentos o enfermedades, cuyo objetivo es el de obtener conclusiones en base a un proceso donde se obtienen datos, procesándose estos a posteriori.*». Hay que tener en cuenta que los estudios para la evaluación de tecnologías sanitarias siguen un procedimiento similar.

### 3.1.2. Clasificación

Dentro de los ensayos clínicos existe una amplia variedad en base a distintos factores. Dicha clasificación queda recogida en la tabla 3.1. El hecho de que existan tantas clases distintas no implica que un estudio deba ser clasificado solamente en una de ellas. Los puntos expuestos a continuación no son sino un resumen de los datos extraídos de diversas fuentes especializadas [IMIM, 2006], [UAM, 2006].

#### 3.1.2.1. Desarrollo clínico

A la hora de presentar un nuevo fármaco no es posible distribuirlo sin haber realizado antes pruebas en humanos. Para ello existe una metodología basada en cuatro fases.

- **Fase I:** Primera fase de las pruebas con personas. Generalmente se realiza en adultos jóvenes de sexo masculino y siempre en grupos inferiores a los 100 integrantes. El fin de esta fase es la de detectar posibles riesgos de toxicidad y así poder ajustar la dosis adecuada del medicamento.
- **Fase II:** Una vez superada la primera fase, se forma un grupo de entre 100 y 200 individuos que presentan la enfermedad que trata el medicamento. En este punto se dividirá el grupo en dos partes.

Por un lado, tendremos aquellos que tomarán el medicamento más efectivo hasta la fecha (un placebo si no existiera). Por otro lado, estarán aquellos que sí tomarán el medicamento en cuestión. La finalidad de esta fase es la de realizar mediciones de la relación eficacia terapéutica/toxicidad, así como establecer la dosis óptima.

- **Fase III:** Una vez superadas las dos primeras fases con resultados satisfactorios se comienza con esta fase. En este punto se involucra a múltiples médicos y se trata entre cientos y miles de pacientes. El objetivo es el de encontrar manifestaciones de toxicidad previamente no detectadas. Además, también se obtienen datos más precisos en la relación seguridad eficacia.

<b>Factor de clasificación</b>	<b>Tipos de ensayos clínicos</b>	
Desarrollo clínico	Fase I	
	Fase II	
	Fase III	
	Fase IV	
Aspecto de la intervención	Explicativos	
	Pragmáticos	
Objetivo	Equivalencia Terapéutica	
	Búsqueda de dosis	
	Dosis-respuesta	
	Concentración-respuesta	
	Bioequivalencia	
	Biodisponibilidad	
	Estudio piloto	
Enmascaramiento	No enmascarado / Abierto	
	Enmascarado / Ciego /Cerrado	Simple Ciego
		Doble ciego
		Triple ciego
Exposición	Paralelo	
	Cruzado	
Control	Controlado	
	No controlado	
Aleatorización	Aleatorizado	
	No aleatorizado	

**Tabla 3.1:** Clasificación de los distintos ensayos clínicos.[[Wikipedia, 2017](#)]

- **Fase IV:** Cuando un estudio está en esta fase se conoce como estudio de farmacovigilancia. En este punto el medicamento ya se habrá comercializado, por lo que en esta fase el fin de este estudio será analizar su eficacia y encontrar nuevos efectos adversos.

### 3.1.2.2. Aspecto de la intervención

En función del aspecto a analizar el estudio puede ser de tipo explicativo o pragmático

- **Explicativo:** Se realiza el estudio en una muestra controlada y homogénea. Dado que se trata de un estudio muy concreto cuyo objetivo es el de valorar la eficacia del tratamiento, no se pueden aplicar las conclusiones a un conjunto generalizado.
- **Pragmático:** A diferencia del anterior este se realiza en una muestra grande y heterogénea. El objetivo es valorar la eficacia de forma similar a como se realizaría en la realidad. En este caso, gracias a la heterogeneidad de la muestra los resultados si son generalizables.

### 3.1.2.3. Objetivo del estudio

En función del objetivo buscado, podemos tener los siguientes tipos de ensayo:

- **Equivalencia Terapéutica:** Demostrar que el tratamiento puesto a estudio es igual o más efectivo que el tratamiento de referencia.
- **Búsqueda de dosis:** Encontrar la dosis idónea para el tratamiento de cada individuo.
- **Concentración-Respuesta:** Relacionar efectos concretos del tratamiento con la concentración de la exposición.
- **Bioequivalencia:** Evaluar la similitud de dos medicamentos con un mismo principio activo administrado con una medida equivalente.
- **Biodisponibilidad:** Analizar la velocidad de un medicamento en obtener el efecto deseado.
- **Estudio piloto:** Preparar el terreno para estudios posteriores.
- **Valoración tecnológica:** Evaluación de tecnologías software o hardware.

#### 3.1.2.4. Enmascaramiento

Con el objetivo de obtener evidencias, a menudo se separan los sujetos a controlar en dos grupos. Aquellos que reciben un tratamiento ya establecido o un placebo (grupo de control), y aquellos sobre los que se prueban los nuevos medicamentos (grupo de experimentación).

De cara a obtener unos resultados óptimos, es importante ocultar a que grupo pertenecen los distintos sujetos. Existen cuatro tipos:

- **Abierto:** Todos los involucrados saben en qué parte del estudio están asignados.
- **Ciego simple:** Cuando el estudiado o el observador desconocen a que parte del estudio pertenecen.
- **Doble ciego:** Ni el sujeto ni el observador conocen su ubicación en el estudio.
- **Triple ciego:** Es igual que el doble ciego solo que en este caso el analista tampoco conoce en que parte del estudio está ubicado. Este tipo de ensayo es el que mayor evidencia científica aporta.

#### 3.1.2.5. Exposición

En función del tipo de exposición a la que se somete a un participante podemos encontrar dos tipos distintos:

- **Paralelo:** Se realiza el estudio de ambos grupos al mismo tiempo
- **Cruzado:** Se administra el tratamiento experimental y de control al mismo individuo en periodos sucesivos.

#### 3.1.2.6. Control

Este tipo de clasificación se basa en el nivel de control del estudio, es decir, si el estudio está controlado o no. En el caso de un estudio controlado, se incluyen al menos dos grupos de pacientes cuyo tratamiento será o el de control o el experimental. En su mayoría, se realizan estudios controlados, ya que es la única forma científicamente válida para evaluar la eficacia de los mismos.

### 3.1.2.7. Aleatorización

Dentro de un ensayo clínico encontramos distintos sujetos a tratar. Con el fin de obtener resultados y análisis válidos se realiza una distribución al azar de los sujetos en los distintos grupos.

### 3.1.3. Tipos de estudios

En función de la temporalidad de los estudios, estos pueden ser de tres tipos.

- **Retrospectivo:** Se analiza en el presente, pero con datos del pasado.
- **Transversal:** Se realiza utilizando los datos de un momento puntual. Sirve a modo de “instantánea” de la situación actual.
- **Prospectivo:** Se realiza en el presente, y se analizan sus datos una vez transcurrido un determinado tiempo.

## 3.2. Soluciones a la gestión de estudios clínicos

Actualmente, existen múltiples soluciones en el dominio de los estudios clínicos. Haciendo una búsqueda en la red podemos ver que existen numerosas compañías que ofrecen soluciones basadas en flujos de trabajo representados mediante *BPMN (Business Process Model Notation)*, como es el caso de Mediri<sup>1</sup> y su plataforma *mTRIAL*.

Aunque no está estrictamente relacionado con la gestión de estudios clínicos, también existen herramientas como *ResearchKit*<sup>2</sup> de Apple que puede utilizarse para recopilar datos de los pacientes a través de aplicaciones móviles, desarrolladas mediante la herramienta.

No obstante, en este proyecto se pretende realizar un software basado en *BPM (Business Process Management)*, y desarrollado mediante una herramienta especializada en ese ámbito, las suites BPM.

---

<sup>1</sup><https://mediri.com/en/software-solutions>

<sup>2</sup><https://www.apple.com/es/researchkit/>

### 3.2.1. Utilización de sistemas BPM en la gestión de estudios clínicos

No es la primera vez que se utilizan sistemas *BPM* para este tipo de soluciones. En 2014 en la Facultad de Informática de San Sebastián se realizó también un proyecto donde se analizaba el uso de este tipo de sistemas en este área [Hidalgo Bruno, 2014]. En este estudio, se demostraba la efectividad del uso de este tipo de sistemas en el dominio en cuestión, es decir, en el dominio de los estudios clínicos.

Teniendo en cuenta la sistemática a seguir para la realización de cualquier tipo de ensayo clínico, puede verse a simple vista que los sistemas *BPM* son una muy buena solución a este tipo de problemas. Eso es debido a la particularidad de este tipo de sistemas que están centrados en procesos.

No obstante, donde realmente brillan este tipo de sistemas es en la facilidad de refinar los procesos, mejorando así su desempeño. Esto se consigue centrando el desarrollo en los procesos, y facilitando la comprensión gracias al uso del estándar de notación *BPMN*. Además, al poder obtener opiniones de los interesados, este tipo de sistemas nos brindan la posibilidad de generar prototipos rápidamente.

## 3.3. Proveedores de las tecnologías BPM

En la actualidad, la competencia entre las distintas suites *BPM* ha ido dejando solamente a algunas de las más fuertes en el mercado. En la figura 3.1 podemos ver un ranking del pasado 2016.

Si nos fijamos en las empresas que se dedican a desarrollar este tipo de sistemas, podemos ver que se encuentran empresas de la talla de IBM y Oracle. Entre los más importantes, también podemos ver la suite *BPM* utilizada en el estudio antes citado [Hidalgo Bruno, 2014] Bizagi, el cual, a pesar de no estar en el cuadrante de los líderes, está bien posicionado por su rendimiento en ejecución.

## 3.4. BPMN

El estándar *BPMN* permite a las empresas visualizar sus procedimientos internos de negocio de forma gráfica, proporcionando una notación estándar para la comunicación de





**Figura 3.1:** Cuadrante mágico sobre suites BPM 2016

procesos. Actualmente se encuentra en la versión 2.0 creada por la *OMG (Object Management Group)*.

Los elementos más importantes que componen este tipo de modelado son los siguientes:

- **Objetos de flujo**

Representan los conceptos a modelar, clasificados en tres áreas distintas: eventos, actividades y compuertas.

- **Objetos de conexión**

Líneas que conectan los objetos de flujo, y pueden ser de tres tipos: flujos de secuencia, flujos de mensaje y asociaciones.

- **Contenedores y compartimentos**

Elementos utilizados para organizar distintos aspectos de un proceso a la hora de realizar un diagrama. Agrupan elementos visualmente en carriles que pueden representar actividades de un rol o agrupaciones por categoría.

- **Artefactos**

Representan información relevante al modelo, pero no elementos individuales dentro del proceso. Los tres tipos existentes son: anotaciones, grupos y procesos.

Podemos encontrar un póster con todos los elementos de la notación obtenida elaborada por la “*Berliner BPM-Offensive*” y publicada en su página web<sup>3</sup>. En caso de querer ahondar más en el estándar BPMN, es posible encontrar los recursos necesarios en la página oficial de la *OMG*<sup>4</sup>.

---

<sup>3</sup>[http://www.bpmb.de/images/BPMN2\\_0\\_Poster\\_ES.pdf](http://www.bpmb.de/images/BPMN2_0_Poster_ES.pdf)

<sup>4</sup><http://www.omg.org/spec/BPMN/2.0/PDF/>



### **Descripción de la situación actual**

---

En este capítulo se describen el punto de partida del proyecto y sus diferentes condicionantes.

#### **4.1. Soluciones previas**

Tal y como se ha comentado en el apartado de antecedentes, anteriormente se realizó un proyecto previo que trató de evaluar la idoneidad de los sistemas BPM en el dominio de los estudios clínicos.

Este proyecto pretende elaborar una nueva solución más genérica en este campo en lugar de centrar los objetivos en simular un estudio previo. Por ello, se parte desde cero a la hora de realizar todo el proceso de desarrollo.

#### **4.2. Puesta en marcha de los sistemas BPM**

Uno de los aspectos que más limitó desde el principio el proyecto, fue la puesta en marcha y evaluación de las distintas tecnologías BPM a probar. Este punto es importante, ya que si la inicialización del propio sistema resultaba compleja, el tiempo de dedicación a esa tarea aumentaba. Teniendo como resultado la aparición de un riesgo importante para el proyecto.

Este es el procedimiento establecido para la puesta en marcha de los distintos sistemas:

1. Creación de una máquina virtual adecuada a las necesidades de cada tecnología.
2. Instalación del sistema de base.<sup>1</sup>
3. Creación de una copia de seguridad.
4. Instalación y configuración de la suite *BPM*.
5. Realización de una copia de seguridad adicional en el caso de que la instalación sea correcta.

El motivo de realizar este procedimiento es que, de no seguir estos pasos, se generan muchos problemas. Esto es debido a que las suites *BPM* instalan una gran cantidad de software que tiene que estar bien configurado, y cualquier error puede suponer que las instalaciones posteriores no funcionen de manera correcta.

### 4.3. Formación previa

El autor de este proyecto ya poseía los conocimientos básicos en materia de *BPM*, adquiridos en la asignatura de Calidad del Software del Grado de Ingeniería Informática. De hecho, ya existía una formación previa en el software de Bizagi y en el estándar *BPMN*.

En vista del procedimiento seguido para aprender el funcionamiento de Bizagi, se decide que se realizarán los talleres de evaluación que las distintas suites *BPM* suelen ofrecer junto a sus productos, tanto para valorarlos como para adquirir conocimientos sobre ellos.

---

<sup>1</sup>En todos los casos el sistema de base fue Windows 7 Professional

### Normas y referencias

---

En este capítulo se describirán las normas, métodos y herramientas utilizados en este proyecto.

#### 5.1. Disposiciones y normas aplicadas

##### 5.1.1. Norma UNE 157801:2007

El contenido de esta memoria toma como base los contenidos establecidos por la norma *UNE 157801:2007* titulada «*Criterios generales para la elaboración de proyectos de sistemas de información*» y producida por *AENOR*. Esta norma está ligada a realizar una correcta presentación de los proyectos a los clientes. La norma divide la realización de un proyecto en tres fases: Definición, construcción e implantación.

Dado que en este proyecto se trata de dar una solución al problema de la gestión de los ensayos clínicos, y que solamente estamos ofreciendo una posible solución al cliente, nos encontramos en la fase de definición.

Es tras presentar el proyecto a el cliente cuando éste decidirá si contratar la construcción del sistema propuesto y su posterior implantación.

### 5.1.2. LOPD: Ley Orgánica de Protección de Datos

Tal y como viene publicado en el boletín oficial del estado [del Estado, 1999] *«la ley Orgánica tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor e intimidad personal y familiar».*

Esta ley es algo muy a tener en cuenta en el ámbito que se mueve este proyecto pues los datos utilizados en el mismo son de carácter sensible por afectar a la intimidad personal. En este caso habría que dejar claro que los datos a almacenar en este sistema son los relativos a la salud de los pacientes.

En base a los datos a almacenar, el nivel de protección de los datos deberá ser de nivel ALTO, aunque en esta primera fase del proyecto no es de aplicación por que no se van a emplear datos reales.

Además de lo anterior, a partir del 25 de mayo de 2018 se deberá cumplir la normativa del Reglamento General de Protección de Datos [AGPD, b], y por ello es conveniente hacer referencia a los contenidos establecidos en la Guía del Reglamento General de Protección de Datos para responsables de tratamiento [AGPD, a].

## 5.2. Bibliografía

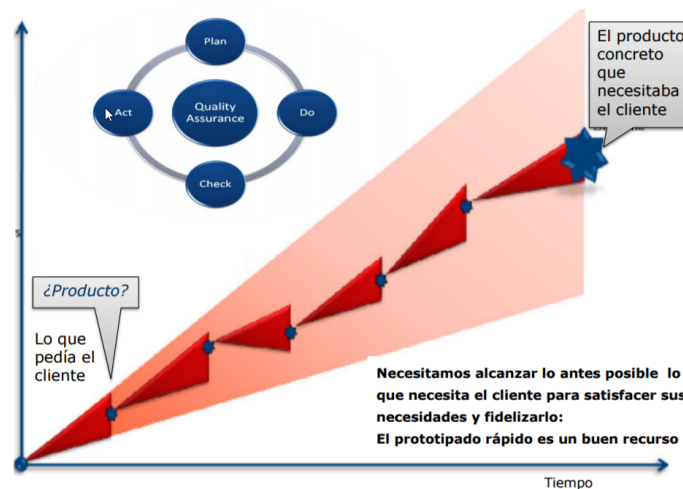
- [AGPD, a] AGPD. Guía del reglamento general de protección de datos para responsables de tratamiento. [https://www.agpd.es/portaIwebAGPD/temas/reglamento/common/pdf/guia\\_rgpd.pdf](https://www.agpd.es/portaIwebAGPD/temas/reglamento/common/pdf/guia_rgpd.pdf). [Online; Ultimo Acceso 05/07/2017].
- [AGPD, b] AGPD. Reglamento general de protección de datos. [https://www.agpd.es/portaIwebAGPD/canaIdocumentacion/legislacion/union\\_europea/reglamentos/common/pdfs/Reglamento\\_UE\\_2016-679\\_Proteccion\\_datos\\_DOUE.pdf](https://www.agpd.es/portaIwebAGPD/canaIdocumentacion/legislacion/union_europea/reglamentos/common/pdfs/Reglamento_UE_2016-679_Proteccion_datos_DOUE.pdf). [Online; Ultimo Acceso 05/07/2017].
- [Bizagi, 2017] Bizagi (2017). Arquitectura del producto. [http://help.bizagi.com/bpm-suite/es/suite\\_producta.htm](http://help.bizagi.com/bpm-suite/es/suite_producta.htm). [Online; Ultimo Acceso 05/07/2017].
- [del Estado, 1999] del Estado, J. (1999). Ley orgánica 15/1999, de 13 de diciembre, de protección de datos de carácter personal. <https://www.boe.es/buscar/pdf/1999/BOE-A-1999-23750-consolidado.pdf>. [Online; Ultimo Acceso 05/07/2017].
- [Durán Quintana, 2004] Durán Quintana, J. c. A. (2004). Investigación en Terapéutica clínica: Ensayos clínicos con medicamentos. *Educación Médica*, 7:51 – 55.
- [Hidalgo Bruno, 2014] Hidalgo Bruno, A. (2014). Sistemas bpm en la gestión de ensayos clínicos.
- [IMIM, 2006] IMIM, I. H. d. M. d. M. (2006). Términos de ensayos clínicos. <https://www.imim.cat/media/upload/arxiu/terminologia.pdf>. [Online; Ultimo Acceso 05/07/2017].
- [UAM, 2006] UAM, U. A. d. M. (2006). Acerca de los ensayos clínicos. <https://www.uam.es/departamentos/medicina/farmacologia/especifica/cfc/acerca.html>. [Online; Ultimo Acceso 05/07/2017].
- [Wikipedia, 2017] Wikipedia (2017). Ensayos clínicos. [https://es.wikipedia.org/wiki/Ensayo\\_cl%C3%ADnico](https://es.wikipedia.org/wiki/Ensayo_cl%C3%ADnico). [Online; Ultimo Acceso 05/07/2017].



## 5.3. Métodos, Herramientas, Modelos, Métricas y Prototipos

### 5.3.1. Modelo de desarrollo basado en prototipos

Dada la complejidad del dominio, y de cara a dar una solución óptima al problema, se realizará un desarrollo basado en prototipos. Esto, sumado a las facilidades en cuanto a desarrollo mediante suites *BPM*, nos permitirá ir adecuando el producto a las necesidades del cliente mediante la sucesión de prototipos cada vez más completos y cercanos al problema a resolver. En este tipo de desarrollo habitualmente se realizan modelos de análisis y desarrollo para no construir lo que no se necesita, evitando de esta forma costes innecesarios.



**Figura 5.1:** Desarrollo basado en prototipos<sup>1</sup>

Realizar este tipo de desarrollo no obstante genera una serie de necesidades:

- Uso de estándares como *UML (Universal Modeling Language)* para la comunicación entre interesados.
- Realización de evaluaciones de las distintas tecnologías y herramientas buscando aquellas que permitan agilizar el desarrollo.
- Utilización de una metodología ágil.

<sup>1</sup>Obtenido de las transparencias de la asignatura “Calidad del Software” y elaboradas por Juan Manuel Pikatza Atxa

- Reelaboración de modelos de análisis y diseño en base a las conclusiones obtenidas en los prototipos.

### 5.3.2. Metodología de desarrollo OpenUp

*OpenUp* es una metodología ágil que cubre todo el ciclo de vida. Para el desarrollo de este proyecto se ha seguido la metodología en su primera versión a petición del cliente. En la actualidad la metodología ha ido evolucionando y refinándose cada vez más, añadiendo nuevos artefactos a la misma, y es por ello que se especifica la versión. Esta metodología está basada en el proceso racional unificado (*RUP*), y la idea principal de la misma es la de realizar un desarrollo de software iterativo que a su vez sea mínimo, completo y extensible.

De cara a explicarlo de una forma sencilla podríamos definir tres puntos clave de este proceso de desarrollo de software como son: *roles*, *tareas* y *artefactos*.

Las personas están categorizadas en roles y realizan tareas que a su vez producen artefactos o dicho de otra forma, entregables internos. Es posible encontrar más información acerca de la metodología en su web<sup>2</sup>.

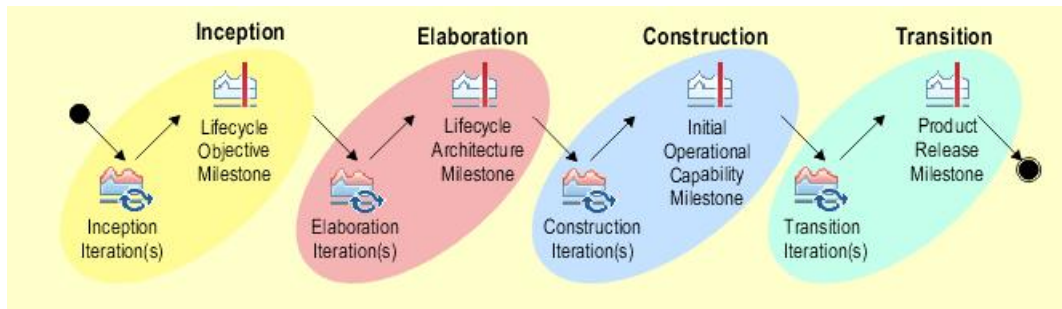
#### 5.3.2.1. Ciclo de vida

El ciclo de vida del desarrollo de un proyecto de software basado en esta metodología se compone de cuatro fases, tal y como podemos ver en la imagen 5.2. Para avanzar de fase se deberán cumplir los requisitos de las mismas:

- **Inicio:** Compresión del problema a solucionar.
- **Elaboración:** Se valida la solución aportada.
- **Construcción:** Desarrollo y verificación incremental de la solución.
- **Transición:** Transferencia al entorno operacional y validación.

---

<sup>2</sup><http://epf.eclipse.org/wikis/openup/>



**Figura 5.2:** Ciclo de vida de *OpenUp*

### 5.3.3. Herramientas y librerías

El listado de herramientas y librerías utilizadas en el desarrollo del proyecto hasta la fecha ha sido el siguiente:

- **Atom:** Editor gratuito compatible con la mayoría de lenguajes actuales y extensible mediante *plugins*.
- **Bcrypt-nodejs:** Librería para encriptar las claves de los usuarios.
- **Bizagi Studio:** Suite de desarrollo de Bizagi que incluye todo lo necesario para elaborar aplicaciones basadas en procesos.
- **Bootstrap:** Framework *CSS* para generar interfaces agradables fácilmente.
- **Express:** Framework para el desarrollo de aplicaciones web y *APIs* (*Application Programming Interface*) basadas en *Node.js*
- **GanttProject:** Aplicación para elaborar los diagramas de gantt necesarios para la memoria.
- **IBM BPM:** Suite de desarrollo de *BPM* de IBM que incluye todo lo necesario para elaborar aplicaciones basadas en procesos.
- **Jade:** Motor de plantillas compatible con express.
- **Json2csv:** Librería para conversión de documentos *JSON* (JavaScript Object Notation) a *CSV* (*Comma Sepparated Values*).
- **Microsoft SQL Server 2008 Express:** Servidor de Base de datos instalado con Bizagi.

- **MongoDB:** Motor de bases de datos *NoSQL*.
- **Mongoose:** Librería para conectar con *MongoDB*.
- **Node.js:** Entorno de ejecución para *JavaScript* que usa un modelo de operaciones entrada/salida sin bloqueo y orientado a eventos, que lo hace liviano y eficiente.
- **Passport:** Librería para gestionar los inicios de sesión.
- **Postman:** Aplicación del navegador *Google Chrome* y que permite consumir *APIs* con el fin de realizar pruebas.
- **Rational Software Architect y StarUML:** Aplicaciones de diseño que utilizan *UML*.
- **Robomongo:** Aplicación para el manejo de la base de datos *mongoDB* de forma visual.
- **SourceTree:** Aplicación para el control de versiones y repositorios de código.
- **Suite de Microsoft Office:** Suite de ofimática de Microsoft para la elaboración de la documentación interna.
- **TeXstudio:** Entorno de desarrollo para la escritura de documentos.
- **VMware Workstation Pro:** Software de virtualización mediante el cual crear máquinas virtuales.
- **Xampp:** Distribución de servidor apache gratuita. En este caso se ha utilizado para aprovechar el servidor de correo *Mercury Mail* que viene preinstalado.



### Definiciones y abreviaturas

---

**AENOR** Asociación Española de Normalización y Certificación.

**API** Del inglés *Application Programming Interface* o interfaz de programación de aplicaciones, es un conjunto de subrutinas, funciones y métodos que permiten hacer uso de un sistema con una cierta capa de abstracción.

**BAM** *Business Activity Monitoring*.

**BPM** La gestión de procesos de negocio (en inglés *Business Process Management*) es una metodología corporativa cuyo objetivo es mejorar el desempeño (eficiencia y eficacia) de la organización a través de la gestión de los procesos de negocio, que se deben diseñar, modelar, organizar, documentar y optimizar de forma continua.

**BPMN** Del inglés *Business Process Model and Notation*, es un lenguaje formal y estandarizado que permite modelar, simular y, eventualmente, ejecutar procesos de negocio. Su sintaxis está basada en elementos gráficos.

**BPMS** *BPMS(Business Process Management System)* es el software generado mediante una suite *BPM*, en el caso de este proyecto en forma de aplicación web.

**BSON** Acrónimo de *Binary JSON*, es el tipo de datos que almacena *MongoDB*.

- CEIC** Comité Ético de Investigación Clínica, cada comunidad autónoma tiene su propio comité. El objetivo de este, es velar por la protección de los derechos, seguridad y bienestar de los sujetos que participen en un ensayo y de ofrecer garantía pública al respecto.
- CSS** Hojas de Estilo en Cascada (*Cascading Style Sheets*) es el lenguaje utilizado para definir el aspecto y estética de documentos *HTML* o *XML*.
- CSV** Del inglés *Comma Separated Values*, es un formato libre para la representación de datos mediante tablas. Cada fila de estos documentos equivale a una fila de la tabla.
- Cuestionario** Cuestionarios clínicos validados para homogeneizar las revisiones sistemáticas de resultados de diferentes estudios clínicos, con el objetivo de obtener una fuente de información científica de mayor valor.
- Gartner** Asesoría de nivel mundial encargada de ayudar a los responsables de las compañías aportando información que les permita conseguir sus objetivos y tomar las decisiones adecuadas.
- HTML** Lenguaje de marcas de hipertexto (*HyperText Markup Language*) es un lenguaje de marcado utilizado para el desarrollo de páginas web.
- HTTP** Abreviatura de *Hypertext Transfer Protocol*, el cual es un protocolo utilizado para las comunicaciones en internet.
- JSON** Del inglés *JavaScript Object Notation*, es un formato ligero de intercambio de datos. Se compone de pares nombre/valor donde el valor puede ser una sola cosa o varias.
- MVC** Modelo Vista Controlador.
- Máquina virtual** Software que simula a un ordenador y puede ejecutar programas como si fuese un ordenador real.
- OMG** *Object Management Group*.
- OpenUp** Es un proceso modelo, extensible y *open source*, dirigido a gestión y desarrollo de proyectos de software basados en un desarrollo iterativo, ágil e incremental apropiado para proyectos pequeños y de bajos recursos; y es aplicable a un conjunto amplio de plataformas y aplicaciones de desarrollo. Los contenidos de esta metodología están basados en *RUP*.

**Procesos y subprocesos** Procesamiento o conjunto de operaciones con el objetivo de obtener un resultado.

**REST** Transferencia de Estado Representacional (en inglés *Representational State Transfer*) es un estilo de arquitectura para el desarrollo de servicios web, con un conjunto de operaciones bien definidas basada en métodos *HTTP*.

**RUP** El Proceso Racional Unificado o *RUP* (por sus siglas en inglés de *Rational Unified Process*) es un proceso de desarrollo de software desarrollado por la empresa Rational Software, actualmente propiedad de IBM.

**Servicio Web** Tecnología que utiliza un conjunto de estándares y protocolos para el intercambio de datos entre aplicaciones.

**SOAP** Del inglés *Simple Object Access Protocol* es un protocolo para servicios web basado en *XML*).

**Subproceso Múltiple** Conjunto de actividades o tareas ejecutadas de forma paralela.

**Suite BPM** Suite *BPM* o Software *BPM*, es el conjunto de servicios y herramientas software que facilitan la administración de procesos de negocio y su ciclo de vida.

**Verbos HTTP** Se refiere a un conjunto de acciones o métodos establecidos por el protocolo *HTTP*.

**XML** Lenguaje de marcado extensible o en inglés *eXtensible Markup Language*. Es una especificación para el diseño de lenguajes de marcado, que permite definir etiquetas personalizadas tanto para organizar como describir datos.





## Requisitos iniciales

---

En este capítulo se recogen los requisitos iniciales capturados durante la fase de inicio del proyecto.

### 7.1. Requisitos funcionales

Entre los requisitos funcionales inicialmente identificados durante las fases de captura de requisitos podemos encontrar las siguientes:

- La solución final será una aplicación web.
- La solución deberá contemplar los elementos más comunes de los estudios clínicos.
- Se deberán poder extraer los datos del estudio en un formato que permita su posterior análisis estadístico.
- Existirán 4 tipos de usuarios
  - Administrador o Investigador principal encargado de la gestión del estudio.
  - Médicos participantes en el estudio.
  - Administrativos o Investigadores auxiliares encargados de realizar la supervisión de los distintos permisos o tareas de índole administrativa.

- Pacientes/Cuidadores participantes en el estudio.
- Se deberá automatizar en la medida de lo posible las tareas que no requieran de interacción humana.
- Habrá que asegurar el rigor del estudio en proceso.
- Se deberá ofrecer algún tipo de control sobre la participación de los médicos en el estudio.
- La aplicación comprenderá las siguientes fases:
  - **Fase de inicio del estudio:** En esta fase se diseñará el estudio a llevar a cabo y se corroborará que se obtienen los permisos necesarios. Además, se establecerán los grupos de médicos que van a formar parte del estudio.
  - **Fase de evolución del estudio:** Aquí se darán de alta los pacientes/cuidadores y se registrarán las visitas de los mismos. Mientras tanto el administrador y los administrativos se dedicarán a controlar y gestionar el estudio.
  - **Fase de finalización del estudio:** En este punto el estudio habrá sido finalizado y no se permitirá añadir nuevos datos ni médicos. Se mostrará un resumen de los distintos participantes y las incidencias ocurridas durante el proceso.

## 7.2. Requisitos no funcionales

Entre los requisitos no funcionales podemos encontrar:

- La solución debe ser adaptable fácilmente a nuevos tipos de estudios clínicos.
- La aplicación desarrollada debe estar sujeta a posibles mejoras del proceso a seguir, con el objetivo de mejorar la eficacia de la aplicación.
- Inicialmente, se implementarán unos formularios concretos, no obstante, se debe diseñar un sistema que permita la inclusión de nuevos formularios fácilmente, ya que en cada estudio a realizar los formularios son distintos.
- A petición del cliente se estableció como requisito de presentación, que este realizado en base a los contenidos predefinidos en la norma *UNE 157801:2007* y siguiendo la metodología de desarrollo *OpenUp*.

- La información del proyecto debe expresarse en estándares que faciliten la transmisión efectiva de toda la información a futuros equipos de trabajo.



## CAPÍTULO 8

---

### Alcance

---

En este capítulo se incluyen de forma clara y concisa todos los entregables de este proyecto.

#### 8.1. Alcance del proyecto

Como resultado de aplicar una norma y una metodología al desarrollo de este proyecto se distinguirán dos elementos distintos. Por un lado, estará la estructura documental del proyecto siguiendo la norma *UNE 157801:2007*. Por otro lado, tendremos los contenidos de la metodología materializados en una web con los artefactos que exige la misma con el objetivo de facilitar el compartimiento de la información.

<https://evdiscover-9fc8c.firebaseio.com/>

Con respecto al último punto, pueden verse los artefactos que se subirán a dicha web en la tabla 8.1, a excepción de los artefactos de *Build e Implementación* pues estarán alojados en la máquina virtual para el desarrollo de este proyecto, y que por motivos de espacio no tiene cabida dentro de la web del proyecto.

<b>Disciplina</b>	<b>Artefactos</b>
Arquitectura	Cuaderno de arquitectura
Desarrollo	Diseño Build Implementación Test de desarrollo
Gestión	Planes de iteración Plan de proyecto Lista de riesgos Lista de elementos de trabajo
Requisitos	Glosario Soporte de especificación de requisitos Casos de uso Modelo de casos de uso Modelo de análisis Visión
Test	Casos de test Test log Test Script

**Tabla 8.1:** Artefactos de la metodología *OpenUp* a presentar

### 8.1.1. Exclusiones

Quedan excluidos del alcance del proyecto los siguientes elementos:

- Realizar un nuevo modelo de análisis tras haber seleccionado las tecnologías y haber refinado los requisitos. Si bien se realizará una primera versión del mismo con el objetivo de entender mejor el problema a resolver, por las restricciones temporales de este proyecto no se ha realizado más que una primera versión. No obstante, de cara a la construcción habría que realizar una nueva versión del mismo.
- Realizar un análisis y posterior diseño del sistema teniendo en cuenta las leyes que afectan a este proyecto. No se realizará dicho análisis, aunque si sería necesario tenerlo en cuenta en posteriores fases del proyecto.
- Quedan excluidos de ser presentados los siguientes apartados de la norma:
  - Estudios con entidad propia

- Anexos de la memoria:
  - Estimación de tamaños y esfuerzos
  - Planes de gestión de la ejecución del proyecto
  - Seguridad

El motivo de realizar esta exclusión es debido al carácter educativo de este proyecto.





### Hipótesis y restricciones

---

En este capítulo se describen las hipótesis de partida del proyecto y las restricciones impuestas para el desarrollo del mismo.

#### 9.1. Hipótesis

La hipótesis de partida de este proyecto consiste en que los *BPMS (Business Process Management System)* son un tipo de software adecuado para la gestión de los estudios clínicos. Esto queda justificado, ya que cualquier estudio debe seguir un proceso concreto para asegurar el rigor del mismo.

La separación de procesos y datos de carácter analítico mitiga la dependencia generada por las tecnologías *BPM*.

#### 9.2. Restricciones

Inicialmente, y debido a las peticiones del cliente, se han impuesto las siguientes restricciones, las cuales han limitado en cierta medida las opciones a tener en cuenta para la realización de algunos elementos relacionados con el proyecto:

- Facilitar el mantenimiento y evolución de la solución promoviendo la mejora continua de procesos gracias al desarrollo de una aplicación basada en un *BPMS*.

- Utilización de una metodología ágil para la elaboración y desarrollo del proyecto (*OpenUp*).
- Presentación del proyecto siguiendo el estándar *UNE 157801:2007*.

### Estudio de alternativas y viabilidad

---

En este capítulo se tratan las alternativas a la solución tenidas en cuenta, criterios de valoración, justificación de la alternativa elegida, y razones para los descartes realizados.

#### 10.1. Suites BPM

A la hora de realizar el estudio de alternativas de las distintas suites se decidió reducir la posibilidad de selección a aquellas aplicaciones de las que se pudiera obtener una licencia que permitiera el desarrollo del prototipo.

Tras realizar un filtrado, las opciones se redujeron a dos. Por un lado, la suite *BPM* de Bizagi, la cual ya se conocía en parte, y por otro lado la de IBM. Una vez reducido el cerco a estas dos opciones, el siguiente paso fue analizar cada una de las herramientas mediante los talleres de evaluación que ambas poseen.

##### 10.1.1. IBM BPM

Esta potente suite desarrollada por IBM incluye muchísimas funcionalidades e integraciones con los servicios de otras aplicaciones de la marca. Esto puede suponer que si se realizará un buen estudio de los distintos componentes necesarios para su implantación, podríamos ampliar las posibilidades con cada componente nuevo conectado.

No obstante, y pese a que el software de esta marca resulta muy bueno, es complicado conseguir que todos los componentes trabajen juntos sin generar problemas. Durante la fase de investigación, se realizaron varias pruebas en las que se pudo apreciar tanto las posibilidades de esta tecnología, como las dificultades de su implantación.

Si bien se realizó el taller de evaluación de la plataforma, hubo que invertir mucho tiempo en poner a punto el sistema. Además, pese a conseguir realizar el taller de evaluación, algunas cosas no funcionaban. Esto sumado a la dificultad a la hora de solucionar fallos del propio software, implicaba un riesgo inherente en el proyecto.

### 10.1.2. Bizagi Studio

Dado que se conocía la herramienta de antemano y que ésta es gratuita, al menos de cara a realizar un prototipo, fue una de las primeras opciones a tener en cuenta. Aunque no es una herramienta tan potente como la de IBM, sí que resulta sencilla de utilizar, y su implantación no requiere de demasiadas configuraciones.

Esta suite está situada en el cuadrante de competidores de Gartner que puede verse en la figura 3.1. Eso, sumado a que tiene una versión gratuita que permite el desarrollo de prototipos sin ánimo de lucro, la convierte en una alternativa ideal para esta primera fase del proyecto.

### 10.1.3. Selección de la suite BPM

En la tabla 10.1 puede verse una comparativa con algunos de los puntos más relevantes para la toma de la decisión.

Finalmente, tras realizar un análisis y dedicar un tiempo a evaluar cada opción, se decidió optar por el software de Bizagi. Esto se debe a que inicialmente se busca desarrollar un prototipo pequeño para darle una primera solución al problema.

En cualquier caso, en ningún momento se debe dejar de lado la solución de IBM, ya que se ha podido comprobar la potencia de la misma. Además, existen numerosas aplicaciones de tipo estadístico de la marca, las cuales podríamos llegar a conectar a nuestro sistema para aumentar las funcionalidades, como por ejemplo *SPSS*<sup>1</sup>.

---

<sup>1</sup><https://www.ibm.com/analytics/es/es/technology/spss/>

<b>Característica</b>	<b>Bizagi</b>	<b>IBM</b>	<b>Observaciones</b>
Posibilidad de obtener licencia	✓	✓	Pese a ser de pago la UPV/EHU tiene una licencia para estudiantes de IBM, como se ha comentado antes, Bizagi ofrece una versión gratuita.
Documentación sencilla y bien organizada	✓	✗	La documentación de IBM pese a ser extensa a menudo redirige a otros componentes de la marca.
Facilidad de puesta en marcha	✓	✗	IBM ha requerido mucho esfuerzo para poder realizar pruebas. Además, no ha sido posible ajustar la configuración todo lo necesario, pese a seguir los manuales.
Reconocimiento profesional	✓	✓	Ambas aparecen en el cuadrado mágico de Gartner
Comunidad Activa y Foros	✓	✓	La comunidad de Bizagi destaca por su actividad y solución de los problemas
Ejemplos para el aprendizaje	✓	✓	Aunque ambas tienen documentación con ejemplos, en el caso de Bizagi es más sencillo encontrarlos.
Trazabilidad de los fallos sencilla	✗	✗	Debido a la gran cantidad de software que se instala, la trazabilidad de los fallos es compleja en ambos sistemas.

**Tabla 10.1:** Comparativa de suites BPM

Debido a la posibilidad de separar cada uno de los componentes necesarios para su implantación, la solución de IBM puede ser adecuada a la hora de realizar una implantación real a gran escala. No obstante, se deberían conseguir técnicos especializados para configurar y afinar los distintos componentes de este sistema.

## 10.2. Cumplimentación de cuestionarios

Con el objetivo de eliminar la dependencia tecnológica de Bizagi en cuanto a la extracción de datos y la implementación de nuevos formularios, se decidió externalizar estas funciones a un componente distinto en forma de aplicación web.

El sistema escogido en este caso para implementar esta parte de la solución debía proveer también de la posibilidad de conexión con Bizagi, preferiblemente mediante servicios web. A la hora de escoger el tipo de servicio a utilizar, se valoró si utilizar servicios *REST* (*Representational State Transfer*) o *SOAP* (*Simple Object Access Protocol*).

En este punto y puesto que se trata de realizar un prototipo y que existe la opción de que los servicios cambien a lo largo de los distintos prototipos, se decidió utilizar *REST*, ya que provee de una mayor flexibilidad y escalabilidad además de sencillez a la hora de su creación, frente a la necesidad de tipado fuerte y definición de los servicios *SOAP*.

### 10.2.1. Framework de desarrollo

A la vista de la necesidad de realizar un desarrollo ágil en este aspecto, se valoraron varios frameworks de desarrollo web, tras realizar un filtrado de los candidatos en base a la facilidad de crear servicios web de manera sencilla. Los candidatos seleccionados y probados fueron los de la tabla 10.2.

En este punto, se decidió optar por *Express* y *Node.js*, pues su facilidad de ampliación mediante módulos y la amplia comunidad de desarrollo lo convierte en una de las mejores opciones.

### 10.2.2. Base de datos

Ante la necesidad de establecer una base de datos para el almacenamiento de los distintos cuestionarios y sus respuestas, se valoraron dos tecnologías distintas en este aspecto, basadas en las opciones más frecuentemente utilizadas.

- **Bases de datos relacionales**

Una base de datos relacional es una colección de datos con relaciones predefinidas entre ellos. Estos elementos se organizan como un conjunto de tablas formadas por

Característica	Express	SlimFrameWork	Observaciones
Tecnología de base	Node.js	PHP	-
Documentación sencilla y bien organizada	✓	✓	-
Facilidad de ampliación mediante módulos	✓	✗	En el caso de Express al estar basado en Node.js la cantidad de paquetes que encontramos de cara a añadir funcionalidades son inmensos.
Creación de servicios web sencilla	✓	✓	-
Comunidad activa	✓	✓	La comunidad alrededor de Express y Node.js es mayor.

**Tabla 10.2:** Comparativa de Frameworks Web

columnas y filas, representando cada fila un valor relacionado con una entidad u objeto. Es obligatorio definir la estructura de cada una de las tablas en su implementación.

#### ■ Bases de datos *NoSQL*

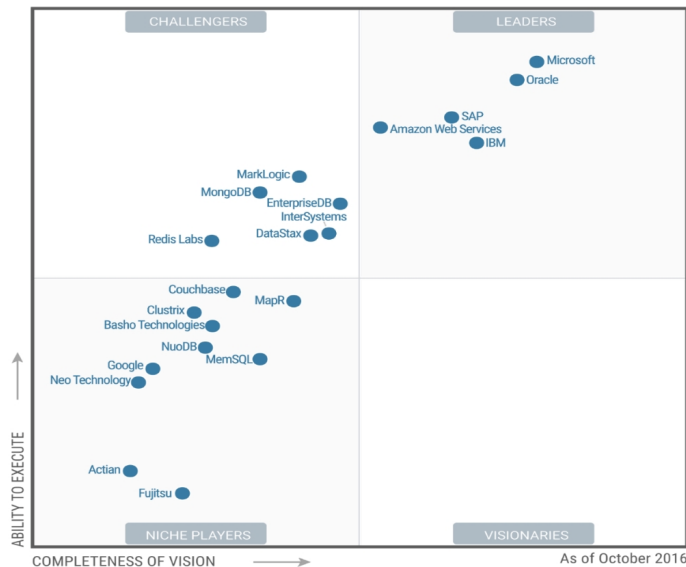
En las bases de datos *NoSQL*, los datos almacenados no requieren estructuras fijas como tablas de las bases de datos relacionales. Esto implica un desarrollo ágil ya que no es necesario estar definiendo nuevas tablas constantemente.

A la hora de valorar qué tipo de tecnología utilizar en este aspecto, lo primero fue analizar el tipo de datos a almacenar. En el caso de este proyecto, al ser los datos variables, y ante la necesidad futura de implementar una gran cantidad de cuestionarios, que pese a mantener una estructura común, pueden variar en cuanto contenidos, se utilizará una base de datos de tipo *NoSql*.

Tras analizar las opciones disponibles en el mercado, se decidió utilizar *MongoDB* pues



es una de las opciones mejor consideradas, y una de las competidoras más fuertes, tal y como podemos ver en la figura 10.1.



**Figura 10.1:** Cuadrante mágico de Gartner sobre bases de datos operacionales.

### 10.3. Estudio de viabilidad

Tras realizar las pruebas necesarias para comprobar las virtudes y defectos de las distintas tecnologías, se decidió que no es viable realizar el primer prototipo que de solución al problema mediante la tecnología de IBM. Esto se debe a que en las primeras pruebas realizadas se invirtió mucho tiempo solucionando distintos problemas de comunicación entre componentes del propio software.

En cualquier caso, gracias a la velocidad de desarrollo que nos permite la tecnología de Bizagi, el desarrollo de prototipos incrementales de forma rápida resulta viable. Además, de cara al uso del software, las interfaces de Bizagi resultan muy intuitivas, y permitirán que los usuarios del sistema se acomoden rápidamente a su uso.

En cuanto al componente relativo a los cuestionarios, y debido a las tecnologías a utilizar, el desarrollo del mismo resulta viable también, pues tanto el tipo de base de datos como la tecnología para desarrollar la web favorecen el desarrollo ágil.

## Descripción del sistema propuesto

---

En este capítulo se describe el sistema propuesto como solución al problema planteado.

### 11.1. Características de la solución

#### 11.1.1. Restricciones

Con el fin de mostrar un prototipo inicial, se han establecido una serie de restricciones para agilizar el proceso de desarrollo. Estas restricciones están justificadas por la gran cantidad de tipos de estudios clínicos existentes

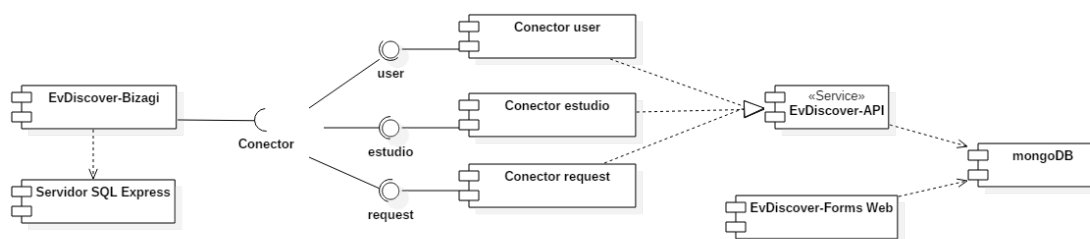
- La aplicación dará soporte a los estudios que cumplan los siguientes requisitos:
  - Estudios de fase II en adelante, por ser aquí donde se realizan pruebas con grupos distintos, y a partir de donde se obtiene una mayor evidencia científica.
  - Estudios controlados donde exista tanto un grupo de control como uno de experimentación.
  - Estudios de tipo prospectivo, ya que son estos los que necesitan más ayuda para controlar los procesos.
  - Estudios aleatorizados, donde no exista restricción sobre quién pertenece a qué grupo.

- Los cuestionarios a pasar al personal médico y a los pacientes se limitan a los establecidos a rellenar en cada visita. Es decir, en cada visita se rellenarán tanto un cuestionario por parte del médico como uno por parte del paciente/cuidador.
- Solamente se podrá asignar un tipo de formulario a cada uno de los tipos de usuario. Esto significa que el médico solo tendrá un tipo de cuestionario, el paciente otro y el cuidador otro.
- En el componente relativo a los cuestionarios, estos no estarán limitados por el rol de usuario ya que las solicitudes de cumplimentación serán generadas desde el proceso definido en Bizagi donde sí se controlará esto.

### 11.1.2. Qué hace la solución

Antes que nada, es importante explicar que la solución propuesta se ha dividido en dos componentes distintos. A saber, uno para el control y gestión de un estudio clínico, y otro para la recolección de datos, vía cuestionarios. De aquí en adelante se hará referencia a ellos como EvDiscover-Bizagi (para el componente de gestión) y EvDiscover-Forms (para el componente de los cuestionarios). A modo de comunicación, entre ambos se ha desarrollado una interfaz de comunicación en forma de *API REST*.

En el diagrama de la figura 11.1 podemos ver la arquitectura general del sistema. Pueden encontrarse información más detallada en la web del proyecto.



**Figura 11.1:** Diagrama de componentes del sistema software

El componente EvDiscover-Bizagi es un *BPMS* desarrollado generado mediante la suite *BPM* de Bizagi, y nos ofrece la posibilidad de gestionar un estudio clínico. Para ello el usuario de tipo administrador deberá crear un nuevo caso (estudio). En este punto todas las tareas asignadas al administrador del estudio serán realizadas por este usuario.

Explicado de forma simple EvDiscover-Bizagi es una aplicación web que irá asignando tareas a los distintos implicados, y estos deberán ir realizando aquellas que vayan apa-

reciendo en su bandeja de entrada. De esta forma el proceso irá avanzando hasta llegar a un punto donde este se finalice. Es posible ver las actividades de dicho proceso en el apartado 11.2.4 de esta memoria.

En este sistema no es posible acceder mediante un registro público en ninguno de los casos, y los usuarios a utilizar se dividen en 5 tipos distintos. En la tabla 11.1 podemos ver los tipos de usuarios existentes y los sistemas en los que están presentes.

Tipo de usuario	EvDiscover-Bizagi	EvDiscover-Forms
Administrador	<b>x</b>	<b>x</b>
Administrativo	<b>x</b>	
Médico	<b>x</b>	<b>x</b>
Paciente		<b>x</b>
Cuidador		<b>x</b>

**Tabla 11.1:** Tipos de usuario utilizables en cada sistema

El motivo de esta separación es que a pesar de que el sistema está compuesto por dos componentes conectados, las utilidades de cada uno de ellos son totalmente distintas, y es por ello que, por ejemplo, un administrativo no tiene cabida en el sistema de EvDiscover-Forms, ya que solamente los usuarios que cumplimenten cuestionarios (médicos, pacientes y cuidadores) o el que extraiga los datos (el administrador) tienen acceso al mismo. De la misma forma no tiene sentido incluir los pacientes/cuidadores en el proceso de gestión, pues a pesar de que sus datos, si existirán, no realizan ninguna acción en el propio proceso. Todos los usuarios del componente relativo a los cuestionarios, son creados directamente desde el proceso elaborado en el componente de gestión. En el componente basado en Bizagi se realiza mediante un panel de administración que genera automáticamente la suite, y a través del cual se realiza la gestión de todos ellos.

En lo que respecta al componente EvDiscover-Forms, su funcionalidad es muy sencilla, limitándose únicamente a dos tareas. Por una parte, tenemos la cumplimentación de cuestionarios que es realizada por los médicos, pacientes y cuidadores. Estos usuarios solamente visualizarán los formularios a rellenar, y tras cumplimentarlos, ya no podrán realizar ninguno más hasta que se genere una nueva solicitud de cumplimentación.

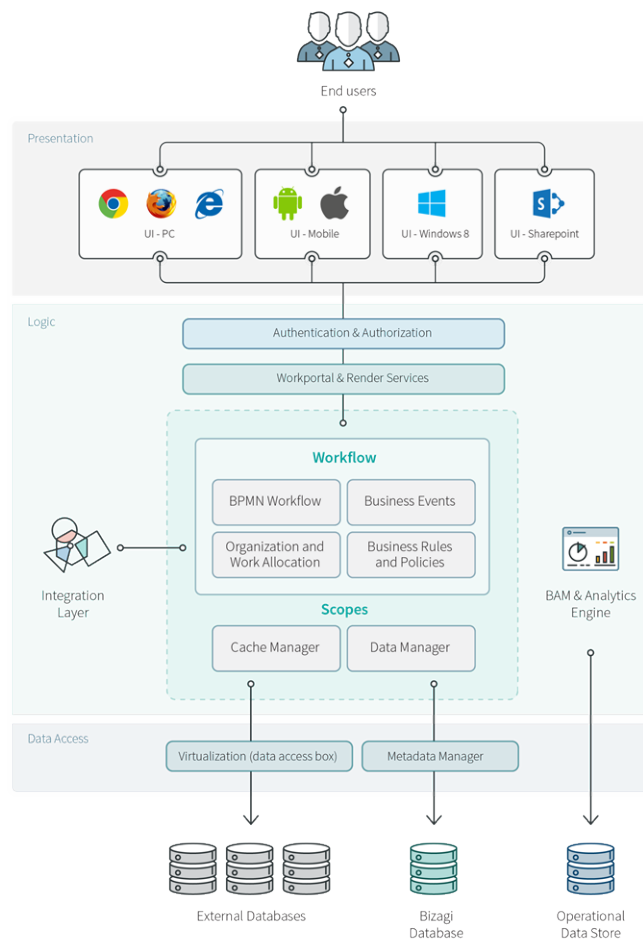
Por otra parte, tenemos al administrador que podrá conectarse al sistema para extraer los datos obtenidos mediante los cuestionarios cumplimentados en cualquier momento. La exportación se realizará en formato CSV, que facilita el análisis posterior de los datos con diferentes aplicaciones software de carácter estadístico.

En el componente de cuestionarios, tanto los usuarios como las solicitudes para cumpli-

mentar un formulario concreto, son generadas desde el proceso de gestión. En el caso de las solicitudes de cumplimentación, se generarán pares de cuestionarios médico/paciente o médico/cuidador en cada una de las visitas.

## 11.2. EvDiscover-Bizagi

### 11.2.1. Arquitectura



**Figura 11.2:** Arquitectura de Bizagi

Pese a la cantidad de componentes que contiene Bizagi, solamente se explicaran los más relevantes. La información ha sido extraída directamente desde la página web de la compañía [Bizagi, 2017].

### **Portal de trabajo**

El portal de trabajo es una aplicación basada en protocolos web para usuarios finales, donde estos podrán ir creando sus propios procesos o atender a las actividades que se les han asignado.

### **Motor de flujo de trabajo BPMN**

El Motor *BPMN* es conocido en Bizagi como el “*Workflow Engine*”, es el encargado de controlar y ejecutar el flujo de proceso.

### **BAM (*Business Activity Monitoring*) y Análisis**

Las opciones de análisis y *BAM* proveen una poderosa opción de informes para indicadores de gestión (para monitorización, control y mejora de los procesos).

### **Administrador de datos**

El motor de datos de Bizagi se conoce como “*EntityManager*”. Es el encargado de administrar la información de forma segura, consistente y sincronizada. El acceso a la base de datos queda completamente transparente a los usuarios.

### **Seguridad y Organización**

Módulo de seguridad para manejar las opciones de autenticación y autorización para los distintos roles.

### **Reglas de negocio**

Bizagi ofrece un motor para la ejecución de reglas de negocio o políticas que dan la lógica al proceso de negocio.

### **Capa de Integración (*Enterprise Mapping Layer*)**

La capa de integración es un módulo robusto, flexible y poderoso que complementa al proceso en soluciones corporativas. En proyectos donde se requiere integrar los procesos con aplicaciones existentes, la capa de integración de Bizagi permite múltiples maneras de integración:

- Servidor de Correo
- Repositorio central de documentos
- Conexión con fuentes de datos externas
- Eventos de negocio

- Integración de Aplicaciones

### **Administración de Carga de Trabajo**

Bizagi permite definir el balanceo de carga de trabajo en la asignación de actividades. Para una optimización de las asignaciones, es importante tener en cuenta algunos criterios como el calendario laboral, el calendario de vacaciones de los usuarios, el uso de delegación de trabajo y reasignaciones.

#### 11.2.2. Análisis

Habitualmente a la hora de realizar un análisis de los requerimientos clásicos, se utilizan los casos de uso como forma de representación para describir las necesidades del sistema. Por ello, pese a que un caso de uso habitualmente se refiere a una sola funcionalidad, en este caso puede convertirse en un subproceso o en una tarea en función del contenido del mismo. En la figura [11.3](#) podemos ver los casos de uso de este componente organizados en las distintas fases del estudio.

En este apartado únicamente se describen los casos de uso identificados y su conversión a una tarea o subproceso. En la web del proyecto en la sección relativa a los requisitos pueden encontrarse los documentos de cada uno de ellos y el modelo de casos de uso.

##### 11.2.2.1. Caso de uso: Crear estudio

**Iniciador del caso:** Administrador

**Precondiciones:** Ninguna.

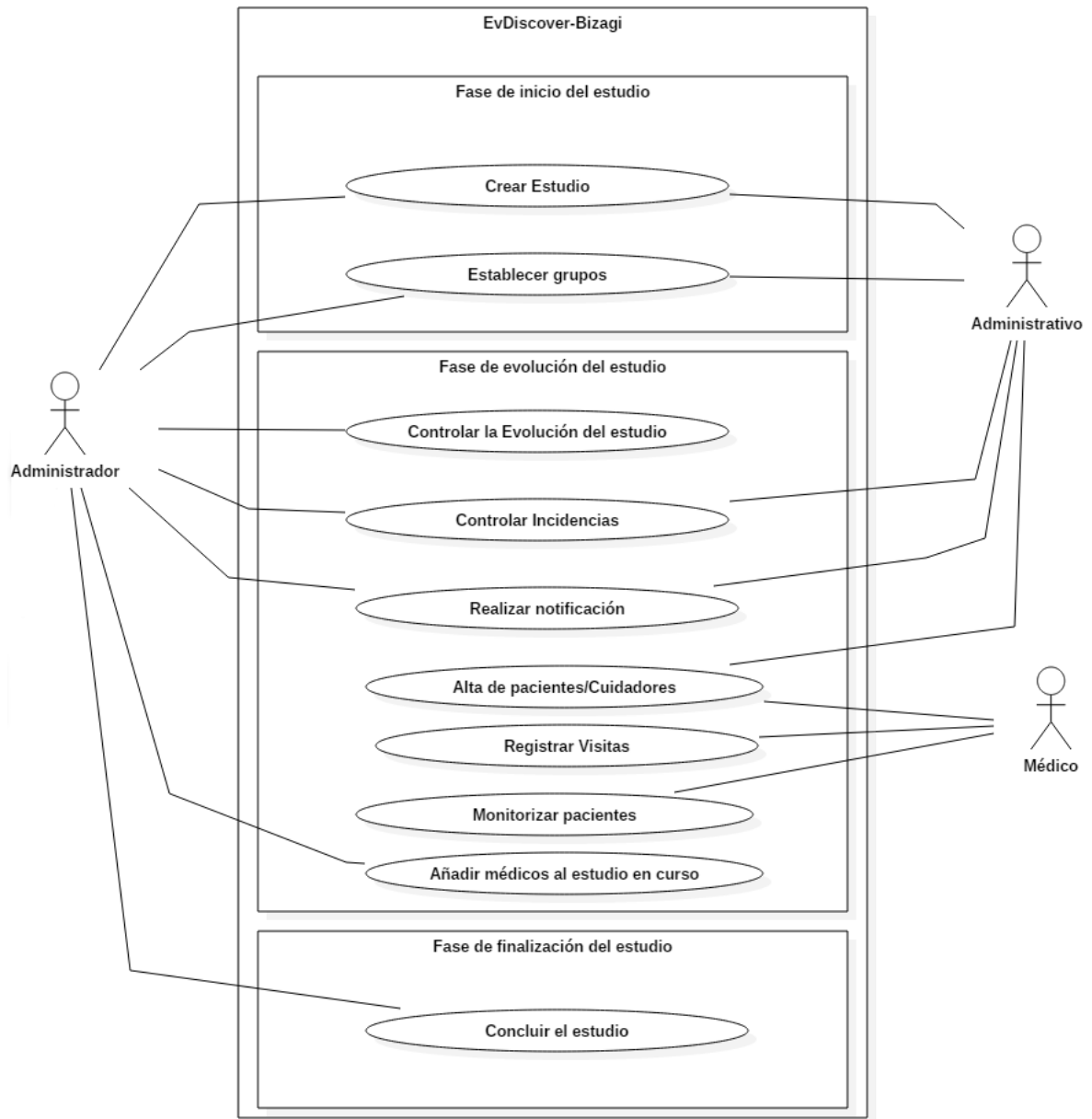
**Post condiciones:** Se ha creado un estudio y se puede pasar a la formación de grupos.

**Actores:** Administrador y Administrativo

#### **Descripción**

Se diseña el estudio clínico a llevar a cabo definiendo valores tales como el nombre, número de médicos por grupo, documentación, descripciones del mismo y formularios a utilizar. Posteriormente se envía el documento de solicitud de permisos y se registra la respuesta de la autoridad pertinente.

Este caso de uso se transforma en un subproceso que está descrito en el apartado [11.2.4.2](#).



**Figura 11.3:** Diagrama de casos de uso de EvDiscover-Bizagi.

#### 11.2.2.2. Caso de uso: Establecer grupos

**Iniciador del caso:** Caso de uso: Crear estudio

**Pre condiciones:** Debe existir un estudio creado previamente con los permisos obtenidos.

**Post condiciones:** Los médicos participantes en el estudio están registrados y listos para realizar el estudio, se habrán establecido las fechas de los eventos formativos y las fechas



de inicio y fin del estudio.

**Actores:** Administrador y Administrativo

### **Descripción**

En este caso de uso se comprenden los pasos a llevar a cabo para el establecimiento de los grupos: registro de candidatos, solicitud de participación, registro de respuestas y formación de los grupos. Además, se establecen las fechas de formación y las de inicio y fin del estudio.

Este caso de uso se transforma en un subproceso que está descrito en el apartado [11.2.4.3](#).

#### 11.2.2.3. Caso de uso: Controlar Evolución del estudio

**Iniciador del caso:** Administrador

**Pre condiciones:** Debe existir un estudio en curso.

**Post condiciones:** Tener el estudio controlado.

**Actores:** Administrador

### **Descripción**

El administrador podrá controlar el estudio para asegurar que la evolución del mismo es correcta. Esto significa poder ver el estado del proceso en todo momento y analizar las tareas realizadas por los distintos participantes.

Este caso de uso viene incluido en el *BPMS*, esto es debido a que el propio software generado incluye por defecto facilidades a la hora de visualizar el estado de los procesos y asegurar el rigor de los mismos.

#### 11.2.2.4. Caso de uso: Alta de pacientes/cuidadores

**Iniciador del caso:** Médico

**Pre condiciones:** El médico está registrado en el sistema, asignado a uno de los grupos del estudio, encontrándose éste en fase de evolución.

**Post condiciones:** El médico ya ha completado el registro de los pacientes que monitorizará a lo largo del estudio y ya puede comenzar a registrar visitas.

**Actores:** Médico

**Descripción**

El médico da de alta a los pacientes que va a monitorizar. En los casos donde el paciente no pueda representarse a sí mismo se deberá añadir un cuidador al paciente. Posteriormente las fichas deberán ser validadas por el administrativo y se asignara un usuario a cada uno de los pacientes/cuidadores.

Este caso de uso se traslada a la primera fase del subproceso que aparece en la sección [11.2.4.7](#).

**11.2.2.5. Caso de uso: Registrar visita****Iniciador del caso: Médico**

**Pre condiciones:** Tanto el paciente como el médico están registrados en el sistema, así mismo el médico se encuentra en la fase de registro de visitas.

**Post condiciones:** Se habrá registrado en el sistema la visita de un paciente en una fecha con los datos introducidos.

**Actores:** Médico

**Descripción**

El médico selecciona un paciente y crea un registro de visita rellenando el diagnóstico, el tratamiento y las observaciones.

Este caso de uso se traslada a la tarea de registro de visitas que puede verse en el subproceso del apartado [11.2.4.7](#).

**11.2.2.6. Caso de uso: Monitorizar pacientes****Iniciador del caso: Médico**

**Pre condiciones:** El médico se encuentra en la fase de registro de visitas.

**Post condiciones:** El médico habrá visualizado las fichas de sus pacientes.

**Actores:** Médico

**Descripción**

El médico puede visualizar los distintos pacientes a su cargo y visualizar el registro de visitas.

Este caso de uso se traslada a la tarea de monitorización de pacientes que puede verse en el subproceso del apartado [11.2.4.7](#).

#### 11.2.2.7. Caso de uso: Controlar incidencias

**Iniciador del caso:** Administrador o administrativo

**Pre condiciones:** Debe existir un estudio en curso en la fase de evolución.

**Post condiciones:** Se ha registrado una nueva incidencia o bien se ha solucionado una incidencia anterior.

**Actores:** Administrador o administrativo

#### **Descripción**

El actor de este caso de uso registra podrá controlar las incidencias ocurridas en el estudio, esto puede suponer el añadir una nueva incidencia o registrar la solución de algunas de las incidencias que no estén cerradas.

Este caso de uso se traslada a la tarea de control de incidencias que puede verse en el subproceso del apartado [11.2.4.5](#).

#### 11.2.2.8. Caso de uso: Realizar notificación

**Iniciador del caso:** Administrador o administrativo

**Pre condiciones:** Debe existir un estudio en curso en la fase de evolución.

**Post condiciones:** Se ha registrado una nueva incidencia o bien se ha solucionado una incidencia anterior.

**Actores:** Administrador o administrativo

#### **Descripción**

Se envía un correo electrónico a los receptores seleccionados con un asunto y un mensaje definido por el usuario.

Este caso de uso se traslada a la tarea de realizar notificación que puede verse en el subproceso del apartado [11.2.4.5](#).

#### 11.2.2.9. Caso de uso: Añadir médicos al estudio en curso

**Iniciador del caso:** Administrador

**Pre condiciones:** Debe existir un estudio en curso en la fase de evolución.

**Post condiciones:** Se ha registrado una nueva incidencia o bien se ha solucionado una incidencia anterior.

**Actores:** Administrador

##### **Descripción**

Permite al administrador añadir nuevos médicos a cualquiera de los grupos del estudio ya sea desde la lista de reserva o añadiendo unos nuevos. Una vez asignado el médico a un grupo este podrá comenzar a usar el sistema.

Este caso de uso se traslada a la tarea de añadir médicos que puede verse en la fase de evolución del proceso principal en el apartado [11.2.4.4](#).

#### 11.2.2.10. Caso de uso: Concluir el estudio

**Iniciador del caso:** Administrador

**Pre condiciones:** Debe existir un estudio en curso en la fase de evolución.

**Post condiciones:** El estudio habrá finalizado.

**Actores:** Administrador

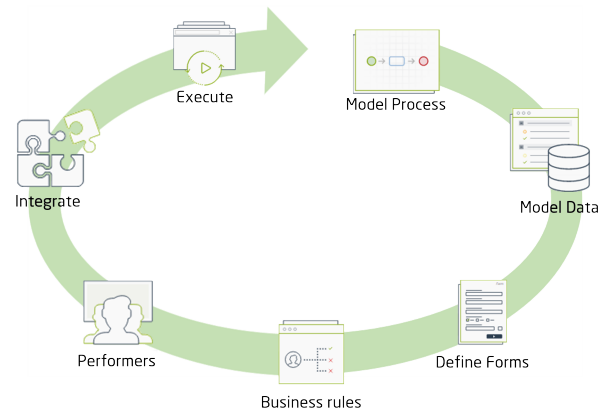
##### **Descripción**

Permite al administrador dar por finalizado el estudio en curso y obtener un resumen de los participantes y las incidencias ocurridas durante el mismo.

Este caso de uso se traslada a la tarea de finalización del estudio que puede verse en la fase de evolución del proceso principal en el apartado [11.2.4.4](#), y la tarea de resumen del estudio del apartado [11.2.4.8](#).

### 11.2.3. Diseño e implementación

A la hora de realizar el diseño e implementación de una aplicación basada en Bizagi, el propio software nos da los pasos a seguir para un correcto desarrollo de los procesos.



**Figura 11.4:** Asistente de procesos de Bizagi Studio.

En los apartados siguientes se describen los pasos a seguir para un correcto diseño del proceso.

#### 11.2.3.1. Modelado de procesos

En este punto se definen los flujos que deberán seguir nuestros procesos mediante el uso de *BPMN*. Esto se realiza mediante el componente *Modeler* que está incluido dentro de la suite *BPM*.

El diseño de los procesos se realiza mediante una interfaz sencilla, permitiéndonos documentar también las partes del proceso que consideremos oportunas para aclarar lo que hace cada una de las tareas.

En la sección [11.2.4](#), dedicada a describir los procesos de esta aplicación, podemos ver cómo quedaría un proceso diseñado.

#### 11.2.3.2. Modelado de datos

A la hora de modelar los datos en Bizagi se trabaja con entidades que contienen atributos o valores donde se almacenan los datos utilizados durante el proceso. Existen distintos tipos de entidades:

- **Maestras:** Estas entidades se muestran en color azul, y es donde Bizagi almacena la información de los procesos. Su información va actualizándose a medida que va avanzando el proceso.

- **Paramétricas:** Estas entidades se muestran en color verde, y son utilizadas para almacenar unos valores predefinidos, como pueden ser, por ejemplo, los tipos de incidencias. Estas pueden ser administradas, bien durante el desarrollo, o bien en tiempo de ejecución, permitiendo por ejemplo al administrador añadir nuevos tipos de incidencia. En cualquier caso, los valores son modificados desde el panel de administración.
- **Stakeholder:** Estas entidades son de color morado y se utilizan para representar a los distintos interesados en el proceso. Siempre incluyen un usuario del sistema y únicamente se utilizan para mejorar la experiencia de usuario.
- **Sistema:** Estas entidades están marcadas con gris y un icono que simboliza un engranaje. Este tipo de entidad pertenece al modelo interno de Bizagi, y contienen información interna, como pueden ser los usuarios.
- **Aplicación:** La entidad de aplicación centraliza la información de cada proceso. Esta entidad se genera automáticamente cuando creamos la aplicación.
- **Proceso:** La entidad de proceso es la entidad principal desde la que el usuario final accede a las distintas entidades del modelo de datos.

En la figura [11.5](#) podemos ver el modelo de datos final elaborado para este proceso.

#### 11.2.3.3. Definición de formularios

En esta parte del asistente tendremos que definir un formulario por cada tarea de tipo manual que tengamos. Los campos a incluir en estos están restringidos a las variables a utilizar en el proceso, es decir a los atributos de las entidades.

Es muy sencillo añadir nuevos formularios ya que basta con arrastrar los atributos que deseamos utilizar en ese punto al formulario concreto. Todo se desarrolla mediante una interfaz similar a la que vemos en la figura [11.6](#). No obstante, pese a parecer sencillo, podemos definir validaciones y restricciones a todos los campos que utilicemos.

Se han adjuntado las interfaces diseñadas para cada una de las tareas en el modelo de proceso que se encuentra en la web del proyecto.

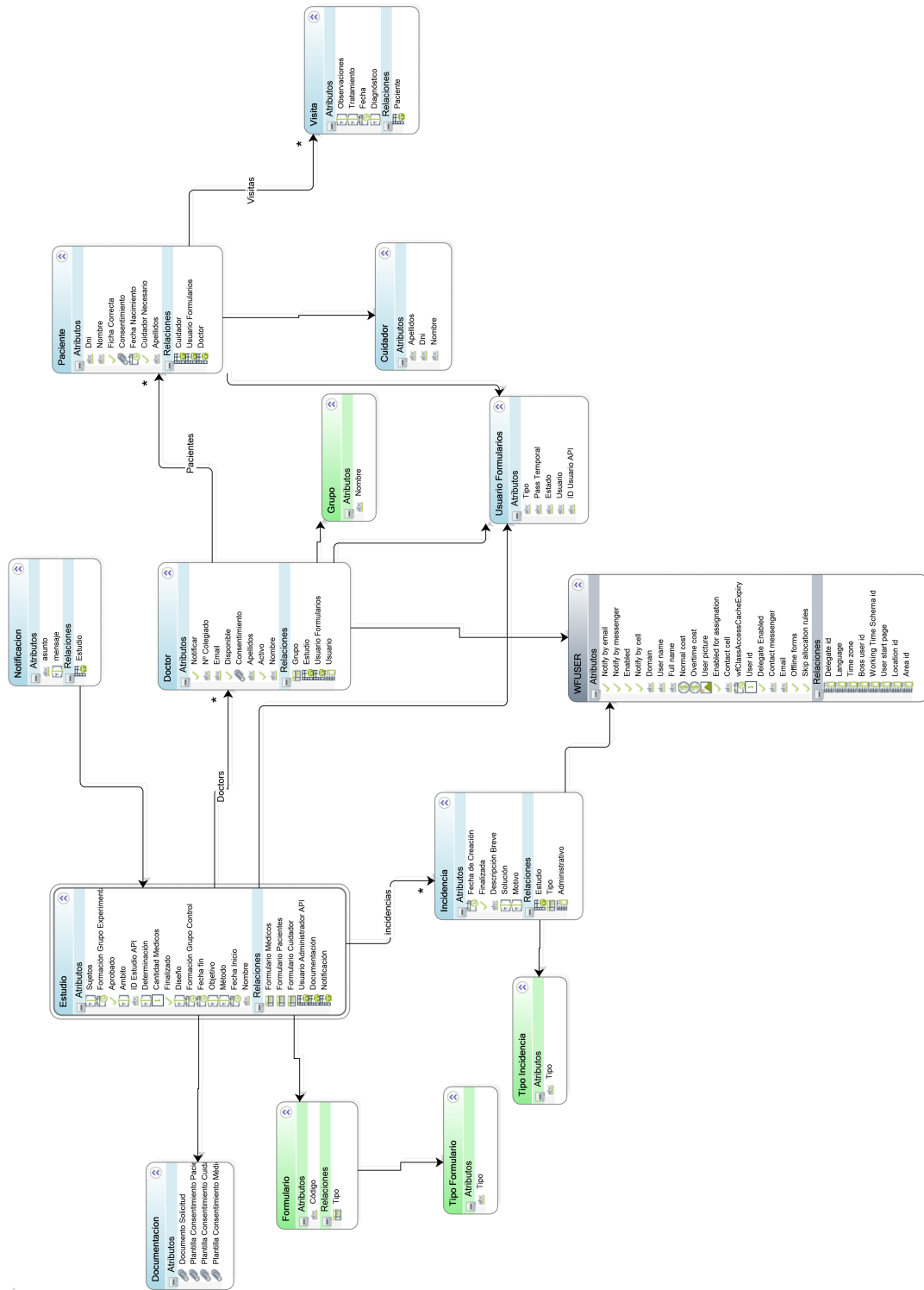
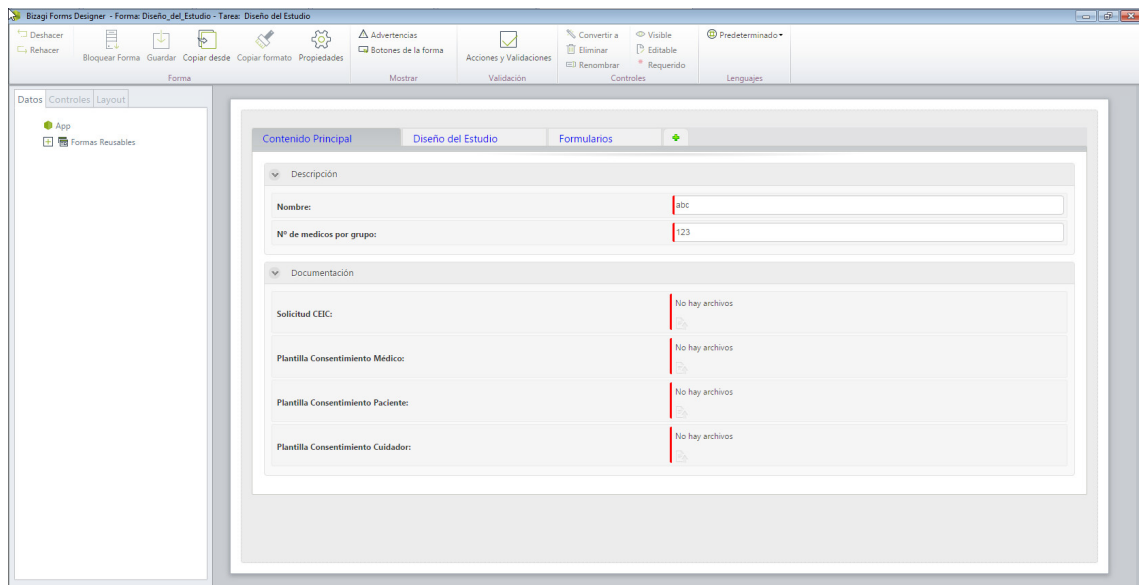


Figura 11.5: Modelo de datos del proceso.



**Figura 11.6:** Editor de formularios de Bizagi Studio.

#### 11.2.3.4. Reglas de negocio

En esta sección del asistente se definen las reglas de negocio que dan forma al sistema. Con esto podemos establecer validaciones a nivel de actividades o decidir el flujo del proceso en base a datos. Esta parte del asistente está dividida en dos secciones:

- **Definir expresiones:** Nos permite controlar la secuencia del flujo decidiendo por qué camino debe seguir el proceso en base a condiciones específicas. Lo utilizaremos también a la hora de realizar filtros en los formularios.
- **Acciones de la actividad (eventos):** Ejecuta los procedimientos necesarios al realizar una tarea como validaciones o acciones a realizar. Por cada tarea se nos permite incluir acciones al entrar y salir en la tarea, así como en las ocasiones que el estado de esta es guardado. Es aquí donde podremos definir acciones como mandar un correo electrónico o ejecutar un script.

A modo de ejemplo en la figura 11.7 podemos ver el código utilizado para filtrar los usuarios que podemos asignar a los médicos.



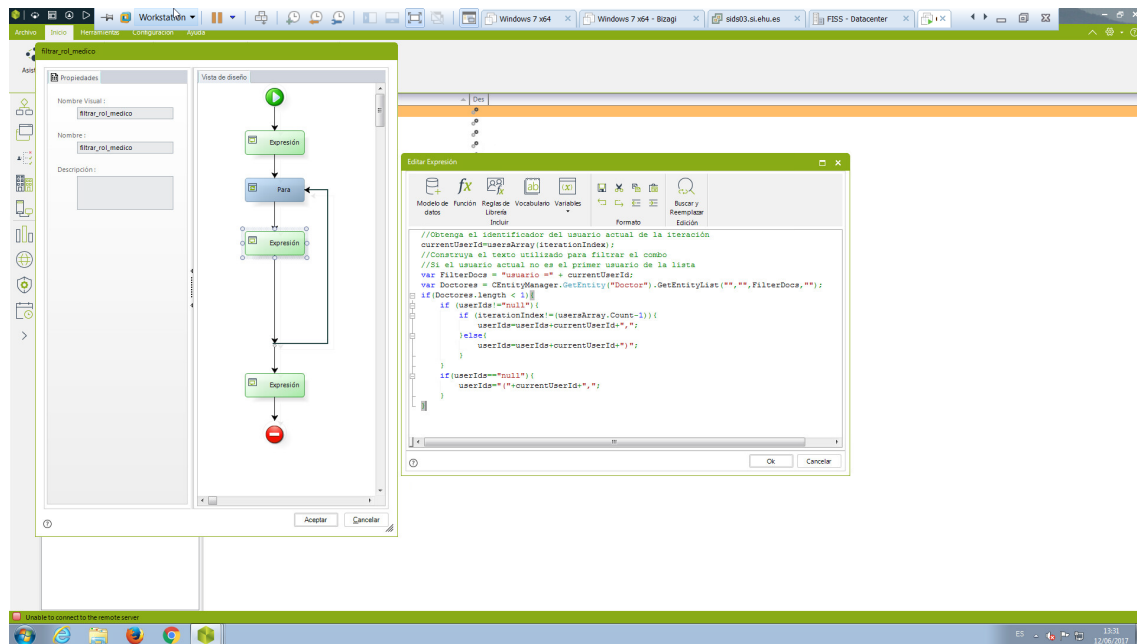


Figura 11.7: Formulario de definición de una regla en Bizagi.

#### 11.2.3.5. Participantes

Es en esta parte del asistente donde definiremos quién realiza que tarea y como se realiza dicha asignación. En Bizagi tenemos cuatro tipos de asignaciones:

- **Por Carga:** La tarea se asigna al usuario con menor carga de trabajo que cumpla los requisitos establecidos. Sin embargo, el sistema verifica primero si alguien en el grupo del usuario que ya ha trabajado en el caso. Si es así, se asignará a él / ella independientemente de la carga de trabajo del usuario en comparación con el resto del grupo.
- **Todos:** Se asigna la tarea a todos los usuarios que cumplan los requisitos, pudiendo coger la tarea cualquiera de ellos en tanto otro no la tenga asignada. Una vez se pinche sobre ella, ésta desaparecerá del buzón del resto de implicados.
- **Secuencial:** cada tarea se asigna equitativamente y de forma secuencial a los usuarios que cumplan los criterios de asignación, independientemente de su carga de trabajo.
- **Primer disponible:** Con este método, las tareas se asignan al primer usuario disponible en base a la zona horaria.

A la hora de especificar qué usuarios pueden tener una tarea asignada, se realiza mediante condiciones. Entre las posibilidades de dichas condiciones, tenemos distintos tipos de filtrado en función de las variables de un usuario, como son: Id de Usuario, área, lugar, rol, habilidad, cargo. No obstante, en este proyecto las más utilizadas han sido el id y el rol.

#### 11.2.3.6. Integrar

En este punto es donde se definen las interfaces de integración con otros sistemas. Más adelante se explicará en que consiste en la sección 11.4 dedicada a la colaboración entre los componentes.

### 11.2.4. Proceso Implementado

Al ser la gestión de un estudio clínico un proceso en sí, se ha decidido que sea un solo proceso el encargado de controlar el desarrollo del mismo. En la figura 11.8 podemos ver el proceso completo. El proceso global está dividido en tres fases distintas que se ejecutan siguiendo un orden específico.

1. Inicio del estudio.
2. Evolución del estudio.
3. Finalización del estudio.

A continuación, se explican en detalle cada uno de las fases, así como su relación con los casos de uso extraídos en la parte de análisis.

#### 11.2.4.1. Fase de inicio del estudio

##### **Descripción**

En esta fase del estudio es necesario diseñarlo y establecer los grupos que posteriormente pasarán a formar parte del estudio. En la figura 11.9 podemos ver una las actividades y subprocesos de esta fase.

##### **Tareas**

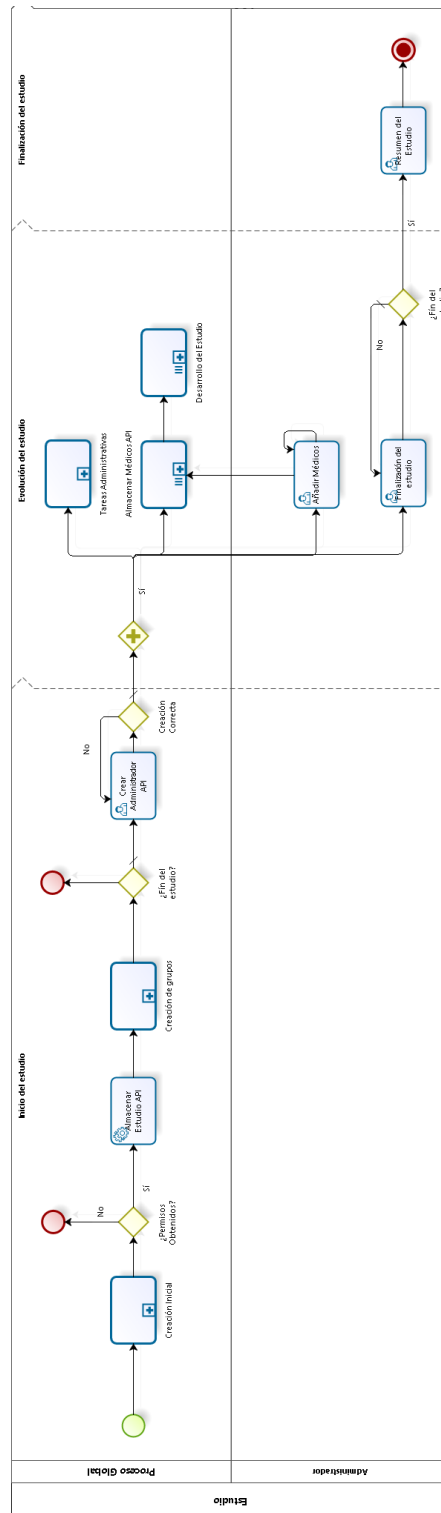
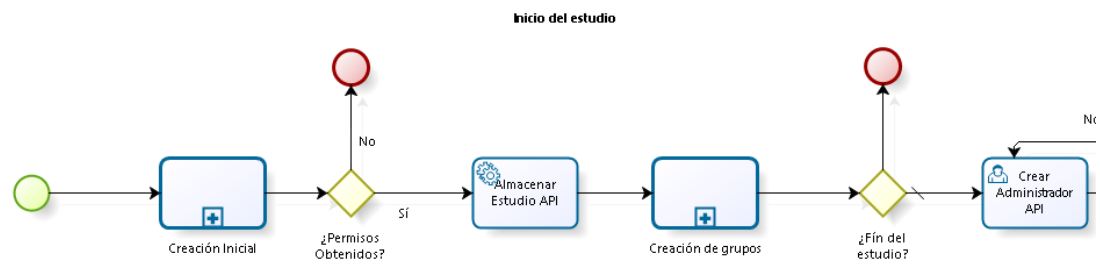


Figura 11.8: Proceso general para el control del estudio.



**Figura 11.9:** Fase de inicio del proceso.

- **Creación inicial:** Esta tarea de tipo subproceso esta descrita en el apartado [11.2.4.2](#).
- **Almacenar Estudio:** Esta tarea de tipo servicio se comunica con el componente EvDiscover-Forms de forma que añade el estudio creado, y guarda el id que le devuelve para posteriores usos. En el diagrama de secuencia de la figura [11.31](#) podemos ver cómo interactúan los dos componentes a través de la API.
- **Creación de grupos:** Esta tarea de tipo subproceso esta descrita en el apartado [11.2.4.3](#).
- **Crear Administrador API:** En esta tarea se define un usuario y una contraseña para el administrador de forma que posteriormente pueda realizarse la extracción de datos. En el diagrama de secuencia de la figura [11.32](#) podemos ver cómo se realiza la comunicación con la API.

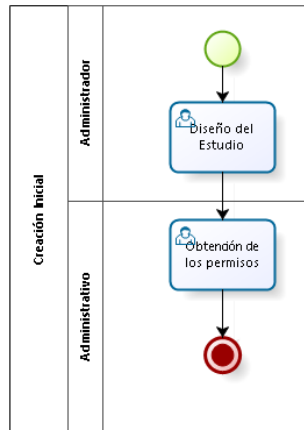
Con esto realizado ya es posible acceder a la web de EvDiscover-Forms para extraer resultados de los formularios. Una vez finalizada esta tarea, el proceso entrara en la siguiente fase del mismo.

#### 11.2.4.2. Subproceso Creación inicial

##### Descripción

Este subproceso que podemos ver en la figura [11.10](#) comprende los pasos necesarios para la creación de un estudio clínico, y es el equivalente al caso de uso *Crear Estudio* identificado durante la captura de requisitos.

##### Tareas



**Figura 11.10:** Subproceso *Creación Inicial*.

- **Diseño del estudio:** En esta tarea el administrador debe realizar el diseño para dar comienzo al estudio incluyendo el tamaño mínimo por grupo y los documentos y cuestionarios científicamente validados, a utilizar entre otras cosas.
- **Obtención de los permisos:** El administrativo debe descargar el documento de solicitud al *CEIC* (Comité Ético de Investigación Clínica), y enviarlo para posteriormente registrar su respuesta. Una vez registrada, el proceso seguirá o terminará dependiendo de si el estudio ha sido aprobado o no.

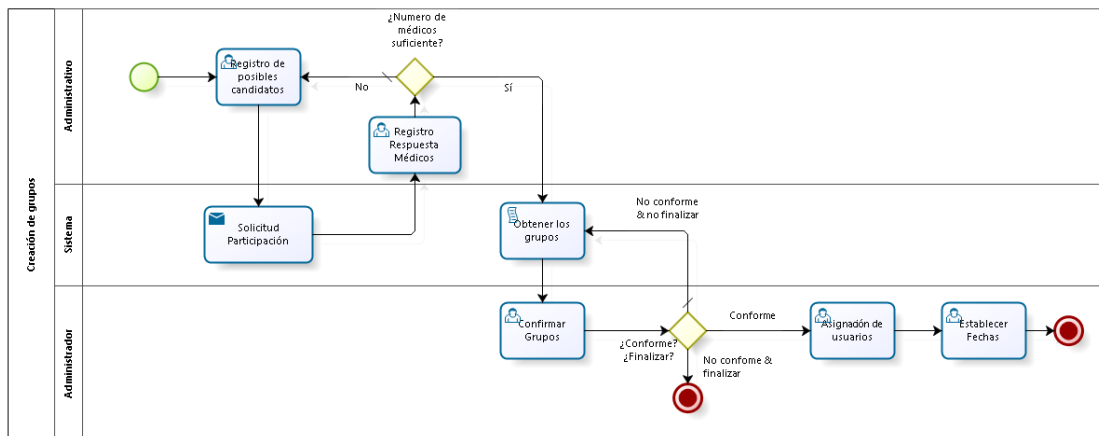
#### 11.2.4.3. Subproceso Creación de grupos

##### Descripción

Este subproceso que podemos ver en la figura 11.11 comprende los pasos necesarios para establecer los grupos de control y experimentación, y es el equivalente al caso de uso *Establecer Grupos* identificado durante la captura de requisitos.

##### Tareas

- **Registro de posibles candidatos:** El administrativo registra los posibles candidatos a participar en el estudio.
- **Solicitud de participación:** Se envía una solicitud de participación a los médicos registrados en la tarea anterior mediante un correo automático con los datos del estudio y un adjunto conteniendo el consentimiento a firmar.

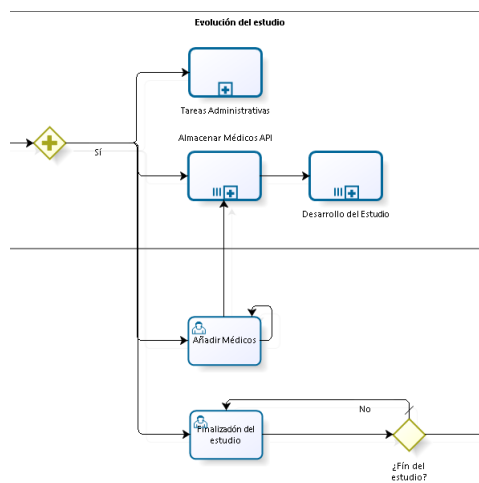


**Figura 11.11:** Subproceso *Creación de grupos*.

- **Registro respuesta médicos:** El administrativo registra las respuestas de los médicos al correo enviado en la tarea anterior. En caso de que confirmen su participación deberá adjuntarse el consentimiento informado de dicho médico. De no haber suficientes médicos se vuelve de nuevo a la tarea de registro de nuevos candidatos.
- **Obtener grupos:** Tarea de tipo script donde el sistema sin necesidad de actuación por parte de usuarios asigna aleatoriamente los participantes a los grupos de control y experimentación en base a el número de médicos por grupo establecido en el diseño del estudio. Los médicos restantes quedan en el grupo de reserva de forma que posteriormente se puedan incorporar al estudio.
- **Confirmar grupos:** El administrador deberá decidir si acepta los grupos formados o no. En caso negativo, tendrá la posibilidad de finalizar el estudio o volver a realizar la obtención de grupos.
- **Asignación de usuarios:** Se asignan los usuarios creados mediante el portal de Bizagi a cada uno de los médicos. El motivo de que sea el administrador el que realiza la asignación es debido a que es el único con permisos de creación de usuarios.
- **Establecer fechas:** El administrador establece la fecha de inicio y fin del estudio, así como la reunión de formación para cada uno de los grupos. Tras establecerlas, se manda automáticamente un correo a cada médico indicando las fechas de inicio y fin, y la reunión de formación a la que debe asistir. Esto último sin decirle en qué grupo ha sido colocado.

#### 11.2.4.4. Fase de evolución

En esta fase del estudio que podemos ver en la figura 11.12 los grupos ya han sido creados, y los médicos ya pueden comenzar a trabajar. En este punto el proceso se ramifica en 4 partes distintas.



**Figura 11.12:** Fase de evolución del proceso.

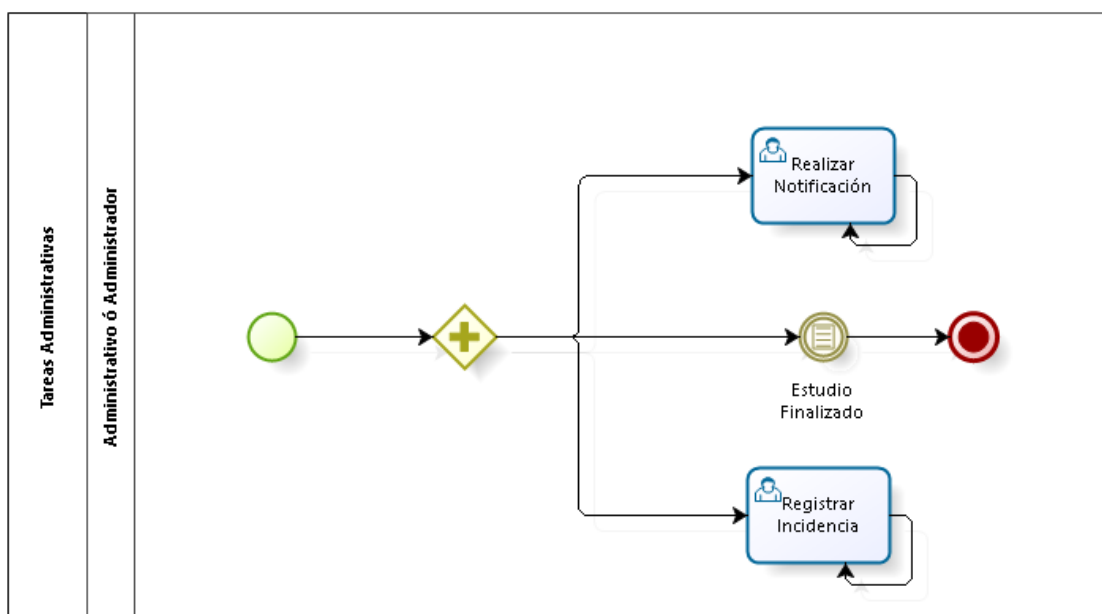
### Tareas

- **Tareas administrativas:** Esta tarea de tipo subproceso esta descrita en el apartado [11.2.4.5](#).
- **Almacenar médicos API:** Esta tarea de tipo subproceso esta descrita en el apartado [11.2.4.6](#).
- **Tareas administrativas:** Esta tarea de tipo subproceso esta descrita en el apartado [11.2.4.7](#).
- **Añadir médicos:** En esta tarea el administrador puede incluir nuevos médicos al estudio si alguno de los médicos no estuviera cumpliendo con su cometido, se dieran bajas o si deseara incluir más simplemente para aumentar el tamaño de la muestra.
- **Finalizar el estudio:** En esta tarea el administrador puede dar el estudio por finalizado. En caso afirmativo el proceso seguiría a la siguiente fase y en caso negativo se volvería a esta tarea.

## 11.2.4.5. Subproceso Tareas administrativas

**Descripción**

Este subproceso activa las tareas administrativas que podrán ser realizadas a lo largo del estudio. En la figura 11.13 podemos ver que existirán dos tipos de tareas que se repetirán continuamente hasta que el estudio sea finalizado. En este supuesto queda claro a qué caso de uso corresponde cada una de las tareas a realizar. Es importante destacar que estas tareas deben ser finalizadas o liberadas mediante la opción que aparece en la web, de forma que otro de los usuarios pueda cogerla.



**Figura 11.13:** Subproceso *Tareas administrativas*

**Tareas**

- **Realizar notificación:** En esta tarea el administrativo o administrador escogerá unos destinatarios y establecer el asunto y el contenido del mensaje. Posteriormente el sistema enviara un mensaje a dichos destinatarios con los parámetros establecidos.
- **Registrar incidencia:** En esta tarea se permite al administrativo o administrador que la coja registrar una incidencia. En esta tarea el usuario podrá visualizar las incidencias registradas y una vez introducidas, darles una solución. Además, se



mostrarán las incidencias en dos apartados. Por un lado, estarán aquellas que no se hayan cerrado, y por el otro un historial con todas las incidencias registradas.

#### 11.2.4.6. Subproceso Almacenar médicos API

##### Descripción

Este subproceso múltiple, que podemos ver en la figura 11.14, se ejecuta tomando como entrada los médicos que han sido asignados al grupo de control o experimentación. Una vez dentro, por cada uno de ellos se realiza el registro del usuario en la aplicación de formularios. Es importante destacar que cada vez que se entra a este subproceso no se ejecuta por cada uno de los médicos que se hayan registrado, sino que también tiene en cuenta que dicha instancia de médico no haya entrado previamente al subproceso.

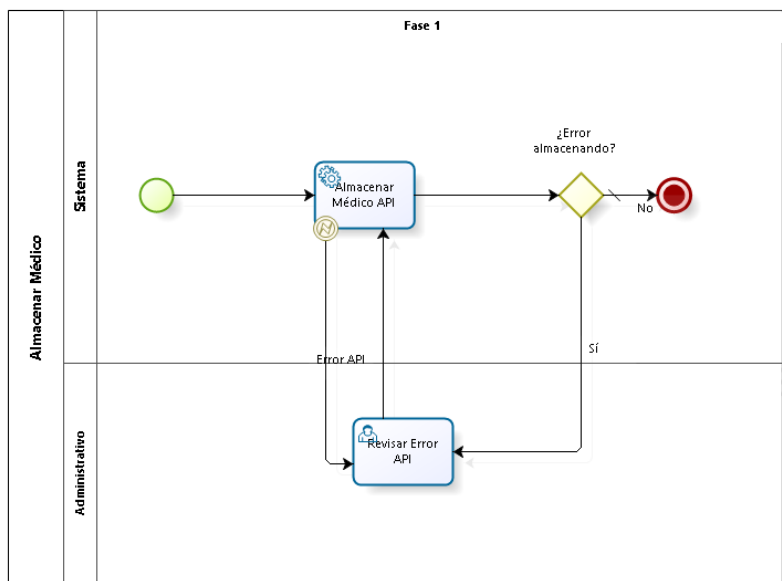


Figura 11.14: Subproceso *Almacenar médicos API*

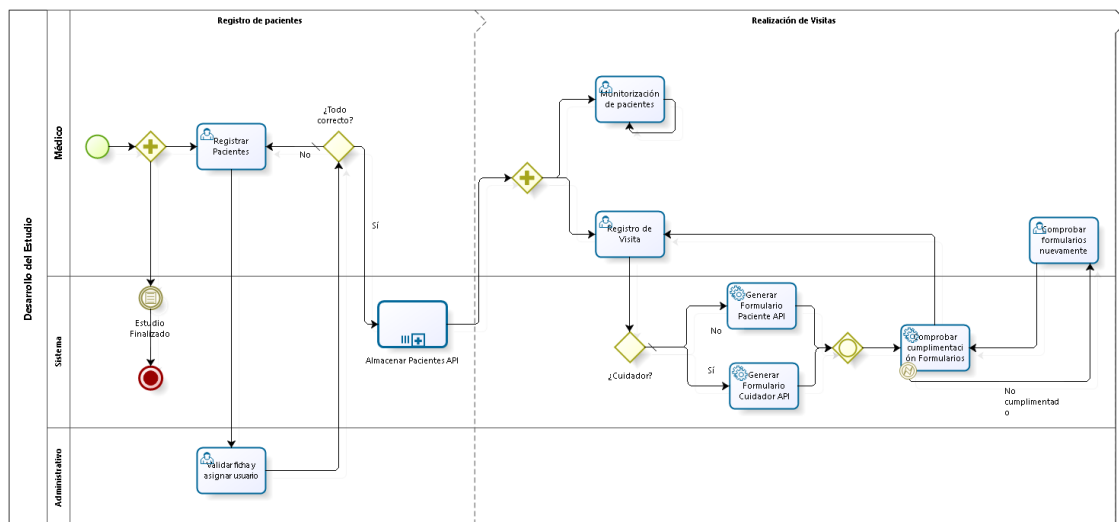
##### Tareas

- **Almacenar médico API:** Esta tarea de servicio se comunica con el componente EvDiscover-Forms y da de alta al usuario en el sistema, el funcionamiento es el mismo que el que podemos ver en el diagrama de secuencia de la figura 11.32.
- **Revisar error API:** Esta tarea solo se realizará cuando exista algún problema con la comunicación entre componentes, o la respuesta de la API nos comunique que existe algún problema relacionado con la existencia del nombre de usuario.

## 11.2.4.7. Subproceso Desarrollo del Estudio

**Descripción**

Este subproceso múltiple, que podemos ver en la figura 11.15, funciona de la misma forma que el que le precede. Dentro de éste es donde cada uno de los médicos comenzara con el registro de pacientes y sus visitas. Este proceso estará continuamente activo hasta que el estudio finalice.



**Figura 11.15:** Subproceso *Desarrollo del estudio*

Este proceso está dividido en dos fases claras. En primer lugar, estará la fase de registro de pacientes. Esta fase solamente finalizará una vez el médico haya registrado al menos un paciente, y el administrativo que compruebe la validez de la ficha y los consentimientos informados y asigne un nombre de usuario para usar en la aplicación de formularios a todos los pacientes.

**Tareas**

- **Registro pacientes:** En esta tarea el médico registra los pacientes y en caso de necesitar un cuidador se también se añaden sus datos. En este punto la decisión de si registrar al cuidador recae sobre el médico. Esta tarea sería el equivalente al caso de uso *Alta de Pacientes/Cuidadores*.
- **Validar ficha y asignar usuario:** En este punto el administrativo que cumplimente esta tarea deberá comprobar que la ficha del paciente es la correcta. Esto supone

revisar el consentimiento informado y si se ha asignado un cuidador al paciente en caso necesario. Además, también deberá asignar un nombre de usuario al paciente para poder usar posteriormente el componente EvDiscover-Forms.

- **Almacenar Pacientes API:** Este subproceso múltiple se ejecuta tomando como entrada los pacientes añadidos por el médico y que hayan sido aprobados por el administrativo. Se ha de destacar que no se accede a este subproceso hasta que todos los pacientes del médico hayan sido validados por el administrativo. El subproceso es similar al realizado para añadir a los médicos y por tanto la comunicación con la API es similar a la vista en el diagrama de secuencia de la figura [11.32](#).

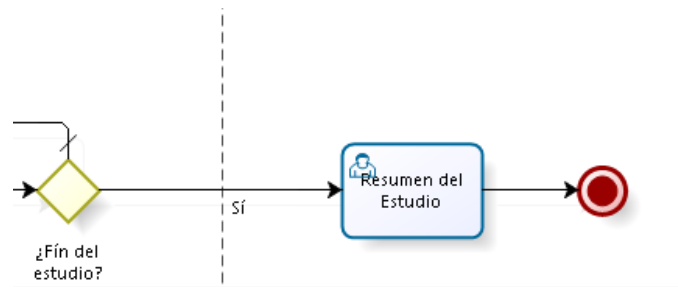
En la segunda fase el médico irá registrando las visitas de los pacientes, y el proceso controlará que se están rellenando los formularios pertinentes.

### Tareas

- **Monitorización de pacientes:** Esta tarea en forma de bucle permitirá al médico visualizar las fichas de los pacientes que tenga a su cargo. En ella podrá visualizar las visitas registradas por cada uno de los pacientes.
- **Registro de visita:** En esta tarea el médico escoge un paciente y rellena los campos requeridos referentes a dicha visita.
- **Generar Formulario:** Aunque son dos tareas de tipo servicio distintas, el funcionamiento es similar en ambas. En estas tareas el sistema se comunica con la aplicación de cuestionarios para generar una solicitud de cumplimentación, tanto al médico como al paciente/cuidador. El diagrama de secuencia relativo a esta comunicación puede verse en la figura [11.33](#).
- **Comprobar Cumplimentación:** Esta tarea de tipo servicio se comunica con el componente de cuestionarios para comprobar que no existen cuestionarios pendientes relativos al paciente. El diagrama de secuencia relativo a esta comunicación puede verse en la figura [11.34](#).
- **Comprobar Formularios Nuevamente:** Esta tarea será realizada por el médico y sirve para volver a realizar la tarea de tipo servicio de comprobar cumplimentación.

#### 11.2.4.8. Fase de finalización del estudio

En esta fase el estudio ya ha sido finalizado, y por tanto sólo queda una tarea a realizar en el mismo. Esta tarea consiste en mostrar un resumen del estudio al administrador pudiendo extraer los datos de los distintos participantes del estudio, así como el registro de incidencias. Tras pasar la tarea se dará por finalizado el estudio.



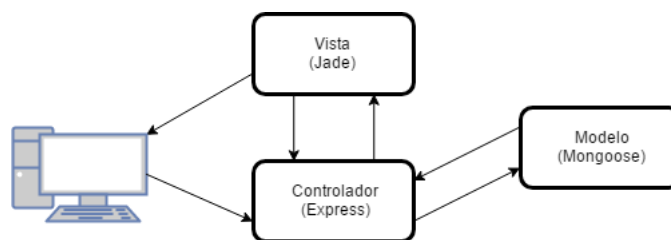
**Figura 11.16:** Fase de finalización del estudio

Con esto el proceso de gestión del estudio quedaría descrito. No obstante, si se desean ver todos los procesos se dispone del diseño publicado en la página web del proyecto.

### 11.3. EvDiscover-Forms

#### 11.3.1. Arquitectura

Con el objetivo de implementar el sistema utilizando el patrón de arquitectura Modelo Vista Controlador (*MVC*), se han utilizado varios componentes que se explicarán más tarde orquestados tal y como se refleja en la figura 11.17.



**Figura 11.17:** Patrón *MVC* con *Express*, *Jade* y *Mongoose*.

## 11.3.2. Tecnologías utilizadas

### 11.3.2.1. Node.js

*Node.js* es un entorno de ejecución para *JavaScript* construido con el motor de *JavaScript* V8 de *Chrome*. Usa un modelo de operaciones entrada/salida sin bloqueo y orientado a eventos, gracias a lo cual es mucho más rápido y eficiente.

Entre sus mayores ventajas, destaca el enorme ecosistema de librerías de código abierto, que nos permite añadir nuevas funcionalidades a nuestros sistemas sin necesidad de reinventar la rueda, ganando así una mayor velocidad de desarrollo.

Para iniciar el servidor solamente necesitamos configurar un script que será el encargado de ponerlo en marcha.

### 11.3.2.2. MongoDB

Esta base de datos es de tipo *NoSQL*, y una de las diferencias más importantes con respecto a las clásicas bases de datos relacionales, es que no es necesario seguir un esquema predeterminado. Ello supone que podemos almacenar registros de distinto tipo en una misma colección. Dichos datos quedan almacenados en formato *BSON* (*Binary JSON*), el cual es muy similar a *JSON*. En la figura 11.18 podemos ver el tipo de estructura.

```
{
  "_id" : ObjectId("594d2030e3757b193874d5b2"),
  "estudio" : ObjectId("594d1dd2ad868b13a0d9182a"),
  "grupo" : "Control",
  "password" : "$2a$08$MRezFB6.QQkhbcCReQvpx.K5SSWNeaDKeI7aWJXGHcbonjwIVkOaK",
  "role" : "Médico",
  "user" : "medico5",
  "formularios" : [],
  "__v" : 0
}
```

**Figura 11.18:** Ejemplo de documento almacenado por mongoDB

Explicado de forma sencilla, estas bases de datos contienen colecciones donde se agrupan documentos como el que ya hemos visto anteriormente. Cada documento puede tener valores distintos, y las colecciones no están restringidas a contener un solo tipo de documento. A la hora de diseñar la parte de cumplimentación de cuestionarios, nos viene bien de cara a almacenar distintos tipos de cuestionarios clínicos en una misma colección. Esto frente al clásico modelo de base de datos relacional supone un enorme avance, pues no necesitaremos especificar una vez tras otra nuevas tablas.

#### 11.3.2.3. Express

Este framework web basado en node.js provee de unas funciones definidas que nos permiten especificar las rutas de nuestra aplicación web, así como el controlador encargado de llevar el control de las mismas.

Gracias a esta capa de abstracción, es muy sencillo realizar aplicaciones web y *APIs* de manera rápida y eficaz, ya que nos permite definir las direcciones de la aplicación en base a verbos *HTTP* (*Hyper Text Transfer Protocol*), lo cual facilita la creación de un servicios web con arquitectura REST, que se basan en el uso de dichos verbos.

#### 11.3.2.4. Mongoose

Gracias a la librería *Mongoose* es posible conectar con una base de datos *MongoDB*, utilizando esquemas para asegurar que el contenido de los documentos que van a ser almacenados corresponde con lo que se desee almacenar. También es una forma de recuperar documentos que cumplan un esquema predeterminado de la base de datos.

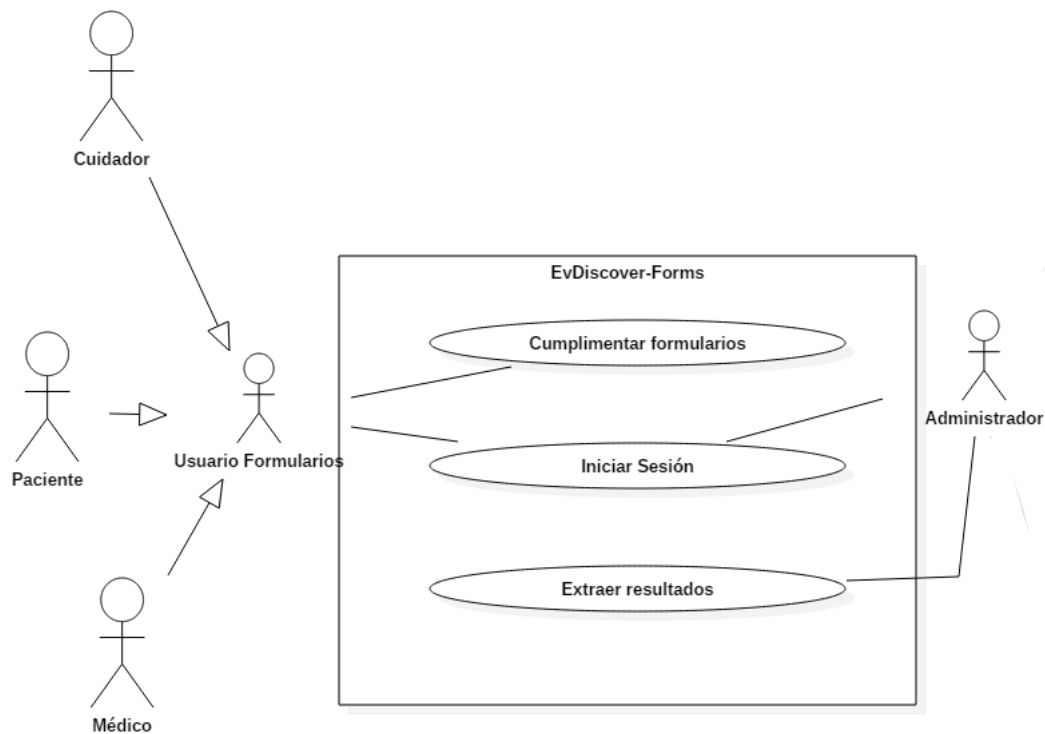
Esta librería nos permite una capa de abstracción para no acceder directamente a la base de datos. Además, el hecho de que permita definir esquemas, no resta la flexibilidad que nos da utilizar un sistema *NoSQL*, sino que nos da aún más ventajas.

#### 11.3.2.5. Interfaces

Para el desarrollo de las interfaces de este componente se ha utilizado el motor de plantillas de *Jade*, el cual cambio su nombre a *Pug* en la versión 2. Gracias a él podremos convertir las plantillas definidas en código *HTML* para el cliente. Además, con el objetivo de ofrecer una interfaz agradable, se ha utilizado el framework *CSS Bootstrap*.

### 11.3.3. Análisis

Los casos de uso relativos a este componente pueden verse en el diagrama de casos de uso de la figura 11.19. Cuando en los casos de uso de este componente se hace referencia a cumplimentar un formulario, a de entenderse en todo momento que se trata de los cuestionarios clínicos.



**Figura 11.19:** Casos de uso de EvDiscover-Forms.

#### 11.3.3.1. Caso de uso: Iniciar sesión

##### **Descripción**

Caso de uso que permite al usuario acceder al sistema.

##### **Actores**

Usuario formularios y Administrador

##### **Precondiciones**

El usuario está en la página de acceso a la aplicación.

##### **Post-Condiciones**

El usuario habrá iniciado sesión en el sistema.

##### **Flujo básico de eventos**

1. El usuario introduce su usuario y contraseña

2. En caso de que las credenciales sean válidas el usuario accede a la página de entrada.

#### 11.3.3.2. Caso de uso: Cumplimentar formularios

##### **Descripción**

Con el objetivo de obtener conclusiones estadísticas, se almacenarán los formularios rellenos por los pacientes y médicos, de manera que más tarde, puedan extraerse conclusiones formales de dichos datos.

##### **Actores**

Usuario formularios

##### **Precondiciones**

El usuario tendrá que haber iniciado sesión y tener un formulario a rellenar pendiente.

##### **Post-Condiciones**

Se tendrá registrado en la base de datos los resultados del formulario realizado por el paciente.

##### **Flujo básico de eventos**

1. El usuario accede al sistema.
2. El usuario cumplimenta el/los formularios pendientes.
3. Se almacenan las respuestas dadas por el usuario.
4. Se informa de que se ha cumplimentado correctamente el formulario

#### 11.3.3.3. Caso de uso: Extraer resultados

##### **Descripción**

Con el objetivo de obtener conclusiones estadísticas se extraerán los resultados de los cuestionarios utilizados durante el estudio en un formato adecuado para su análisis.

##### **Actores**

Administrador



### Precondiciones

El usuario tendrá que haber iniciado sesión y deben existir formularios con resultados.

### Post-Condiciones

Se habrá obtenido un fichero *CSV* con los datos de las respuestas al formulario.

### Flujo básico de eventos

1. El usuario accede al sistema.
2. Se selecciona el formulario del cual queremos obtener los datos.
3. Se obtiene el fichero con los datos.

## 11.3.4. Diseño

### 11.3.4.1. Modelo de datos

Dado que estamos usando una base de datos documental, se ha decidido representar el modelo de datos mediante un diagrama de clases que representa las relaciones entre los distintos documentos a almacenar. Para asegurarnos de que los documentos almacenados cumplen con lo especificado, se hace uso de modelos de *mongoose*. En la figura 11.20 podemos ver la estructura que se ha creado.

Existen tres colecciones:

- **Estudios:** Se almacenarán documento del tipo estudio.
- **Users:** Se almacenan los usuarios a utilizar en el sistema. Dentro de cada uno de estos documentos podemos encontrar peticiones de formulario que representan los cuestionarios que tiene pendientes cada usuario.
- **Formularios:** Se almacenan los datos de las respuestas de los distintos cuestionarios.

En el apartado 11.3.5.4 se explicará cómo se definen nuevos cuestionarios para el sistema, pues se ha diseñado un sistema que permite incluir nuevos formularios con cierta facilidad.

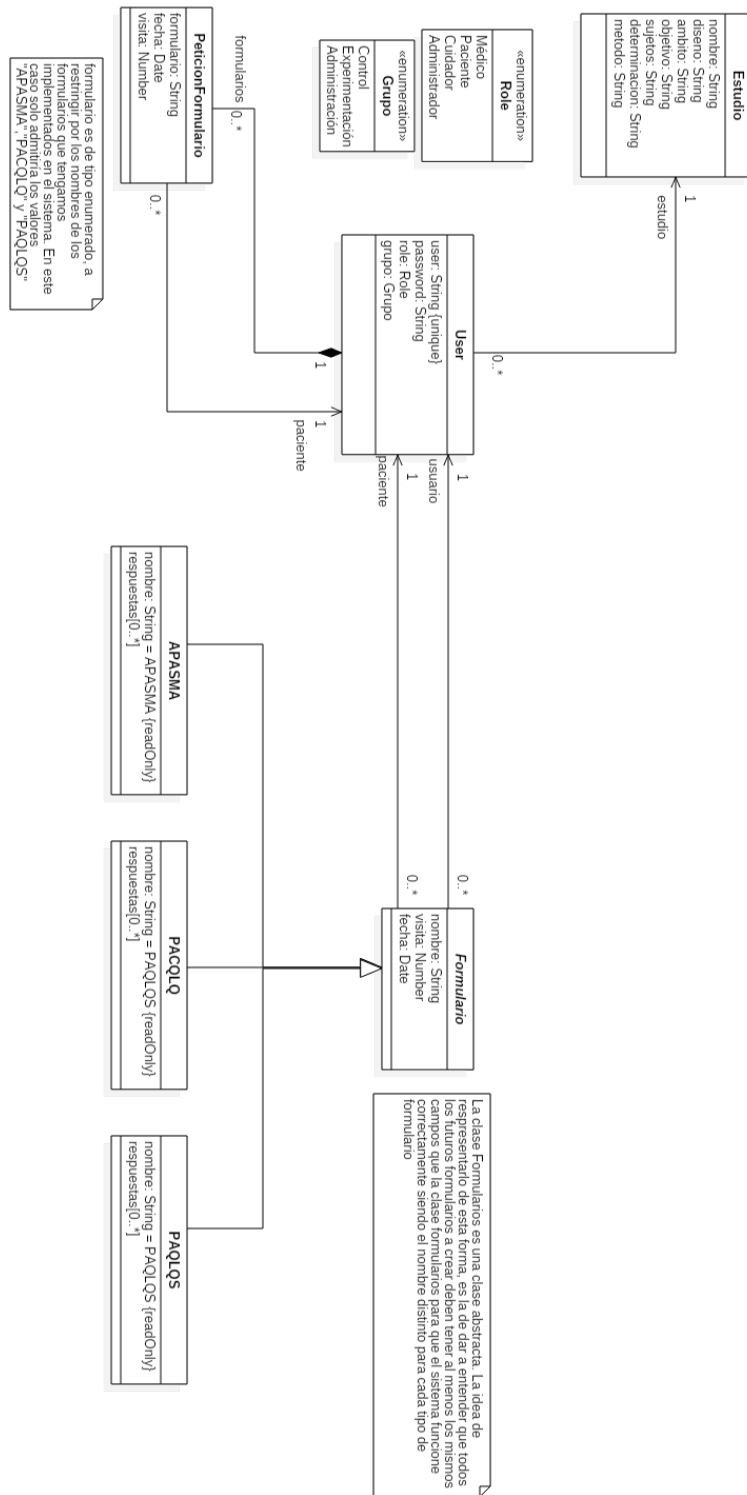
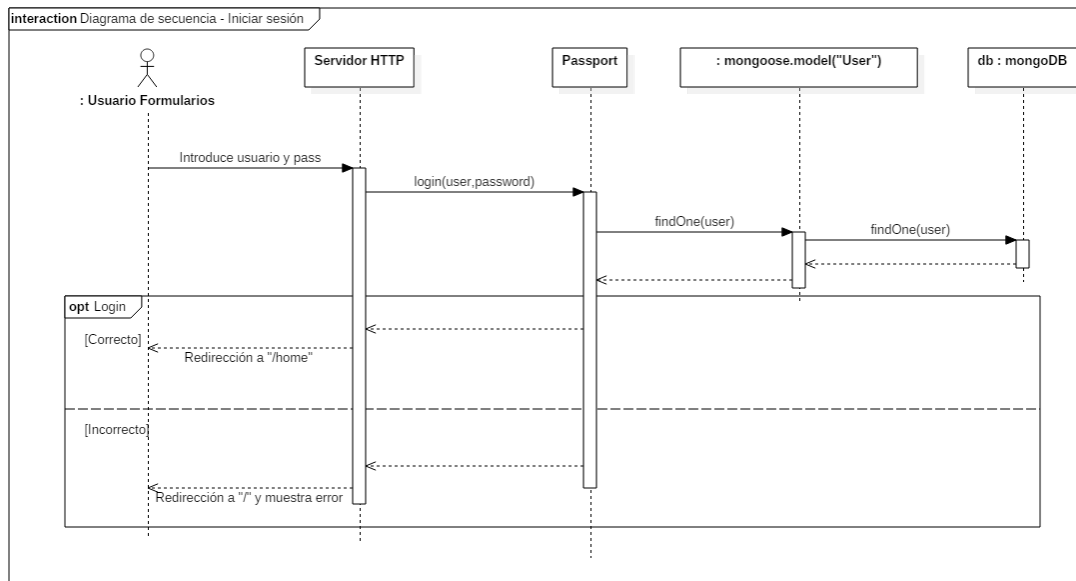


Figura 11.20: Modelo de datos de EvDiscover-Forms.

### 11.3.4.2. Diseño del caso de uso: Iniciar sesión

En el diagrama de la figura 11.21 podemos ver la realización del caso de uso relativo a iniciar sesión en el sistema. En este caso el diagrama de secuencia sería igual para todos los actores del sistema.



**Figura 11.21:** Diagrama de secuencia de iniciar sesión.

La interfaz diseñada para este caso de uso puede verse en la figura puede verse en la figura 11.22.

**Figura 11.22:** Interfaz del caso de uso iniciar sesión.

11.3.4.3. Diseño del caso de uso: Cumplimentar formularios

En el diagrama de la figura 11.23 podemos ver la realización del caso de uso relativo a la cumplimentación de un cuestionario. Es el mismo tipo de secuencia en todos los casos, pues los cuestionarios son indiferentes al usuario que los rellena. No obstante, será necesario que exista una solicitud de los mismos.

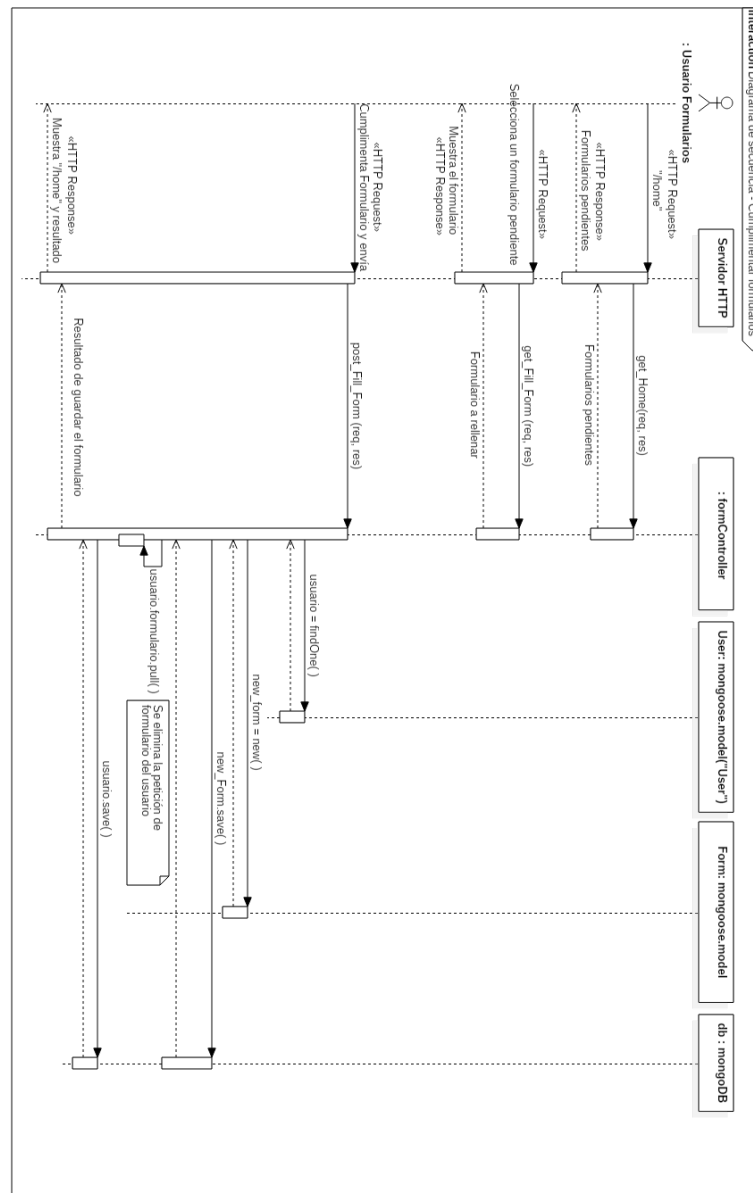
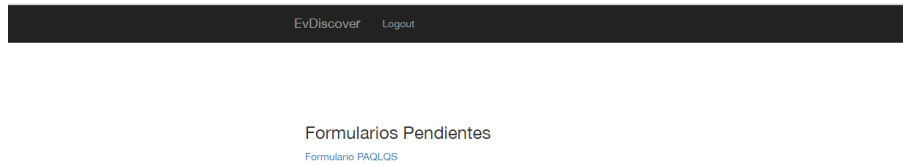


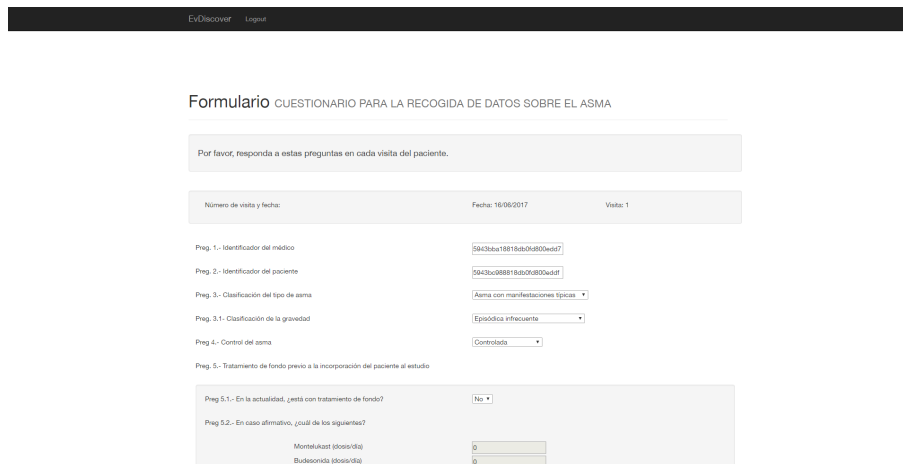
Figura 11.23: Diagrama de secuencia de cumplimentar formularios.

La interfaz consiste en una sencilla ventana donde aparecerán los formularios pendientes

a modo de enlaces [11.24](#), y que posteriormente nos redirigirán a un formulario como el del ejemplo de la figura [11.25](#).



**Figura 11.24:** Interfaz del caso de uso cumplimentar formularios.



**Figura 11.25:** Interfaz de ejemplo de uno de los formularios implementados.

#### 11.3.4.4. Diseño del caso de uso: Extraer datos

En el diagrama de la figura [11.26](#) podemos ver la realización del caso de extraer los datos de un formulario. El diseño de la interfaz puede verse en la figura [11.27](#), y consiste únicamente en una serie de enlaces a los datos de los formularios, que tras pinchar en ellos, descargarán los ficheros con las respuestas a los cuestionarios.

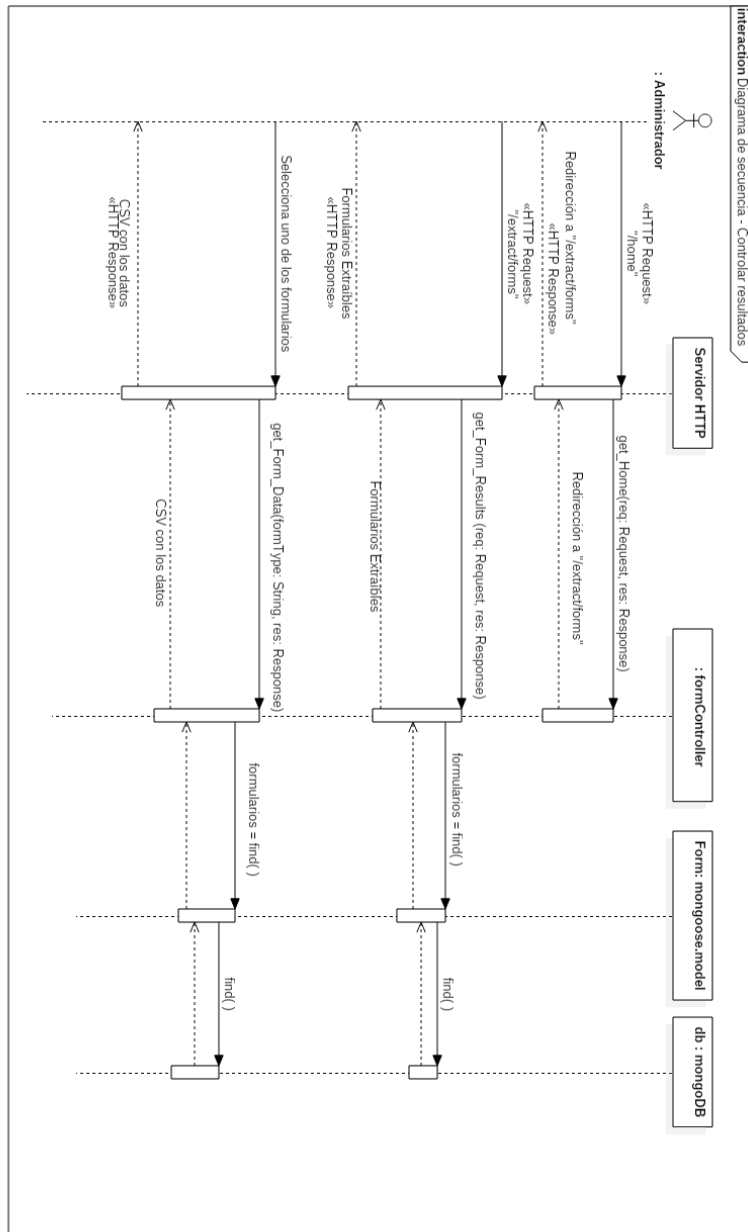


Figura 11.26: Diagrama de secuencia de extraer datos.



**Figura 11.27:** Interfaz de usuario para la extracción de datos.

### 11.3.5. Implementación

En este apartado se muestran algunos de los elementos más significativos a la hora de realizar la implementación de este componente.

#### 11.3.5.1. Enrutamiento y Middlewares

Gracias a Express, el enlace entre direcciones url y controladores, resulta realmente sencillo definir una nueva ruta ya que lo único que necesitamos realizar es algo similar a lo siguiente.

```
app.route('/home').all(middleware.funcion).get(controlador.get_Home);
```

En el ejemplo mostrado, lo que estamos haciendo es permitir que nuestra aplicación gestione la dirección “/home” y atienda las peticiones *HTTP (Hyper Text Transfer Protocol)* de tipo *GET*. Esto es muy útil a la hora de crear nuevas rutas o exponer las del servicio *REST*, que están basadas en los métodos *HTTP*.

Otro de los elementos que aparece en el ejemplo, es una función de tipo *middleware*. Las funciones de *middleware* son funciones que tienen acceso a la solicitud del cliente web, a la respuesta del servidor y a la siguiente función de *middleware* incluida en nuestra aplicación. Normalmente se llama a la siguiente función mediante una variable denominada *next*. En nuestro caso, ejecutaría el *middleware*, y si este así lo decidiera, la petición pasaría al controlador.

En este proyecto esta tipo de funciones se han utilizado varias veces con el objetivo de

limitar el acceso a las distintas partes del sistema, o dicho de otra forma, para controlar si un usuario ha accedido al sistema o no.

#### 11.3.5.2. Modelos

En lo que respecta al modelo de datos utilizado, es de tipo documental y por tanto podríamos llegar a guardar cualquier tipo de contenido dentro de las colecciones. Con el objetivo de controlar efectivamente los datos almacenados en este proyecto se utilizan modelos de *mongoose*.

Los modelos de *mongoose* nos permiten generar contenedores con los requisitos deseados, y que posteriormente nos ayudaran a recuperar o guardar dichos modelos. En el código de la figura 11.28 podemos ver un ejemplo de un modelo llamado persona, que necesariamente tendrá un nombre y unos apellidos.

```
var mongoose = require('mongoose');

var myschema = mongoose.Schema({
  nombre: { type: String, required: true},
  apellidos :{ type: String, required: true}
});

module.exports = mongoose.model("persona",formSchema, "Coleccion");
```

**Figura 11.28:** Ejemplo de un modelo de *mongoose*

Es interesante señalar que cuando usamos una base de datos de tipo *NoSQL*, las consultas pueden llegar a ser más complejas, porque dependiendo de la situación, nos podemos encontrar que una consulta nos puede devolver distintos tipos de documentos. Gracias a estos modelos, las búsquedas son mucho más sencillas, pues nos permitirá obtener todos los documentos que cumplan al menos las restricciones del modelo, y sobre las que además también podremos realizar filtros.

A la hora de guardar los datos en la base de datos, ocurre algo similar, pues gracias al modelo, podremos asegurarnos de que solo se guardan los campos definidos en el modelo y con las restricciones impuestas en él.



### 11.3.5.3. Vistas

En este sistema las vistas son generadas utilizando el motor de plantillas *Jade* (actualmente llamado *Pug*). Estas plantillas nos permiten incluir código *javascript* en ellas para generar dinámicamente el contenido, en base a los parámetros pasados por el controlador.

Explicado de forma sencilla y con un ejemplo, podríamos programar la plantilla de un cuestionario para que en función del número de la visita, incluyera o no unas preguntas. Esto es algo que viene muy bien en este dominio en concreto, pues en algunos cuestionarios pasados a los médicos o pacientes en la primera visita, se ingresan algunos datos que en otras no. En la figura 11.29 podemos ver un ejemplo de la conversión que se realiza de un archivo *Jade* (parte izquierda) a un *HTML* (parte derecha).

```
html
head
  title Titulo
body
-for (var i = 0; i<2; ++i)
  p Parrafo #{i}
```

```
<html>
  <head>
    <title>Titulo</title>
  </head>
  <body>
    <p>Parrafo 0</p>
    <p>Parrafo 1</p>
  </body>
</html>
```

**Figura 11.29:** Ejemplo de conversión *Jade* a *HTML*

### 11.3.5.4. Creando nuevos formularios

Aunque no existe un concepto de herencia, se ha desarrollado una estructura base para poder implementar nuevos formularios sin demasiado esfuerzo. Para implementar un nuevo formulario solamente habría que seguir los siguientes pasos:

1. Generar un modelo de *mongoose* con al menos los siguientes campos.
  - Nombre: Cada nuevo tipo de cuestionario deberá tener su propio nombre.
  - Visita: Número de la visita del paciente.
  - Usuario: Identificador del usuario que rellena el cuestionario
  - Paciente: Identificador del paciente al que hace referencia el cuestionario.
2. Generar la vista en *Jade* del nuevo formulario donde se tengan en cuenta al menos los campos descritos en el paso anterior.

3. Añadir en el fichero de configuración de formularios un nuevo registro indicando el modelo, vista y los campos a utilizar como columnas del *CSV*.
4. Añadir el nombre del formulario a la lista de enumerados de las solicitudes.

## 11.4. Integración

La integración entre los dos sistemas se ha realizado a través de una serie de direcciones URL establecidas en EvDiscover-Forms, con el objetivo de exponer una serie de recursos de dicha aplicación. A la hora de exponer los recursos, *Express* nos facilita mucho las cosas, pues es muy sencillo asignar una URL a un método *HTTP* concreto, y a su vez enlazarlo al controlador encargado de realizar dicha acción.

URL	Método HTTP	Descripción
/api/estudios	POST	Dar de alta un estudio
/api/users	POST	Dar de alta un usuario en el sistema
/api/requests	POST	Añadir una solicitud de formulario a un par paciente/médico
/api/requests/pacienteId	GET	Obtener si existen solicitudes pendientes para un paciente concreto

**Tabla 11.2:** Tabla de direcciones de la *API* de EvDiscover-Forms.

A la hora de consumir estos métodos es importante dejar claro que se utiliza el formato *JSON* para el intercambio de mensajes entre ambos sistemas, lo cual facilita mucho el posterior procesamiento de los mensajes.

En Bizagi es necesario definir un conector para poder realizar conexiones con el exterior. El proceso de creación de un nuevo conector es muy sencillo y se realiza a través del propio Bizagi Studio. Para realizar este proceso solamente hay que acceder a la sección de integración del asistente, y una vez ahí, seleccionaremos la tarea de tipo servicio que queramos utilizar.

La primera vez nos pedirá la ruta a utilizar, tras esa primera creación el conector quedará guardado por si quisiéramos usarlo en un futuro. Una vez establecida la ruta, tendremos

que especificar los parámetros a enviar y los que se van a recibir. Todo esto se realiza de forma visual e intuitiva conectando los valores entre si, tal y como podemos ver en la figura 11.30.

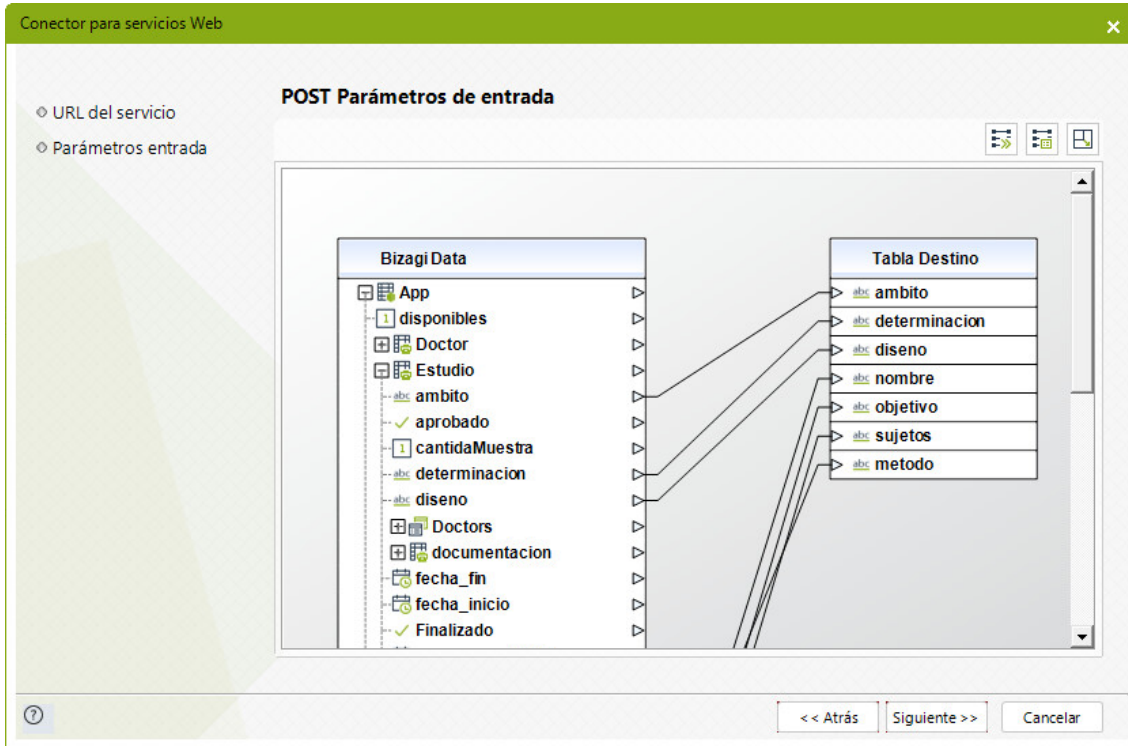


Figura 11.30: Ventana de configuración de datos de entrada del conector.

En los siguientes diagramas de secuencia (11.31, 11.32, 11.33 y 11.34), podemos ver las distintas interacciones realizadas entre los dos componentes del sistema.

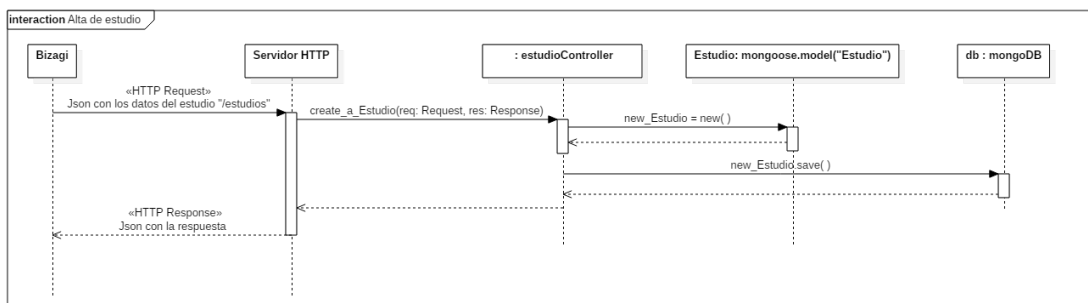
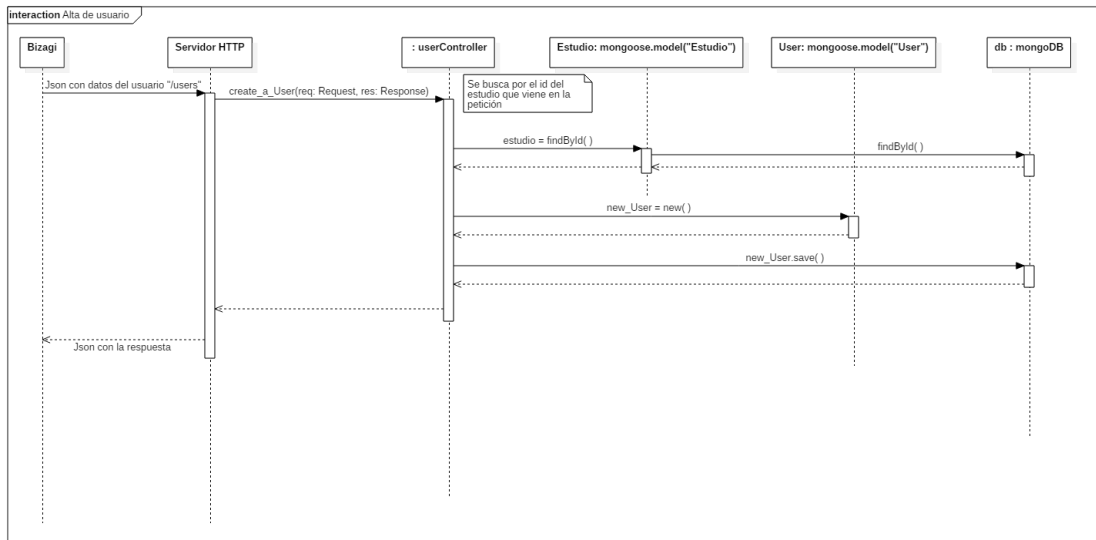
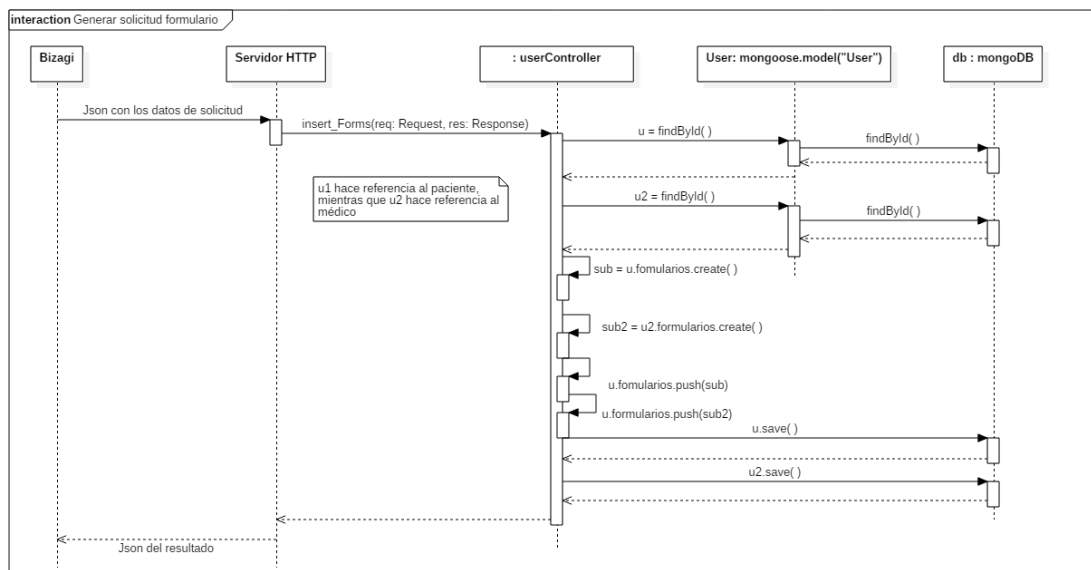


Figura 11.31: Diagrama de secuencia de alta de estudio en la API.



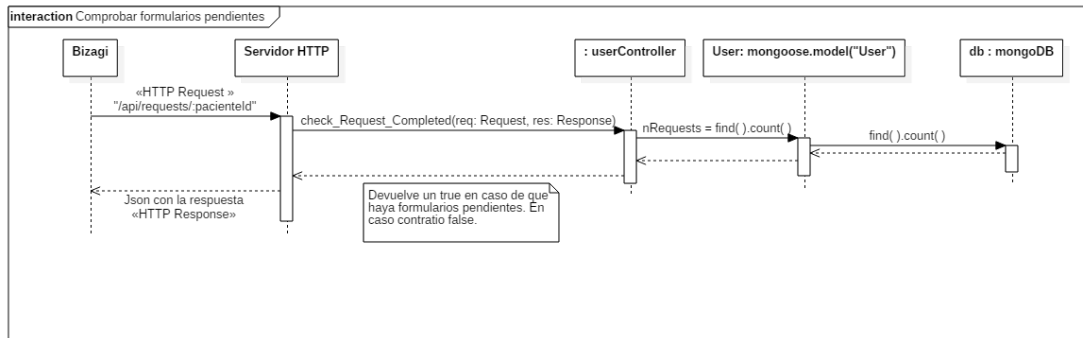
**Figura 11.32:** Diagrama de secuencia de alta de usuario.



**Figura 11.33:** Diagrama de secuencia de generación de solicitud de formulario

## 11.5. Ventajas y desventajas

Con el objetivo de poder apreciar las características de esta solución en particular en el dominio de la gestión de estudios clínicos, se exponen a continuación una serie de ventajas y desventajas de la solución planteada.



**Figura 11.34:** Diagrama de secuencia de comprobación de formularios pendientes

### 11.5.1. Ventajas

Entre las ventajas de este sistema podemos encontrar dos tipos, unas relacionadas con el uso de sistema *BPM* y otras de la solución en particular.

Relacionadas con el uso de un *BPMS*:

- Mejora continua de los procesos.
- Reducción de los tiempos entre actividades.
- Monitorización y trazabilidad de los procesos
- Automatización de las tareas.
- Optimización de los recursos.

En cuanto a las ventajas derivadas de la solución planteada.

- Al estar la aplicación separada en dos componentes distintos cualquiera de los dos puede ser sustituido en caso de encontrar una solución que aporte mejoras sustanciales.
- Al estar la aplicación de cuestionarios separada del *BPMS* es mucho más simple añadir nuevos formularios al sistema, disminuyendo además la dependencia tecnológica de Bizagi

### 11.5.2. Desventajas

Las desventajas de esta solución son las siguientes:

- Tanto administradores como administrativos deben tener cuidado de las acciones realizadas, pues son los responsables de realizar tareas de mucho peso, que de no ser realizadas correctamente pueden echar a perder el proceso. Esto se debe a que una vez avanzado el proceso no es posible dar marcha atrás en el mismo.
- La no posibilidad de volver hacia atrás en un proceso es un factor muy a tener en cuenta, pues si ocurren equivocaciones graves por parte del personal no será posible volver hacia atrás. Ante esto, la funcionalidad implementada para poder registrar las incidencias puede ser una solución.
- La división en dos componentes distintos implica que hay un doble esfuerzo en cuanto a desarrollo.

## 11.6. Conclusiones

A raíz de todo lo visto anteriormente, a continuación se exponen una serie de conclusiones relativas al proyecto. En líneas generales se podría decir que se han cumplido los objetivos, y que con el tiempo y procesos de refinamiento adecuados, podría convertirse en un producto viable para su uso. También comentar que se han cumplido los requisitos establecidos en el inicio.

Si bien a lo largo de este proyecto se han descubierto las múltiples ventajas del uso de suites *BPM*, también se han encontrado deficiencias a la hora de realizar los desarrollos, pues a menudo han surgido problemas debidos al software utilizado para el desarrollo de esta parte. Cuando se utiliza un software tan completo y complejo como es Bizagi para la elaboración de un producto, nos podemos encontrar con muchos problemas y fallos que normalmente no tendríamos en un entorno de desarrollo normal.

En algunos casos se han encontrado problemas a la hora de modificar los procesos creados inicialmente, en concreto al utilizar elementos de tipo evento y subprocesos. Cuando se han intentado eliminar dichos eventos o subprocesos, el sistema lanzaba errores y no permitía quitar los elementos del modelo de proceso. No obstante, ante esta situación se pueden plantear dos soluciones. Marcar los errores como tal y dejarlos fuera del flujo del

proceso, o si se hubiera dispuesto de la licencia comercial de Bizagi, contactar con el servicio de soporte para arreglar dicho fallo.

A pesar de los fallos y problemas encontrados a la hora de desarrollar el sistema mediante la suite escogida, se ha podido comprobar que ésta en efecto es muy completa.

Otra de las conclusiones obtenidas mediante este proyecto es que el uso de una norma y una metodología han facilitado muchas cosas a la hora de saber cuál era el siguiente paso a realizar en el desarrollo, así como seguir las fases ha facilitado comprender mejor el problema a solucionar.

### 11.6.1. Líneas futuras

En base a los resultados obtenidos mediante el prototipo, se proponen las mejoras que a continuación mencionaremos divididas en cada uno de los componentes. En cualquier caso, para ambos sistemas, una de las siguientes mejoras con mayor importancia sería asegurar el cumplimiento de lo establecido en la LOPD.

#### 11.6.1.1. Gestión del proceso de un ensayo clínico

- A la vista de las limitaciones encontradas en Bizagi, realizar la implementación utilizando el software de IBM. Es importante tener en cuenta que ésto supondría la necesidad de adquirir especialistas de la plataforma para la correcta configuración de la misma, así como personal que ayudara a mantener toda la infraestructura necesaria.
- Modificar de alguna manera el proceso, de forma que se permita generar solicitudes de cuestionarios más allá de los tres formularios a definir al diseñar el estudio. Ello significaría poder pasar más de un formulario a un usuario del sistema, o incluso poder pasar un cuestionario en un punto concreto del estudio.
- Añadir la posibilidad de escoger el tipo de estudio a realizar y en función de ello realizar un proceso u otro. Esto sería interesante de cara a poder controlar todos los tipos de estudio personalizando el tipo de proceso por cada uno de los distintos tipos.
- Añadir nuevos tipos de limitaciones a la muestra necesaria. En este proyecto solo se controla el número mínimo de médicos por grupo y se establece el mínimo

de pacientes a 1. No obstante, se propone una mayor especificidad del tamaño de muestra, pudiendo escoger entre número de pacientes, número de médicos por grupo o incluso una combinación de ambas.

- Posibilidad de obtener los médicos directamente desde una base de datos en base a filtros establecidos en el estudio. Aunque esta característica podría ser especialmente interesante, implicaría establecer una base de datos y los mecanismos adecuados para el filtrado de los médicos.
- Utilizar el nombre del estudio como referencia en el portal de trabajo en lugar de un número. Esto permitiría una mayor facilidad a la hora de saber a qué estudio pertenece una determinada tarea.
- Asignación de las tareas de los administrativos a aquellos cuyo jefe sea el administrador. Esto supondría un mayor control sobre que administrativos pueden colaborar en un estudio.
- Utilización de las entidades de tipo *StakeHolder* de forma que pueda personalizarse la experiencia.

#### 11.6.1.2. Aplicación de cuestionarios clínicos

- Permitir la creación de nuevos cuestionarios en el sistema por medio de la subida de un archivo de configuración. Esto facilitaría muchas cosas a la hora de añadir nuevos formularios al sistema. No obstante, también requeriría un control y definición de cómo crearlos muy completo y limitaría en gran medida la personalización de los mismos.
- Permitir añadir solicitudes de cumplimentación solo a un usuario, y no a un par paciente/médico como se realiza actualmente. Actualmente el sistema si soporta esta funcionalidad, pero debido a la restricción de interacción con Bizagi las solicitudes siempre se generan en pares. Bastaría con añadir un nuevo método a la *API*, ya que el modelo de dominio si que soportaría dicha acción.
- Permitir la gestión de los usuarios de un estudio concreto al administrador con el objetivo de eliminar solicitudes de formulario erróneas o restablecimiento de contraseñas para los distintos usuarios.





## CAPÍTULO 12

---

### Análisis de riesgos

---

<b>Probabilidad</b>	<b>Alta</b>
<b>Descripción</b>	Pueden darse dependencias tecnológicas debido al tipo de tecnología a utilizar.
<b>Impacto</b>	Grave
<b>Mitigación</b>	Separar en distintos componentes aquellas funcionalidades diferentes. Gracias a esto podremos sustituir cualquiera de los componentes en caso de vernos limitados por la tecnología.

**Tabla 12.1:** Riesgo de dependencia tecnológica.

<b>Probabilidad</b>	<b>Muy alta</b>
<b>Descripción</b>	Puede darse que las herramientas utilizadas para el desarrollo se actualicen, perdiendo la compatibilidad con el proyecto desarrollado.
<b>Impacto</b>	Grave
<b>Mitigación</b>	Desactivar actualizaciones automáticas. Analizar las ventajas/desventajas de instalar las actualizaciones y en caso de efectuarlas realizar copias de seguridad antes de aplicarlas. Crear un repositorio con los instaladores.

**Tabla 12.2:** Riesgo de actualizaciones del software de desarrollo.

<b>Probabilidad</b>	<b>Alta</b>
<b>Descripción</b>	Problemas con la duración del proyecto. Esto implicaría quedarse sin recursos para la producción del mismo y problemas en las entregas al cliente.
<b>Impacto</b>	Grave
<b>Mitigación</b>	Realizar un desarrollo iterativo mediante prototipos que se puedan ir mostrando al cliente.

**Tabla 12.3:** Riesgo de duración del proyecto.

<b>Probabilidad</b>	<b>Media</b>
<b>Descripción</b>	Problemas con la caída del servidor imposibilitando el trabajo a los distintos implicados.
<b>Impacto</b>	Medio
<b>Mitigación</b>	Crear un duplicado del servidor para asistir a los usuarios en caso de fallo del servidor original.

**Tabla 12.4:** Riesgo de caídas del servidor.

<b>Probabilidad</b>	<b>Muy Alta</b>
<b>Descripción</b>	El no cumplimiento de la legislación vigente en materia de protección de datos supondría sanciones.
<b>Impacto</b>	Muy Grave
<b>Mitigación</b>	Realizar un análisis a fondo de los requisitos a cumplir e implementar las medidas necesarias antes de pasar a la fase de construcción.

**Tabla 12.5:** Riesgo por incumplimiento de legislación.

### Organizaciones y gestión del proyecto

---

En este apartado se explica la sistemática a utilizar para la ejecución del proyecto con el objeto de ayudar en la gestión de las futuras fases.

Para llevar a cabo la gestión de este proyecto se ha utilizado la metodología ágil *OpenUp* para todas las fases del proyecto. Aunque en este caso el proyecto ha sido realizado por una única persona, la propia metodología ya contempla qué implicados son los que tienen que realizar que tareas. La metodología utilizada en este desarrollo propone el establecimiento de una serie de iteraciones con objetivos prefijados, que darán como resultado versiones del sistema que podrán ser entregadas.

Al inicio de todas las iteraciones es necesario actualizar una serie de artefactos relativos a la gestión del proyecto:

- Plan de proyecto: En este plan se irán añadiendo los distintos hitos y objetivos a cumplir en cada una de las iteraciones planificadas.
- Plan de iteración: En este artefacto se añade el plan para una iteración concreta y al finalizar la iteración se realiza una evaluación.
- Lista de elementos de trabajo: Se incluyen las estimaciones de las actividades a realizar en la iteración y posteriormente se registran las horas invertidas.
- Lista de riesgos: Se incluyen los riesgos identificados hasta la fecha.

Al inicio de cada una de estas iteraciones se realiza un refinamiento de los requisitos necesarios para la cumplimentación de los objetivos de esa iteración. Posteriormente, se pasa a realizar el análisis, diseño, implementación y pruebas de los objetivos establecidos para ese ciclo en concreto.

Con el objetivo de que todos puedan trabajar y colaborar en el proyecto, se propone que el sistema de información este alojado en la nube en todo momento, y sea posible editar los documentos desde el mismo, facilitado así la colaboración entre miembros.

En base al resultado obtenido en este proyecto, se estima que se debería formar un equipo completo para el desarrollo de este producto, y que estuviera compuesto por los siguientes responsables.

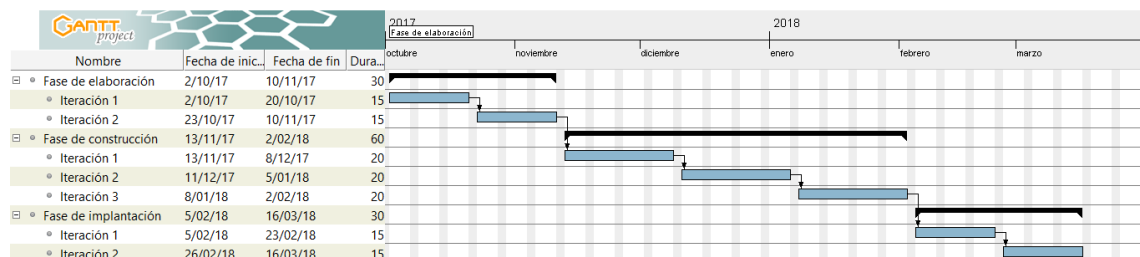
- Directores de proyecto : 2 (1 por componente)
- Arquitecto de Software: 1
- Desarrolladores : 4 (2 por componente)
- Testers: 4 (2 por componente)

En cualquier caso, de seguir la metodología *OpenUp* esta contiene todos los aspectos que pueden facilitar al equipo trabajar de una forma ordenada y correcta. Como recomendación y a la vista de que la metodología utilizada ha sido actualizada se aconseja que el equipo utilice la nueva versión pues además de incluir los contenidos utilizados en la primera añada nuevos artefactos y procesos que pueden mejorar la calidad del proyecto de manera sustancial.

## Planificación temporal

En este apartado se presenta la planificación temporal diseñada para la elaboración de este proyecto tras su aceptación.

En vista de las mejoras restantes a realizar para obtener una mayor calidad y la necesidad de realizar pruebas extensas con usuarios antes de la puesta en marcha se plantean las fechas indicadas en el diagrama de gantt de la figura 14.1.



**Figura 14.1:** Diagrama de gantt con las fechas estimadas para el desarrollo final

Para obtener una mejor calidad y adaptar la solución propuesta al cumplimiento de la legislación vigente, se plantea realizar dos iteraciones de elaboración adicionales. Tras eso se comenzará con la fase de construcción y finalmente la implantación del sistema.

En la tabla 14.1 podemos ver la estimación propuesta para la construcción de este proyecto. A pesar de que pueda parecer que son muchas horas, habría que tener en cuenta que el análisis en lo que respecta a los datos de carácter personal, es vital en este caso. En total estaríamos ante una dedicación estimada de unas 1220 h.

<b>Fase de elaboración</b>	
<b>Tarea</b>	<b>Estimación(horas)</b>
Refinamiento de los requisitos	20h
Análisis para el cumplimiento legislativo	50h
Remodelación del diseño	50h
Implementación de las mejoras	200h
Tests	20h
<b>Fase de construcción</b>	
<b>Ambiente desarrollo</b>	
<b>Tarea</b>	<b>Estimación(horas)</b>
Refinamiento de los requisitos	30h
Implementación final	300h
Tests	30h
<b>Ambiente de pruebas</b>	
<b>Tarea</b>	<b>Estimación(horas)</b>
Simulaciones	50h
Tests dirigidos	50h
<b>Ambiente de producción</b>	
<b>Tarea</b>	<b>Estimación(horas)</b>
Ajustes finales	50h
Pruebas con usuarios reales	200h
<b>Fase de implantación</b>	
<b>Tarea</b>	<b>Estimación(horas)</b>
Configuración de servidores	20h
Configuración de seguridad	50h
Formación a usuarios	100h

**Tabla 14.1:** Estimación temporal de las siguientes fases del proyecto siguiendo la metodología *OpenUP*

## CAPÍTULO 15

---

### Presupuesto

---

En este capítulo se presenta un presupuesto que ha sido elaborado en base a las planificaciones y la organización propuesta.

El presupuesto propuesto para el concepto de personal lo podemos ver en la tabla 15.1, el precio por hora de cada uno de los implicados está planteado en base al Acuerdo Marco 26/2010 del Ministerio de Economía y Hacienda obtenido del Director de este proyecto.

<b>Puesto</b>	<b>Horas</b>	<b>Precio (€/hora)</b>	<b>Total</b>
Directores de proyecto	100h	100€/h	10.000,00.-€
Arquitecto	100h	70€/h	7.000,00.-€
Desarrolladores	670h	35€/h	23.450,00.-€
Tester	150h	35€/h	5.250,00.-€
<b>Total</b>			<b>45.700,00.-€</b>

**Tabla 15.1:** Presupuesto para el personal

Dado que también será necesario adquirir una serie de licencias para el correcto desarrollo del proyecto, también se incluyen los presupuestos en la tabla 15.2. Éstos han sido extraídos directamente de las páginas web de cada uno de los productos.



<b>Recurso</b>	<b>Proveedor</b>	<b>Precio</b>
Bizagi Engine	Bizagi	725,00.-€
Mercury mail	David Harris	68,00.-€
Office 365	Microsoft	1.419,00.-€
StarUML	MKLab	89,00,00.-€
<b>Total</b>		<b>2.301,00.-€</b>

**Tabla 15.2:** Presupuesto para material

## **PARTE II**

### **Elaboración del proyecto**



## Plan inicial del proyecto

---

### 1.1. Alcance

#### 1.1.1. Alcance mínimo

- La solución debe estar realizada al menos en parte mediante un *BPMS*.
- Web con la documentación de la metodología *OpenUp*.
- La memoria debe estar realizada en base a la norma *UNE 157801:2007*.

La aplicación web a desarrollar deberá comprender como mínimo los siguientes aspectos:

- La solución final será un prototipo de aplicación web y deberá contemplar los elementos más comunes de los estudios clínicos.
- La aplicación dará soporte a los estudios que cumplan los siguientes requisitos:
  - Fase II en adelante.
  - Estudios controlados donde exista un grupo de control y uno de experimentación.
  - Estudios de tipo prospectivo, ya que son éstos los que necesitan más ayuda para controlar los procesos.

- Estudios aleatorizados donde no esté restringido quién pertenece a que grupo.
- Existirán 5 tipos de usuarios
  - Administrador o investigador principal encargado de la gestión del estudio.
  - Analista encargado de realizar un informe con las conclusiones extraídas de los datos del estudio.
  - Administrativos o investigadores auxiliares encargados de realizar la supervisión de los distintos permisos o tareas de índole administrativa.
  - Médicos participantes en el estudio.
  - Pacientes/Cuidadores participantes en el estudio.
- Exportación de los datos de carácter analítico en cualquier momento del estudio.
- Se deberán poder extraer los datos del estudio en un formato habitual de las herramientas de análisis.
- Se dividirá la aplicación en tres fases:
  - **Fase de inicio del estudio:**
    - Creación del estudio por parte del administrador.
    - Obtención de permisos por parte del administrativo.
    - Creación de los grupos del estudio.
  - **Fase de evolución del estudio:**
    - Control del proceso por parte del administrador en todo momento.
    - Realización de notificaciones y registro de incidencias por parte del administrativo/administrador.
    - Registro de pacientes/cuidadores y visitas por parte de los médicos.
    - Cumplimentación de cuestionarios por parte de pacientes, cuidadores y médicos.
  - **Fase de finalización del estudio:**
    - Extracción de resúmenes de incidencias y participantes por parte del administrador.

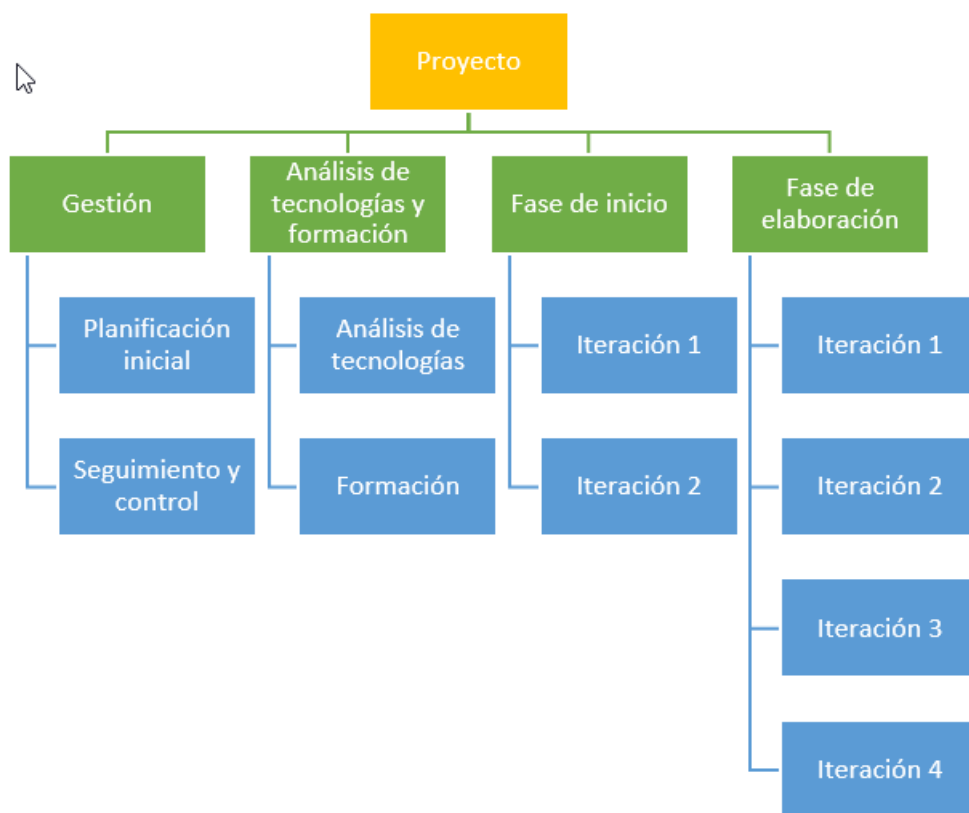
### 1.1.2. Ampliación del alcance

En caso de completar el alcance mínimo establecido se propone realizar las siguientes ampliaciones

- Añadir la posibilidad al médico de ver las fichas de sus pacientes junto con un historial de sus visitas.
- Añadir la posibilidad de incluir médicos una vez comenzado el estudio.

## 1.2. WBS/EDT

En la figura 1.1 podemos ver el *WBS/EDT* del proyecto en cuestión.



**Figura 1.1:** WBS/EDT del proyecto

## 1.3. Planificación temporal

### 1.3.1. Entregables

Dada la metodología y la norma a utilizar en este proyecto tendremos diferentes tipos de entregables. Algunos estarán relacionados con la entrega al cliente del proyecto mientras que otros en cambio estarán más relacionados con la gestión interna que se realiza del mismo. A continuación, se expone la lista de entregables.

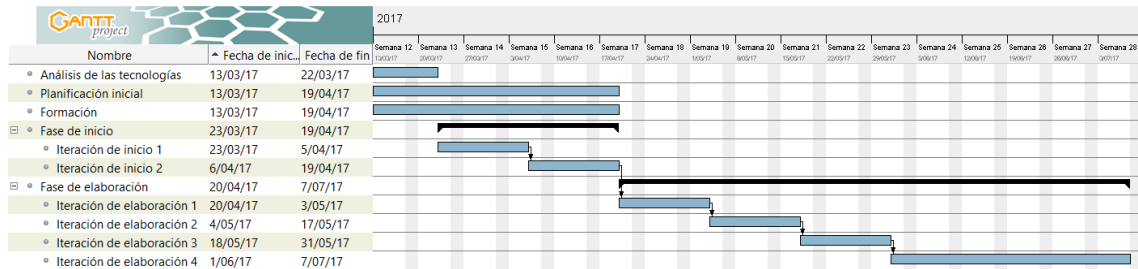
- Máquina virtual con el prototipo del sistema propuesto como solución.
- Memoria del proyecto siguiendo la norma *AENOR UNE 157801:2007* para la presentación de proyectos.
- Página web con la documentación producida por la metodología *OpenUp*.

### 1.3.2. Hitos

- Finalización de la planificación inicial. (19/04/2017)
- Fin del análisis de tecnologías. (22/03/2017)
- Fin de la fase de inicio. (19/04/2017)
- Fin de la fase de elaboración. (28/05/2017)
- Entrega final del proyecto. (29/06/2017)

### 1.3.3. Diagrama de Gantt

Los periodos definidos para el desarrollo de las distintas fases de este proyecto quedan reflejados en el diagrama de Gantt de la figura 1.2. Tal y como puede verse, la última iteración de elaboración tiene una mayor duración, ya que será en ésta en la que se elabore la memoria y se prepare la documentación a presentar.



**Figura 1.2:** Diagrama Gantt inicial del proyecto

### 1.3.4. Estimación inicial

Debido al modelo de desarrollo basado en prototipos, se ha decidido realizar las estimaciones en base al coste global de cada iteración. Entre los documentos de gestión del anexo, pueden verse las estimaciones y dedicaciones por cada una de las tareas realizadas. En la tabla 1.1 se puede encontrar las estimaciones realizadas.

Paquete	Estimación (horas)
<b>Gestión</b>	
Planificación	10
Seguimiento y control	20
<b>Alternativas y formación</b>	
Alternativas	30
Formación	30
<b>Fase de inicio</b>	
Iteración 1	30
Iteración 2	20
<b>Fase de elaboración</b>	
Iteración 1	50
Iteración 2	50
Iteración 3	50
Iteración 4	110
<b>Total</b>	<b>400</b>

**Tabla 1.1:** Tabla de estimaciones

Estas estimaciones son totalmente orientativas, pues dada la metodología a aplicar, se realizarán planificaciones parciales al inicio de cada iteración. Tal y como puede apreciarse, se ha estimado una menor duración a las iteraciones de inicio debido a que se estará realizando al mismo tiempo el análisis de alternativas y formación.



## 1.4. Plan de comunicación y seguimiento

Con el objetivo de llevar un seguimiento adecuado del proyecto, se realizarán reuniones de evaluación tanto al inicio como al final de cada iteración, con el objetivo de verificar que se van cumpliendo los objetivos establecidos para cada una de las iteraciones, y fijar los objetivos de las siguientes. Todas las comunicaciones fuera de las reuniones establecidas para la evaluación del trabajo realizado, se harán vía correo electrónico.

## 1.5. Plan para la gestión de la calidad

- **Documentación** Con el fin de ofrecer una documentación de calidad, se utilizarán en la medida de lo posible las plantillas de los artefactos que ofrece la metodología a aplicar.
- **Memoria** La memoria estará realizada en base a la norma *UNE 157801:2007*, que establece una serie de contenidos para cada uno de los apartados de la misma, y gracias a ello nos aseguraremos que los contenidos básicos aparezcan.
- **Producto** Con el objetivo de encontrar una solución de calidad al problema a solventar, se desarrollarán prototipos funcionales en cada una de las iteraciones. De esta forma podremos refinar los requisitos en cualquier momento.

## 1.6. Selección de la suite BPM

Dado que el tipo de tecnología a utilizar es compleja y que implica realizar una serie de puestas en marcha y configuraciones, se ha decidido que dedicará un tiempo razonable a evaluar cada una de las herramientas mediante los talleres de evaluación que los proveedores de este tipo de software suelen ofrecer.

## 1.7. Sistema de información

Se ha decidido que el sistema de información del proyecto sea alojado en *Google Drive* por la facilidad y comodidad de acceder a los contenidos desde cualquier sitio y las he-

ramientas que provee para la creación de copias de seguridad como es *Google Takeout*. A continuación se expone la estructura del mismo.

- **Recursos:** En esta carpeta se irán guardando los recursos que puedan ser útiles tanto para el aprendizaje de las tecnologías necesarias, recursos provistos por el Director, imágenes o documentos que puedan ser utilizados para la memoria, o la presentación de la defensa.
- **Documentación:** En esta carpeta se guardará la memoria escrita, la presentación a utilizar en la defensa del proyecto y la página web con los contenidos de metodología *OpenUp*.
- **Public:** Carpeta donde almacenar los recursos propios a compartir con el Director de proyecto.

Además, y dado que se utilizarán máquinas virtuales para el desarrollo de este proyecto, estas serán almacenadas en un disco duro externo a modo de copia de seguridad.

## 1.8. Gestión de los riesgos

En la esta sección se recogen los riesgos más significativos detectados para el desarrollo de este proyecto, así como un plan de contingencia para los mismos. Los riesgos más evidentes son los siguientes:

- Aprendizaje tecnológico. (Tabla 1.2)
- Pérdida de datos. (Tabla 1.3)
- No cumplir con los requisitos y plazos. (Tabla 1.4)

<b>Probabilidad</b>	<b>Media</b>
<b>Descripción</b>	Desconocimiento de las herramientas a utilizar para la elaboración del proyecto.
<b>Impacto</b>	Grave. El tiempo dedicado al aprendizaje puede lastrar al desarrollo global del proyecto.
<b>Mitigación</b>	Una vez comprendidas las funciones básicas de las herramientas a utilizar, se irá profundizando en ellas a lo largo del proyecto.

**Tabla 1.2:** Riesgo derivado del Aprendizaje Tecnológico.

<b>Probabilidad</b>	<b>Baja</b>
<b>Descripción</b>	Siempre es posible que puedan suceder percances como que se rompa el equipo o borremos algún dato sin querer.
<b>Impacto</b>	Muy Grave. Se perderían todos los avances realizados en el proyecto, perjudicando a todas las áreas del mismo.
<b>Mitigación</b>	Se realizarán copias de seguridad diarias de la máquina virtual que contenga la aplicación.

**Tabla 1.3:** Riesgo derivado de la pérdida de datos

<b>Probabilidad</b>	<b>Media</b>
<b>Descripción</b>	No poder obtener el resultado esperado en las fechas establecidas.
<b>Impacto</b>	Muy Grave. La falta de tiempo, o la no cumplimentación de los objetivos en plazo, puede suponer no lograr presentar un proyecto de calidad o la ausencia de un prototipo viable para su uso.
<b>Mitigación</b>	Se irán realizando iterativos prototipos de manera que se puedan ir alcanzando los objetivos generales del proyecto e ir refinándolo en sucesivas iteraciones. Además, se ha establecido la fecha final del proyecto a una fecha adelantada que permite cierto margen de modificación.

**Tabla 1.4:** Tabla de riesgos derivada de requisitos y plazos.

### Seguimiento y control

---

En este capítulo se describe el trabajo realizado en este proyecto, y las desviaciones resultantes con respecto a la planificación inicial.

#### 2.1. Gestión del alcance

El alcance final del proyecto comprende todos los puntos incluidos en el alcance inicial y la ampliación del mismo, expuestos en la planificación inicial, a excepción de algunos cambios comentados a continuación:

- En el momento de estar realizando la fase de análisis de alternativas se descubrió que las suites BPM están bastante limitadas en algunos aspectos. Por ello, y tras identificar un riesgo claro debido a la dependencia tecnológica, se decide realizar la separación del software a desarrollar en dos componentes distintos.
- En la reunión del 26/04/2017 se decide eliminar el rol del analista del sistema pues pese a que sí que suelen intervenir, éste no participa en el proceso de gestión sino que más bien ayuda posteriormente a interpretar los datos.

## 2.2. Objetivos por iteración

Dada la metodología aplicada en este proyecto se han realizado planificaciones en cada una de las iteraciones que se han llevado a cabo. Esto ha supuesto que, en ocasiones, se replanificaran las dedicaciones de las mismas en base a los objetivos establecidos. En la tabla 2.1 podemos ver los objetivos establecidos en cada una de ellas.

Fase	Iteración	Objetivos
Inicio	1	<ul style="list-style-type: none"> <li>■ Analizar el problema.</li> <li>■ Comprender el problema.</li> <li>■ Captura de requisitos. Modelo de casos de uso.</li> </ul>
Inicio	2	<ul style="list-style-type: none"> <li>■ Analizar el problema.</li> <li>■ Comprender el problema.</li> <li>■ Refinamiento de los requisitos. Modelo de análisis.</li> </ul>
Elaboración	3	<ul style="list-style-type: none"> <li>■ Obtención de un prototipo inicial para la primera fase de inicio del estudio y comunicación con la <i>API</i>.</li> </ul>
Elaboración	4	<ul style="list-style-type: none"> <li>■ Obtención de un prototipo funcional con todos los casos de uso establecidos en el alcance inicial cubiertos.</li> </ul>
Elaboración	5	<ul style="list-style-type: none"> <li>■ Obtención de un prototipo funcional con los casos de uso de la ampliación del alcance.</li> <li>■ Mejora del proceso de extracción de datos.</li> </ul>
Elaboración	6	<ul style="list-style-type: none"> <li>■ Elaboración de la memoria y preparación de los documentos a presentar</li> </ul>

**Tabla 2.1:** Objetivos por iteración

Si se desea ver más información sobre cada iteración y los elementos de trabajo asignados esto puede verse en la web de la documentación del proyecto.

## 2.3. Desviaciones

### 2.3.1. Desviaciones en los plazos

En el diagrama de gantt de la figura 2.1 podemos ver que han ocurrido desviaciones en lo que respecta a fechas. Esto se debe a que inicialmente no se tomó en cuenta que la segunda iteración de elaboración comprendía fechas de carácter festivo, lo cual supuso un retraso global en el proyecto.

Otro de los desvíos a tener en cuenta, es la reducción de la duración de la primera y tercera iteración, así como la ampliación de la segunda. Ello se debe a que dado el tiempo disponible se decidió establecer una mayor carga en la segunda iteración de cara a poder realizar las mejoras planteadas en la ampliación del alcance.

### 2.3.2. Desviaciones en las estimaciones

En la tabla 2.2 podemos ver las desviaciones en horas con respecto de la planificación inicial. Las dedicaciones asignadas inicialmente no han variado de una manera significativa, a excepción de la segunda y tercera iteración de elaboración, donde podemos ver una sobrecarga muy alta que no se contempló al principio.

Esta carga es debida a que se decidió invertir algo más de tiempo en esta iteración con el objetivo de poder realizar una ampliación del alcance en la tercera iteración. No obstante, esto no supuso un problema ya que la dedicación en exceso quedo compensada con la dedicación sobrante de la tercera iteración. Cabe decir que esta reasignación de horas si que fue planificada al inicio de la segunda iteración de elaboración.

Si bien no existe una desviación muy grande con respecto a la estimación global realizada, cabe destacar que se ha debido a la imposición de objetivos en cada una de las iteraciones, así como de las tareas a realizar en cada una de ellas. Ésto sumado al desarrollo basado en prototipos nos ha permitido ajustar las estimaciones en cada una de las iteraciones. En el artefacto *Lista de elementos de trabajo* de la web podemos ver las tareas realizadas en cada una de las iteraciones con sus correspondientes estimaciones y dedicaciones.

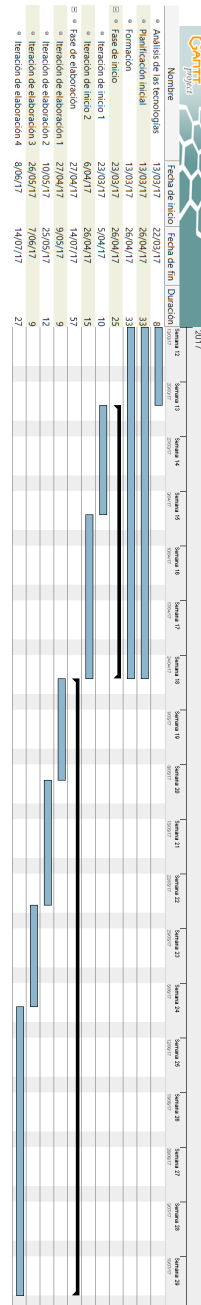


Figura 2.1: Diagrama Gantt final del proyecto

<b>Paquete</b>	<b>Estimación</b>	<b>Dedicación</b>	<b>Diferencia</b>
<b><i>Gestión</i></b>			
Planificación	10	10	0
Seguimiento y control	20	20	0
<b><i>Alternativas y formación</i></b>			
Alternativas	30	30	0
Formación	30	30	0
<b><i>Fase de inicio</i></b>			
Iteración 1	30	31	0
Iteración 2	20	21	0
<b><i>Fase de elaboración</i></b>			
Iteración 1	50	53	+3
Iteración 2	50	74	+24
Iteración 3	50	33	-17
Iteración 4	110	108	-2
<b>Total</b>	<b>400</b>	<b>410</b>	<b>+10</b>

**Tabla 2.2:** Tabla de dedicaciones(en horas)





# **Anexos**



---

## Índice de la web del proyecto

---

### Parte I - Presentación al cliente

1. Memoria .....	WEB
2. Anexos: Detalles técnicos de la solución .....	WEB
A. Análisis y diseño del sistema .....	WEB
A.1. Análisis .....	WEB
A.1.1 Modelo de análisis .....	WEB
A.2. Diseño .....	WEB
A.2.1 Cuaderno de arquitectura .....	WEB
A.2.2 Modelo de datos de Bizagi .....	WEB
A.2.3 Modelo de proceso .....	WEB
A.2.4 Modelo de diseño .....	WEB
A.3. Prototipo .....	WEB
A.3.1 Implementación .....	WEB
A.3.2 Tests .....	WEB
B. Estimación de tamaños y esfuerzos .....	(No incluido)
C. Planes de gestión de la ejecución del proyecto .....	(No incluido)
D. Seguridad .....	(No incluido)

3. Requisitos del sistema .....	<a href="#">WEB</a>
3.1. Glosario .....	<a href="#">WEB</a>
3.2. Soporte de especificación de requisitos .....	<a href="#">WEB</a>
3.3. Casos de uso .....	<a href="#">WEB</a>
3.4. Modelos de captura de requisitos .....	<a href="#">WEB</a>
3.4.1. Modelo de casos de uso v0.1 .....	<a href="#">WEB</a>
3.4.1. Modelo de casos de uso v1.0 .....	<a href="#">WEB</a>
3.5. Visión .....	<a href="#">WEB</a>
4. Presupuesto .....	<a href="#">WEB</a>
5. Estudios con entidad propia .....	(No incluido)

## **Parte II - Elaboración del proyecto**

### 1. Anexos:

A. Actas .....	<a href="#">WEB</a>
B. Plan del proyecto .....	<a href="#">WEB</a>
C. Planes de iteración .....	<a href="#">WEB</a>
C.1 Plan de iteración 1 .....	<a href="#">WEB</a>
C.2 Plan de iteración 2 .....	<a href="#">WEB</a>
C.3 Plan de iteración 3 .....	<a href="#">WEB</a>
C.4 Plan de iteración 4 .....	<a href="#">WEB</a>
C.5 Plan de iteración 5 .....	<a href="#">WEB</a>
C.6 Plan de iteración 6 .....	<a href="#">WEB</a>
D. Lista de elementos de trabajo .....	<a href="#">WEB</a>
E. Lista de riesgos .....	<a href="#">WEB</a>

## **ANEXO B**

---

**Actas**

---

## Acta de reunión

23/03/2017

15:30

UPV/EHU

<b>TIPO DE REUNIÓN</b>	Reunión de Seguimiento y Organización
<b>ORGANIZADOR</b>	Juan Manuel Pikatza
<b>APUNTADOR</b>	Ioritz Cabero Llama
<b>ASISTENTES</b>	Juan Manuel Pikatza Ioritz Cabero Llama

### Temas del orden del día

30 MINUTOS DECISIÓN SOBRE EL SISTEMA BPM A UTILIZAR

<b>DISCUSIÓN</b>		
	Una vez realizado un primer acercamiento a dos de los distintos sistemas BPM se debe tomar la decisión de escoger uno pues pronto se acabará la fase de adaptación a las tecnologías.	
	Se debe decidir entre el sistema BPM de IBM o el de Bizagi.	
<b>CONCLUSIONES</b>		
	Se utilizará el sistema de Bizagi para el desarrollo del proyecto, pues el otro sistema requiere de un refinamiento para el que no se dispone de tiempo.	
	Se planteará la solución de IBM a modo de alternativa.	
<b>PLANES DE ACCIÓN</b>	<b>RESPONSABLE</b>	<b>PLAZO</b>

15 MINUTOS DECISIÓN SOBRE EL TIPO DE SERVICIOS A UTILIZAR

<b>DISCUSIÓN</b>		
	Se pretende minimizar la dependencia del sistema BPM mediante la separación del software en dos componentes distintos conectados mediante un servicio web.	
	Decisión entre utilizar servicios REST o SOAP.	
<b>CONCLUSIONES</b>		
	Para un primer acercamiento al prototipo a desarrollar se ha decidido realizar los servicios utilizando el modelo de arquitectura REST.	
<b>PLANES DE ACCIÓN</b>	<b>RESPONSABLE</b>	<b>PLAZO</b>

45 MINUTOS SIGUIENTES PASOS A SEGUIR EN EL PROYECTO

<b>DISCUSIÓN</b>		
	Se debe comenzar a preparar la Iteración de inicio, así como terminar los apartados más necesarios para el comienzo correcto del proyecto.	
<b>CONCLUSIONES</b>		
	Se fijarán unos plazos para elaborar las versiones iniciales de algunos documentos a elaborar antes del fin de la primera semana de la iteración de inicio	
<b>PLANES DE ACCIÓN</b>	<b>RESPONSABLE</b>	<b>PLAZO</b>
Presentación de las funcionalidades iniciales a desarrollar.	Ioritz Cabero	30 de Marzo

## Acta de reunión

05/04/2017

9:00

UPV/EHU

<b>TIPO DE REUNIÓN</b>	Reunión de Seguimiento y Organización
<b>ORGANIZADOR</b>	Juan Manuel Pikatza
<b>APUNTADOR</b>	Ioritz Cabero Llama
<b>ASISTENTES</b>	Juan Manuel Pikatza Ioritz Cabero Llama

### Temas del orden del día

10 MINUTOS NOMBRIAMIENTO DEL PROYECTO

<b>DISCUSIÓN</b>	
Se habla acerca del nombre a asignar al proyecto de cara a ir rellenando de una forma más adecuada los distintos elementos de la documentación.	
<b>CONCLUSIONES</b>	
El proyecto se conocerá de esta fecha en adelante como "EvDiscover" que proviene del inglés "Evidence Discover"	

30 MINUTOS REVISIÓN Y EVALUACIÓN DE LAS TAREAS DE LA PRIMERA ITERACIÓN DE INICIO

<b>DISCUSIÓN</b>		
Se deben revisar los contenidos elaborados como resultado de la primera iteración de inicio.		
<b>CONCLUSIONES</b>		
Se han revisado y los distintos documentos y se han identificado algunos problemas de rápida solución.		
<b>PLANES DE ACCIÓN</b>	<b>RESPONSABLE</b>	<b>PLAZO</b>
Realizar la corrección de los errores identificados	Ioritz Cabero	23 de Abril

20 MINUTOS SIGUIENTES PASOS A SEGUIR EN EL PROYECTO Y PUNTOS A TENER EN CUENTA

<b>DISCUSIÓN</b>		
Se debe realizar el modelo de análisis y comenzar con el cuaderno de arquitectura, de cara a comenzar con las iteraciones de elaboración.		
Se discutirán elementos a tener en cuenta en el futuro para obtener una mejor implementación.		
<b>CONCLUSIONES</b>		
Se fija el plazo de entrega para la documentación restante de la segunda iteración de inicio.		
Se propone CSV como formato de extracción de datos para los respuestas a los formularios.		
<b>PLANES DE ACCIÓN</b>	<b>RESPONSABLE</b>	<b>PLAZO</b>
Modelo de análisis (Sin tener en cuenta la separación de componentes ni la tecnología.	Ioritz Cabero	23 de Abril
Principio del cuaderno de arquitectura	Ioritz Cabero	23 de Abril



## Acta de reunión

26/04/2017

11:00

UPV/EHU

<b>TIPO DE REUNIÓN</b>	Reunión de Seguimiento y Organización
<b>ORGANIZADOR</b>	Juan Manuel Pikatza
<b>APUNTADOR</b>	Ioritz Cabero Llama
<b>ASISTENTES</b>	Juan Manuel Pikatza Ioritz Cabero Llama

### Temas del orden del día

1H 30 MINUTOS REVISIÓN Y EVALUACIÓN DE LAS TAREAS DE LA SEGUNDA ITERACIÓN DE INICIO

<b>DISCUSIÓN</b>	
	Se deben revisar los contenidos elaborados como resultado de la segunda iteración de inicio.
<b>CONCLUSIONES</b>	
	Se han completado los objetivos de la iteración.

30 MINUTOS SIGUIENTES PASOS A SEGUIR EN EL PROYECTO Y PUNTOS A TENER EN CUENTA

<b>DISCUSIÓN</b>		
	Se deben decidir las funcionalidades que debe tener el prototipo tras la primera iteración de inicio.	
	Valoración de la necesidad de tener el rol de analista en el sistema.	
<b>CONCLUSIONES</b>		
	Se diseñarán e implementarán los casos de uso relativos a la fase de inicio del estudio, así como la versión inicial de la API REST. Esto nos permitirá refinar los requisitos relativos a esta fase.	
	Se eliminan el rol de analista pues no es necesario en este sistema.	
<b>PLANES DE ACCIÓN</b>	<b>RESPONSABLE</b>	<b>PLAZO</b>
Desarrollar los prototipos iniciales de la API y la fase de inicio del proceso	Ioritz	08/05/2017

## Acta de reunión

09/05/2017

11:00

UPV/EHU

TIPO DE REUNIÓN	Reunión de Seguimiento y Organización
ORGANIZADOR	Juan Manuel Pikatza
APUNTADOR	Ioritz Cabero Llama
ASISTENTES	Juan Manuel Pikatza Ioritz Cabero Llama

### Temas del orden del día

1H 30 MINUTOS

REVISIÓN Y EVALUACIÓN DE LAS TAREAS DE LA  
PRIMERA ITERACIÓN DE ELABORACIÓN

DISCUSIÓN		
	Se deben revisar los contenidos elaborados como resultado de la segunda iteración de inicio.	
CONCLUSIONES		
	Se han completado los objetivos de la iteración.	
PLANES DE ACCIÓN	RESPONSABLE	PLAZO
	Ioritz Cabero	

30 MINUTOS

FUNCIONALIDADES A IMPLEMENTAR EN LA  
SEGUNDA ITERACIÓN DE ELABORACIÓN

DISCUSIÓN		
	Casos de uso a implementar para la segunda iteración de elaboración.	
CONCLUSIONES		
	Inicialmente deberán implementarse los casos de uso relativos al desarrollo del estudio.	
	En el caso de los formularios se intentará dar una solución genérica, no obstante, si debido a las limitaciones de la tecnología utilizada no fuera posible. Se implementará un formulario específico a modo de ejemplo.	
PLANES DE ACCIÓN	RESPONSABLE	PLAZO
Comenzar con las implementaciones de los casos de uso de desarrollo del estudio.	Ioritz Cabero	

## Acta de reunión

25/05/2017

15:00

UPV/EHU

TIPO DE REUNIÓN	Reunión de Seguimiento y Organización
ORGANIZADOR	Juan Manuel Pikatza
APUNTADOR	Ioritz Cabero Llama
ASISTENTES	Juan Manuel Pikatza Ioritz Cabero Llama

### Temas del orden del día

1H 30 MINUTOS

REVISIÓN Y EVALUACIÓN DE LAS TAREAS DE LA  
SEGUNDA ITERACIÓN DE ELABORACIÓN

DISCUSIÓN		
	Se deben revisar los contenidos elaborados como resultado de la segunda iteración de elaboración.	
CONCLUSIONES		
	Se han completado los objetivos de la iteración. Se han encontrado algunos fallos menores relacionados con la extracción de datos.	
PLANES DE ACCIÓN	RESPONSABLE	PLAZO
Revisión de Fallos en el sistema y ampliación de los datos extraídos.	Ioritz Cabero	06/06/2017

30 MINUTOS

TAREAS PARA LA TERCERA ITERACIÓN DE ELABORACIÓN

DISCUSIÓN		
	Se debe decidir cuáles serán las tareas en esta tercera iteración de elaboración, con respecto al diseño se ha encontrado que no es viable documentarlo todo usando el mismo programa ya que ambos componentes no son iguales.	
CONCLUSIONES		
	Se dispone de tiempo para realizar la ampliación del alcance.	
	Se documentara la parte basada en Bizagi añadiendo elementos adjuntos al proceso exportado que genera este.	
	Creación del modelo de diseño mediante alguna herramienta de Modelado.	
	Implementación de formularios reales de un estudio previo.	
PLANES DE ACCIÓN	RESPONSABLE	PLAZO
Documentar el diseño y el proceso	Ioritz Cabero	06/06/2017
Incluir los casos de uso contemplados en la ampliación del alcance	Ioritz Cabero	06/06/2017
Implementar unos formularios validados a modo de ejemplo	Ioritz Cabero	06/06/2017

## Acta de reunión

07/06/2017

9:00

UPV/EHU

<b>TIPO DE REUNIÓN</b>	Reunión de Seguimiento y Organización
<b>ORGANIZADOR</b>	Juan Manuel Pikatza
<b>APUNTADOR</b>	Ioritz Cabero Llama
<b>ASISTENTES</b>	Juan Manuel Pikatza Ioritz Cabero Llama

### Temas del orden del día

1H 30 MINUTOS

REVISIÓN Y EVALUACIÓN DE LAS TAREAS DE LA  
TERCERA ITERACIÓN DE ELABORACIÓN

<b>DISCUSIÓN</b>	
	Se deben revisar los contenidos elaborados como resultado de la tercera iteración de elaboración.
<b>CONCLUSIONES</b>	
	Se han completado los objetivos de la iteración.

30 MINUTOS

TAREAS PARA LA CUARTA ITERACIÓN DE ELABORACIÓN

<b>DISCUSIÓN</b>		
	Se debe decidir cuáles son los siguientes pasos a seguir en esta iteración	
<b>CONCLUSIONES</b>		
	Se debe realizar la memoria y revisar toda la documentación.	
	Se subirá el software desarrollado a una máquina virtual alojada en la universidad.	
	Creación de la web que dará soporte a la presentación del proyecto.	
<b>PLANES DE ACCIÓN</b>	<b>RESPONSABLE</b>	<b>PLAZO</b>
Realizar una primera versión de la memoria	Ioritz Cabero	19/06/2017
Realizar una primera versión de la página web con la documentación	Ioritz Cabero	23/06/2017
Subir el resultado del proyecto a una máquina virtual en la universidad.	Ioritz Cabero	23/06/2017

## Acta de reunión

28/06/2017

9:00

UPV/EHU

<b>TIPO DE REUNIÓN</b>	Reunión de Seguimiento y Organización
<b>ORGANIZADOR</b>	Juan Manuel Pikatza
<b>APUNTADOR</b>	Ioritz Cabero Llama
<b>ASISTENTES</b>	Juan Manuel Pikatza Ioritz Cabero Llama

### Temas del orden del día

1H 30 MINUTOS

REVISIÓN DE LOS CONTENIDOS DE LA MEMORIA

<b>DISCUSIÓN</b>		
Se comentan las revisiones a realizar en la memoria.		
<b>CONCLUSIONES</b>		
Se debe revisar en base a las indicaciones del director.		
Añadir un índice para los anexos de la página web.		
Modificación de la estructura de la web contenedora de la documentación para adecuarla a la norma utilizada.		
<b>PLANES DE ACCIÓN</b>	<b>RESPONSABLE</b>	<b>PLAZO</b>
Realizar una segunda versión de la memoria	Ioritz Cabero	07/07/2017
Realizar una segunda versión de la página web con la documentación	Ioritz Cabero	07/07/2017

## Acta de reunión

14/07/2017

9:00

UPV/EHU

TIPO DE REUNIÓN	Reunión de Seguimiento y Finalización
ORGANIZADOR	Juan Manuel Pikatza
APUNTADOR	Ioritz Cabero Llama
ASISTENTES	Juan Manuel Pikatza Ioritz Cabero Llama

### Temas del orden del día

1H REVISIÓN DE LOS CONTENIDOS DE LA MEMORIA Y LA  
WEB CON LA DOCUMENTACIÓN

DISCUSIÓN	
Se deben analizar todos los entregables realizados.	
CONCLUSIONES	
El contenido de la memoria es adecuado.	
El contenido de la web es adecuado.	
Se han cumplido los objetivos del proyecto y este puede darse por finalizado.	

