

TRABAJO FIN DE MÁSTER

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA DIAGNÓSTICOS MÉDICOS A TRAVÉS DE ANÁLISIS CLÍNICOS BASADO EN ÁRBOLES DE DECISIÓN

Alumno: Torres García, David
Titulación: Máster Universitario en
Ingeniería de
Telecomunicación

Directora:
Ana Eva Ibarrola Armendariz

Curso Académico:
2015/2016

Índice

Resumen.....	5
Lista de tablas.....	8
Lista de ilustraciones.....	9
Lista de acrónimos.....	11
1. Introducción.....	12
2. Objetivos.....	16
3. Alcance.....	17
3.1. Alcance del proyecto.....	17
3.2. Alcance de los resultados del proyecto.....	20
4. Beneficios.....	22
4.1. Beneficios técnicos.....	22
4.2. Beneficios económicos.....	23
4.3. Beneficios sociales.....	24
5. Estado del arte.....	26
5.1. Aspectos relacionados con la medicina.....	26
5.2. Aspectos relacionados con la tecnología.....	28
6. Análisis de alternativas.....	30
6.1. Alternativas para la lógica de programa.....	30
6.2. Alternativas para el almacenamiento de datos.....	33
6.3. Alternativas para la codificación de la información médica.....	36
6.4. Alternativas para el algoritmo diagnóstico.....	38
6.5. Criterios de selección.....	41
6.5.1. Criterios de selección de la lógica de programa.....	41
6.5.2. Criterios de selección del almacenamiento de datos.....	43
6.5.3. Criterios de selección de la codificación de la información médica.....	45
6.5.4. Criterios de selección del algoritmo diagnóstico.....	46
6.6. Selección de la solución.....	48
6.6.1. Selección de la lógica de programa.....	48

6.6.2.	<i>Selección del almacenamiento de datos</i>	49
6.6.3.	<i>Selección de la codificación de la información médica</i>	49
6.6.4.	<i>Selección del algoritmo diagnóstico</i>	50
6.7.	<i>Descripción de la solución</i>	51
7.	<i>Análisis de riesgos</i>	52
8.	<i>Metodología</i>	55
8.1.	<i>Ciclo de vida del proyecto</i>	55
8.2.	<i>Fases del proyecto</i>	59
8.2.1.	<i>Especificación</i>	59
8.2.1.1.	<i>Especificaciones funcionales</i>	59
8.2.1.2.	<i>Especificaciones de gestión y mantenimiento</i>	64
8.2.1.3.	<i>Especificaciones de diseño</i>	64
8.2.2.	<i>Diseño de la arquitectura - Diseño de Alto Nivel</i>	66
8.2.2.1.	<i>Arquitectura general</i>	66
8.2.2.2.	<i>Módulos</i>	70
8.2.2.2.1.	<i>Módulo de almacenamiento de los datos</i>	70
8.2.2.2.2.	<i>Módulo de tratamiento de datos</i>	74
8.2.2.2.3.	<i>Módulo de recolección automatizada de los datos</i>	75
8.2.2.2.4.	<i>Módulo de sistema de cuestionarios</i>	75
8.2.2.2.5.	<i>Módulo de Gestión de Usuarios y Generación de Gráficas y Estadísticos</i> ...	77
8.2.2.2.6.	<i>Módulo de interfaz de usuario</i>	80
8.2.3.	<i>Diseño detallado – Diseño de Bajo Nivel</i>	83
8.2.3.1.	<i>Módulo de almacenamiento de los datos</i>	83
8.2.3.2.	<i>Módulo de tratamiento de datos</i>	91
8.2.3.3.	<i>Módulo de recolección automatizada de los datos</i>	96
8.2.3.4.	<i>Módulo de sistema de cuestionarios</i>	98
8.2.3.5.	<i>Módulo de Gestión de Usuarios y Generación de Gráficas y Estadísticos</i>	100
8.2.3.6.	<i>Módulo de interfaz de usuario</i>	105
8.2.4.	<i>Codificación</i>	110
8.2.5.	<i>Pruebas unitarias</i>	112
8.2.6.	<i>Integración</i>	118

8.2.7. Validación	120
9. Planificación.....	121
9.1. Equipo de trabajo.....	121
9.2. Hitos del Proyecto	121
9.3. Paquetes de trabajo y desglose por tareas	122
9.4. Planificación estimada del proyecto	125
9.5. Diagrama de Gantt del proyecto	126
10. Cálculos.....	127
11. Presupuesto	128
11.1. Precios unitarios	128
11.1.1. Mano de obra.....	128
11.1.2. Material amortizable	128
11.1.3. Material fungible.....	129
11.2. Coste de Recursos Humanos.....	129
11.3. Coste de Recursos Materiales	130
11.4. Resumen final del coste del proyecto.....	131
12. Conclusiones	132
13. Bibliografía	134
Anexo I: Normativa aplicable.....	136

Resumen

Este proyecto tiene como objetivo el diseño e implementación de un sistema que permita inferir posibles diagnósticos médicos mediante datos obtenidos de análisis clínicos. Este sistema pretende estimar posibles diagnósticos a través de analíticas médicas, principalmente utilizando análisis de sangre, para presentarle al usuario información de los parámetros alterados y las posibles causas por las que dichos parámetros pueden sufrir dicha alteración.

El sistema permitirá guardar un histórico de la evolución que sigue el paciente, así como sacar estadísticos de los parámetros alterados o de interés para el paciente, además de dar la posibilidad de hacer un seguimiento de los mismos.

Por otro lado, el sistema posibilitará la captación de los datos para su posterior procesado, tanto de forma manual, es decir introducida por el usuario desde una interfaz, como automatizada, a través de un fichero *JSON* (en formato *FHIR*), permitiendo agilizar las labores de recolección de los datos.

Una vez recibidos los parámetros de las analíticas el sistema utilizará una lógica interna basada en árboles de decisión para mostrar información relacionada con los parámetros alterados y sugerir las posibles causas de dichas alteraciones. Para realizar estos árboles de decisión se hará uso del algoritmo *ID3*.

El sistema propuesto se ha diseñado en base a las especificaciones enunciadas en los estándares *Fast Healthcare Interoperability Resources (FHIR)*, para el intercambio y almacenamiento electrónico de información en salud, *Logical Observation Identifiers Names and Codes (LOINC)*, para la nomenclatura y codificación de los parámetros de salud y, la clasificación internacional de enfermedades en su décima versión (*CIE-10*) para la codificación de patologías publicada por la Organización Mundial de la Salud (*OMS*).

Palabras clave: *FHIR*, *LOINC*, *CIE-10*, *ID3*

Proiektu honen helburua honakoa da: analisi klinikoetatik lortutako datuen bitartez diagnostiko medikoak ondorioztatzea baimentzen duen sistema berri baten diseinu eta inplementazioa. Sistema honek, analisi medikoen bidez, batez ere odol-analisiaren bitartez, diagnostiko posible desberdinak lortzea bilatzen du. Aldatutako parametroen informazioa eta aipaturiko parametroek jasan dezaketen kausa posibleak estimatzea bilatzen du, erabiltzaileari emateko informazio guzti hau.

Sistemak pazientearen eboluzioaren historikoa gordetzea baimenduko du, baita aztoratutako parametroak edo pazientearentzako intereseko datu estatistikoak eta beren segimendua egiteko aukera ematea ere.

Bestalde, sistemak posible egingo du ondoren prozesatuko diren datuak hartzea, bai eskuzko eran (erabiltzaileak berak interfazetik sartutakoak), bai era automatizatuan, *JSON*-a fitxategi baten bitartez (*FHIR*-a formatuan), datuen biltze lanak bizkortzea baimenduz.

Behin analisisien parametroak jasota, sistemak barne-logika erabiliko du. Logika hau erabaki-zuhaitzetan oinarrituko da, aldatutako parametroen informazioa eta hauen aldaketen kausa posibleak iradokitzeke. Erabaki-zuhaitz hauek egiteko *ID3* algoritmoa erabiliko da.

Proposatutako sistema hau ezaugarri estandar desberdinak kontutan harturik diseinatu da. Hala nola, Fast Healthcare Interoperability Resources (*FHIR*) –osasuneko informazioaren truke eta metatze elektronikorako-, Logical Observation Identifiers Names and Codes (*LOINC*) – osasun parametroen izendapen eta kodifikaziorako- eta gaixotasunetako nazioarteko 10.edizioaren sailkapena (*CIE-10*) – Osasunaren Mundu Erakundea (*OME*) argitaratutako patologia desberdinen kodifikaziorako.

Gako-hitzak: *FHIR*, *LOINC*, *CIE-10*, *ID3*

The objective of this project is the design and implementation of a system that allows obtaining medical diagnoses based on data from medical examinations. This system intends to estimate possible diagnoses through medical examinations mainly by using blood tests, to show the user information about altered parameters and the possible reasons why those parameters might change.

This system will make it possible to save a log of the patient's progress as well as getting statistical data from the altered parameters or the ones of interest for the patient. Moreover, a monitoring of all the parameters is also possible.

Furthermore, the system will enable obtaining the information in order to process them later both manually, meaning that the user enters the data by an interface, and automatically with a *JSON* file (in *FHIR* format), speeding up the data collection.

Once all the parameters from the medical examinations have been collected, the system will use an internal logic based on decision trees to show information related to the parameters altered and it will suggest the possible reasons of those alterations. To generate those decision trees, the *ID3* algorithm will be used.

The proposed system has been designed based on the specifications of the standards Fast Healthcare Interoperability Resources (*FHIR*), for the electronic exchange and storage of health information, Logical Observation Identifiers Names and Codes (*LOINC*), for the nomenclature and coding of health parameters and the International Statistical Classification of Diseases and Related Health Problems in its 10th version (*ICD-10*) for the coding of diseases published by the World Health Organization (*WHO*).

Keywords: *FHIR*, *LOINC*, *CIE-10*, *ID3*

Lista de tablas

Tabla 1. Resumen ponderación de la selección de la lógica de programa	42
Tabla 2. Resumen ponderación de la selección del almacenamiento de datos	44
Tabla 3. Resumen ponderación de la selección de la codificación de la información médica.....	46
Tabla 4. Resumen ponderación de la selección del algoritmo diagnóstico.....	47
Tabla 5. Comparativa de alternativas para la lógica de programa	48
Tabla 6. Comparativa de alternativas para el almacenamiento de datos	49
Tabla 7. Comparativa de alternativas para la codificación de la información médica.....	50
Tabla 8. Comparativa de alternativas para el algoritmo diagnóstico.....	50
Tabla 9. Estimación de los riesgos.....	53
Tabla 10. Tablas de la BBDD. Módulo de almacenamiento de datos.....	84
Tabla 11. Tabla Parametros.....	84
Tabla 12. Miembros del equipo de trabajo.....	121
Tabla 13. Hitos del Proyecto	122
Tabla 14. Paquetes de trabajo del Proyecto.....	123
Tabla 15. Desglose de tareas del Proyecto.....	125
Tabla 16. Precio unitario de la mano de obra	128
Tabla 17. Precio unitario del material amortizable.....	129
Tabla 18. Precio unitario del material fungible	129
Tabla 19. Coste de recursos humanos.....	130
Tabla 20. Coste de recursos materiales.....	130

Lista de ilustraciones

Ilustración 1. Grafico prevalencia (%) de problemas crónicos por edad y sexo.....	14
Ilustración 2. Árbol de decisión para diagnosticar anemia.....	27
Ilustración 3. Modelo Lineal	55
Ilustración 4. Modelo Espiral.....	56
Ilustración 5. Modelo Incremental.....	56
Ilustración 6. Modelo en V.....	57
Ilustración 7. Arquitectura del sistema.....	60
Ilustración 8. Arquitectura básica del sistema.....	67
Ilustración 9. Flujo cliente-servidor	67
Ilustración 10. Arquitectura Modelo-Vista-Controlador	69
Ilustración 11. Diagrama entidad-relación de la base de datos del sistema.....	71
Ilustración 12. Diagnóstico de anemia ferropénica en base al algoritmo ID3.....	74
Ilustración 13. Recolección automatizada de datos	75
Ilustración 14. Comunicación entre módulos, módulo de cuestionarios.....	77
Ilustración 15. Funciones principales del sistema.....	81
Ilustración 16. Fragmento de código. Módulo de tratamiento de datos. Obtención de cookies. .91	
Ilustración 17. Fragmento de código. Módulo de tratamiento de datos. Generación del JSON(FHIR).....	92
Ilustración 18. Fragmento de código. Módulo de tratamiento de datos. Parseo y detección de parámetros alterados	93
Ilustración 19. Diagrama de flujo. Módulo de tratamiento de datos.....	94
Ilustración 20. Fragmento de código. Módulo de tratamiento de datos. Diagnóstico realizado. .95	
Ilustración 21. Fragmento de código. Módulo de recolección automatizada de los datos. Subida del fichero.	96
Ilustración 22. Interfaz de usuario. módulo de recolección automatizada de datos.	97
Ilustración 23. Diagrama de flujo. Módulo de recolección automatizada de datos.....	97
Ilustración 24. Módulo de sistema de cuestionario. Interfaz de las preguntas.....	99

Ilustración 25. Módulo de sistema de cuestionario. Resultado del cuestionario.	100
Ilustración 26. Módulo de Gestión de Usuarios y Generación de Gráficas y Estadísticos. Formulario de registro.	101
Ilustración 27. Módulo de Gestión de Usuarios y Generación de Gráficas y Estadísticos. Envío a la capa de datos.	101
Ilustración 28. Fragmento de código de la capa de datos que registra al usuario.	102
Ilustración 29. Generación de los arrays con los valores del parámetro y la fecha.	103
Ilustración 30. Fragmento de código que genera las gráficas.	103
Ilustración 31. Grafica ejemplo obtenida.	104
Ilustración 32. Pantalla de Bienvenida.	106
Ilustración 33. Fragmento de código del slider de bienvenida.	106
Ilustración 34. Interfaz de la analítica.	107
Ilustración 35. Información extra de los parámetros.	107
Ilustración 36. Paginas asociadas.	108
Ilustración 37. Fragmento de código de los servicios.	108
Ilustración 38. Noticias del sector salud.	109
Ilustración 39. Arquitectura formada por 3 capas	111
Ilustración 40. Prueba módulo de recolección automatizada de datos. Ficheros.	113
Ilustración 41. Prueba del módulo de recolección automatizada de datos. Formato.	114
Ilustración 42. Formulario de registro.	115
Ilustración 43. Usuario registrado en la BBDD.	115
Ilustración 44. Acceso simultaneo al módulo de almacenamiento de datos	116
Ilustración 45. Diagrama Gantt del proyecto	126
Ilustración 46. reporte obtenido de la herramienta GTMETRIX.	127

Lista de acrónimos

HL7	Health Level Seven
LOINC	Logical Observation Identifiers Names and Codes
FHIR	Fast Healthcare Interoperability Resources
SW	Software
TIC	Tecnologías de la información y la comunicación
XML	eXtensible Markup Language
JSON	JavaScript Object Notation
RDF	Resource Description Framework
DOM	Document Object Model
SSS	Server-side scripting
JSF	JavaServer Faces
JSP	JavaServer Pages
XUL	XML-based User-interface Language
DML	Data Manipulation Language
DDL	Data Definition Language
MVCC	Multiversion concurrency control
SGBDOO	Sistema gestor de bases de datos orientados a objeto
BBDD	Base de datos
MVC	Modelo Vista Controlador
PL	Presentation Layer
BL	Business Layer
DL	Data Layer
HTML	HyperText Markup Language
CSS	Cascading Style Sheets

1. Introducción

La utilización de las TIC en el mundo de la salud y la medicina ha sido hasta la actualidad, escaso y poco fructífero. Sin embargo, el desarrollo de nuevas tecnologías y el avance en las comunicaciones, unido a la preocupación actual por parte de varios gobiernos de países occidentales por el futuro de la atención sanitaria; ha propiciado que estos dos mundos se unan, con lo que se están llevando a cabo una serie de medidas importantes al respecto. Estas medidas contemplan tanto aspectos de calidad como de gasto en recursos, por lo que abordan desde programas piloto a cambios operativos destinados a hacer más eficiente el sistema sanitario sin mermar la calidad del mismo.

Se ha demostrado que, la combinación de las nuevas tecnologías y las telecomunicaciones en los procesos sanitarios, consigue mejoras no sólo en la prevención de enfermedades, sino también en el seguimiento, actuación médica e incluso recopilación de datos que pueden ser útiles para futuros tratamientos y diagnósticos. Todo ello ha propiciado que se hable profusamente de lo que hoy se conoce como e-Health. **e-Health**^[1] es la clave del cambio en el modelo sanitario que estamos viendo, en el que se prioriza al paciente, haciéndole más consciente de su salud e implicándole en su recuperación, consiguiendo ahorros económicos y reducción de tiempos de espera en los sistemas sanitarios.

En los últimos tiempos, se ha notado un cambio en las prioridades de las personas y el cuidado de la salud ha entrado a formar parte de una de las necesidades básicas más demandadas. Este cambio en las necesidades, sumado a lo que se conoce como *mHealth* (uso de dispositivos móviles y de nuevas tecnologías en el sector de la salud, tanto para pacientes, como para profesionales de la medicina), ha propiciado un empoderamiento del paciente de forma que su recuperación sea más rápida y su seguimiento posterior más efectivo.

El empoderamiento del paciente consiste en hacer más consciente al paciente sobre su estado de salud, así como de los efectos de las enfermedades en su cuerpo. Se consigue que el paciente sea capaz de decidir y tomar decisiones sobre su salud. Este empoderamiento también permite personalizar los tratamientos y adaptarlos al estilo de vida de cada paciente. Con lo que se consigue un nuevo modelo menos paternalista que la medicina tradicional, basado en la participación del paciente. El nuevo paradigma basado en e- Health permite aligerar el sistema sanitario, principalmente gracias al ahorro en los tratamientos y seguimiento de los pacientes crónicos, que son los que más recursos consumen del sistema, debido a sus brotes y recaídas y que, con mayor autocontrol, consiguen reducir la frecuencia de sus dolencias.

Las enfermedades crónicas tienen un peso cada vez mayor en nuestro país, debido en gran medida al aumento de la esperanza de vida y el cambio en los estilos de vida. Además, los costes indirectos de la cronicidad se concentran en la población en edad de trabajar. Un 45% de la población española mayor de 16 años padece al menos un proceso crónico. Las enfermedades crónicas afectan a más de 19 millones de personas en España, de los cuales más de 11 millones son mujeres; estas enfermedades se centran principalmente en mayores de 55 años y aumenta con la edad. A continuación se muestra una gráfica extraída de la publicación *“Estrategia para el Abordaje de la Cronicidad en el Sistema Nacional de Salud”*^[2] del Ministerio de sanidad, servicios sociales e igualdad, del año 2012, que desglosa la prevalencia de problemas crónicas por edad y sexo.

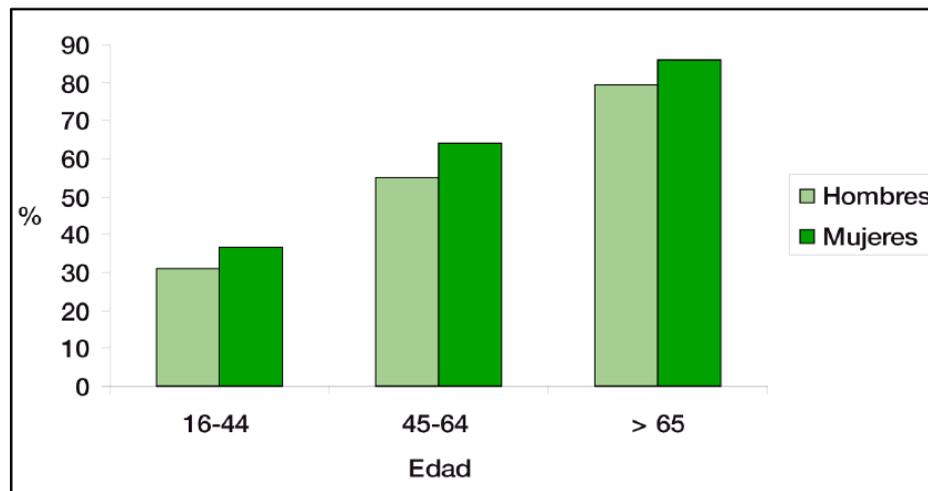


Ilustración 1. Gráfico prevalencia (%) de problemas crónicos por edad y sexo

Otro dato importante a tener en cuenta, extraído de la publicación anterior, es que las enfermedades crónicas son la causa del 80% de las consultas de Atención Primaria. Según los datos de la EES 2009, el 47% de las personas mayores de 65 años acudió a una consulta médica en las cuatro semanas previas a la encuesta. Según datos del informe *Soluciones para la gestión de la cronicidad*, elaborado por la Sociedad Española de Directivos de la Salud (SEDISA), el 80% del gasto sanitario en España se dedica a atender las enfermedades crónicas: cardiovasculares, diabetes, cáncer, dolencias psiquiátricas, pulmonares, etc. Por todo esto, estas enfermedades, repercuten en un aumento del gasto de los servicios sanitarios y de las familias. Además, la esperanza de vida aumenta, y por tanto, convivimos más años con enfermedades que requieren una serie de controles y cuidados periódicos con su correspondiente gasto asociado.

Otra cuestión importante y que está de actualidad, es el tratamiento y transmisión de datos referentes a salud. En el ámbito del tratamiento de datos de salud no existe, actualmente, un gran número de estándares mundialmente aceptados, y uno de los más importantes que, entre otras compañías, va adoptar la empresa Apple en sus futuros dispositivos, y en el que también se apoyará este proyecto para el almacenamiento y tratamiento de los datos, es el estándar FHIR

(Fast Healthcare Interoperability Resources). *FHIR*^[3] es un marco de estándares de última generación creado por *HL7*^[4] (Health Level Seven) que proporcionan un conjunto de estándares para facilitar el intercambio electrónico de información clínica.

En este contexto es donde surge este proyecto, y más concretamente, en el mundo de la salud electrónica o e-Health. Este proyecto persigue cubrir una necesidad actual, reducir el gasto en salud pública, principalmente para enfermos con patologías crónicas. El objetivo principal es facilitar un diagnóstico precoz y sencillo a partir de análisis clínicos.

La lógica interna utilizará el algoritmo ID3, un algoritmo que es muy utilizado en el ámbito de la medicina y de la inteligencia artificial y que se basa en árboles de decisión para la obtención de un resultado binario (positivo o negativo).

La obtención de los datos para su posterior procesado se puede realizar de forma manual por parte del usuario (a través de la interfaz que se le proporciona), o través de un sistema automatizado de recolección de datos que facilita obtención de dichos datos, para su posterior tratamiento. Para obtener estos datos de forma automatizada el sistema se apoya en el estándar FHIR, que junto con la codificación de los parámetros siguiendo el estándar LOINC^[5] (estándar que ha surgido especialmente para la codificación de resultados de laboratorio), facilita en gran medida la interoperabilidad del sistema, por una parte, consigo mismo, es decir, con posibles ampliaciones futuras del mismo, mediante la creación de nuevo módulos que le proporcionen nuevas funcionalidades y, por otra parte, con otras aplicaciones/sistemas que puedan hacer uso de dichos estándares.

2. Objetivos

El objetivo principal de este proyecto es el diseño e implementación de un sistema que realice posibles diagnósticos médicos basándose en análisis clínicos por medio de árboles de decisión. Este sistema estará formado por una serie de módulos que tienen como objetivo agilizar tanto la recolección de los datos que llegan desde los usuarios como el posterior procesamiento de los mismos para obtener y presentarle al usuario un posible diagnóstico de una forma rápida y clara. Esta captura de los datos se podrá realizar tanto de forma manual, es decir, los datos serán introducidos por el usuario a través de un formulario presentado en la interfaz, como automatizada, en la que los datos serán cargados a través de un fichero siguiendo un formato estandarizado.

El sistema pretende, además, proporcionar un sistema de backup de análisis realizados tanto para pacientes, de forma que estos puedan guardar sus análisis para seguir una evolución de los mismos y obtener datos estadísticos de los parámetros de interés para el tratamiento de sus patologías, como para médicos y clínicas que quieran realizar un histórico de los análisis de todos los pacientes tratados. Esto les posibilitaría poder visualizar la información de un determinado paciente de una forma rápida, estructurada y sencilla.

Como objetivo final, el sistema permitirá realizar labores de Big Data de tal forma que permita simular patrones de variabilidad de parámetros en sangre. De esta forma posibilitará en futuros estudios médicos, comprobar si la relación entre la existencia de uno o más parámetros en cantidades alteradas en la sangre puede ser debido a la existencia de una determinada enfermedad o sigue patrones de edad, sexo o raza.

3. Alcance

Este apartado se dividirá a su vez en dos secciones: el alcance del proyecto y alcance de los resultados del proyecto:

3.1. Alcance del proyecto

El objetivo global del proyecto consiste en el diseño e implementación de un sistema que permita obtener diagnósticos médicos a través de análisis clínicos utilizando árboles de decisión. El sistema además proporcionará un interfaz web intuitiva y fácil de manejar, de forma que los usuarios puedan obtener información sobre su estado de salud de forma rápida y comprensible. Dicho objetivo global se desglosará a su vez en una serie de objetivos específicos:

➤ Diseño de la arquitectura general del sistema

Para el diseño de la arquitectura general del sistema, se ha subdividido éste en una serie de módulos, en concreto el sistema está compuesto por 6 módulos principales. El módulo de almacenamiento de datos, el módulo de tratamiento de datos, el módulo de generación de encuestas, el módulo interfaz de usuario, el módulo de recolección automatizada de datos y el módulo de gestión de usuarios y generación de gráficas y estadísticos. A continuación se detallan las funciones principales de cada uno de ellos.

➤ Diseño del módulo de almacenamiento de los datos

Este módulo permitirá el almacenamiento de los datos y facilitará el posterior procesado de los mismos. Para diseñarlo de forma adecuada, será necesario tener en cuenta el tipo de datos que se van a tratar y la sensibilidad de los mismos, puesto que son datos médicos referentes a la salud de las personas. Por tanto, la seguridad

del sistema será un aspecto fundamental a tener en cuenta de cara a garantizar que el tratamiento y almacenamiento de los datos cumple con los estándares de protección de datos^[6] (extracción de la LOPD referente al tratamiento de datos de salud en el anexo 4).

➤ **Diseño del módulo de tratamiento de datos**

Este módulo permitirá el tratamiento ágil de los datos utilizando para ello, el algoritmo de decisión ID3, llegando a un diagnóstico lo más cercano posible a la realidad. Este módulo estará dividido en dos submódulos, el primer submódulo se encargará del tratamiento de los datos introducidos por el usuario, de forma manual o automatizada (haciendo uso del módulo de recolección automatizada de datos), sobre los parámetros de interés en su análisis, llegando a un primer diagnóstico que únicamente tenga en cuenta las sustancias que se encuentran en demasía o escasez en su organismo y el segundo submódulo se encargará de tratar los datos que le proporcione el módulo de generación de cuestionarios para afinar el resultado y obtener un diagnóstico más ajustado.

➤ **Diseño del módulo de recolección automatizada de los datos**

Este módulo permitirá la recolección de los datos de una forma automatizada siguiendo una estructura de datos que cumpla con los estándares de salud referentes al almacenamiento, intercambio y gestión de información médica.

➤ **Diseño del módulo de sistema de cuestionarios**

Se desarrollará un sistema de cuestionarios que permita, por una parte, afinar los resultados de los diagnósticos realizados por el sistema para obtener un resultado lo más cercano posible a la realidad, y por otra, sirva como soporte para aportar valor añadido a las labores de Big Data.

➤ **Diseño del módulo de Gestión de Usuarios y Generación de Gráficas y Estadísticos**

Este módulo permitirá, por una parte, gestionar si los usuarios están registrados o no, ofreciendo en cada caso unos servicios u otros, además de gestionar el registro de estos, y por otra parte, permitirá generar estadísticos y gráficas, posibilitando el realizar un seguimiento de la evolución de los parámetros de interés de una forma visual y sencilla. El sistema permitirá inicialmente, escoger uno o varios parámetros y a partir de ellos, creará una gráfica de puntos con los valores que han tenido en los sucesivos análisis del paciente.

➤ **Diseño de la arquitectura de la información**

Un paso previo al diseño de la interfaz de usuario, es el diseño de la arquitectura de la información, que sin ser en si un módulo del sistema, es importante para el correcto diseño de la interfaz de usuario. Para el diseño de la arquitectura de la información hay que tener en cuenta que el diseño estará centrado o enfocado en los usuarios. Este diseño se basará en las necesidades del usuario haciendo énfasis en la usabilidad y comprensión del sistema por parte de los usuarios. Durante el diseño y desarrollo de este módulo también se evitará la infoxicación de los usuarios, es decir, sobresaturar a estos con demasiada información, ya que de otra forma el sistema se volvería complicado para el uso por parte de los usuarios sin demasiados conocimientos técnicos o de medicina. También en esta parte del proyecto se tendrá estudiar, organizar y rotular el sitio web que alojará el sistema de tal forma que sea usable, accesible y encontrable por los usuarios.

➤ **Diseño del módulo de interfaz de usuario**

El diseño de la interfaz web, como se ha comentado anteriormente, está íntimamente relacionado con el diseño de la arquitectura de la información. Una vez diseñada esta arquitectura, es decir, posicionados los elementos gráficos que tendrá la interfaz gráfica de la herramienta, lo siguiente será diseñar y construir la forma en la que se visualizarán estos elementos. Se pretende que la interfaz sea muy sencilla de tal forma que pueda ser utilizada tanto por personas con conocimientos médicos, como sin estos conocimientos. Además, la interfaz también se diseñará para que sea *responsive*, es decir, que se amolde a cualquier tipo de dispositivo o de pantalla (*Smartphone, Tablet*, ordenadores de sobremesa, etc.) sin que se vea perjudicada su sencillez y claridad.

3.2. *Alcance de los resultados del proyecto*

En el apartado anterior se ha descrito el alcance global del proyecto. En este apartado se describirá cual es el alcance del trabajo y de qué forma permitirá la consecución de los objetivos planteados.

En cuanto al alcance de los resultados del proyecto se pueden marcar los siguientes objetivos:

- Interfaz de usuario sencilla e intuitiva: Se pretende que la interfaz de usuario sea lo más clara e intuitiva posible de tal forma que cualquier tipo de perfil de usuario pueda localizar de manera rápida que desea realizar en el sistema.
- Formato de almacenamiento de los datos: Se pretende que los datos sean almacenados siguiendo un estándar internacional para el tratamiento y almacenamiento de datos médicos, y que permita al

sistema interactuar con otros desarrollos (se analizará en el análisis de alternativas).

- Sistema escalable: El sistema estará diseñado de tal forma que la incorporación de un parámetro nuevo (ya sea de análisis de sangre, orina o cualquier otra prueba diagnóstica), no varíe la estructura del sistema y de forma automatizada sea seleccionable y procesable por el mismo.
- Sistema eficiente: Se pretende además que el sistema en global, así como cada uno de los módulos de los que está compuesto, hagan un uso eficiente de los recursos de que disponen.

Logrando los objetivos anteriormente descritos se conseguirá un sistema que proporcionará tanto a usuarios como a médicos, por un lado, agilizar los diagnósticos médicos para en caso de sufrir alguna patología comenzar con el tratamiento lo antes posible y por otro un sistema de almacenamiento y generación de estadísticos para los análisis clínicos de los pacientes.

4. Beneficios

Este proyecto consiste en el diseño e implementación de un sistema para diagnósticos médicos a través de análisis clínicos basada en árboles de decisión. En este apartado se analizarán los beneficios del mismo desde tres puntos de vista:

1. Beneficios técnicos
2. Beneficios económicos
3. Beneficios sociales

4.1. Beneficios técnicos

Como beneficios técnicos cabe resaltar que el sistema permitirá dotar a los pacientes con suficiente información sobre su estado de salud para que sean capaces de tomar sus propias decisiones, cara a un seguimiento más adecuado sobre su tratamiento y la capacidad de acelerar su mejoría.

El tratamiento y almacenamiento de los datos se realizará en un formato estándar creado para el intercambio y tratamiento de información de salud entre diferentes plataformas, como es el formato *FHIR* y más concretamente utilizando el formato de codificación *JSON* (el estándar define que la estructuración de los datos puede ser a través de *XML*, *JSON* o mediante un grafo *RDF*). El usar este estándar facilitará en gran medida la interconexión del sistema con otras posibles herramientas que también hagan uso de dicho estándar y su uso en herramientas de análisis de *BIG DATA*.

Se persigue que el sistema sea escalable de tal forma que permita la incorporación de nuevos parámetros de forma automatizada con lo que siempre será posible su actualización evitando que quede desfasado. El diseño se ha realizado de tal forma que el sistema siga una estructura modular, con lo que se

permite la ampliación de los diferentes módulos del sistema o la incorporación de nuevos módulos, sin que ello repercuta de forma negativa en el funcionamiento global del mismo.

Por último, la estructura del sistema que se diseñará e implementará para este proyecto se centrará básicamente en parámetros existentes en la sangre, pero gracias al diseño flexible del sistema, en futuras actualizaciones se podrían incorporar parámetros existentes en cualquier tipo de análisis (orina, médula, etc.) de forma automatizada.

4.2. Beneficios económicos

Este sistema proporciona grandes beneficios económicos, ya que persigue reducir en gran medida el gasto en salud pública, principalmente para enfermos con patologías crónicas, que son los que mayor gasto suponen a la sanidad pública. Este objetivo se pretende conseguir evitando que dichos pacientes tengan que acudir a las consultas simplemente para llevar un seguimiento de su enfermedad.

El coste derivado del mantenimiento del sistema, gracias a la modularidad y flexibilidad de este, será reducido, ya que, tras la integración de todos los módulos, el seguimiento y control del sistema se podrá hacer de una forma centralizada.

El hecho de ser un sistema flexible, no solo reporta los beneficios técnicos anteriormente citados. También permitirá que las ampliaciones o posibles modificaciones futuras se realicen de forma rápida y sencilla, de manera que se reduzcan las horas de trabajo necesarias, con el consiguiente ahorro económico.

Por último, hay que considerar los beneficios que reportará a la empresa una vez finalizado el desarrollo del sistema. Por una parte, se pretende aplicar técnicas de SEO de forma que el posicionamiento en buscadores sea el adecuado, con lo que los beneficios provendrían principalmente de publicidad digital y, por otra parte,

en futuras ampliaciones se pretende ofrecer un servicio Premium con mayor número de opciones a disposición del paciente (como poder ser, disponer de la herramienta sin publicidad, posibilidad de consultar a un médico online, etc.), para aquellos que se abonen a este servicio pagando una cuota mensual.

4.3. Beneficios sociales

Además de los beneficios técnicos y económicos mencionados anteriormente, este proyecto cuenta también con una serie de beneficios sociales que se detallarán a continuación.

La creación de este sistema facilitará la vida en gran medida a los pacientes que sufren enfermedades crónicas, evitándoles los desplazamientos necesarios para llegar a la consulta cuando lo único que pretenden es estar informados sobre su evolución. Además, el hecho de proporcionarles una interfaz intuitiva y sencilla no solo pensando en la gestión o administración del sistema, sino también en los usuarios, favorece el que cualquier tipo de usuario, tenga el perfil que tenga, pueda utilizar la plataforma sin mayores dificultades. Con este proyecto también se pretende incidir en el empoderamiento de los pacientes, es decir, dotarles de toda la información necesaria para que tengan un tratamiento y una evolución lo más favorables posible.

Además de esto, los médicos también se podrán beneficiar en gran medida del sistema, pudiendo obtener históricos de la evolución de sus pacientes, gestionar de forma centralizada toda la información y obtener las estadísticas sobre los parámetros de interés para el seguimiento de dichos pacientes. Con el presente proyecto también se obtienen grandes beneficios para la salud pública de tal forma que se puedan controlar la evolución de determinadas enfermedades en los pacientes y si un determinado fármaco o tratamiento es más o menos efectivo para

la evolución de la misma (controlando como afectan a la sangre la ingesta de dichos fármacos).

Cabe destacar que la ejecución de este proyecto aporta una intensa formación en diferentes tecnologías y técnicas de gran relevancia en la actualidad como son PHP, JSON, SEO (On-page, Off-page, linkbuilding), Usabilidad, marketing digital.

5. Estado del arte

Para la elaboración de este apartado se han dividido los estudios previos realizados bajo dos temáticas bien diferenciadas:

1. Aspectos relacionados con la medicina.
2. Aspectos relacionados con la tecnología.

5.1. Aspectos relacionados con la medicina

La búsqueda de información se centró inicialmente en términos relacionados con análisis clínicos y los parámetros asociados a este tipo de exámenes. El objetivo de esta búsqueda era localizar, primeramente, si existía algún estándar relacionado con los análisis que se realizan en los laboratorios médicos y que parámetros miden comúnmente estos análisis y, verificar la relación que hay entre las patologías que sufren los pacientes con la variación de dichos parámetros.

Para esta búsqueda se recurrió principalmente, que no exclusivamente, a la bibliografía que se explica a continuación:

- **Hemograma – Manual de Interpretación: Renato Failace y colaboradores**

Un hemograma, también conocido con el nombre de conteo sanguíneo completo (CBC; por sus siglas en inglés), es una prueba para contabilizar la cantidad de células sanguíneas. Existen tres tipos de células en la sangre: glóbulos rojos, glóbulos blancos y plaquetas.

En este libro se realiza un estudio hematológico completo, que representa uno de los elementos básicos en la evaluación diagnóstica de los pacientes. Cabe

destacar, que este libro muestra una interpretación básica de un análisis clínico basándose únicamente en la serie roja, la serie blanca y la serie plaquetar. Además, aporta valores de referencia para un hemograma, clasificando estos datos en función del sexo, la raza y la edad.

- Manual Práctico de Hematología Clínica: Miguel, A. Sanz Alonso, Enric Cabrerias I Pons

En este libro, se clasifican una gran variedad de patologías, desde la descripción de cada enfermedad, a cómo se puede llegar a un posible diagnóstico, a través de un hemograma. Centrándose en los resultados, el autor muestra una serie de árboles de decisión ejemplo, que pueden ayudar a llegar a un veredicto acertado si tenemos en cuenta la variación de los parámetros en sangre. A continuación se muestra uno de los árboles de decisión extraído de este libro, de cómo se puede llegar a diagnosticar una anemia, a modo de ejemplo:

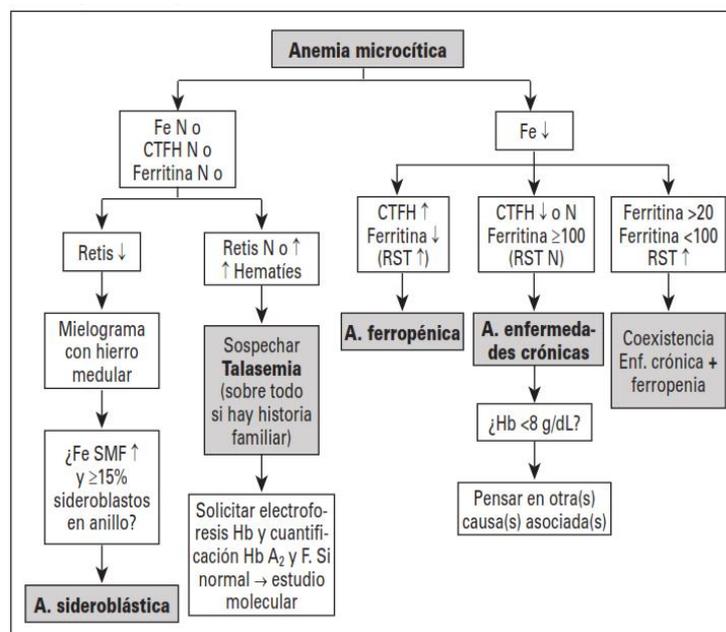


Ilustración 2. Árbol de decisión para diagnosticar anemia

- *Carencias en el apartado médico*

Tras la búsqueda en esta y otra bibliografía, se determinó que no existía ningún patrón común a la hora de realizar análisis por parte de los laboratorios médicos y que cada laboratorio escogía que parámetros medir y que valores de referencia escoger para determinar si un parámetro estaba alterado o no.

5.2. Aspectos relacionados con la tecnología

En este apartado, la búsqueda se centró principalmente en la localizar que herramientas había disponibles en el mercado que tuvieran relación con el diagnóstico médico y con las analíticas. Se presenta a continuación un listado de algunas herramientas existentes en la actualidad que tienen relación con el diagnóstico médico, las analíticas y las enfermedades en general:

- *GESMED*^[7]: es una herramienta que permite a médicos y clínicas realizar entre otras funciones, gestiones médicas con informes e historias, reservar citas online y otro tipo de gestiones médicas, como facturación a mutuas o pacientes, etc.
- *MEDLINE*^[8]: es una base de datos de bibliografía médica producida por la Biblioteca Nacional de Medicina de los Estados Unidos. Contiene alrededor de 15 millones de artículos de aproximadamente 5.000 publicaciones, cubriendo las áreas de biomedicina y salud desde 1950.
- *SYMCA*^[9]: es una aplicación que permite a los usuarios discernir entre varias enfermedades al ingresar síntomas como fiebre, tos, dolor e inflamación, y recibir un diagnóstico instantáneo. Básicamente, no es más que un generador de diagnósticos (más bien probabilidades de diagnósticos) a partir de ciertos datos que va introduciendo el usuario en el programa, como sexo, edad, síntomas, antecedentes personales, etc.

- MEDIKTOR^[10]: es una herramienta interactiva, capaz de analizar los síntomas y evaluar el estado de salud del usuario. Además, permite chatear con un médico en tiempo real.

En cuanto al almacenamiento y tratamiento de los datos médicos, en los últimos días ha surgido una noticia que indicaba que importantes empresas, la empresa Apple entre ellas, van a adoptar en sus futuros modelos de *Smartphones* y *Smartwatches* el estándar FHIR para el tratamiento e intercambio de información médica, con el objetivo de garantizar la interoperabilidad de sus dispositivos con las herramientas medicas existentes en el mercado.

- Carencias en el apartado tecnológico

Si bien se localizó un gran número de herramientas que daban todo tipo de información sobre patologías y enfermedades, o diagnósticos en base a enfermedades ya existentes en los pacientes, no se encontró ninguna que realizara diagnósticos médicos basados en la alteración de parámetros que se miden en los análisis clínicos.

6. Análisis de alternativas

Para conseguir los objetivos que se han definido en apartados anteriores es necesario tener en cuenta una serie de alternativas, así como fijar unos criterios para la elección de cada una de ellas. A continuación se presenta el análisis de alternativas desarrollado para la realización del proyecto. Este análisis se centra en varias cuestiones:

- Lógica de programa
- Almacenamiento de datos
- Codificación de la información médica
- Algoritmo diagnóstico

La principal decisión corresponde a la lógica de programa y el almacenamiento de los datos. Similar peso adquiere la elección del algoritmo diagnóstico ya que será el que de fluidez al sistema a la hora de obtener un diagnóstico de forma ágil.

6.1. Alternativas para la lógica de programa

En cuanto a la lógica de programa, se han analizado una serie de alternativas como son: ASP.NET, PHP y JSF con JSP.

Alternativa 1: ASP.NET

ASP.NET^[11] es un framework para aplicaciones web desarrollado y comercializado por Microsoft. ASP.NET es un modelo de desarrollo Web unificado que incluye los servicios necesarios para crear aplicaciones Web empresariales con el código mínimo. Forma parte de .NET y al codificar las aplicaciones ASP.NET tiene acceso a las clases de .NET. El código de las

aplicaciones puede escribirse en cualquier lenguaje compatible con el Common Language Runtime (CLR), entre ellos Microsoft Visual Basic, C#, JScript .NET y J#.

Las ventajas de esta alternativa son:

- El framework se encarga de detectar el tipo de navegador utilizado por el cliente y determinar la versión HTML que soporta.
- Se puede almacenar en la caché del servidor tanto páginas enteras, como controles personalizados o simples variables, reduciendo enormemente el consumo de recursos.

Las desventajas de esta alternativa son:

- Control limitado sobre HTML. Los ID son generados por el servidor, con lo que se complica el acceso a elementos del DOM a través de javascript.
- No es adecuado para las pruebas unitarias.
- No es fácil de leer e interpretar y por tanto es complejo de implementar.

Alternativa 2: PHP

PHP^[12] es un lenguaje de programación de uso general de código del lado del servidor (SSS). Originalmente estaba diseñado para el desarrollo web de contenido dinámico. También incluye un CLI que puede ser usado en aplicaciones gráficas independientes.

Las ventajas de esta alternativa son:

- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad.

- Capacidad de expandir su potencial utilizando una enorme cantidad de módulos.
- Puede utilizarse para generar módulos binarios CGI.
- Permite aplicar técnicas de programación orientada a objetos.

Las desventajas de esta alternativa son:

- Configuración complicada para evitar brechas de seguridad.
- Legibilidad del código un poco complicada al mezclarse sentencias HTML y PHP.
- Todo el trabajo lo realiza el navegador por tanto puede ser ineficiente cuando las solicitudes son muy elevadas.

Alternativa 3: JSF con JSP

JavaServer Faces (JSF)^[13] es un framework para aplicaciones Java basadas en web enfocado al desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JavaServer Pages (JSP)^[14] como la tecnología que permite hacer el despliegue de las páginas, pero también se puede utilizar otras tecnologías como XUL.

Las ventajas de esta alternativa son:

- Facilita que el código esté bien estructurado.
- Dispone de una gran cantidad de funciones y librerías.

Las desventajas de esta alternativa son:

- La codificación de interfaces elaboradas conlleva muchas líneas de código.
- Su naturaleza como estándar hace que la evolución de JSF no sea tan rápida como pueda ser la de otros entornos.

6.2. Alternativas para el almacenamiento de datos

En cuanto al almacenamiento de datos, se han analizado una serie de alternativas como son: SQL Server, MySQL, Oracle y PostgreSQL.

Alternativa 1: SQL Server

SQL Server^[15] es un sistema de bases de datos del modelo relacional, desarrollado por la empresa Microsoft. El lenguaje de desarrollo utilizado es Transact-SQL (TSQL), una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos (DML), crear tablas y definir relaciones entre ellas (DDL).

Las ventajas de esta alternativa son:

- Seguridad alta ya que permite gestionar los permisos independientemente (permisos a nivel de servidor, a nivel de tablas, etc.).
- Permite la agregación de otros servidores SQL Server para realizar una compartición de datos.

Las desventajas de esta alternativa son:

- Uso excesivo de recursos.
- Bloqueos a nivel de página altos.
- Tamaños de página fijos y pequeños.

Alternativa 2: MySQL

MySQL^[16] es un sistema de gestión de base de datos relacional (RDBMS) de código abierto, basado en lenguaje de consulta estructurado (SQL). A pesar de que se puede utilizar en una amplia gama de aplicaciones, MySQL se asocia más con las aplicaciones basadas en web y es un componente importante de una pila empresarial de código abierto llamado LAMP (Linux – Apache – MySQL - PHP).

Las ventajas de esta alternativa son:

- Sin límites en los tamaños de los registros.
- Gran cantidad de tipos de datos.
- Posibilidad de crear y configurar usuarios, asignando a cada uno de ellos permisos diferentes.
- Facilidad de exportación e importación de datos, incluso de la base de datos completa.

Las desventajas de esta alternativa son:

- Los privilegios de una tabla no se eliminan automáticamente al borrar la tabla (es necesario usar el comando REVOKE).
- Las Foreign Key no las trata de forma diferente al resto de los campos.

Alternativa 3: Oracle

Oracle Database es un sistema de gestión de base de datos de tipo objeto-relacional. Oracle es básicamente una herramienta cliente/servidor para la gestión de base de datos. Desarrollado sobre Oracle Database ha sido diseñada para que las organizaciones puedan controlar y gestionar grandes volúmenes de datos no

estructurados en un único repositorio con el objetivo de reducir los costes y los riesgos asociados a la pérdida de información.

Las ventajas de esta alternativa son:

- Permite el uso de particiones para la mejora de la eficiencia.
- Multiplataforma.
- Dispone de una integridad referencial declarativa bastante potente.

Las desventajas de esta alternativa son:

- Configuración bastante tediosa.
- Alta curva de aprendizaje.
- Coste muy elevado.

Alternativa 4: PostgreSQL

PostgreSQL^[17] es un gestor de bases de datos orientadas a objetos (SGBDOO) muy conocido y usado en entornos de software libre porque cumple los estándares SQL92 y SQL99, y también por el conjunto de funcionalidades avanzadas que soporta, lo que lo sitúa al mismo o a un mejor nivel que muchos SGBD comerciales.

Las ventajas de esta alternativa son:

- Diseñado para ambientes de alto volumen utilizando una estrategia de almacenamiento de filas llamada MVCC (Multiversion concurrency control), consigue mejor respuesta en grandes volúmenes.
- Dispone de múltiples herramientas gráficas de diseño y administración de bases de datos.
- Buen sistema de seguridad mediante la gestión de usuarios, grupos de usuarios y contraseñas.

- Gran capacidad de almacenamiento.

Las desventajas de esta alternativa son:

- Lento en inserción y actualización de datos.
- Consumo alto de recursos.
- Sentencias y comandos nada intuitivos.

6.3. *Alternativas para la codificación de la información médica*

En cuanto a la codificación de la información médica, se han analizado una serie de alternativas como son: SNOMED-CT, LOINC y CIE-10.

Alternativa 1: SNOMED-CT

SNOMED-CT^[18] (Systematized Nomenclature of Medicine – Clinical Terms) proporciona todos los conceptos que alguna vez se hayan expresado en el dominio de la medicina sin ambigüedad. Cubre todo el espectro del dominio de la salud gracias a sus más de 300.000 conceptos, junto con la capacidad de combinarlos y relacionarlos. Esta terminología es mantenida y distribuida por la International Health Terminology Standards Development Organisation (IHTSDO).

Las ventajas de esta alternativa son:

- Cubre prácticamente todo el espectro de terminología médica.
- Estándar ampliamente usado.
- Permite conceptos que son construidos a través de la combinación de varios conceptos.

Las desventajas de esta alternativa son:

- Dificultades cuando se tienen que combinar varias codificaciones para obtener una definición.
- Baja consistencia cuando se necesita distinguir entre conceptos similares, con lo que disminuye la velocidad de decisión.

Alternativa 2: LOINC

LOINC (Logical Observation Identifiers Names and Codes) proporciona una clasificación completa de observaciones clínicas en medicina. Se utiliza especialmente para codificar resultados de laboratorio. Desde su creación ha evolucionado y actualmente se puede utilizar también para codificar exámenes físicos u otras observaciones clínicas.

Las ventajas de esta alternativa son:

- Consistencia alta para codificar un determinado concepto.
- Gran cantidad de términos sin la necesidad de combinarlos.
- Compatibilidad alta con otros estándares relacionados con la medicina.

Las desventajas de esta alternativa son:

- No permite combinar codificaciones para obtener una definición que no esté definida en el estándar.

Alternativa 3: CIE-10

CIE-10^[7] (Clasificación Internacional Estadística de Enfermedades y Problemas de Salud Relacionados) proporciona una codificación completa para las enfermedades, así como para una amplia variedad de signos, síntomas, hallazgos anormales, denuncias, circunstancias sociales y causas externas de daños o enfermedad.

Las ventajas de esta alternativa son:

- Amplia aceptación en el mundo de la medicina.
- Precisión y adaptación al estado del arte actual en el ámbito clínico.

Las desventajas de esta alternativa son:

- Limitados a enfermedades, patologías y síntomas (no codifica parámetros y sustancias medicas).
- En la actualidad conviven varias versiones de CIE con diferentes nomenclaturas para un mismo parámetro.

6.4. Alternativas para el algoritmo diagnóstico

En cuanto al algoritmo diagnóstico, se han analizado dos alternativas como son: el algoritmo C4.5 y el algoritmo ID3, ambos algoritmos basados en la generación de árboles de decisión.

Alternativa 1: Algoritmo C4.5

El algoritmo C4.5^[8] genera un árbol de decisión a partir de los datos mediante particiones realizadas recursivamente. El algoritmo considera todas las pruebas posibles que pueden dividir el conjunto de datos y selecciona la prueba que resulta

en la mayor ganancia de información. Para cada atributo discreto, se considera una prueba con n resultados, siendo n el número de valores posibles que puede tomar el atributo. Para cada atributo continuo, se realiza una prueba binaria sobre cada uno de los valores que toma el atributo en los datos. En cada nodo, el sistema debe decidir cuál prueba escoge para dividir los datos.

Las ventajas de esta alternativa son:

- Permite manejar atributos continuos y discretos.
- Permite en manejo de atributos con diferentes valores (no solo con soluciones binarias).
- Permite el entrenamiento con valores faltantes.
- Se pueden utilizar técnicas de “poda” para obtener árboles mas simplificados.

Las desventajas de esta alternativa son:

- Puede llegar a ser lento en la convergencia a un resultado, ya que se generan árboles de grandes dimensiones.
- Favorece indirectamente a aquellos atributos con muchos valores.

Alternativa 2: Algoritmo ID3

El algoritmo ID3^[9] permite construir un árbol de decisión que explique cada instancia de la secuencia de entrada de la manera más compacta posible a partir de una tabla de inducción. En cada momento elige el mejor atributo dependiendo de una determinada heurística. Los árboles de decisión que forma el algoritmo ID3 están formados por nodos (nombres o identificadores de los atributos), ramas (posibles valores del atributo asociado al nodo), hojas (conjuntos ya clasificados de ejemplos y etiquetados con el nombre de una clase) y como datos están los

atributos (factores que influyen la clasificación o decisión). ID3 examina todos los atributos y escoge el de máxima ganancia, forma la ramificación y usa el mismo proceso recursivamente para formar sub-árboles a partir de los v nodos generados.

Las ventajas de esta alternativa son:

- Algoritmo rápido en converger a una solución.
- Generalmente se generan árboles pequeños.
- El conjunto de datos forma un conjunto de reglas fácilmente entendibles.
- Implementación sencilla.

Las desventajas de esta alternativa son:

- Solo se comprueba un atributo en cada paso.
- Clasificar datos continuos puede ser costoso computacionalmente hablando.

6.5. Criterios de selección

6.5.1. Criterios de selección de la lógica de programa

En cuanto a la lógica de programa utilizada, los factores más importantes que se han tenido en cuenta para elegir una u otra opción han sido:

1. Rendimiento (30%):

Este criterio mide la eficiencia en cuanto al manejo de los recursos por parte del servidor que alojará la aplicación. Cuanto mejor sea la eficiencia en cuanto al uso de los recursos, menores serán los tiempos de respuesta y mayor la capacidad del servidor para manejar los datos, lo que se traducirá en una mejor experiencia del por parte usuario a la hora de utilizar el sistema.

2. Escalabilidad (20%):

Este criterio valora la capacidad de crecimiento y ampliación, sin que la funcionalidad se vea afectada negativamente. Dada la continua evolución que sufren las aplicaciones durante su vida útil, es importante que las modificaciones o posibles ampliaciones de la misma se realicen de forma sencilla, ejerciendo el menor impacto sobre el funcionamiento del sistema.

3. Facilidad de programación (15%):

Este criterio valora la facilidad con la que se realiza la implementación del sistema, además se valorará positivamente aquella solución cuya curva de aprendizaje sea menor, reduciendo el tiempo necesario para el desarrollo del sistema.

4. Presentación (15%):

Este criterio valora las funcionalidades y características que ofrece el nivel de presentación de cada alternativa. Se valorará por una parte el número de funcionalidades útiles para el sistema y, por otra parte, la facilidad para la separación entre la capa de presentación y el resto de capas.

5. Licencia (15%):

Este criterio valora el tipo de licencia que dispone cada una de las alternativas. Se valorará el coste económico de la licencia, así como la libertad que otorga cada una de ellas.

6. Soporte (5%):

Este criterio valora el mantenimiento actual de que dispone cada una de las soluciones. Se valorará positivamente la que disponga de un soporte actual mayor, teniendo en cuenta también, el número de plataformas en las que se puede implementar la solución.

CRITERIOS	PESO
1. Rendimiento	30%
2. Escalabilidad	20%
3. Facilidad de programación	15%
4. Presentación	15%
5. Licencia	15%
6. Soporte	5%

Tabla 1. Resumen ponderación de la selección de la lógica de programa

6.5.2. *Criterios de selección del almacenamiento de datos*

En cuanto al almacenamiento de los datos utilizado, los factores más importantes que se han tenido en cuenta para elegir una u otra opción han sido:

1. **Funcionalidades (30%):**

Este criterio valora como se adaptan las funcionalidades que proporciona cada alternativa con el objetivo de cumplir los objetivos establecidos para el proyecto. Se tendrá en cuenta, por una parte, las funcionalidades básicas para cumplir con las especificaciones del proyecto y, por otro, se valora positivamente la existencia de funciones alternativas.

2. **Escalabilidad (25%):**

Este criterio valora la capacidad de crecimiento y ampliación, sin que la funcionalidad se vea afectada negativamente. Dada la continua evolución que sufren las aplicaciones durante su vida útil, es importante que las modificaciones o posibles ampliaciones de la misma se realicen de forma sencilla, ejerciendo el menor impacto sobre el funcionamiento del sistema.

3. **Rendimiento (20%):**

Este criterio mide la eficiencia en cuanto al manejo de los recursos por parte del servidor que alojará la aplicación. Cuanto mejor sea la eficiencia en cuanto al uso de los recursos, menores serán los tiempos de respuesta y mayor la capacidad del servidor para manejar los datos, lo que se traducirá en una mejor experiencia del por parte usuario a la hora de utilizar el sistema.

4. Sencillez (10%):

Este criterio valora la sencillez a la hora de la implementación. Se valorará positivamente aquella alternativa que conlleve una curva de aprendizaje menor, suponiendo un ahorro en el coste debido a una reducción del tiempo necesario para el desarrollo.

5. Licencia (10%):

Este criterio valora el tipo de licencia que dispone cada una de las alternativas. Se valorará el coste económico de la licencia, así como la libertad que otorga cada una de ellas.

6. Soporte (5%):

Este criterio valora el mantenimiento actual de que dispone cada una de las soluciones. Se valorará positivamente la que disponga de un soporte actual mayor, teniendo en cuenta también, el número de plataformas en las que se puede implementar la solución.

CRITERIOS	PESO
1. Rendimiento	30%
2. Escalabilidad	25%
3. Facilidad de programación	20%
4. Presentación	10%
5. Licencia	10%
6. Soporte	5%

Tabla 2. Resumen ponderación de la selección del almacenamiento de datos

6.5.3. *Criterios de selección de la codificación de la información médica*

En cuanto a la codificación de la información médica utilizada, los factores más importantes que se han tenido en cuenta para elegir una u otra opción han sido:

1. **Ajuste (50%):**

Este criterio valora la eficiencia en la adaptación de los códigos. Se valorará negativamente el exceso de códigos (no necesarios por el sistema), debido a que esto se traduce en un aumento del tiempo necesario para la codificación de cada parámetro.

2. **Ambigüedad (30%):**

Este criterio valora la ambigüedad generada a la hora de identificar cada uno de los parámetros. Se valorará positivamente aquella solución que genere menor ambigüedad al identificar un parámetro.

3. **Posibilidades (15%):**

Este criterio valora el abanico de posibilidades de codificación. Se valorará positivamente aquella solución que disponga de un mayor número de soluciones, no solo el número de soluciones propias, sino también la posibilidad de generar nuevas mediante la combinación de las existentes.

4. Licencia (5%):

Este criterio valora el tipo de licencia que dispone cada una de las alternativas. Se valorará el coste económico que conlleva la obtención de la BBDD con los códigos de cada parámetro.

CRITERIOS	PESO
1. Ajuste	50%
2. Ambigüedad	30%
3. Posibilidades	15%
4. Licencia	5%

Tabla 3. Resumen ponderación de la selección de la codificación de la información médica

6.5.4. *Criterios de selección del algoritmo diagnóstico*

En cuanto al algoritmo diagnóstico utilizado, los factores más importantes que se han tenido en cuenta para elegir una u otra opción han sido:

1. Velocidad (35%):

Este criterio valora la velocidad de convergencia hacia una solución. Se valorará positivamente aquella solución que sea más eficiente a la hora de obtener una solución, lo más ajustada a la realidad posible.

2. Facilidad de programación (30%):

Este criterio valora la facilidad con la que se realiza la implementación del algoritmo, además se valorará positivamente aquella solución cuya curva de

aprendizaje sea menor, reduciendo el tiempo necesario para el desarrollo del mismo.

3. Rendimiento (15%):

Este criterio mide la eficiencia en cuanto al manejo de los recursos por parte del servidor cuando se está ejecutando un diagnóstico. Cuanto mejor sea la eficiencia en cuanto al uso de los recursos, menores serán los tiempos de respuesta y mayor la capacidad del servidor para manejar los datos.

4. Escalabilidad (10%):

Este criterio valora la capacidad de crecimiento y ampliación, sin que la funcionalidad se vea afectada negativamente. Se valorará positivamente aquel algoritmo que se adapte mejor a los cambios en el sistema, sin la necesidad de variar la codificación del mismo.

5. Complejidad (10%):

Este criterio cuantifica los caminos generados para la obtención de una solución. Se valorará positivamente aquella solución que genere árboles de decisión más pequeños y con ellos más entendibles a simple vista.

CRITERIOS	PESO
1. Velocidad	35%
2. Facilidad de programación	30%
3. Rendimiento	15%
4. Escalabilidad	10%
5. Complejidad	10%

Tabla 4. Resumen ponderación de la selección del algoritmo diagnóstico

6.6. Selección de la solución

Una vez identificados los criterios y su ponderación, se procede a realizar la comparativa de las diferentes alternativas estudiadas. La alternativa con la media más alta representa la solución más adecuada, siendo ésta la elegida.

6.6.1. Selección de la lógica de programa

Teniendo en cuenta las ponderaciones realizadas para la lógica de programa anteriormente mencionadas, se muestran a continuación una tabla resumen de las puntuaciones de cada alternativa:

LÓGICA DE PROGRAMA	PESO	ASP.NET	PHP	JSF con JSP
1. Rendimiento	30%	8	10	9
2. Escalabilidad	20%	9	8	9
3. Facilidad de programación	15%	6	10	7
4. Presentación	15%	9	7	8
5. Licencia	15%	6	10	9
6. Soporte	5%	6	10	9
TOTAL	100%	7,65	9,15	8,55

Tabla 5. Comparativa de alternativas para la lógica de programa

Como se puede apreciar, a la vista de los factores que se han tenido en cuenta para la elección de una lógica de programa u otra, la opción más adecuada para este proyecto es PHP.

6.6.2. Selección del almacenamiento de datos

Teniendo en cuenta las ponderaciones realizadas para el almacenamiento de datos anteriormente mencionadas, se muestran a continuación una tabla resumen de las puntuaciones de cada alternativa:

ALMACENAMIENTO DE DATOS	PESO	SQL Server	MySQL	Oracle	PostgreSQL
1. Rendimiento	30%	6	9	10	6
2. Escalabilidad	25%	9	8	8	7
3. Facilidad de programación	20%	8	9	7	7
4. Presentación	10%	7	8	7	7
5. Licencia	10%	6	9	5	9
6. Soporte	5%	6	9	9	9
TOTAL	100%	7,25	8,65	8,05	7

Tabla 6. Comparativa de alternativas para el almacenamiento de datos

Como se puede apreciar, a la vista de los factores que se han tenido en cuenta para la elección de un tipo de almacenamiento de datos u otro, la opción más adecuada para este proyecto es MySQL.

6.6.3. Selección de la codificación de la información médica

Teniendo en cuenta las ponderaciones realizadas para la codificación de la información médica anteriormente mencionadas, se muestran a continuación una tabla resumen de las puntuaciones de cada alternativa:

CODIFICACIÓN DE LA INFORMACIÓN MÉDICA	PESO	SNOMED-CT	LOINC	CIE-10
1. Ajuste	50%	7	8	6
2. Ambigüedad	30%	7	10	9
3. Posibilidades	15%	10	9	8
4. Licencia	5%	7	8	10
TOTAL	100%	7,45	8,75	7,4

Tabla 7. Comparativa de alternativas para la codificación de la información médica

Como se puede apreciar, a la vista de los factores que se han tenido en cuenta para la elección de un tipo de codificación de la información médica u otro, la opción más adecuada para este proyecto es LOINC.

6.6.4. Selección del algoritmo diagnóstico

Teniendo en cuenta las ponderaciones realizadas para el algoritmo diagnóstico anteriormente mencionadas, se muestran a continuación una tabla resumen de las puntuaciones de cada alternativa:

ALGORITMO DIAGNÓSTICO	PESO	C4.5	ID3
1. Velocidad	35%	8	9
2. Facilidad de programación	30%	5	6
3. Rendimiento	15%	9	9
4. Escalabilidad	10%	8	8
5. Complejidad	10%	8	9
TOTAL	100%	7,25	8

Tabla 8. Comparativa de alternativas para el algoritmo diagnóstico

Como se puede apreciar, a la vista de los factores que se han tenido en cuenta para la elección de un algoritmo diagnóstico u otro, la opción más adecuada para este proyecto es el algoritmo ID3.

6.7. Descripción de la solución

Una vez presentadas las alternativas de diseño más importantes y elegidas las soluciones más adecuadas, se realizará una breve descripción general de la solución elegida.

En primer lugar, el sistema será implementado sobre la plataforma PHP, que formará el núcleo del sistema. Además, se apoyará sobre otras tecnologías para la creación de la interfaz gráfica como es Bootstrap, JQuery, HTML o CSS, para las comunicaciones se utilizará AJAX.

En cuanto al almacenamiento de los datos, se utilizarán bases de datos relacionales utilizando el sistema de gestión MySQL, que proporciona la potencia y flexibilidad necesaria para cumplir con los objetivos expuestos anteriormente para el presente proyecto. Cabe destacar que el motor de almacenamiento de datos que se utilizará para todas las tablas es InnoDB, por ser un diseño en el que se prioriza la integridad y la utilización de relaciones entre tablas.

Para la codificación de los parámetros médicos, se utilizará la base de datos y estándar internacional LOINC, por ser la que más se ajusta a las necesidades del sistema y una de las más ampliamente usadas en el mundo.

Por último, para el algoritmo encargado de obtener los diagnósticos se utilizará ID3 por ser un algoritmo sencillo de implementar y que obtiene resultados que son fácilmente identificables visualmente.

En base a dichas soluciones adoptadas se definirá la arquitectura general del sistema. También cada uno de los módulos que lo conforman en el diseño de alto nivel en posteriores apartados.

7. Análisis de riesgos

Para la realización del análisis de riesgos se ha utilizado como base el documento “MAGERIT v.3: Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información”^[10] en el cual se proponen 3 técnicas muy específicas para el análisis y gestión de riesgos; En el presente proyecto se utilizará el método de tablas, por ser el más gráfico a la hora de evaluar y analizar los riesgos a los que se enfrenta el presente proyecto.

A la hora de llevar a cabo el desarrollo del proyecto, existen diversos riesgos que pueden afectar tanto al diseño como al funcionamiento del mismo. Aunque ninguno de ellos suponga peligro grave para la ejecución del mismo, existen algunos aspectos que han de ser controlados pues podrían desviarlo de su correcto funcionamiento. Al igual que se ha hecho en apartados anteriores, se van a diferenciar los posibles riesgos a los que se enfrenta el proyecto, en función, de aspectos puramente médicos y aspectos tecnológicos. Los posibles riesgos son los siguientes:

- **Aspectos médicos**

1. *Evolución de los análisis clínicos*: una evolución en los análisis clínicos puede provocar, por ejemplo, un cambio en el formato en el que se presentan los análisis clínicos, e incluso que no se presenten mediante valores numéricos, por ejemplo, que se tengan en cuenta la forma o el volumen de los propios parámetros.
2. *Evolución del sistema sanitario*: una evolución del sistema sanitario puede provocar, por ejemplo, que los análisis se realicen de tal forma que no se utilicen valores de referencia para la realización de los análisis, sino que directamente se analiza que sustancias se encuentran con valores incorrectos.

3. Evolución del diagnóstico de enfermedades: una evolución en la forma de diagnosticar enfermedades puede provocar, por ejemplo, que sean necesarios además de los valores de los parámetros en sangre, características como la forma o el volumen de las células para diagnosticar la variación de una sustancia en concreto.
4. Cambios en los patrones de referencia: una evolución en los patrones de referencia puede provocar, por ejemplo, que los valores de referencia que actualmente se usan, cambien.

- **Aspectos tecnológicos**

5. Evolución de las tecnologías actuales: un cambio en los estándares de salud actuales, puede provocar que sean necesarios pequeños cambios en la estructuración de los datos, para su posterior tratamiento y procesado.
6. Fallos en el servidor: un fallo en el funcionamiento del servidor, puede retrasar la ejecución del proyecto, así como la prestación del servicio correspondiente,

A continuación se presenta la tabla que analiza la estimación de los riesgos anteriormente mencionados:

Riesgo		Probabilidad				
		Muy Bajo	Bajo	Medio	Alto	Muy Alto
Impacto	Muy Alto		1			
	Alto			6		
	Medio	2				5
	Bajo		4		3	
	Muy Bajo					

Tabla 9. Estimación de los riesgos

Como se puede apreciar en la tabla, una evolución en los análisis clínicos tiene una probabilidad baja de suceder, sin embargo, el impacto en el proyecto sería muy alto. En cambio, la evaluación de las tecnologías actuales tiene una probabilidad muy alta de suceder, sin embargo, el impacto en el proyecto sería medio, puesto que el sistema está diseñado de tal forma que sea flexible y se adapte con facilidad a los cambios.

8. Metodología

En este apartado se detallarán cada una de las fases que ha seguido el proyecto, desde las especificaciones previas del sistema hasta la validación del mismo, y se desarrollará cada una de ellas siguiendo una metodología en V.

8.1. Ciclo de vida del proyecto

Dentro del gran abanico de metodologías y modelos que un proyecto SW puede adoptar durante su ciclo de vida destacan:

- 1) **Modelo lineal:** este modelo proporciona un enfoque secuencial para el desarrollo software, comenzando en una fase de definición de los requerimientos del sistema y continua por el análisis y diseño, implementación y pruebas, integración para terminar en la operación y mantenimiento.

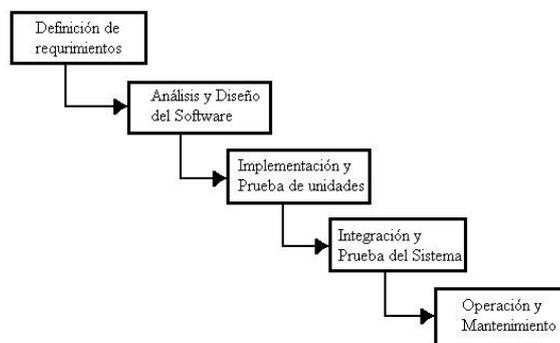


Ilustración 3. Modelo Lineal

- 2) **Modelo espiral:** este modelo proporciona un enfoque iterativo para el desarrollo software. En cada iteración hay que tener en cuenta y controlar los objetivos del proyecto, corroborar que los objetivos se pueden lograr de forma exitosa, desarrollar y verificar el proceso realizado y si el resultado no es el esperado planificar como realizar las mejoras necesarias.

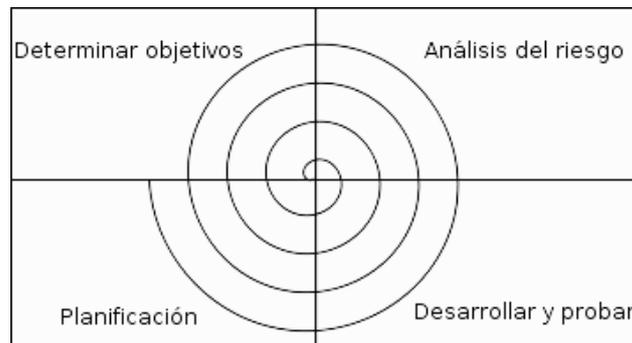


Ilustración 4. Modelo Espiral

3) **Modelo incremental:** al igual que el modelo espiral este modelo proporciona un enfoque iterativo. Se comienza con un análisis de los requisitos para terminar en las pruebas, si se detectan errores o el diseño no es el adecuado, se genera una segunda iteración y así sucesivamente. Es una mejora con respecto al método lineal puesto que permite retroceder en las fases (se corrigen en las posteriores iteraciones).

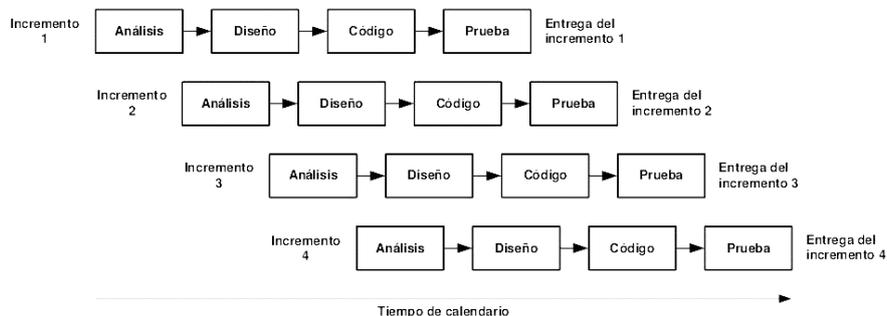


Ilustración 5. Modelo Incremental

- 4) Para el desarrollo del proyecto que nos ocupa se ha utilizado un **modelo en V**.

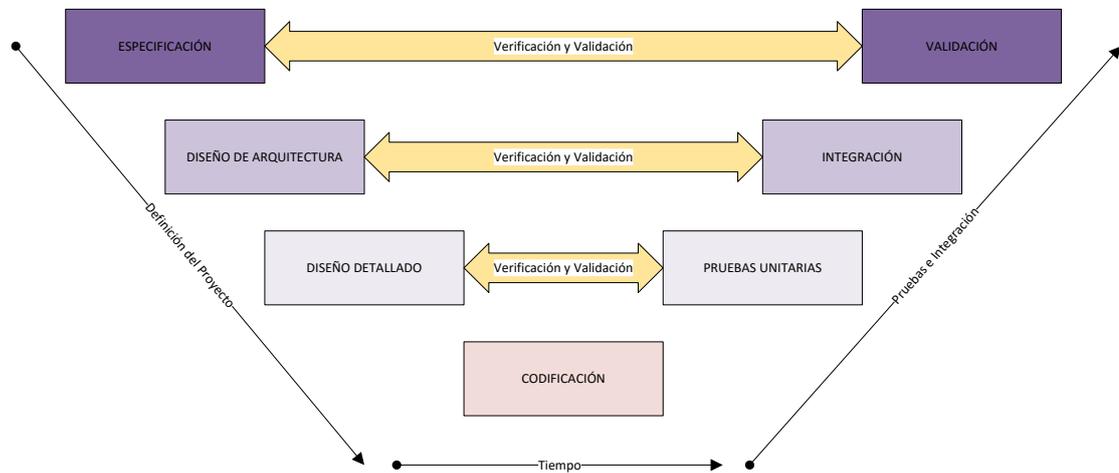


Ilustración 6. Modelo en V

Este modelo proporciona a los proyectos SW la capacidad de corregir errores en cada fase de desarrollo del mismo sin tener que esperar a las etapas finales para que sean rectificadas. Es decir, no se espera hasta la consecución del proyecto para realizar las pruebas, sino que, con la codificación de cada módulo/fase, se realizan pruebas unitarias para validar el correcto funcionamiento de cada uno. Las fases que tiene un modelo en V son las siguientes:

- 1) **Especificación:** *en esta fase se fijarán los requerimientos que deberá tener el sistema para cumplir correctamente con los objetivos. Para la elaboración de las especificaciones del proyecto se dividirán en 3 tipos las especificaciones: especificaciones funcionales, especificaciones de gestión y mantenimiento y especificaciones de diseño.*
- 2) **Diseño de la arquitectura:** *en esta fase se define la estructura general que va a tener el sistema. También se conoce como diseño de alto nivel y en él*

se describen los componentes principales y el modo en que interaccionan entre sí para satisfacer los requerimientos del sistema.

- 3) **Diseño detallado:** *en esta fase se incide en la estructura de cada uno de los modelos en los que se divide el sistema. También se conoce como diseño de bajo nivel y en él se produce un diseño completo y listo para ser programado de tal forma que se verifique que se satisfacen los requerimientos para los que se diseñó el sistema.*
- 4) **Codificación:** *en esta fase se procede al desarrollo y codificación de cada una de los módulos/componentes diseñados y definidos en la fase de diseño detallado o diseño de bajo nivel.*
- 5) **Pruebas unitarias:** *en esta fase se realizan las correspondientes pruebas a cada uno de los módulos desarrollados en la fase anterior de tal forma que se valide el correcto funcionamiento de cada uno de ellos de forma individual, es decir, sin tener en cuenta el resto de los módulos cuando se valida uno de ellos.*
- 6) **Integración:** *en esta fase se procede a la integración de cada uno de los módulos, previamente validados de manera independiente, con el resto de los módulos que componen el sistema completo, tal y como se diseñó y definió en la fase diseño de la arquitectura o diseño de alto nivel.*
- 7) **Validación:** *en esta fase se realizan las pruebas necesarias para validar el sistema completo compuesto por todos los módulos integrados y probados por separado en las fases anteriores, además se realiza un análisis para corroborar que se cumplen los objetivos para los que fue diseñado el sistema en la fase de especificación.*

8.2. Fases del proyecto

A continuación se especificará cada una de las fases que ha seguido el proyecto durante su desarrollo siguiendo el modelo citado en el apartado anterior.

8.2.1. Especificación

Como se ha comentado anteriormente, en este apartado se muestran los requerimientos que el sistema debe cumplir para alcanzar los objetivos expuestos en apartados anteriores. Las especificaciones del proyecto se dividirán en los siguientes apartados:

1. Especificaciones funcionales.
2. Especificaciones de gestión y mantenimiento.
3. Especificaciones de diseño.

8.2.1.1. Especificaciones funcionales

En este apartado, se describirán las especificaciones o requisitos que se establecerán para el diseño y desarrollo del sistema propuesto en el presente proyecto, de forma que se logre la consecución de los objetivos citados en apartados anteriores. Cabe destacar que el sistema está compuesto por diferentes subsistemas o módulos y a continuación se describen cada uno de ellos analizando los requisitos funcionales que los conforman y su relación con los otros módulos. A continuación se muestra una imagen de la arquitectura general del sistema, con los diferentes módulos que la conforman:

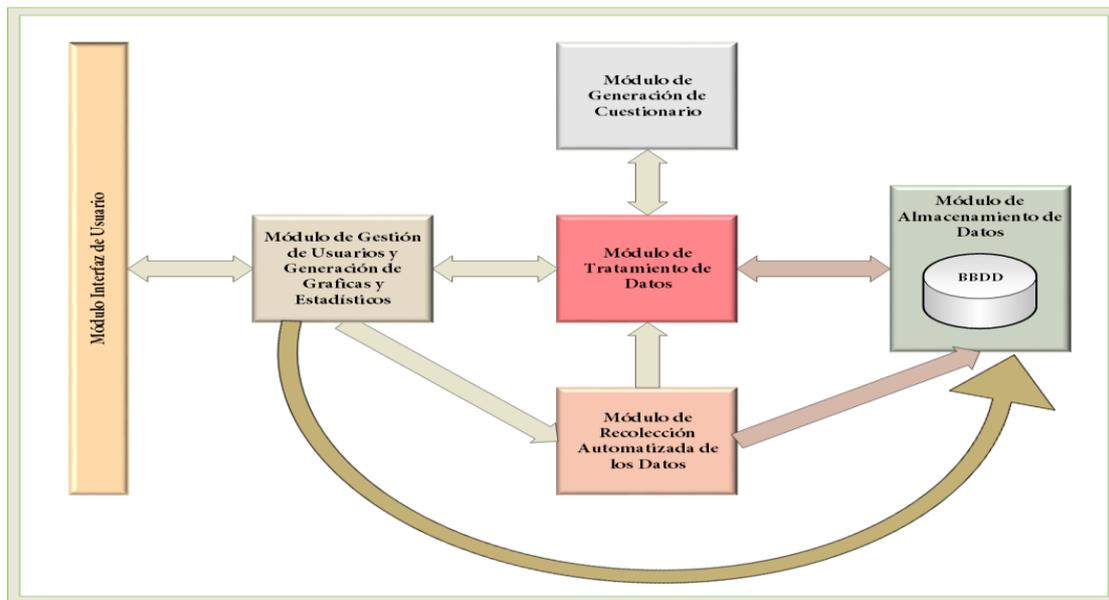


Ilustración 7. Arquitectura del sistema

Como se puede apreciar el sistema está compuesto por seis módulos principales. El módulo de almacenamiento de datos, el módulo de tratamiento de datos, el módulo de generación de encuestas, el módulo interfaz de usuario, el módulo de recolección automatizada de datos y el módulo de gestión de usuarios y generación de gráficas y estadísticos. A continuación se enumeran un resumen de las funcionalidades y relaciones de cada uno de ellos.

➤ **Módulo de almacenamiento de los datos**

○ **Funcionalidad**

Se encarga, por una parte, de perpetuar los datos, con los análisis clínicos que van introduciendo los usuarios y, por otra, de almacenar los datos de parámetros, unidades de medida, rangos, etc. necesarios para que el sistema funcione.

- Relación con otros módulos

Este módulo tiene relación con el módulo de tratamiento de datos para proporcionarle los datos necesarios para la obtención de los posibles diagnósticos y, además, para almacenar los datos relativos a los diagnósticos realizados por parte de los usuarios registrados. También, tiene relación con el módulo de recolección automatizada de datos permitiendo a los usuarios guardar sus análisis para posteriores usos.

➤ **Módulo de tratamiento de datos**

- Funcionalidad

Se encarga de obtener un posible diagnóstico en base a los datos proporcionados, bien por el módulo de obtención de automatizada de datos, o bien introducidos por el usuario a través de la interfaz gráfica.

- Relación con otros módulos

Este módulo tiene relación, como se ha comentado anteriormente, con el módulo de almacenamiento de datos, con el módulo de recolección automatizada de los datos para proporcionar a los usuarios que no están registrados diagnósticos de forma automatizada. Este módulo también está relacionado con el módulo de generación de cuestionarios, para tratar de afinar los diagnósticos en base a las respuestas del usuario y, por último, con el módulo de gestión de usuarios y generación de gráficas y estadísticos que será el que haga las veces de interfaz entre este y la interfaz que visualiza el usuario.

➤ **Módulo de recolección automatizada de los datos**

○ Funcionalidad

Se encarga de facilitar al sistema la recolección de los datos referentes a análisis clínicos, proporcionando una interfaz que permite subir estos datos a un servidor para posteriores usos.

○ Relación con otros módulos

Este módulo está relacionado, como se ha comentado anteriormente, con el módulo de almacenamiento de datos, con el módulo de tratamiento de datos y con el módulo de gestión de usuarios y generación de gráficas y estadísticos.

➤ **Módulo de sistema de cuestionarios**

○ Funcionalidad

Se encarga de ajustar los diagnósticos realizados por el módulo de tratamiento de datos, en base a los resultados obtenidos, interactúa con el usuario realizando un cuestionario y, en base a las respuestas de este, evalúa cuál de las posibles patologías tiene más posibilidades de padecer.

○ Relación con otros módulos

Este módulo está relacionado con la interfaz de usuario, módulo de gestión de usuarios y generación de gráficas y estadísticos y el módulo de tratamiento de datos, de forma que gestione si los datos que van a ser tratados o cargados han de ser guardados en el sistema de almacenamiento (porque es un usuario registrado) o únicamente han de ser tratados.

➤ **Módulo de Gestión de Usuarios y Generación de Gráficas y Estadísticos**

○ Funcionalidad

Se encarga, por una parte, de gestionar el registro de los usuarios en el sistema y proporcionarles los análisis e información referente a sus perfiles, y por otra generar gráficas en base a diferentes analíticas.

○ Relación con otros módulos

Este módulo está relacionado con la interfaz de usuario, haciendo las veces de interfaz a su vez, entre esta y el módulo de tratamiento de datos, también está relacionado con el módulo de recolección automatizada de los datos, como se ha comentado anteriormente, y con el módulo de almacenamiento de datos, para gestionar el registro de los usuarios.

➤ **Módulo de interfaz de usuario**

○ Funcionalidad

Se encarga de proporcionar a los usuarios una forma sencilla y manejable de utilizar el sistema.

○ Relación con otros módulos

Este módulo está relacionado con el módulo de gestión de usuarios y generación de gráficas y estadísticos y será lo que el usuario vea del sistema.

8.2.1.2. Especificaciones de gestión y mantenimiento

En este apartado se detallan los requisitos no funcionales del sistema. Entre ellos destacan la fiabilidad, mantenibilidad, disponibilidad y portabilidad del sistema.

- ***Fiabilidad:*** el sistema deberá representar de una manera fiable, los resultados de los diagnósticos en base a las analíticas de los usuarios. De igual forma, estos datos deberán permanecer el sistema almacenados, el tiempo que el usuario exija, con una correcta estructuración.
- ***Mantenibilidad:*** el mantenimiento deberá ser sencillo y para ello se ha elegido un diseño modular.
- ***Disponibilidad:*** el sistema deberá estar disponible 24 horas al día, los 365 días del año, siempre que el servidor donde se alberga esté funcionando correctamente.
- ***Portabilidad:*** el sistema deberá funcionar en cualquier sistema operativo. Por otra parte, se podrá acceder al él desde cualquier navegador, ya sea en versión de escritorio o en versión móvil.

8.2.1.3. Especificaciones de diseño

Al margen de las especificaciones funcionales que se han marcado en el apartado anterior, también se plantean una serie de requerimientos desde el punto de vista de diseño del sistema. Los requisitos que constituyen las especificaciones del diseño son los siguientes:

- ***Escalabilidad y Flexibilidad:*** de esta manera se facilitará la inclusión de posibles nuevas funcionalidades o mejoras en los módulos ya existentes con el fin de agilizar dicha integración.

- **Modular:** la integración de nuevos módulos deberá seguir la estructura de los existentes. Para ello, se deberán generar las nuevas funcionalidades en nuevos módulos o agregarlas en los módulos ya existentes. Gracias a esto, se facilitará la comprensión del diseño que se ha realizado, así como la localización de los cambios que se introduzcan.
- **Interactiva:** se dispondrá de una interfaz clara y manejable, de forma que se facilite la navegación a usuarios con un perfil menos técnico para que puedan navegar por el portal web de una manera sencilla e intuitiva.
- **Adaptable:** la interfaz se deberá ajustar a cualquier tipo de pantalla, así como a cualquier navegador.

8.2.2. *Diseño de la arquitectura - Diseño de Alto Nivel*

En este apartado se definirá la arquitectura general del sistema, así como los componentes principales que lo conforman.

8.2.2.1. *Arquitectura general*

Según lo detallado anteriormente, se ha procedido a la creación del diseño del sistema para diagnósticos médicos. Para ello, se ha diseñado e implementado el sistema siguiendo las especificaciones definidas para la plataforma PHP. Los principales motivos para la utilización de esta plataforma para el desarrollo del sistema son los siguientes:

- Liberación para el desarrollador de las tareas de bajo nivel, ya que se encuentran dentro de las propias APIs de la plataforma PHP.
- Existencia de una gran variedad de Frameworks y librerías que incrementan las utilidades o tareas de bajo nivel.
- PHP como lenguaje de programación, está claramente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- Énfasis en el uso del patrón MVC, el cual permite separar la parte lógica de la presentación.

A continuación se presenta a modo de ejemplo, un escenario en que dos usuarios acceden al sistema a través de dos equipos diferentes. Como se puede observar en la siguiente imagen el sistema funcionará en un servidor web y, por tanto, este servidor recibirá peticiones de los clientes bajo una estructura cliente-servidor típica. Además, el servidor estará conectado a su vez a una BBDD para almacenar la información. Cabe mencionar que se ha optado por una arquitectura

cliente-servidor porque permite la intercomunicación con múltiples clientes y es la forma más simple de controlar el sistema.

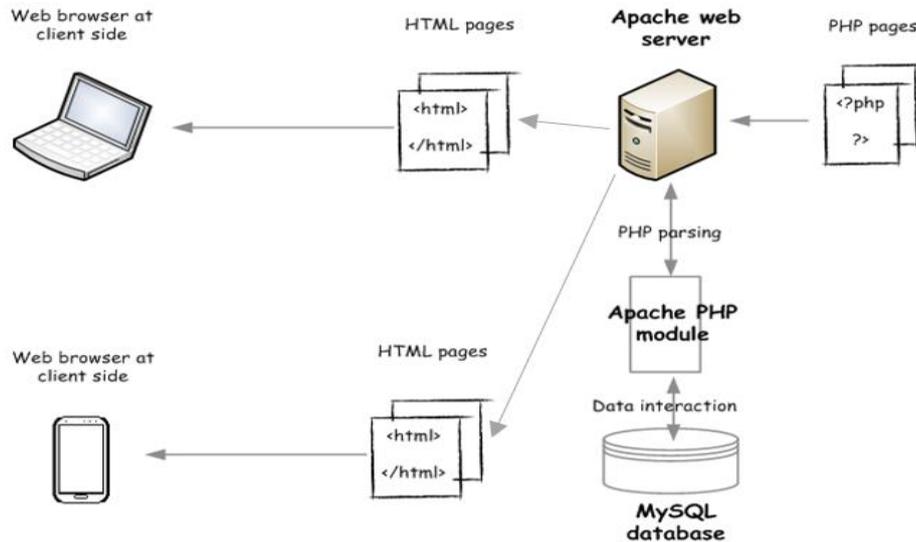


Ilustración 8. Arquitectura básica del sistema

En la siguiente imagen se muestra un ejemplo de flujo cliente-servidor cuando un cliente se conecta a la herramienta:

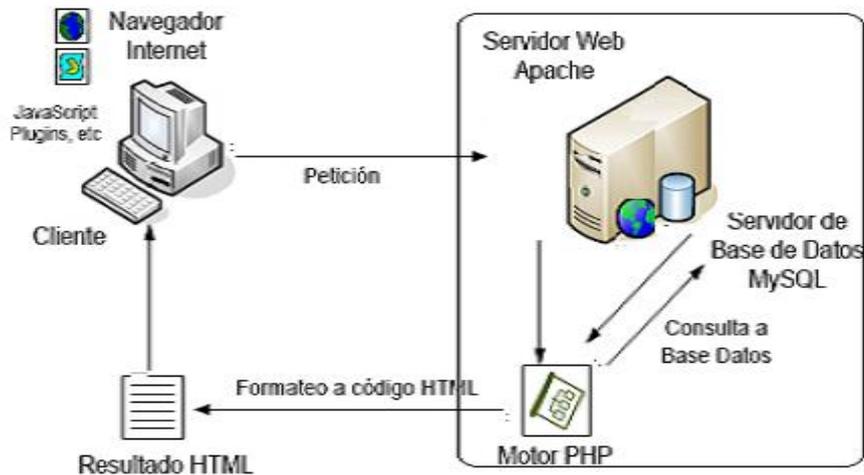


Ilustración 9. Flujo cliente-servidor

En el explorador se visualizarán las diferentes páginas de la herramienta utilizando para ello *HTML* que contendrán partes tanto estáticas como dinámicas. Para dar dinamismo a las paginas, en el navegador del cliente, se usará *JQuery*, que

es una librería de JavaScript que permite interactuar con los documentos HTML, manipular el DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica *AJAX* a páginas web. Por otra parte, además de HTML y JQuery, para la visualización de las paginas en dispositivos con tamaños de pantalla reducida se utilizará *Bootstrap*, que implementa una gran variedad de hojas de estilo CSS que facilitan la codificación para cualquier tipo de pantalla.

Como se ha comentado anteriormente, el uso de la plataforma PHP permite la utilización de patrón de diseño MVC, el cual permite separar la parte lógica de la presentación de la herramienta. Por ello, el sistema se ha diseñado y desarrollado fundamentado en este modelo durante su construcción. Este patrón de arquitectura software se basa en la idea de reutilización de código y separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

Para conseguir estas características, el patrón MVC se fundamenta en la creación de tres capas distintas: modelo, vista y controlador.

- **Modelo:** representa los datos del programa. Se encarga de manipular estos datos de forma coherente y ofrecerlos a la aplicación a medida que los va requiriendo. Esta capa no debe tener conocimiento ninguno ni del controlador ni de las vistas.
- **Vista:** es la representación visual de los datos. Serán en sí las páginas que el usuario contemple y es la capa mediante la que este interactuará con la herramienta. Para representar los datos se hará uso de ficheros *.php* que establecerán cómo es la interacción con dichos datos, pasando siempre por la capa del controlador.
- **Controlador:** hace de intermediario entre la vista y el modelo. Ofrece respuesta frente a eventos e invoca peticiones al modelo. También modifica

la vista cambiando la forma en que se presentan los datos que el modelo proporciona.

En la siguiente imagen se proporciona un ciclo básico que cumple toda herramienta que esté basada en el patrón MVC.

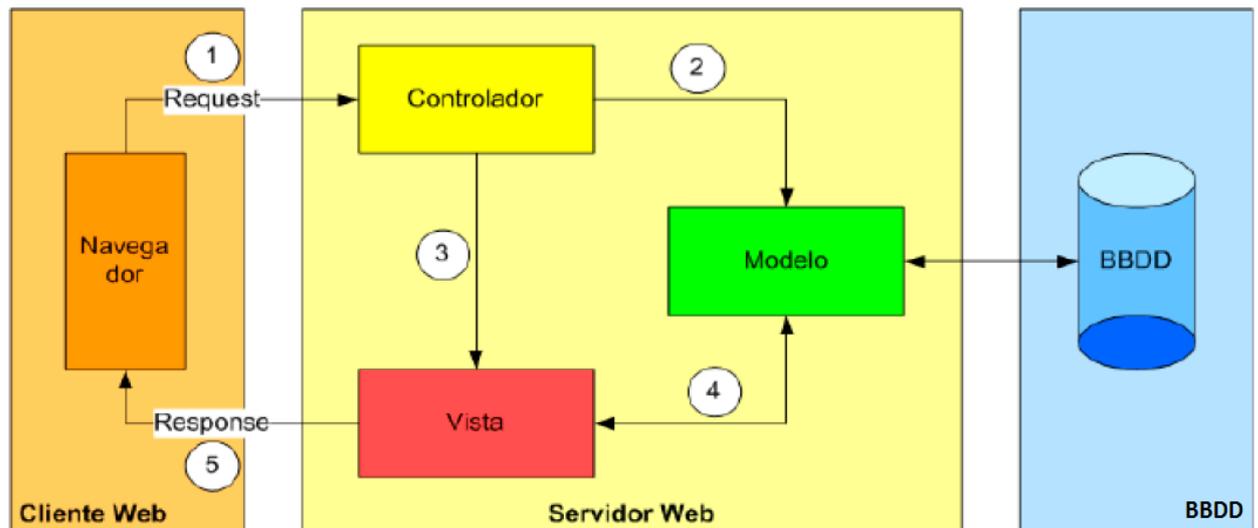


Ilustración 10. Arquitectura Modelo-Vista-Controlador

A continuación se detalla el significado de cada iteración representado numéricamente en la anterior imagen:

1. En primer lugar, el usuario debe realizar algún tipo de petición en un request, es decir, alguna acción en el navegador.
2. El controlador capturaré la petición y realizará diferentes operaciones. Es probable que necesiten obtener información de la base de datos, por lo que se hará uso de la capa modelo.
3. Una vez el controlador lleve a cabo las acciones, deberá escoger dónde se debe redirigir la petición. Será necesaria la capa vista para ofrecer al usuario los datos requeridos.
4. El modelo se encargará de proporcionar los datos que la vista necesite en cada momento.

5. El resultado será devuelto en un response al cliente y vuelve a iniciarse el ciclo en caso de ser necesario.

Las principales ventajas que proporciona trabajar con un patrón como este son la separación de los datos de su representación gráfica y la simpleza a la hora de añadir y modificar nuevas vistas para que se adapten según las necesidades del modelo. Otras características que ofrece este patrón son la independencia entre las tres capas, la escalabilidad que proporciona el poder actuar en cualquiera de las capas y la separación de funciones que facilita el mantenimiento.

8.2.2.2. Módulos

En este apartado se describen el diseño de cada uno de los módulos que componen la arquitectura general del sistema.

8.2.2.2.1. *Módulo de almacenamiento de los datos*

Este módulo es el encargado de almacenar los datos relativos al sistema. Estos datos se almacenan en una base de datos relacional y en el caso de los datos referentes a los análisis, para su almacenamiento en la base de datos se hará uso del estándar FHIR.

Las bases de datos relacionales, permiten almacenar y tratar datos que están relacionados entre sí. El funcionamiento de una base de datos que sigue un modelo relacional consiste en establecer y procesar relaciones entre entidades. A continuación se muestra el modelo entidad-relación para la base de datos creada para el sistema.

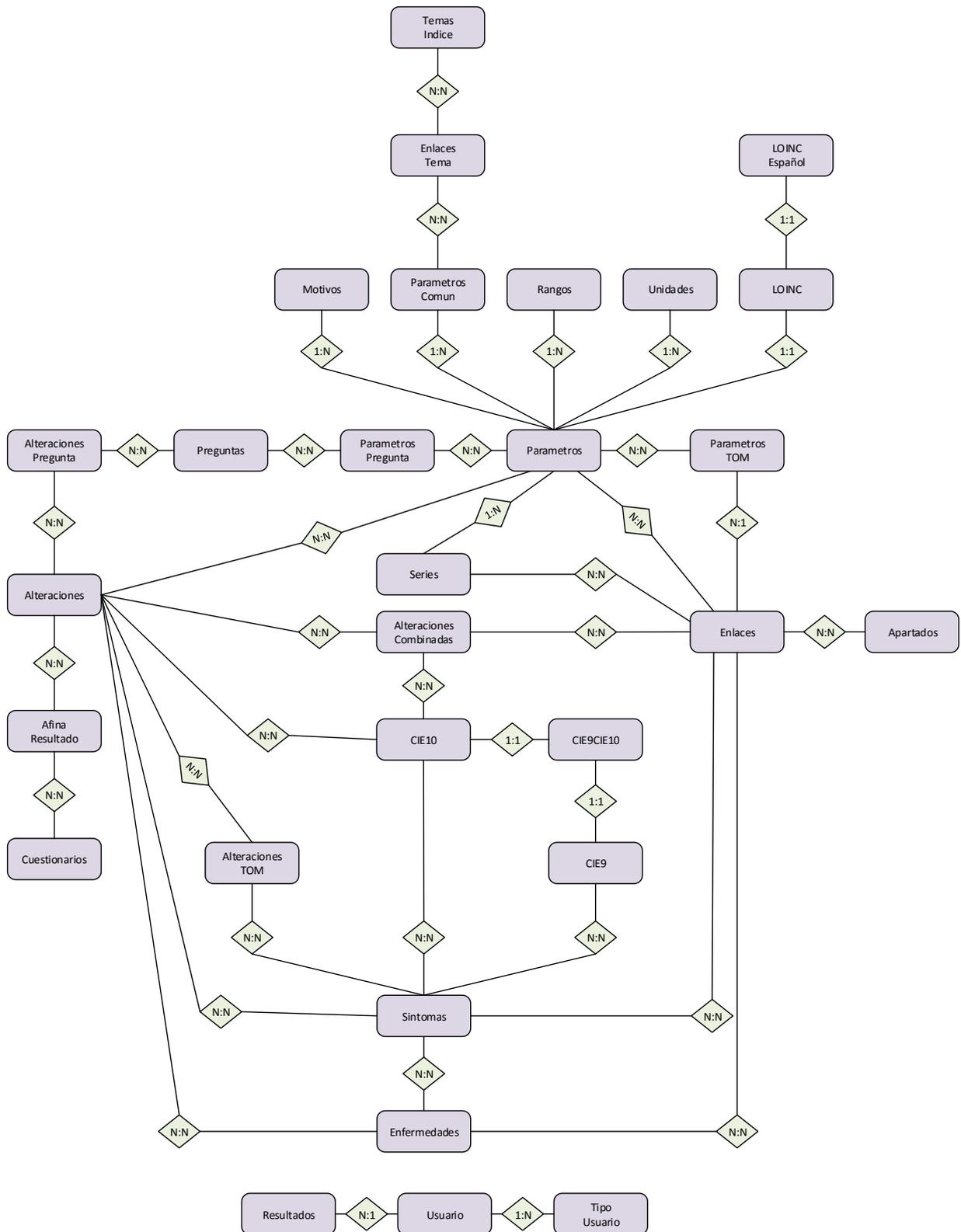


Ilustración 11. Diagrama entidad-relación de la base de datos del sistema

Como se ha comentado en apartados anteriores, el diseño de todo el sistema se ha realizado siguiendo una estructura modular, por ello, el añadir nuevas entidades para proporcionar nuevas funcionalidades, no perjudica el funcionamiento de las entidades actuales.

A continuación se detallan las funciones de las entidades más importantes que conforman el módulo de almacenamiento de datos.

- ***Parámetros***

Esta entidad almacena los datos referentes a los parámetros que se pueden medir en una analítica. De cada parámetro se almacena, además de su nombre, una breve descripción explicativa sobre qué indica dicho parámetro, las unidades de medida en la que habitualmente se mide, además de un flag que indica si es un parámetro común o que habitualmente se mida en los análisis clínicos o no.

- ***Rangos***

Esta entidad almacena los datos referentes a los valores en torno a los que un parámetro se puede considerar que está en unos valores correctos. Para cada parámetro se almacena un valor máximo y un mínimo, diferenciando a los usuarios por su edad, su raza y su sexo.

- ***Unidades***

Esta entidad almacena las unidades en las que se pueden medir cada uno de los parámetros, así como una conversión de cada una de estas unidades a una escala diferente.

- ***Alteraciones***

Esta entidad almacena los datos referentes a las diferentes alteraciones que se pueden sufrir debido a la variación de alguno de los parámetros existentes en una analítica.

- ***CIE10***

Esta entidad almacena la clasificación internacional de enfermedades en su versión 10, cada enfermedad esta codificada con un número del tipo “X00.00” que se utiliza para la identificación de cada enfermedad.

- ***LOINC***

Esta entidad almacena todos los códigos LOINC con los que se pueden nombrar a cada uno de los parámetros existentes en el sistema.

- ***Usuario***

Esta entidad almacena el perfil de cada uno de los usuarios. De cada uno, se almacena su nombre, su fecha de nacimiento, su email, su sexo, su raza, además de una contraseña para acceder a su perfil.

- ***Resultados***

Esta entidad almacena los resultados de los diagnósticos de cada usuario. Para el almacenamiento de los datos se utiliza el formato FHIR.

- ***Cuestiones***

Esta entidad almacena las diferentes cuestiones que se pueden hacer para el ajuste de un diagnóstico.

- *Afina Resultado*

Esta entidad relaciona las diferentes cuestiones con las alteraciones, de forma que se ajuste si es lógico o no el mostrar cada alteración en base a las respuestas obtenidas de las cuestiones.

8.2.2.2.2. Módulo de tratamiento de datos

Este módulo se encarga de, a partir de los datos captados, bien por parte del módulo de recolección automatizado de datos, o bien a través de la interfaz del usuario, obtener un diagnóstico en base a los datos obtenidos. Para la obtención del diagnóstico se hace uso del algoritmo ID3. A continuación se muestra un diagrama ejemplo para el diagnóstico de una enfermedad (*Anemia Ferropénica*).

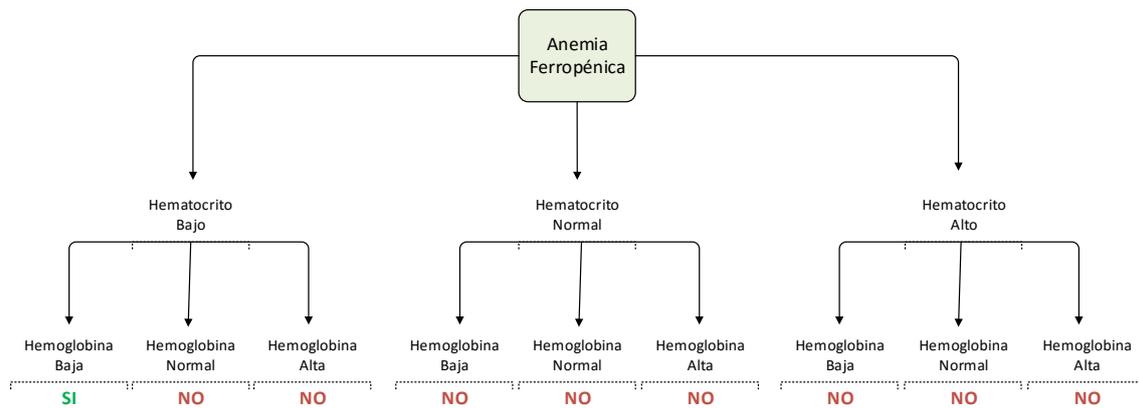


Ilustración 12. Diagnóstico de anemia ferropénica en base al algoritmo ID3

Este módulo inicialmente analiza todos los parámetros que están desviados de rango y mediante el algoritmo anteriormente mencionado, se comprueba cuáles de las posibles enfermedades cumplen con los requisitos. Como se puede apreciar del diagrama anterior, para diagnosticar la enfermedad *Anemia ferropénica* será necesario que se cumpla que tanto los *hematocritos* como la *hemoglobina* estén en niveles por debajo de lo normal.

8.2.2.2.3. Módulo de recolección automatizada de los datos

Este módulo se encarga de automatizar la recolección de los datos referentes a analíticas, proporcionando a los usuarios un interfaz en la que seleccionar el fichero *.json* alojado en su sistema y utilizando el método POST del protocolo HTTP lo alojará en el servidor. Además, si el usuario está registrado, también guarda el nombre del fichero generado asociado al usuario en la base de datos para la generación de estadísticos si el usuario así lo necesita. Para la generación del nombre del fichero en caso de que el usuario este registrado, se utiliza el nombre del usuario, unido al Id que se autogeneró cuando éste se registró, seguido de una barra baja y el número de analíticas que tiene actualmente subidas. A continuación se muestra un diagrama ejemplo de la subida de un fichero.

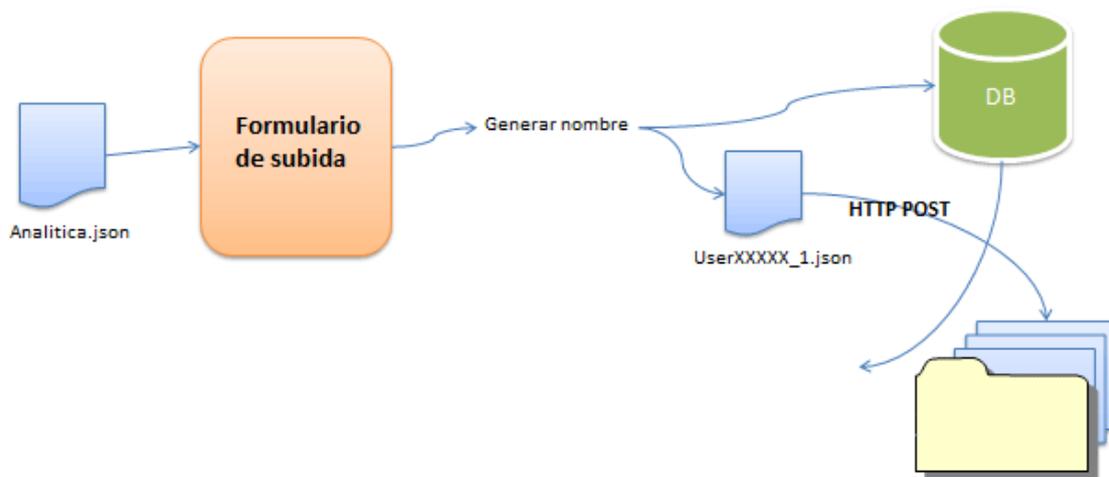


Ilustración 13. Recolección automatizada de datos

8.2.2.2.4. Módulo de sistema de cuestionarios

Este módulo contiene las funcionalidades para la realización de las encuestas que se encargan de ajustar los diagnósticos. Este módulo se encarga de, una vez realizado el diagnóstico y presentadas al usuario las posibles enfermedades que éste puede padecer en base a su análisis, presentar una serie de preguntas que

permitirán reducir el número de posibilidades que se le muestran. También es función de este módulo recoger las respuestas que el usuario proporcione y tratarlas para estimar cuál de las enfermedades es más posible que padezca.

En primer lugar, el módulo recibe del módulo de gestión cuáles son las posibles enfermedades en base al resultado de la analítica. Estos datos contienen además del nombre de las enfermedades, un identificador para su búsqueda en la base de datos. Con estos datos, el módulo accede al sistema de cuestiones de la base de datos a buscar que cuestiones están relacionadas con las enfermedades que resultan más probables.

Tras recolectar las preguntas asociadas a las distintas enfermedades, le traspasa dichas cuestiones al módulo de interfaz de usuario, que se las presenta al usuario. Tras la respuesta de cada cuestión, el módulo de cuestiones registra la respuesta (positiva o negativa). Tras la respuesta de todas las cuestiones este módulo evalúa la posibilidad de padecer cada enfermedad.

A continuación se presenta un diagrama resumen del funcionamiento de este módulo.

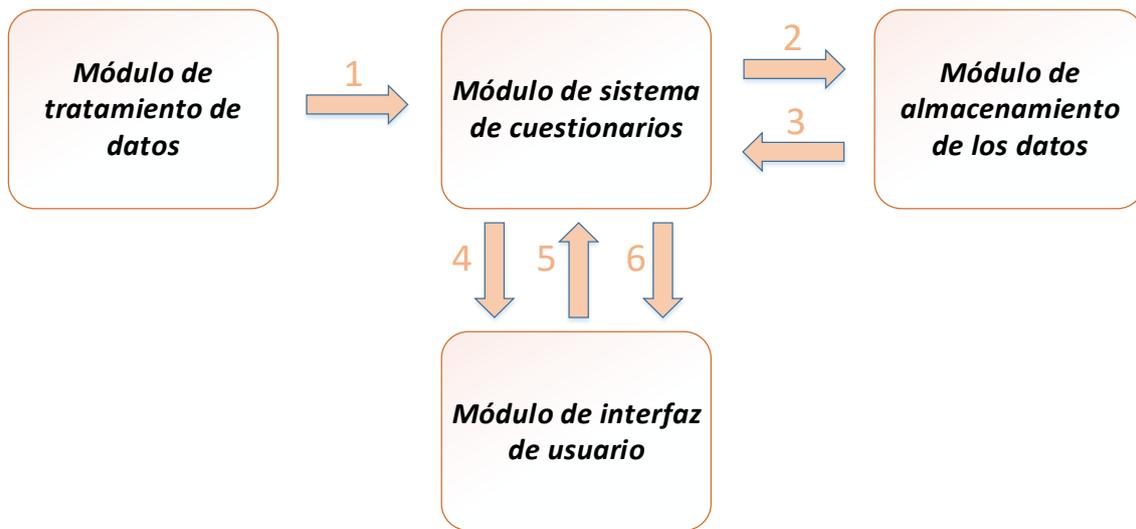


Ilustración 14. Comunicación entre módulos, módulo de cuestionarios

8.2.2.2.5. Módulo de Gestión de Usuarios y Generación de Gráficas y Estadísticos

Este módulo, por una parte, se encarga de las funciones relacionadas con el manejo de usuarios. Estas funciones proporcionan las herramientas necesarias para el registro de los usuarios y gestión de los perfiles de usuario. Por otra, se encarga de las funciones necesarias para la generación de gráficas de parámetros, tomando como base diferentes analíticas de un usuario.

A continuación se describen en detalle las funciones principales de este módulo diferenciando entre el sub-módulo de gestión de usuarios y el de generación de gráficas.

Sub-módulo de gestión de usuarios

- Registro de usuarios

El registro de usuarios se lleva a cabo a través de un formulario que se presenta a través de la interfaz de usuario. Los datos que se almacenan de cada usuario son los siguientes:

- Tipo de usuario: identifica qué tipo de usuario es el que se está registrando, las opciones son, usuario particular, médico o laboratorio clínico. En función del perfil del usuario se tendrá acceso únicamente a analíticas propias o de varios usuarios, en caso de ser un laboratorio clínico o un médico.
- Nombre y apellidos: se utiliza para generar el fichero FHIR que contiene las analíticas del paciente e identificarlo en caso de tener que operar en herramientas externas.
- Email: utilizado para identificar unívocamente al usuario en el sistema, además de utilizarlo para *logearse* y acceder a las funciones que poseen los usuarios registrados.
- Sexo, raza y fecha de nacimiento: estos tres campos, son necesarios para tomar los valores de referencia oportunos de cada parámetro.
- Contraseña. se utiliza en el proceso de autenticación para verificar la identidad del usuario. La contraseña se almacena realizando un HASH (SHA256), por tanto, es desconocida para los administradores del sistema.
- Id del médico: un número de identificación del médico para, en caso de estar bajo la supervisión de dicho médico, este tendrá acceso a los análisis del paciente.

- Id de la clínica: un número de identificación de una clínica. Cada clínica puede tener asociados diferentes médicos y gestionar a su vez todos los pacientes que dependan de estos.

Además de almacenar estos datos, este módulo también verifica que los datos se introduzcan correctamente, que los emails no se repitan y estén en un formato correcto, que las fechas de nacimiento se introduzcan correctamente y que no se introduzcan caracteres no permitidos (especiales) en ninguno de los campos.

- Gestión de perfiles

Este módulo también se encarga de presentar datos de analíticas pasadas almacenadas en el sistema, solo a los legítimos propietarios de estas analíticas y a personas autorizadas, en caso de que el seguimiento lo realice un médico o una clínica. De esta forma se garantiza que ningún usuario accede sin autorización a datos que no le pertenecen. Esta gestión es realizada mediante número de identificación en cada analítica, de tal forma que solo podrá acceder a una analítica aquel usuario que posea el mismo número de identificación con el que la analítica fue registrada en el sistema, y en su caso del médico o clínica que tenga asociado este usuario.

Sub-módulo de generación de gráficas

- Generación de gráficas

Cada usuario podrá optar por elegir alguno de los parámetros, de los que se posean más de una analítica, y obtener una evolución visual del desarrollo de dicho parámetro. Este módulo extrae de las diferentes analíticas almacenadas en formato FHIR, tanto la fecha en la que se realizó la analítica como el valor cuantificado para el parámetro en cuestión y lo muestra en un diagrama ordenado por fecha.

8.2.2.2.6. *Módulo de interfaz de usuario*

Este módulo se encarga de la interacción con los usuarios de la aplicación del sistema. Las funciones principales de este módulo se pueden resumir en: presentar los datos al usuario y capturar las acciones que éste realiza en el sistema.

Una de las especificaciones principales para el presente proyecto es el desarrollo de una interfaz sencilla y que se adapte a cualquier dispositivo, que permite a cualquier usuario, tenga el perfil que tenga, interactuar con el sistema de manera fácil.

A continuación se detallan las principales funciones que este módulo proporciona.

- **Presentar los datos al usuario**

Una de las tareas principales de este módulo es presentar el sistema al usuario y permitirle navegar por las diferentes funcionalidades que este proporciona. Este módulo se implementa mediante una interfaz web, que utiliza tanto partes dinámicas como estáticas sobre las que el usuario puede actuar para realizar diferentes acciones. A continuación se muestra un pantallazo de la herramienta donde se muestran las principales acciones de que dispone el sistema.

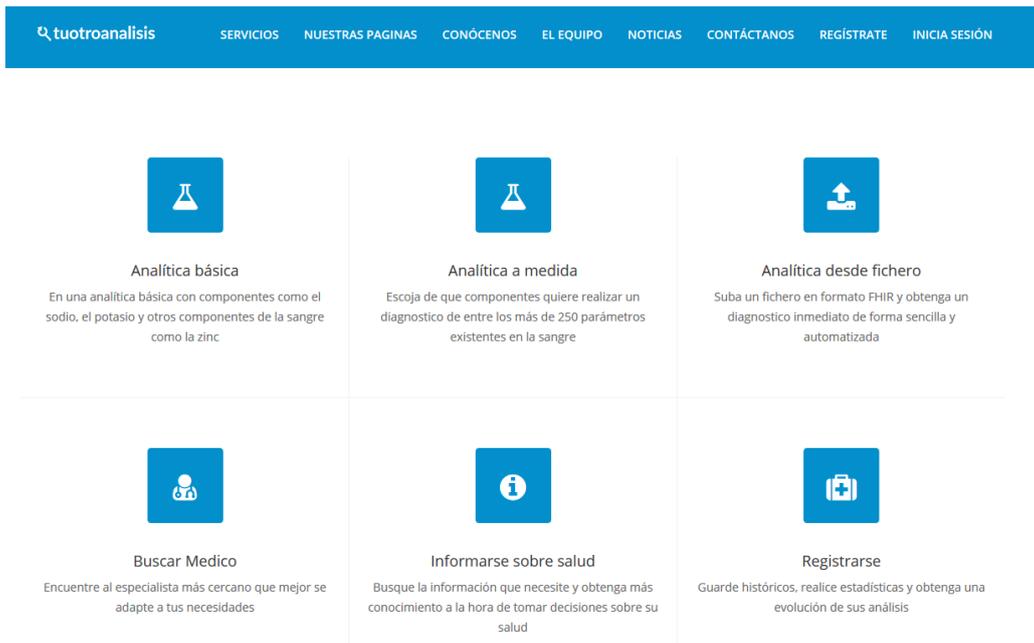


Ilustración 15. Funciones principales del sistema

Algunas de las funciones que dispone el usuario son las siguientes:

- **Análítica básica:** proporciona al usuario la posibilidad de introducir valores sobre una analítica estándar generada en el presente proyecto.
- **Análítica a medida:** proporciona al usuario la posibilidad de elegir qué parámetros analizar de su analítica.
- **Análítica desde fichero:** esta opción conecta con el módulo de recolección automatizada de los datos para subir un fichero al servidor.
- **Registrarse:** permite al usuario registrarse en el sistema.

Funciones extra (externas a éste proyecto):

- **Buscar médico:** permite al usuario buscar en un directorio médico (*ABCmédico.com*), el médico que puede tratar su enfermedad.
- **Informarse sobre salud:** permite al usuario acceder a una enciclopedia de salud (*tuotromédico.com*) para obtener más información sobre patologías, parámetros en sangre, etc.

- ***Captar las acciones del usuario***

Otra función de este módulo es recoger las interacciones que el usuario realiza en la interfaz y generar las acciones pertinentes. Se debe garantizar que la interacción con la interfaz sea lo más sencilla e intuitiva posible, lo que conlleva una presentación clara de las acciones que el usuario puede realizar. Para realizar esto, se aplican técnicas de usabilidad que garanticen que los botones y demás elementos de la interfaz se localicen de forma correcta y visible para el usuario.

8.2.3. *Diseño detallado – Diseño de Bajo Nivel*

En este apartado se describe la implementación del sistema. Para ello se describe en detalle la implementación de cada uno de los módulos que componen el sistema.

- Módulo de almacenamiento de los datos
- Módulo de tratamiento de datos
- Módulo de recolección automatizada de los datos
- Módulo de sistema de cuestionarios
- Módulo de Gestión de Usuarios y Generación de Gráficas y Estadísticos
- Módulo de interfaz de usuario

8.2.3.1. Módulo de almacenamiento de los datos

Este módulo está formado por una base de datos relacional compuesta por 29 tablas cuyos nombres se detallan a continuación:

AfinaResultado
Alteraciones
AlteracionesCombinadas
AlteracionesPregunta
AlteracionesTom
Apartados
CIE10
CIE9
CIE9CIE10
Cuestiones
Enfermedades
Enlaces
EnlacesXTema
LOINC
LOINC_ES
Motivos

Parametros
ParametrosComun
ParametrosPregunta
ParametrosTom
Preguntas
Rangas
Resultados
Series
Sintomas
temaXindice
TipoUsuario
Unidades
Usuarios

Tabla 10. Tablas de la BBDD. Módulo de almacenamiento de datos.

A continuación se describen en detalle los campos de las tablas más importantes y las características de estos.

- **Tabla Parametros**

Campo	Tipo	NULL	Key	Default	Extra	Character set
<i>Id</i>	int (8)	NO	PRI	-	Auto_increment	-
<i>IdSeries</i>	int (8)	SI	-	NULL	-	-
<i>LOINC</i>	varchar (255)	SI	-	NULL	-	utf8
<i>Definicion</i>	varchar (255)	NO	-	-	-	utf8
<i>Descripcion</i>	text	SI	-	NULL	-	utf8
<i>Abreviatura</i>	varchar (255)	SI	-	NULL	-	utf8
<i>Sinonimo</i>	varchar (255)	SI	-	NULL	-	utf8
<i>DefinicionIngles</i>	varchar (255)	SI	-	NULL	-	utf8
<i>DescripcionIngles</i>	text	SI	-	NULL	-	utf8
<i>AbrevituraIngles</i>	varchar (255)	SI	-	NULL	-	utf8
<i>SinonimoIngles</i>	varchar (255)	SI	-	NULL	-	utf8
<i>Unidades</i>	varchar (255)	SI	-	NULL	-	utf8
<i>UnidadesMostrar</i>	varchar (255)	SI	-	NULL	-	utf8
<i>Especial</i>	varchar (255)	SI	-	NULL	-	utf8
<i>Comun</i>	int (8)	SI	-	NULL	-	-
<i>OrdenComun</i>	int (8)	SI	-	NULL	-	-
<i>Enlace</i>	varchar (255)	SI	-	NULL	-	utf8

Tabla 11. Tabla Parametros

A continuación se muestra una breve descripción de cada campo:

- **Id:** número que identifica unívocamente al parámetro.
- **Idseries:** serie a la que pertenece el parámetro (serie roja, serie blanca, etc.).
- **Loinc:** código loinc que identifica globalmente al parámetro.
- **Definicion:** nombre del parámetro.
- **Descripcion:** descripción con información extra de cada parámetro.
- **Abreviatura:** siglas con las que se puede denotar al parámetro.
- **Sinonimo:** otras formas de denominar al parámetro.
- **Definicioningles:** el nombre del parámetro en inglés.
- **Descripcioningles:** descripción con información extra en inglés.
- **Abrevituraingles:** siglas del parámetro en inglés.
- **Sinonimoingles:** otro nombre para el parámetro en inglés.
- **Unidades:** unidades en las que habitualmente se mide el parámetro.
- **Unidadesmostrar:** unidades tal y como se muestran en la interfaz de usuario.
- **Especial:** característica especial del parámetro (por ejemplo, si es un parámetro obtenido solo en una vena o en una arteria, etc.).
- **Comun:** flag para denotar que el parámetro pertenece a un análisis común.
- **Ordencomun:** orden de aparición en la interfaz de usuario en caso de ser un parámetro que pertenece a un análisis común.
- **Enlace:** url a una enciclopedia médica, con más información sobre el parámetro.

- **Rangos**

Campo	Tipo	NULL	Key	Default	Extra	Character set
<i>Id</i>	int (8)	NO	PRI	-	Auto_increment	-
<i>Sexo</i>	varchar (255)	SI	-	NULL	-	utf8
<i>Raza</i>	varchar (255)	SI	-	NULL	-	utf8
<i>Periodo</i>	varchar (255)	SI	-	NULL	-	utf8
<i>PeriodoMinimo</i>	int (8)	SI	-	NULL	-	-
<i>PeriodoMaximo</i>	int (8)	SI	-	NULL	-	-
<i>IdParametro</i>	int (8)	NO	-	-	-	-
<i>Parametro</i>	varchar (255)	SI	-	NULL	-	utf8
<i>RangoMinimo</i>	float (10)	NO	-	-	-	-
<i>RangoMaximo</i>	float (10)	NO	-	-	-	-
<i>Unidades</i>	varchar (255)	SI	-	NULL	-	utf8

A continuación se muestra una breve descripción de cada campo:

- **Id:** número que identifica unívocamente a un rango.
- **Sexo:** sexo de la persona a la que pertenece la analítica.

- **Raza:** raza de la persona a la que pertenece la analítica.
- **Periodo:** diferencia en días/meses/años de la realización de la analítica y de la edad del paciente.
- **Periodominimo:** rango mínimo asociado al periodo.
- **Periodomaximo:** rango máximo asociado al periodo.
- **Idparametro:** id del parámetro asociado a este rango.
- **Parametro:** nombre del parámetro asociado a este rango.
- **Rangominimo:** valor mínimo para que el parámetro se considere correcto.
- **Rangomaximo:** valor máximo para que el parámetro se considere correcto.
- **Unidades:** unidades asociados a los valores rango mínimo y máximo.

- **Unidades**

Campo	Tipo	NULL	Key	Default	Extra	Character set
<i>Id</i>	int (8)	NO	PRI	-	Auto_increment	-
<i>IdParametro</i>	int (8)	NO	-	-	-	-
<i>NombreParametro</i>	varchar (255)	SI	-	NULL	-	utf8
<i>Unidad</i>	varchar (255)	NO	-	-	-	utf8
<i>UnidadReferencia</i>	varchar (255)	NO	-	-	-	utf8
<i>Multiplicando</i>	varchar (255)	NO	-	-	-	utf8
<i>Prioridades</i>	varchar (255)	SI	-	NULL	-	utf8

A continuación se muestra una breve descripción de cada campo:

- **Id:** número que identifica unívocamente a una unidad.
- **Idparametro:** id del parámetro asociado a una unidad.
- **Nombreparametro:** nombre del parámetro asociado a una unidad.
- **Unidad:** unidad de medida asociada a un parámetro.
- **Unidadreferencia:** unidad en la que el parámetro ha sido registrado en la tabla parámetros.
- **Multiplicando:** unidad por la que se ha de multiplicar la unidad para obtener la unidad de referencia.
- **Prioridades:** marca, dentro de las posibles unidades de un mismo parámetro, cual es la de referencia.

- **Alteraciones**

Campo	Tipo	NULL	Key	Default	Extra	Character set
<i>Id</i>	int (8)	NO	PRI	-	Auto_increment	-
<i>IdParametro</i>	int (8)	NO	-	-	-	-
<i>Tipo</i>	varchar (255)	NO	-	-	-	utf8
<i>Motivo</i>	varchar (255)	NO	-	-	-	utf8
<i>Nombre</i>	varchar (255)	NO	-	-	-	utf8
<i>Sinonimo</i>	varchar (255)	SI	-	NULL	-	utf8
<i>NombreParametro</i>	varchar (255)	SI	-	NULL	-	utf8
<i>Unidades</i>	varchar (255)	SI	-	NULL	-	utf8
<i>IdCie10</i>	int (8)	SI	-	NULL	-	-

A continuación se muestra una breve descripción de cada campo:

- ***Id***: número que identifica unívocamente a una alteración.
- ***Idparametro***: id del parámetro asociado a esa alteración.
- ***Tipo***: marca que tipo de variación tiene ese parámetro (comúnmente marca cantidad).
- ***Motivo***: marca si es derivado de un exceso a de una escasez del parámetro.
- ***Nombre***: nombre de la alteración.
- ***Sinonimo***: otro nombre con el que se puede denominar a la alteración.
- ***Nombreparametro***: nombre del parámetro asociado a la alteración.
- ***Unidades***: unidades en las que esta medido ese parámetro para dar la variación.
- ***Idcie10***: id asociado a la tabla con la clasificación internacional de enfermedades en su versión 10, que contiene la alteración.

- **CIE10**

Campo	Tipo	NULL	Key	Default	Extra	Character set
<i>Id</i>	int (8)	NO	PRI	-	-	-
<i>IdMadre</i>	int (8)	NO	-	-	-	-
<i>Nivel</i>	int (8)	NO	-	-	-	-
<i>CIE10</i>	varchar (255)	NO	-	-	-	utf8
<i>Descripcion</i>	varchar (255)	NO	-	-	-	utf8
<i>RanMin</i>	varchar (255)	SI	-	NULL	-	utf8
<i>RanMax</i>	varchar (255)	SI	-	NULL	-	utf8
<i>LetraIni</i>	varchar (255)	SI	-	NULL	-	utf8
<i>LetraFin</i>	varchar (255)	SI	-	NULL	-	utf8

A continuación se muestra una breve descripción de cada campo:

- **Id:** número que identifica unívocamente a una enfermedad dentro del cie-10.
- **Idmadre:** número que identifica unívocamente la enfermedad de nivel superior dentro del cie-10.
- **Nivel:** nivel dentro de la jerarquía de clasificación de enfermedades.
- **Cie10:** código identificativo con los números y letras que clasificación a la enfermedad.
- **Descripcion:** nombre de la enfermedad.
- **Ranmin:** número menor que identifica el rango de valores dentro del código cie-10.
- **Ranmax:** número mayor que identifica el rango de valores dentro del código cie-10.
- **Letraini:** letra menor que identifica el rango de valores dentro del código cie-10.
- **Letrafin:** letra mayor que identifica el rango de valores dentro del código cie-10.

- **LOINC**

Esta tabla está compuesta por un total de 46 campos A continuación se detallan los 4 más importantes utilizados en el presente proyecto.

Campo	Tipo	NULL	Key	Default	Extra	Character set
LOINC_NUM	int (8)	NO	PRI	-	Auto_increment	-
LOINC_NUM_ESP	int (8)	NO	-	-	-	-
COMPONENT	varchar (255)	SI	-	NULL	-	utf8
SYSTEM	varchar (255)	NO	-	-	-	utf8

A continuación se muestra una breve descripción de cada campo:

- **Loinc_num:** código loinc que codifica el parámetro.
- **Loinc_num_esp:** código loinc que codifica el parámetro en su versión española.
- **Component:** nombre del parámetro dentro de la base de datos.
- **System:** localización del parámetro, por ejemplo, si es un parámetro obtenido en sangre, en la médula, procedente de un cultivo, etc.

- **Usuario**

Campo	Tipo	NULL	Key	Default	Extra	Character set
<i>Id</i>	int (8)	NO	PRI	-	Auto_increment	-
<i>IdTipoUsuario</i>	int (8)	NO	-	-	-	-
<i>Nombre</i>	varchar (255)	NO	-	-	-	utf8
<i>Apellidos</i>	varchar (255)	SI	-	NULL	-	utf8
<i>Email</i>	varchar (255)	NO	-	-	-	utf8
<i>Sexo</i>	varchar (255)	NO	-	-	-	utf8
<i>Raza</i>	varchar (255)	NO	-	-	-	utf8
<i>Contrasena</i>	text	NO	-	-	-	utf8
<i>FechaNacimiento</i>	varchar (255)	NO	-	-	-	utf8
<i>IdMédico</i>	int (8)	SI	-	NULL	-	-
<i>IdClínica</i>	int (8)	SI	-	NULL	-	-

A continuación se muestra una breve descripción de cada campo:

- ***Id***: número que identifica unívocamente al usuario.
- ***Idtipousuario***: tipo de usuario que se está registrando (particular, médico o clínica).
- ***Nombre***: nombre del usuario.
- ***Apellidos***: apellidos del usuario.
- ***Email***: email del usuario (ha de ser único en el sistema).
- ***Sexo***: sexo del usuario.
- ***Raza***: Raza del usuario.
- ***Contrasena***: contraseña con la que accede al sistema (hash sha-256).
- ***Fechanacimiento***: fecha de nacimiento del usuario.
- ***Idmédico***: en caso de estar gestionado por un médico, número que identifica unívocamente al médico.
- ***Idclínica***: en caso de estar gestionado por una clínica, número que identifica unívocamente a la clínica.

- **Resultados**

Campo	Tipo	NULL	Key	Default	Extra	Character set
<i>Id</i>	int (8)	NO	PRI	-	Auto_increment	-
<i>IdPaciente</i>	int (8)	NO	-	-	-	-
<i>IdMédico</i>	int (8)	NO	-	-	-	-
<i>IdClínica</i>	int (8)	NO	-	-	-	-
<i>FechaAnálisis</i>	date	NO	-	-	-	-
<i>NombreAnálisis</i>	varchar (255)	NO	-	-	-	utf8
<i>Análisis</i>	longtext	NO	-	-	-	utf8

A continuación se muestra una breve descripción de cada campo:

- **Id:** número que identifica unívocamente una analítica.
- **Idpaciente:** número de identificación del usuario al que pertenece la analítica.
- **Idmédico:** en caso de estar gestionado por un médico, da acceso a este para acceder a una analítica.
- **Idclínica:** en caso de estar gestionado por una clínica, da acceso a ésta para acceder a una analítica.
- **Fechaanálisis:** fecha en la que se realizó la analítica.
- **Nombreaanálisis:** nombre de la analítica asociado al usuario.
- **Análisis:** contiene el contenido del json en formato fhir.

- **Cuestiones**

Campo	Tipo	NULL	Key	Default	Extra	Character set
<i>Id</i>	int (8)	NO	PRI	-	Auto_increment	-
<i>Cuestion</i>	varchar (255)	NO	-	-	-	utf8

A continuación se muestra una breve descripción de cada campo:

- **Id:** número que identifica unívocamente una cuestión.
- **Cuestion:** almacena las diferentes cuestiones posibles.

- **Afina Resultado**

Campo	Tipo	NULL	Key	Default	Extra	Character set
<i>IdCuestion</i>	int (8)	NO	-	-	-	-
<i>Respuesta</i>	int (8)	NO	-	-	-	-
<i>IdAlteracion</i>	int (8)	NO	-	-	-	-
<i>Confirmado</i>	varchar (255)	NO	-	-	-	-

A continuación se muestra una breve descripción de cada campo:

- **Idcuestion:** número de identificación de la cuestión.
- **Respuesta:** respuesta que se ha de dar para considerarla valida.
- **Idalteracion:** número de identificación de la enfermedad asociada a una cuestión.
- **Confirmado:** peso de la pregunta que se tiene que aplicar en caso de ser validada.

8.2.3.2. Módulo de tratamiento de datos

Este módulo inicialmente captura los datos de los parámetros de la analítica enviados por la interfaz de usuario. Tras obtener los datos, el primer paso es convertir esos datos a formato JSON siguiendo el estándar FHIR; si es un usuario registrado se obtendrá de las *cookies* (el usuario tiene que estar previamente logeado) su nombre, apellidos, el id con el que se registró en el sistema y de ser necesario, el laboratorio que realizó la analítica y la URL de éste. A continuación se muestra una imagen de la obtención de estos parámetros de las Cookies del usuario.

```
$nombre=$_COOKIE["nombre"];
$apellidos=$_COOKIE["Apellidos"];
$idPaciente=$_COOKIE["idPaciente"];
$LaboratorioAnálisis=$_COOKIE["Laboratorio"];
$URLLaboratorioAnálisis=$_COOKIE["URLLab"];
```

Ilustración 16. Fragmento de código. Módulo de tratamiento de datos. Obtención de cookies.

Posteriormente se construye el fichero con todos los parámetros y los resultados obtenidos en la analítica de cada uno, así como los valores óptimos (rango máximo y mínimo). Cabe mencionar, que los parámetros que se encuentren alterados se señalizan según marca el estándar FHIR (N: Normal, L: Bajo, H: Alto). A continuación se muestra un fragmento del código encargado de realizar la codificación en formato FHIR.

```

array_push($pilaJSON, '
{
  "resourceType": "Observation",
  "id": "'. $valor. '",
  "status": "final",
  "code": {
    "coding": [
      {
        "system": "http://loinc.org",
        "code": "'. $loinc. '",
        "display": "'. $DesLOINC. '"
      }
    ],
    "text": "'. $definicion. '"
  },
  "subject": {
    "reference": "Paciente/' . $nombre. '_'. $idPaciente. '"
  },
  "performer": [
    {
      "reference": "Organization/1832473e-2fe0-452d-abe9-3cdb9879522f",
      "display": "'. $LaboratorioAnalisis. '"
    }
  ],
  "valueQuantity": {
    "value": "'. $CuantificadoParametros[$i]. '",
    "unit": "'. $UnidadesParam. '",
    "system": "http://unitsofmeasure.org",
    "code": "'. $UnidadesParam. '"
  },
  "interpretation": {
    "coding": [
      {
        "system": "http://hl7.org/fhir/v2/0078",
        "code": "'. $NivelParametros. '"
      }
    ]
  },
  "referenceRange": [
    {
      "low": {
        "value": "'. $RangoMinimo. '",
        "unit": "'. $UnidadesParam. '",
        "system": "http://unitsofmeasure.org",
        "code": "'. $UnidadesParam. '"
      },
      "high": {
        "value": "'. $RangoMaximo. '",

```

Ilustración 17. Fragmento de código. Módulo de tratamiento de datos. Generación del JSON(FHIR).

Seguido, este módulo se encarga de guardar el fichero JSON en la base de datos y, asignar en la *tabla resultados*, el nombre del paciente y el nombre del fichero generado. Una vez guardado el fichero, el siguiente paso es parsearlo y detectar cuáles son los parámetros que se encuentran alterados. Para realizar esto se genera un bucle que recorre el fichero JSON y comprueba el campo *code* de cada parámetro, si el campo es distinto de “N”, implica que ese parámetro está en una cantidad incorrecta. A continuación se muestra un fragmento de código en el que se realiza el parseado del fichero JSON (FHIR).

```
$jto=json_decode($JsonPaciente);
$ParametrosJson=$jto->{'contained'};
for($s=0;$s<count($ParametrosJson);$s++){
    $valorCuantificado=( $ParametrosJson[$s]->{'valueQuantity'});
    $valorMedido=( $valorCuantificado->{'value'});
    $idAlteradoPara=( $ParametrosJson[$s]->{'id'});
    $valorCodigoParam=( $ParametrosJson[$s]->{'interpretation'});
    $codigoParametro=( $valorCodigoParam->{'coding'});
    $valorCodigo=( $codigoParametro[0]);
    $desviacionParametro=( $valorCodigo->{'code'});
    if($desviacionParametro!="N"){
        $ParametrosIncorrectos[]=array( 'IdParametroAlterado' => $idAlteradoPara,
                                        'valorParametroCuantificado' => $valorMedido,
                                        'CodigoDesviacion' => $desviacionParametro
                                    );
    }
}
}
```

Ilustración 18. Fragmento de código. Módulo de tratamiento de datos. Parseo y detección de parámetros alterados.

Una vez detectados los parámetros alterados (se guardan en un array el nombre el parámetro alterado, el id de éste y el código de la desviación del estándar FHIR), se accede al módulo de almacenamiento de datos y se genera un ciclo en el que se va comprobando cuales son las posibles alteraciones y alteraciones combinadas derivadas de los parámetros alterados, así como enfermedades, en base a los *Ids* de éstos. En el caso de las alteraciones combinadas, se van escogiendo los parámetros de dos en dos y se comprueba si existe la combinación de ambos en la tabla *alteracionesCombinadas* y si no es así, se escoge otra pareja de parámetros, no escogida con anterioridad. Para explicar este proceso, A continuación se presenta un diagrama de flujo del funcionamiento de este módulo.

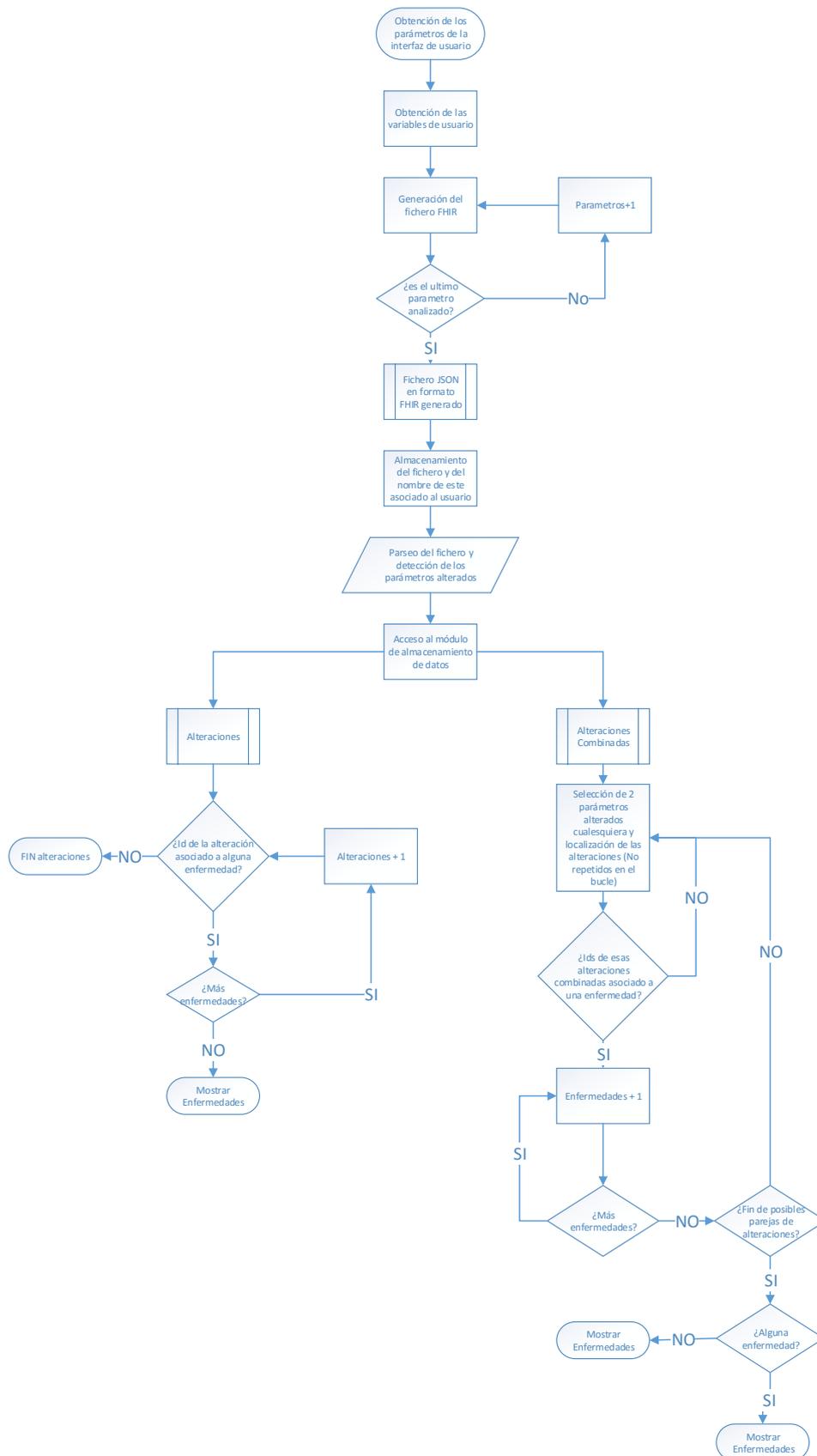


Ilustración 19. Diagrama de flujo. Módulo de tratamiento de datos.

A continuación se muestra un resultado tras un diagnóstico, con un parámetro en sangre alterado (Sodio).

The screenshot displays a web application interface with a blue header containing navigation links: tuotroanálisis, SERVICIOS, NUESTRAS PAGINAS, CONÓCENOS, EL EQUIPO, NOTICIAS, CONTÁCTANOS, REGÍSTRATE, and INICIA SESIÓN. The main content area is white and contains the following sections:

- Resultado de su analítica**: Los valores obtenidos de los siguientes parámetros están alterados:
 - Sodio en la sangre por debajo de lo normal
 - Su valor es: 125 mmol/L
 - Los valores de referencia se sitúan entre 137 y 147 mmol/L
 - A su situación se le denomina HIPONATREMIA
- Hiponatremia**: La hiponatremia es una afección en la cual la cantidad de sodio (sal) en la sangre es más baja de lo normal. El sodio (Na) es el ión positivo que se encuentra principalmente, fuera de las células, en los fluidos extracelulares del cuerpo humano y es muy importante para mantener la presión arterial. El sodio también se necesita para que los nervios, los músculos y otros tejidos corporales trabajen apropiadamente. La concentración de sodio en la sangre es el resultado entre la entrada del mismo por la dieta y su salida a través del filtrado del riñón y su salida por la orina.
- Para qué se realiza el análisis de sodio en sangre**: Se utiliza para evaluar la función de los riñones, de las diferentes hormonas que lo regulan, y situaciones de la regulación de líquidos en el cuerpo humano.
- Causas externas que pueden alterar los valores de sodio en sangre**:
 - Diarrea
 - Vómitos
 - Exceso de sudoración
 - Intervenciones recientes de cirugía
 - Medicamentos
 - Diuréticos
 - Vasopresina
- Posibles enfermedades que cursan con hiponatremia**: Los niveles disminuidos de Sodio en la sangre pueden indicar:
 - Ascitis
 - Cetoacidosis diabética
 - Derrame pleural
 - Enfermedad de Addison
 - Enfermedades renales
 - Edema periférico
 - Insuficiencia cardíaca congestiva
 - Síndrome de secreción de hormona antidiurética

Below the text is a blue button labeled "Ajustar Resultado". At the bottom of the page is a blue footer with the text "tuotroanálisis" and a row of social media icons: email, Twitter, Facebook, LinkedIn, and Telegram.

Ilustración 20. Fragmento de código. Módulo de tratamiento de datos. Diagnóstico realizado.

8.2.3.3. Módulo de recolección automatizada de los datos

Este módulo proporciona una interfaz entre el usuario y el módulo de tratamiento de datos, que facilita y automatiza la obtención de los datos referentes a los diagnósticos. Inicialmente el usuario debe seleccionar un fichero de su sistema de archivos y tras esto, el módulo valida el formato del fichero (solo se permiten ficheros en formato JSON (FHIR)). Posteriormente, el módulo guarda el fichero en el sistema de archivos del servidor y el nombre dicho fichero asociado al usuario (el usuario debe estar registrado para utilizar esta funcionalidad del sistema) en la base de datos. El código de este módulo está localizado en 2 paquetes, como se explica en la fase de integración. A continuación se muestra un fragmento del código encargado del guardado del fichero en el servidor alojado en la capa de datos, una imagen de la interfaz de usuario y un diagrama de flujo que explica el funcionamiento de este módulo.

```
<?php
//datos del archivo
$nombre_archivo = $_FILES['FicheroUp']['name'];
$tipo_archivo = $_FILES['FicheroUp']['type'];
$tamano_archivo = $_FILES['FicheroUp']['size'];
$temporal=$_FILES['FicheroUp']['tmp_name'];
echo $nombre_archivo."<br>".$tipo_archivo."<br>";
//compruebo si las características del archivo son las que deseo
if (!(strpos($tipo_archivo, "octet-stream"))) {
    echo "La extensión del archivo no es correcta. <br><br><table><tr><td><li>Se permiten archivos .json<br></td></tr></table>";
}else{
    echo $temporal."<br>";
if (move_uploaded_file($temporal, "ficheros/".$nombre_archivo)){

    echo "El archivo ha sido cargado correctamente.";
}else{
    echo "Ocurrió algún error al subir el fichero. No pudo guardarse.";
}
}
?>
```

Ilustración 21. Fragmento de código. Módulo de recolección automatizada de los datos. Subida del fichero.

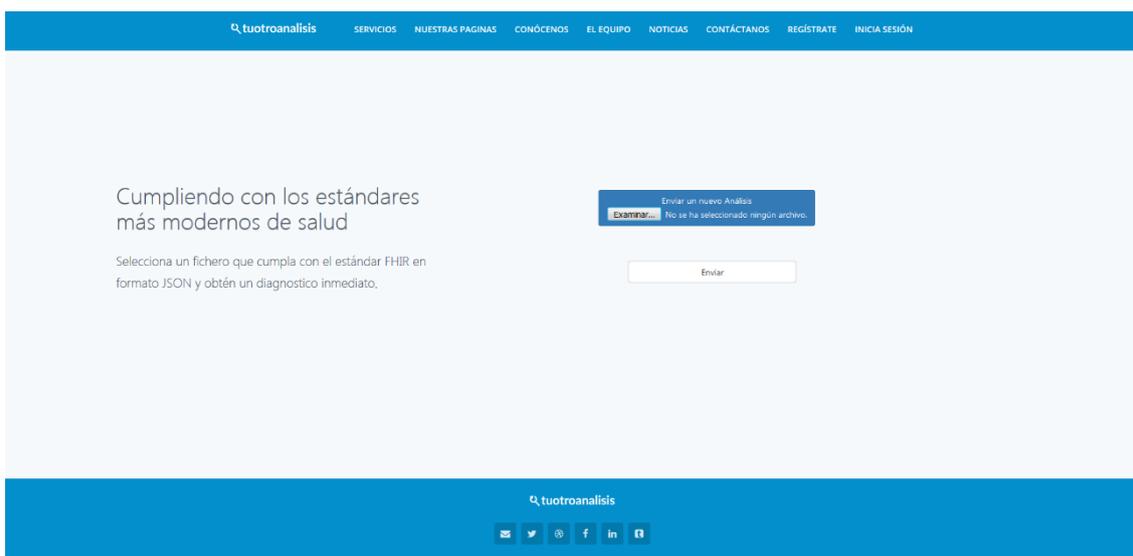


Ilustración 22. Interfaz de usuario. módulo de recolección automatizada de datos.



Ilustración 23. Diagrama de flujo. Módulo de recolección automatizada de datos.

8.2.3.4. Módulo de sistema de cuestionarios

El módulo de sistema de cuestionarios se encarga de afinar los resultados obtenidos por el módulo de tratamiento de datos, a través de una serie de cuestiones realizadas al usuario.

Para su funcionamiento se utilizan 2 tablas del módulo de almacenamiento de datos, *AfinaResultado* y *Cuestiones*, explicadas en el diseño del correspondiente módulo. Inicialmente, se accede a la tabla *AfinaResultado* y se extraen de ella todos Id de las cuestiones asociadas a las enfermedades que el módulo de tratamiento de datos ha diagnosticado previamente, así como, el peso (columna *Confirmado* de la tabla *AfinaResultado*) y cuando se aplica dicho peso, es decir, si la respuesta es afirmativa o negativa (columna *Respuesta* de la tabla *AfinaResultado*).

Posteriormente, se recogen los resultados de las respuestas del usuario y se aplica el peso a aquellas preguntas que coinciden con la respuesta asociada. Por último, se pasa la enfermedad que más peso tiene a la interfaz de usuario para mostrársela al paciente.

A continuación se muestra un ejemplo de cuestionario obtenido en base a la desviación del parámetro del ejemplo mostrado en el diseño de bajo nivel del módulo de tratamiento de datos.

tuotroanálisis SERVICIOS NUESTRAS PAGINAS CONÓCENOS EL EQUIPO NOTICIAS CONTÁCTANOS REGÍSTRATE INICIA SESIÓN

¿Tiene o ha tenido fiebre?

Si No

¿Tiene o ha tenido dificultad para respirar?

Si No

¿Tiene o ha tenido dolor torácico?

Si No

¿Tiene o ha tenido hipo?

Si No

Finalizar

tuotroanálisis

✉ 🐦 🌐 f in 📷

Ilustración 24. Módulo de sistema de cuestionario. Interfaz de las preguntas.

Tras las respuestas afirmativas en las 4 cuestiones anteriores, se obtiene un resultado más ajustado que el ejemplo mostrado en la imagen del módulo de tratamiento de datos. A continuación se muestra una imagen del resultado de dicho cuestionario.

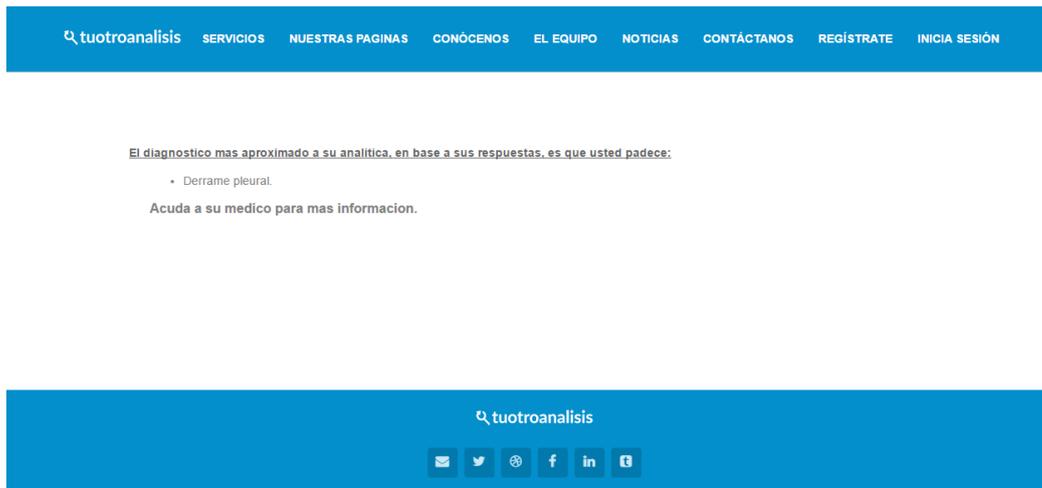


Ilustración 25. Módulo de sistema de cuestionario. Resultado del cuestionario.

8.2.3.5. Módulo de Gestión de Usuarios y Generación de Gráficas y Estadísticos

Este módulo realiza dos funciones, permite el registro y gestión de los usuarios y, muestra gráficas de alguno de los parámetros existentes en la analítica de un paciente.

- Gestión de usuarios:

En la página principal del sistema, se facilita al usuario un formulario de registro para que introduzca los datos necesarios para el diagnóstico, almacenamiento y gestión de sus análisis. A continuación se muestra una imagen de dicho formulario.

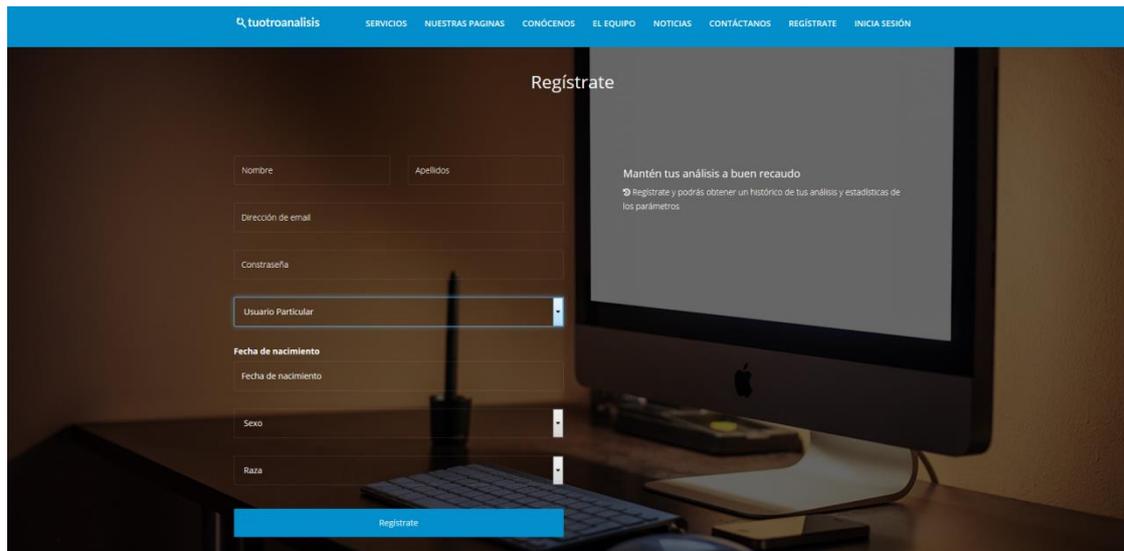


Ilustración 26. Módulo de Gestión de Usuarios y Generación de Gráficas y Estadísticos. Formulario de registro.

Tras introducir el usuario todos los valores, se comprueba mediante atributos de *HTML5* que los campos están rellenos y que tienen un formato correcto (por ejemplo, que el email contenga una @, un dominio, etc.). Una vez validados los datos, se serializan y mediante el método `$.getJSON` de *AJAX*, se envían a la capa de datos para que ésta se encargue de registrar al usuario. A continuación se muestra una imagen del código encargado de serializar los datos y enviarlos a la capa de datos y, una imagen del código alojado en la base de datos que se encarga de registrar al usuario.

```
function registraUsu(){
    var variables=$("#registercontactform").serialize();
    $.getJSON("http://www.tuotroanalysis.com/registraUsuarioNuevo.php?" + variables, function(jsonData){
        $.each(jsonData, function(i,data){
            var insertado=data.insertado;
            if(insertado=="false"){
                alert("Problemas al registrase, inténtelo de nuevo");
            }
            else{
                alert("Registrado correctamente, inicie sesión");
            }
        });
    });
}
```

Ilustración 27. Módulo de Gestión de Usuarios y Generación de Gráficas y Estadísticos. Envío a la capa de datos.

```
$name=$_GET["nombre"];
$apelli=$_GET["apellidos"];
$email=$_GET["email"];
$contra=$_GET["contrasena"];
$contrasena=hash('sha256', $contra);
$perfil=$_GET["perfilUsuario"];
$anonaci=$_GET["nacimiento"];
$sexo=$_GET["sexo"];
$raza=$_GET["raza"];
if($perfil=="Particular"){ $idperfil="1";}
elseif($perfil=="Medico"){ $idperfil="2";}
elseif($perfil=="Clinica"){ $idperfil="3";}
$sql="INSERT INTO Usuarios (IdTipoUsuario,Nombre,Apellidos,Email,Sexo,Raza,Contrasena,FechaNacimiento,idMedico,idClinica) VALUES
('$idperfil','$name','$apelli','$email','$sexo','$raza','$contrasena','$anonaci',NULL,NULL)";
$result = mysqli_query ($link,$sql);
if( $result ){ $insertado[] = array('insertado'=>"true"); }
else{ $insertado[] = array('insertado'=>"false"); }
echo json_encode($insertado);
mysqli_close($link);
?>
```

Ilustración 28. Fragmento de código de la capa de datos que registra al usuario.

Como se puede apreciar en la imagen (señalizado con una flecha azul), para almacenar la contraseña del usuario se realiza un HASH (SHA-256) de esta.

- Generación de gráficos:

Para la generación de las gráficas, este módulo utiliza la librería *HighCharts* de JavaScript. Inicialmente se parsea cada una de las analíticas almacenadas del usuario (con el mismo método que se utiliza y se ha explicado en el módulo de tratamiento de datos y se extrae de ellos, tanto la fecha en la que se realizaron las analíticas, como el valor cuantificado del parámetro elegido por el usuario y se almacena en un array bidimensional (cada valor cuantificado, asociado con su fecha). Posteriormente, se convierte la fecha en un *Timestamp* [sello de tiempo que devuelve la fecha medida como el número de segundos desde la Época Unix (1 de enero de 1970 00:00:00 GMT)] y se separa en 2 arrays del mismo número de posiciones (la librería *HighCharts* requiere recibir de esta forma los datos). A continuación se muestra un fragmento de código de este proceso.

```
$valorCuantificado;  
$fechaAnalisis;  
for($i = 0 ;$i<count($bidiDatos);$i++){  
    $valorCuantificado[$i]= $bidiDatos[$i][1];  
    $tiempo= $bidiDatos[$i][2];  
    $fechaUnix = new DateTime($tiempo);  
    $fechaAnalisis[$i] = $fechaUnix->getTimestamp()*1000;  
}
```

Ilustración 29. Generación de los arrays con los valores del parámetro y la fecha.

Por último, se le pasan a la librería *HighCharts* los parámetros de las analíticas, así como el nombre que queremos darle a la gráfica, entre otros parámetros. A continuación se muestra un fragmento de código de este proceso y el resultado de la gráfica obtenida.

```
<script>  
chartCPU = new Highcharts.StockChart({  
    chart: {  
        renderTo: 'muestraGrafica'  
    },  
    rangeSelector: {  
        enabled: false  
    },  
    title: {  
        text: '<?php echo $nombreParametro;?>'  
    },  
    xAxis: {  
        type: 'datetime'  
    },  
    yAxis: {  
        minPadding: 0.5,  
        maxPadding: 0.5,  
        title: {  
            text: 'rango',  
            margin: 10  
        }  
    },  
    series: [{  
        name: 'valor',  
        data: (function() {  
            var data = [];  
            <?php  
            for($i = 0 ;$i<count($valorCuantificado);$i++){  
                ?>  
                data.push(['<?php echo $fechaAnalisis[$i];?>,<?php echo $valorCuantificado[$i];?>'];  
                <?php } ?>  
            return data;  
        })()  
    },  
    credits: {  
        enabled: false  
    }  
});  
</script>
```

Ilustración 30. Fragmento de código que genera las gráficas.

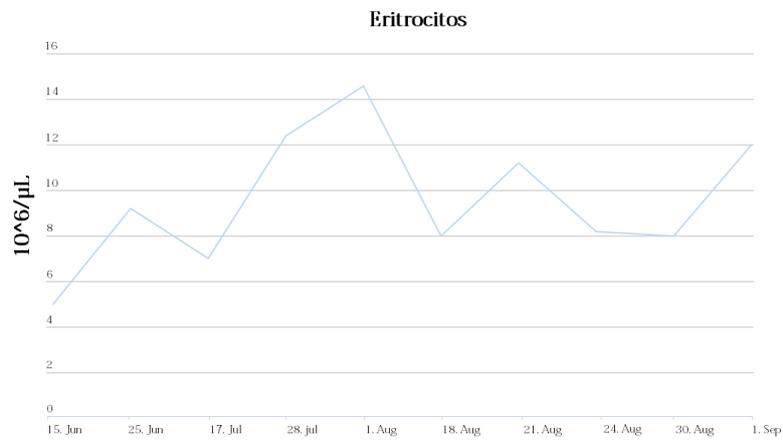


Ilustración 31. Grafica ejemplo obtenida.

8.2.3.6. Módulo de interfaz de usuario

Este módulo posibilita al usuario navegar a través del sistema. Para la codificación, se ha utilizado principalmente:

- *HTML5*: para la generación de las páginas y elementos visuales con los que el usuario puede interactuar.
- *BOOTSTRAP*: para generar elementos que se acoplen a cualquier tipo de pantalla.
- *CSS3*: para dar estilos a los diferentes elementos de las páginas.
- *JQUERY*: para dinamizar la interfaz y permitir generar elementos que interactúen con el usuario.
- *AJAX*: para comunicar el módulo de interfaz de usuario con la capa de datos.

A continuación se muestran unas imágenes con los elementos más importantes de la interfaz de usuario. Cabe mencionar, que los servicios que ofrece el sistema han sido descritos en el diseño de alto nivel del presente módulo.

- Pantalla de bienvenida: al dar en el botón comenzar, se desliza la interfaz de forma automática hasta los servicios que ofrece el sistema.



Ilustración 32. Pantalla de Bienvenida.

A continuación se muestra un fragmento de código del *carousel*(*slider*) mostrado en la imagen anterior.

```
<header id="home">
  <div id="home-slider" class="carousel slide carousel-fade" data-ride="carousel">
    <div class="carousel-inner">
      <div class="item active" style="background-image: url(images/slider/1.jpg)">
        <div class="caption">
          <h1 class="animated fadeInLeftBig">Bienvenido a <span>Tu Otro Analisis</span></h1>
          <p class="animated fadeInRightBig">La mejor forma de estar informado</p>
          <a data-scroll class="btn btn-start animated fadeInUpBig" href="#services">Comenzar</a>
        </div>
      </div>
      <div class="item" style="background-image: url(images/slider/2.jpg)">
        <div class="caption">
          <h1 class="animated fadeInLeftBig">Mantén a buen recaudo<span> tus análisis</span></h1>
          <p class="animated fadeInRightBig">Y estate al día de tu evolución</p>
          <a data-scroll class="btn btn-start animated fadeInUpBig" href="#services">Comenzar</a>
        </div>
      </div>
      <a class="left-control" href="#home-slider" data-slide="prev"><i class="fa fa-angle-left"></i></a>
      <a class="right-control" href="#home-slider" data-slide="next"><i class="fa fa-angle-right"></i></a>
      <a id="tohash" href="#services"><i class="fa fa-angle-down"></i></a>
    </div>
  </div>
```

Ilustración 33. Fragmento de código del slider de bienvenida.

- Análítica: interfaz en la que el usuario puede introducir los valores de su analítica, se le facilita la opción de introducir el valor, deslizando con el ratón/dedo o introducirlo a través del teclado. Adicionalmente, se le posibilita el cambiar la escala de las unidades (cada laboratorio escoge en que unidades ofrece las analíticas a sus pacientes).

Info	Parámetro	Rango	Valor Actual	Elegir Unidades	Eliminar
?	Eritrocitos	4.950	4.950	106/μL	x
?	Hemoglobina	146.000	146.000	g/L	x
?	Hematocritos	45.500	45.500	%	x
?	VCM	88.000	88.000	μ3	x
?	HCM	28.500	28.500	pg	x
?	CHCM	33.000	33.000	%	x
?	RDW	13.000	13.000	%	x

Info	Parámetro	Rango	Valor Actual	Elegir Unidades	Eliminar
?	Plaquetas	245.000	245.000	103/μL	x
?	VPM	8.800	8.800	μ3	x

Ilustración 34. Interfaz de la analítica.

Además, deslizando el ratón/dedo por encima de la interrogación situada a la izquierda de cada parámetro se ofrece al usuario información sobre dicho parámetro. A continuación se muestra una imagen que ilustra esto.

Info	Parámetro	Rango	Valor Actual	Elegir Unidades	Eliminar
?	Eritrocitos			106/μL	x
?	Hemoglobina			g/L	x
?	Hematocritos	45.500	45.500	%	x
?	VCM	88.000	88.000	μ3	x
?	HCM	28.500	28.500	pg	x
?	CHCM	33.000	33.000	%	x

Ilustración 35. Información extra de los parámetros.

Como elementos extra, *externos al presente proyecto*, también se ofrecen en la interfaz una serie de servicios:

- Otras páginas asociadas al sistema: ofrece la posibilidad al usuario de encontrar información sobre enfermedades y los médicos que los tratan en diferentes países.



Ilustración 36. Páginas asociadas.

A continuación se muestra un fragmento de código, asociado a la imagen anterior.

```
<section id="services">
  <div class="container">
    <div class="heading wow fadeInUp" data-wow-duration="1000ms" data-wow-delay="300ms">
      <div class="row">
        <div class="text-center col-sm-8 col-sm-offset-2">
          <h2>Nuestros Servicios</h2>
        </div>
      </div>
    </div>
    <div class="text-center our-services">
      <div class="row">
        <div class="col-sm-4 wow fadeInDown" data-wow-duration="1000ms" data-wow-delay="300ms">
          <a href="InicioBasico.php"><div class="service-icon">
            <i class="fa fa-flask"></i>
          </div></a>
          <div class="service-info">
            <h3>Analítica básica</h3>
            <p>En una analítica básica con componentes como el sodio, el potasio y otros componentes de la sangre como la zinc</p>
          </div>
        </div>
        <div class="col-sm-4 wow fadeInDown" data-wow-duration="1000ms" data-wow-delay="450ms">
          <div class="service-icon">
            <i class="fa fa-flask"></i>
          </div>
          <div class="service-info">
            <h3>Analítica a medida</h3>
            <p>Escoja de que componentes quiere realizar un diagnóstico de entre los más de 250 parámetros existentes en la sangre</p>
          </div>
        </div>
        <div class="col-sm-4 wow fadeInDown" data-wow-duration="1000ms" data-wow-delay="550ms">
          <div class="service-icon">
            <i class="fa fa-flask"></i>
          </div>
        </div>
      </div>
    </div>
  </div>
</section>
```

Ilustración 37. Fragmento de código de los servicios.

- Noticias sobre salud: también se ofrece al usuario, la posibilidad de estar informado con noticias sobre el sector de la medicina.



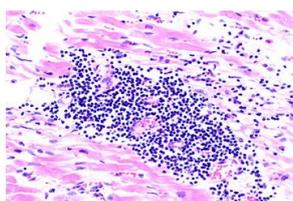
Noticias



Diez consejos para controlar la diabetes

JUNIO 03, 2016 EN DIABETES

La Organización Mundial de la Salud (OMS) estima que unos 350 millones de personas en todo el mundo tienen diabetes y considera que la cifra podría duplicarse en las próximas dos décadas.



Hacer frente a las enfermedades no transmisibles

DICIEMBRE 30, 2015 EN HIPERTENSION

Las cardiopatías, la diabetes y el cáncer suman dos tercios de las muertes en el planeta. No solo destruyen vidas; también obstaculizan el crecimiento económico.



Ingerir bebidas muy calientes "probablemente" cause cáncer, según la OMS

JUNIO 15, 2016 EN CANCER

Una revisión de más de mil estudios concluye que no hay evidencia de que el café ni el mate sean carcinógenos ingeridos a una temperatura normal.

Ilustración 38. Noticias del sector salud.

8.2.4. Codificación

En esta fase se ha procedido a la programación de los módulos anteriormente expuestos. Cabe mencionar que el sistema ha sido codificado realizando una separación por capas, esto es, se ha dividido la arquitectura en 3 capas:

- ***La capa de negocio (BL):*** en esta capa se reciben las peticiones del usuario y se envían las respuestas tras el procesado de las mismas. Esta capa se comunica con la capa de presentación para recibir las solicitudes y presentar los resultados y con la capa de datos para solicitar los datos necesarios.
- ***La capa de datos (DL):*** esta capa es donde residen los datos y es la encargada de acceder a los mismos. Esta capa accederá a los datos para recuperarlos o guardarlos cuando se lo indique la capa de negocio.
- ***La capa de presentación (PL):*** esta capa es conocida también como interfaz gráfica y presenta el sistema al usuario, comunicándole a este la información que devuelve la capa de negocio.

El dividir la arquitectura en estas tres capas o tiers proporciona los siguientes beneficios:

- Los elementos críticos de la lógica de negocio están ubicados más cerca de la capa de datos, con lo que se consigue que solo los datos realmente necesarios se presenten al usuario.
- Mayor flexibilidad y modularidad.
- Escalabilidad, lo que facilita añadir recursos para soportar mayor número de usuarios.
- Facilidades de desarrollo y administración, lo que facilita la reutilización de componentes y el aislamiento frente a cambios en otras capas

A continuación se muestra la arquitectura básica de la herramienta a nivel lógico, dividida en tres capas.

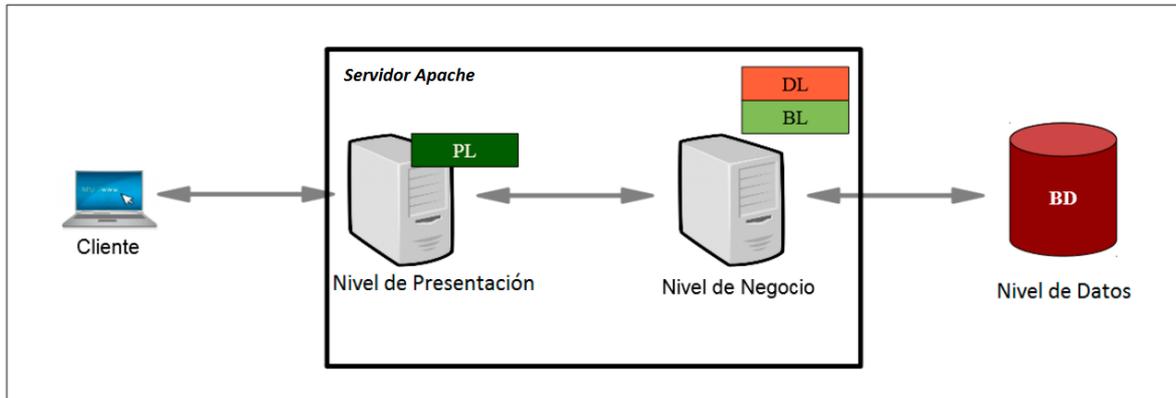


Ilustración 39.Arquitectura formada por 3 capas

8.2.5. Pruebas unitarias

Para la realización de las pruebas unitarias de cada uno de los módulos, se ha utilizado las siguientes técnicas para validar el correcto funcionamiento de cada uno de ellos:

- ***Pruebas de Caja Negra:*** se centran principalmente en probar las funcionalidades de un módulo; se limitan a introducir datos a la entrada del módulo y estudiar las salidas correspondientes, sin importar lo que ocurre en el interior.
- ***Pruebas de usabilidad:*** se centra en la medida de la facilidad de uso de un de un módulo. Se encarga de todo lo que influya en el éxito y la satisfacción del usuario.
- ***Pruebas de estrés:*** se centra en determinar la solidez de la aplicación en momentos de carga extrema y ayuda a determinar si la aplicación rendirá lo suficiente en caso sobrecarga.
- ***Pruebas de revisión de código:*** consiste en una revisión sistemática del código, de forma que se estructure correctamente y se mejore la calidad de este.

En concreto para los módulos de recolección automatizada de los datos, tratamiento de datos y gestión de usuarios y generación de gráficas y estadísticos se han realizado principalmente pruebas de caja negra. Para el módulo de almacenamiento de los datos se han realizado pruebas de estrés. Para el módulo de sistema de cuestionarios se han realizado principalmente pruebas de revisión de código y pruebas de caja negra. Para el módulo de interfaz de usuario se han realizado principalmente pruebas de usabilidad.

Módulo de recolección automatizada de los datos

En el módulo de recolección automatizada de los datos se han realizado pruebas subiendo varios ficheros al servidor y comprobando que la subida se realizaba correctamente y que conservan el formato y la estructura.

En la siguiente imagen se puede apreciar que la subida de varios ficheros se ha realizado de forma satisfactoria.



Ilustración 40. Prueba módulo de recolección automatizada de datos. Ficheros.

Además se comprobó que mantenía correctamente el formato. A continuación se muestra un fragmento del fichero .json subido.

```
    },
    "subject": {
      "reference": "Paciente/David_1"
    },
    "performer": [
      {
        "reference": "Organization/1832473e-2fe0-452d-abe9-3cdb9879522f",
        "display": "Laboratorio Falso de Pruebas, Inc"
      }
    ],
    "valueQuantity": {
      "value": 4.95,
      "unit": "10<sup>6</sup>/µL",
      "system": "http://unitsofmeasure.org",
      "code": "10<sup>6</sup>/µL"
    },
    "interpretation": {
      "coding": [
        {
          "system": "http://hl7.org/fhir/v2/0078",
          "code": "N"
        }
      ]
    },
    "referenceRange": [
      {
        "low": {
          "value": 4.200,
          "unit": "10<sup>6</sup>/µL",
          "system": "http://unitsofmeasure.org",
          "code": "10<sup>6</sup>/µL"
        },
        "high": {
          "value": 5.700,
          "unit": "10<sup>6</sup>/µL",
          "system": "http://unitsofmeasure.org",
          "code": "10<sup>6</sup>/µL"
        }
      }
    ]
  }
}
```

Ilustración 41. Prueba del módulo de recolección automatizada de datos. Formato.

Módulo de tratamiento de datos

En las pruebas de este módulo se han introducido valores alterados en el formulario de adquisición manual de datos y se ha comprobado que la salida correspondía con el diagnóstico derivado de la variación de dichos parámetros. Se comprobó que las salidas eran correctas y que se obtenían los valores deseados.

Módulo de gestión de usuarios y generación de gráficas y estadísticos

En las pruebas de este módulo se realizaron por una parte registros de usuarios y se comprobó que se almacenaban correctamente en el módulo de almacenamiento de datos y por otro se generaron valores aleatorios para

comprobar que las gráficas se realizaban correctamente. A continuación se muestra un ejemplo de un usuario que se ha registrado correctamente.

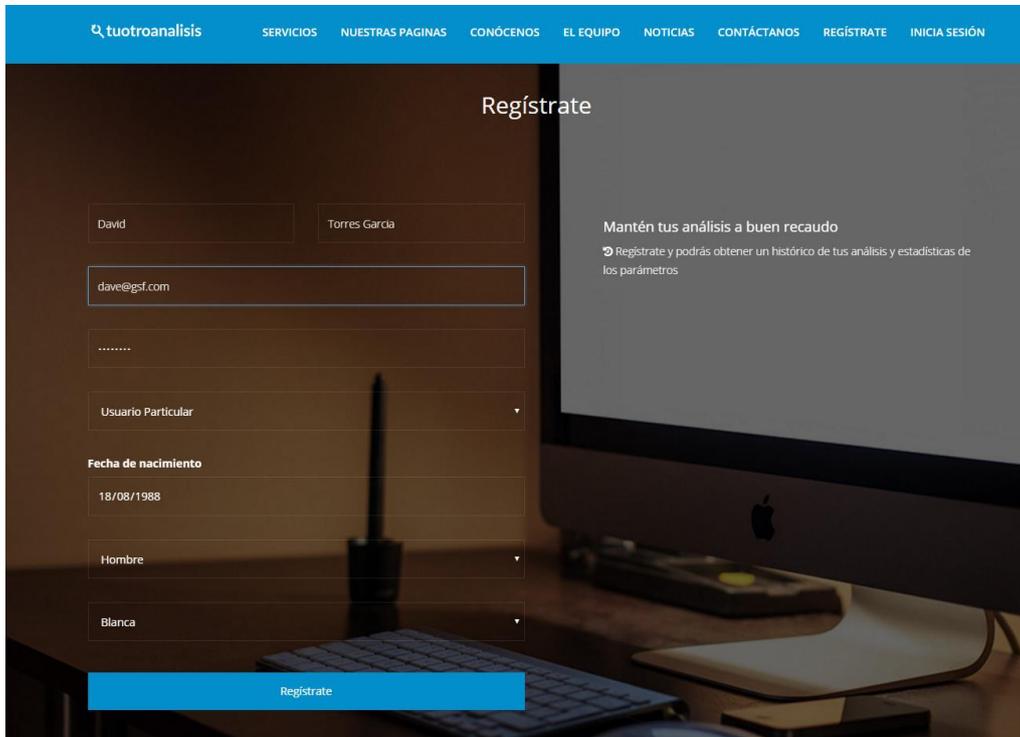


Ilustración 42. Formulario de registro.

Id	IdTipoUsuario	Nombre	Apellidos	Email	Sexo	Raza	Contraseña	FechaNacimiento	IdMedico	IdClinica
1	1	David	Torres García	dave@gsf.com	Hombre	Blanca	e30044298f1c149a8bf4c0996fb92427ae11e4649b934ca495991b7852b055	1988-08-18	(null)	(null)

Ilustración 43. Usuario registrado en la BBDD.

Módulo de almacenamiento de los datos

Para las pruebas de estrés de este módulo se ha utilizado la herramienta *Apache JMeter*, ya que esta herramienta permite realizar peticiones simultáneas, simulando ser un gran número de usuarios. Cabe mencionar, que estas pruebas se han realizado sobre un subdominio del sistema en el que se realiza un acceso al módulo de almacenamiento de datos. A continuación se muestra una imagen con una prueba realizada con 500 usuarios simultáneos en un tiempo de 20 segundos.

Muestra #	Nombre del hilo	Etiqueta	Tiempo de Muestra (ms)	Estado	Bytes	Latency	Connect Time(ms)
505	Grupo de Hilos 1-465	Petición HTTP	109	✔	44209	54	24
506	Grupo de Hilos 1-466	Petición HTTP	103	✔	44209	51	24
507	Grupo de Hilos 1-467	Petición HTTP	112	✔	44209	59	30
508	Grupo de Hilos 1-468	Petición HTTP	104	✔	44209	56	25
509	Grupo de Hilos 1-469	Petición HTTP	104	✔	44209	51	25
510	Grupo de Hilos 1-470	Petición HTTP	110	✔	44209	57	27
511	Grupo de Hilos 1-471	Petición HTTP	109	✔	44209	55	26
512	Grupo de Hilos 1-472	Petición HTTP	113	✔	44209	56	26
513	Grupo de Hilos 1-473	Petición HTTP	108	✔	44209	53	23
514	Grupo de Hilos 1-474	Petición HTTP	105	✔	44209	52	25
515	Grupo de Hilos 1-475	Petición HTTP	109	✔	44209	56	28
516	Grupo de Hilos 1-476	Petición HTTP	117	✔	44209	63	25
517	Grupo de Hilos 1-477	Petición HTTP	107	✔	44209	53	26
518	Grupo de Hilos 1-478	Petición HTTP	112	✔	44209	55	25
519	Grupo de Hilos 1-479	Petición HTTP	111	✔	44209	57	28
520	Grupo de Hilos 1-480	Petición HTTP	109	✔	44209	55	25
521	Grupo de Hilos 1-481	Petición HTTP	105	✔	44209	53	25
522	Grupo de Hilos 1-482	Petición HTTP	110	✔	44209	57	29
523	Grupo de Hilos 1-483	Petición HTTP	110	✔	44209	55	26
524	Grupo de Hilos 1-484	Petición HTTP	109	✔	44209	53	27
525	Grupo de Hilos 1-485	Petición HTTP	113	✔	44209	58	28
526	Grupo de Hilos 1-486	Petición HTTP	111	✔	44209	57	29
527	Grupo de Hilos 1-487	Petición HTTP	107	✔	44209	54	24
528	Grupo de Hilos 1-488	Petición HTTP	118	✔	44209	57	28
529	Grupo de Hilos 1-489	Petición HTTP	110	✔	44209	55	26
530	Grupo de Hilos 1-490	Petición HTTP	118	✔	44209	63	33
531	Grupo de Hilos 1-491	Petición HTTP	113	✔	44209	57	28
532	Grupo de Hilos 1-492	Petición HTTP	114	✔	44209	62	25
533	Grupo de Hilos 1-493	Petición HTTP	112	✔	44209	56	28
534	Grupo de Hilos 1-494	Petición HTTP	109	✔	44209	56	25
535	Grupo de Hilos 1-495	Petición HTTP	111	✔	44209	56	26
536	Grupo de Hilos 1-496	Petición HTTP	111	✔	44209	56	26
537	Grupo de Hilos 1-497	Petición HTTP	114	✔	44209	62	29
538	Grupo de Hilos 1-498	Petición HTTP	111	✔	44209	54	25
539	Grupo de Hilos 1-499	Petición HTTP	108	✔	44209	56	27
540	Grupo de Hilos 1-500	Petición HTTP	108	✔	44209	57	26

Ilustración 44. Acceso simultaneo al módulo de almacenamiento de datos

Como se puede apreciar en la imagen anterior, todas las peticiones se han realizado correctamente y en una media de aproximadamente 0.1 segundos.

Módulo de sistema de cuestionarios

Para la realización de las pruebas de este módulo se ha realizado, por una parte, una revisión de la estructura del código. Tras la revisión del código de este módulo se ha comprobado que no existían errores de síntesis y que la estructuración era la correcta. Y por otra, se han generado una serie de respuestas (afirmativas y negativas) para varias preguntas y se ha comprobado que la salida del módulo era la esperada, es decir, que las enfermedades que se ajustaban tras las respuestas de dichas preguntas eran las más idóneas.

Módulo de interfaz de usuario

En cuanto a las pruebas del módulo de interfaz de usuario, como se ha comentado anteriormente, se han realizado pruebas de usabilidad, comprobando que la localización de los diferentes botones y elementos con los que el usuario puede interactuar, era la adecuada y que el uso de la interfaz era intuitivo y sencillo.

Tras estas pruebas se ha concluido que la interfaz es sencilla y usable por cualquier tipo de perfil de usuario.

8.2.6. Integración

En esta fase se han integrado los distintos módulos para conformar el sistema que se contempla en este proyecto. Se han creado varios paquetes para distribuir los elementos que conforman cada módulo.

Para el *Módulo de recolección automatizada de los datos* se han creado dos paquetes. Uno aloja los ficheros correspondientes a la capa de presentación (PL), ya que será necesario mostrar una interfaz en la que el usuario pueda seleccionar su fichero. Otro corresponde con la capa de datos (DL), puesto que será necesario que acceda al sistema de ficheros para almacenar los .json.

Para el *Módulo de tratamiento de datos* se han creado dos paquetes. Uno aloja los ficheros correspondientes a la capa de presentación (PL), ya que será necesario mostrar los resultados del diagnóstico al usuario. Otro, los correspondientes a la capa de negocio (BL).

Para el *Módulo de gestión de usuarios y generación de gráficas y estadísticos* se han creado tres paquetes. El primero aloja los ficheros pertenecientes a la capa de presentación (PL) para proporcionar un formulario donde los usuarios se puedan registrar y, además, para que los usuarios puedan navegar en sus analíticas. El segundo aloja los ficheros pertenecientes a la capa de datos (DL) para, por una parte, registrar a los usuarios y, por otra, para obtener los datos de las analíticas de estos. El tercero aloja los ficheros pertenecientes a la capa de negocio (BL) que hacen de intermediarios entre la capa de presentación y la de datos.

Para el *Módulo de sistema de cuestionarios* se han creado tres paquetes. Uno aloja los ficheros correspondientes a la capa de presentación (PL), para presentarle las cuestiones a los usuarios. El segundo paquete aloja los ficheros pertenecientes a la capa de datos (DL), para acceder al sistema de almacenamiento y extraer las

correspondientes cuestiones. El tercero aloja los ficheros pertenecientes a la capa de negocio (BL) que hacen de intermediarios entre la capa de presentación y la de datos.

Para el *Módulo de interfaz de usuario* se ha creado un paquete que aloja los ficheros pertenecientes a la capa de presentación (PL).

8.2.7. Validación

La validación del sistema se ha realizado mediante pruebas de integración, que son aquellas que se realizan en el ámbito del desarrollo de software una vez se ha validado cada módulo mediante las pruebas unitarias. Estas pruebas de integración consisten en verificar que el conjunto de módulos que conforman la herramienta funciona como se espera. El tipo de pruebas que se han realizado para ello han sido pruebas de interfaz de usuario, que son aquellas que generan un conjunto de eventos de la interfaz de usuario tales como teclear, hacer click con el ratón e interactuar de otras formas con la herramienta. Con cada evento se observan los cambios resultantes en la interfaz de usuario, validando que el comportamiento del programa sea el correcto. Tras la realización de estas pruebas se ha verificado que el comportamiento del sistema era el esperado y que se consigue, tras la introducción de diferentes valores de parámetros existentes en una analítica, un diagnóstico sobre qué valores están alterados y posibles enfermedades debidas a la variación de dichos parámetros.

9. Planificación

En este apartado se describe la planificación que se va a seguir para la realización del proyecto '*Diseño e implementación de un sistema para diagnósticos médicos a través de análisis clínicos basado en árboles de decisión*'. En él se describe el equipo de trabajo, los hitos del proyecto, la planificación estimada, así como los paquetes de trabajo y las tareas. También se adjunta un diagrama Gantt que ofrece una información visual sobre los plazos de cada uno de los paquetes y tareas.

9.1. Equipo de trabajo

Nombre	Empresa	Responsabilidad	Iniciales
Ana Eva Ibarrola Armendariz	E.T.S.I. de Bilbao	Directora del proyecto	AEIA
Fernando Martínez Sáez	Pulsomed S.A.	Ingeniero Senior	FMS
David Torres García	E.T.S.I. de Bilbao Pulsomed S.A.	Ingeniero Junior	DTG

Tabla 12. Miembros del equipo de trabajo

9.2. Hitos del Proyecto

Los hitos reflejan las reuniones principales de toma de decisiones y seguimiento del estado, que se llevan a cabo a lo largo del proyecto. Los hitos solamente representan reuniones en puntos críticos, no representan todos los puntos de control establecidos en la gestión del proyecto. A continuación se describen los hitos más relevantes del proyecto.

Hito	Fecha	Descripción
H1	15 de octubre de 2015	Inicio del proyecto
H2	15 de diciembre de 2016	Prototipo de análisis clínico
H3	10 de marzo de 2016	Revisión del diseño de cada módulo
H4	20 de junio de 2016	Revisión del desarrollo de cada módulo
H5	4 de julio de 2016	Revisión del resultado de las pruebas unitarias
H6	15 de septiembre de 2016	Revisión de la integración y resultado de las pruebas finales
H7	29 de septiembre de 2016	Entrega de la documentación final del proyecto

Tabla 13. Hitos del Proyecto

9.3. Paquetes de trabajo y desglose por tareas

A continuación se exponen las tareas que han sido necesarias para llevar a cabo el proyecto, incluyendo cada una de ellas en el paquete de trabajo correspondiente. Los paquetes de trabajo y las tareas han sido las siguientes:

a. Paquetes de trabajo del proyecto

Abreviatura	Paquete de Trabajo	Descripción	Duración (Horas)
P1	Análisis Previo	Estudio de los conceptos médicos para la realización del proyecto, así como generación de un análisis clínico común.	176
P2	Diseño del sistema	Diseño de la arquitectura general, así como el diseño de cada uno de los módulos que conforman el sistema.	240
P3	Desarrollo del sistema	Desarrollo de cada uno de los módulos que conforman el sistema.	288

P4	Pruebas unitarias	Prueba y análisis del correcto funcionamiento de cada uno de los módulos.	40
P5	Integración, validación y pruebas	Integración de todos los módulos, así como la validación de la herramienta y las pruebas finales.	212
P6	Documentación	Documentación final del TFM.	228

Tabla 14. Paquetes de trabajo del Proyecto

b. Desglose de tareas del proyecto

Paquete de Trabajo	Abreviatura	Nombre	Duración (Horas)
P1	T1	Análisis y estudio de los parámetros existentes en la sangre y definición de las especificaciones requeridas	48
P1	T2	Estudio de la relación entre los parámetros alterados y la existencia de patologías y definición de las especificaciones requeridas	40
P1	T3	Estudio de los parámetros más comunes en los exámenes de laboratorio	60
P1	T4	Generación de un prototipo de análisis clínico común	28
P2	T5	Diseño de la arquitectura general	44
P2	T6	Diseño de la base de datos	36
P2	T7	Diseño del sistema de recolección de datos automatizado	20
P2	T8	Diseño del sistema de gestión de gestión de usuarios	32
P2	T9	Diseño del sistema del sistema de tratamiento de datos	40
P2	T10	Diseño del sistema de cuestionarios	20
P2	T11	Diseño del sistema generador de gráficas	16

P2	T12	Diseño de la interfaz de usuario	32
P3	T13	Implementación y población de la Base de datos	64
P3	T14	Desarrollo del sistema de recolección de datos automatizado	24
P3	T15	Desarrollo del sistema de gestión de gestión de usuarios	44
P3	T16	Desarrollo del sistema del sistema de tratamiento de datos	56
P3	T17	Desarrollo del sistema de cuestionarios	24
P3	T18	Desarrollo del sistema generador de gráficas	24
P3	T19	Desarrollo de la interfaz de usuario	52
P4	T20	Pruebas del sistema de recolección de datos automatizado	2
P4	T21	Pruebas del sistema de gestión de gestión de usuarios	12
P4	T22	Pruebas del sistema del sistema de tratamiento de datos	18
P4	T23	Pruebas del sistema de cuestionarios	4
P4	T24	Pruebas del sistema Generador de gráficas	4
P5	T25	Integración del sistema de tratamiento de datos con la base de datos	56
P5	T26	Integración de la interfaz de usuario con la base de datos	52
P5	T27	Integración de la interfaz de usuario con el sistema de recolección de datos automatizado	24
P5	T28	Integración de la interfaz de usuario con el sistema de gestión de gestión de usuarios	28
P5	T29	Integración de la interfaz de usuario con el sistema Generador de Gráficas	24

P5	T30	Integración del sistema de tratamiento de datos e interfaz de usuario con el sistema de cuestionarios	28
P6	T31	Documentación del proyecto	228

Tabla 15. Desglose de tareas del Proyecto

9.4. Planificación estimada del proyecto

El tiempo total estimado para la realización del proyecto, teniendo en cuenta el tiempo necesario para cada paquete de trabajo, es de aproximadamente once meses y medio y un total de 1184 horas, a partir de su fecha de comienzo del 15 de octubre de 2015.

9.5. Diagrama de Gantt del proyecto

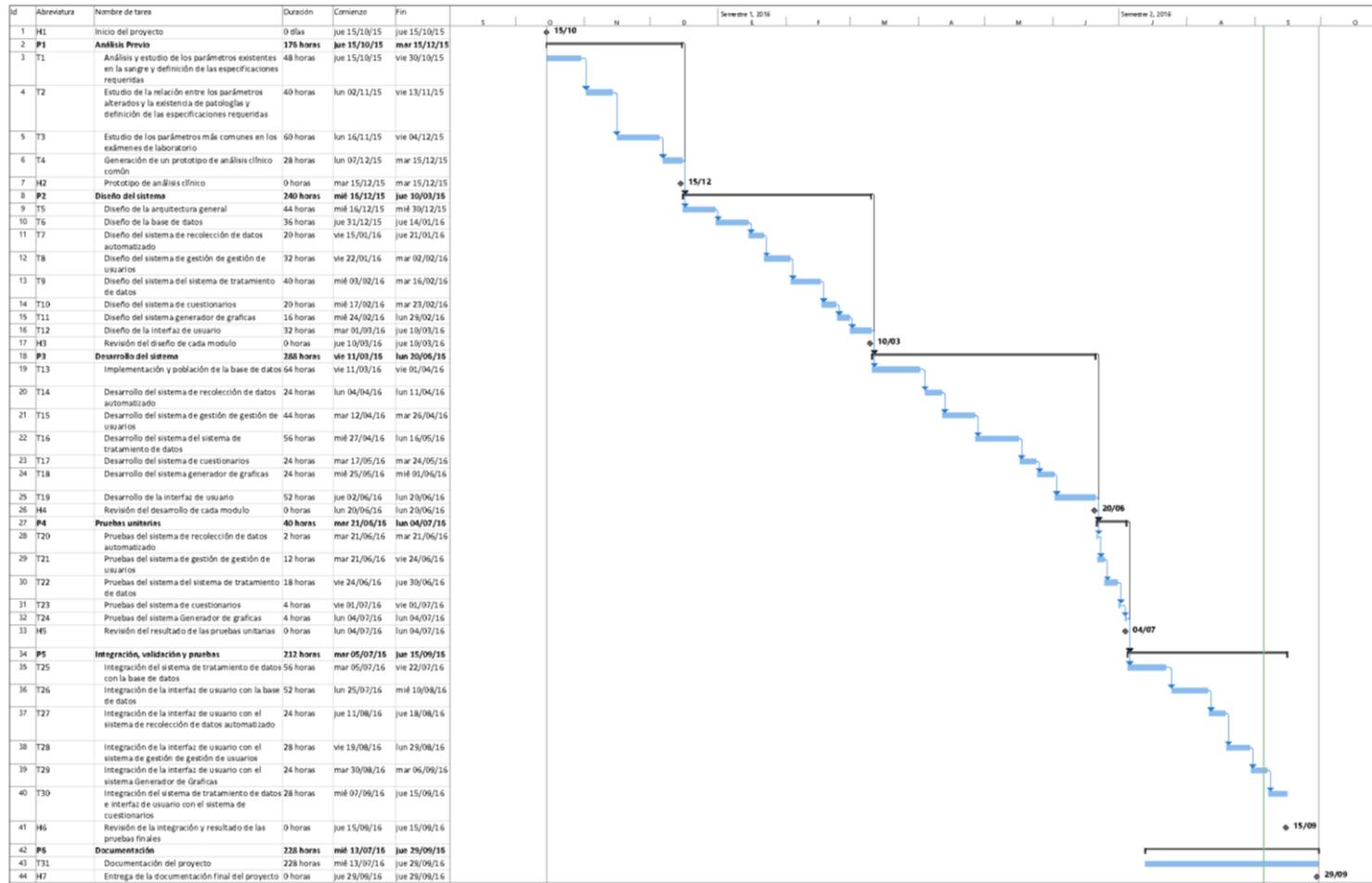


Ilustración 45. Diagrama Gantt del proyecto

10. Cálculos

Para la realización de este apartado se ha utilizado la herramienta GTMETRIX. Una herramienta que permite medir entre otros parámetros, la velocidad de acceso y carga de diferentes dominios. Además, se obtiene un código de colores en función de los parámetros que se obtienen. Se obtiene un color rojo cuando el valor en cuestión es bastante negativo, un color amarillo para aquellos valores mejorables y un color verde cuando el parámetro en cuestión se sitúa en un valor correcto.

Concretamente se van a medir el número de request necesarios para la visualización de una página completa del sistema, el tiempo de carga de dicha página, el YSlow (parámetro que se basa en un conjunto de 34 reglas de optimización con las que puedes mejorar el rendimiento) y el tamaño total de ficheros descargados.

A continuación se muestra una imagen de los resultados obtenidos para la página en la que se introducen los valores de los análisis.

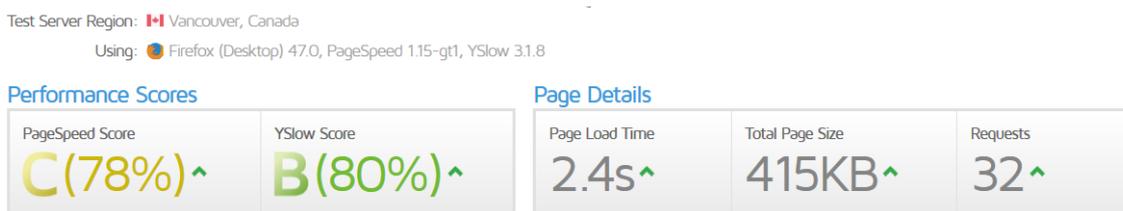


Ilustración 46. reporte obtenido de la herramienta GTMETRIX.

Como se puede apreciar en la imagen anterior, se obtienen valores verdes, para el tiempo de carga (2,4 segundos), el número de request necesarios para la carga de todos los ficheros (32 request), el YSlow (score a 80%) y el tamaño total de la página (415KB). Por otra parte, se obtienen valores amarillos para la velocidad de la página (score a 78%), parámetro que deberá ser objetivo de futuras modificaciones del sistema para mejorarlo en la medida de lo posible.

11. Presupuesto

En este apartado se detallan los aspectos económicos referentes a la realización del proyecto para la realización del proyecto '*Diseño e implementación de un sistema para diagnósticos médicos a través de análisis clínicos basado en árboles de decisión*'.

En este apartado se estiman los recursos económicos a destinar para el proyecto en función de los recursos humanos necesarios y del material amortizable y fungible.

11.1. Precios unitarios

11.1.1. Mano de obra

Código	Nombre	Responsabilidad	Tasa Horaria
AEIA	Ana Eva Ibarrola Armendariz	Directora del proyecto	60 €/hora
FMS	Fernando Martínez Sáez	Ingeniero Senior	40 €/hora
DTG	David Torres García	Ingeniero Junior	30 €/hora

Tabla 16. Precio unitario de la mano de obra

11.1.2. Material amortizable

Código	Descripción	Precio	Amortización	Precio Unitario
AM1	PC Sobremesa - Asus M32CD-SP023T	599 €	3 años	0.022 €/hora
AM2	Licencia Windows 10 Pro	279 €	1 año	0.031 €/hora
AM3	Licencia Suite Office 2016	179 €	1 año	0.02 €/hora
AM4	Monitor - Dell 210-AFOW	95 €	3 años	0.003 €/hora
AM5	Ratón - Asus 90-XB2P00MU0000	15,70 €	4 años	0.0004 €/hora

AM6	Teclado - Logitech K120 Business	12,33 €	4 años	0.0003 €/hora
-----	----------------------------------	---------	--------	---------------

Tabla 17. Precio unitario del material amortizable

11.1.3. Material fungible

Código	Descripción	Precio
MF1	Material de oficina	400 €
MF2	Conexión a internet (11, 5 meses)	345 €
MF3	Electricidad (11,5 meses)	1150 €

Tabla 18. Precio unitario del material fungible

11.2. Coste de Recursos Humanos

Para calcular el coste de las horas internas se van han desglosar los paquetes de trabajo en función de las personas que han sido necesarias para llevar a cabo las tareas de cada paquete y el número de horas invertido por cada persona. De esta forma, el coste debido a los recursos humanos ha sido el siguiente:

Paquete de trabajo	Responsable	Horas invertidas	Coste
P1	AEIA	10	600 €
	FMS	46	1840 €
	DTG	120	3600 €
P2	AEIA	5	300 €
	FMS	10	400 €
	DTG	225	6750 €
P3	DTG	288	8640 €
P4	FMS	5	200 €
	DTG	35	1050 €
P5	FMS	7	280 €
	DTG	205	6150 €

P6	AEIA	20	1200 €
	DTG	208	6240 €
Total			37250 €

Tabla 19. Coste de recursos humanos

11.3. Coste de Recursos Materiales

Para calcular el coste de los recursos materiales, se dividirá por una parte el material amortizable y por otro el material fungible.

Código del material	Cantidad	Horas de uso	Precio Unitario	Coste total
<i>Amortizable</i>				
AM1	1	1081	0.022 €/hora	23,78 €
AM2	1	1081	0.031 €/hora	33,51 €
AM3	1	300	0.02 €/hora	6 €
AM4	2	1081	0.003 €/hora	6,49 €
AM5	1	1081	0.0004 €/hora	0,43 €
AM6	1	1081	0.0003 €/hora	0,32 €
<i>Total amortizable</i>				69,53 €
<i>Fungible</i>				
MF1	-	-	-	400 €
MF2	-	-	-	345 €
MF3	-	-	-	1150 €
<i>Total fungible</i>				1895 €
Total				1965,54 €

Tabla 20. Coste de recursos materiales

11.4. Resumen final del coste del proyecto

En la siguiente tabla se expresa el coste final del proyecto atendiendo a cada una de las partidas e incluyendo los gastos e impuestos derivados de la ejecución del mismo.

Concepto	Coste total
<i>Total Recursos Humanos</i>	37250 €
<i>Total Recursos Materiales</i>	1965,54 €
<i>Total Parcial</i>	39.216,54 €
<i>Imprevistos (2%)</i>	784,31 €
<i>TOTAL</i>	39.999,85 €

El presupuesto final para la validación e implementación final del proyecto '*Diseño e implementación de un sistema para diagnósticos médicos a través de análisis clínicos basado en árboles de decisión*' asciende a la cantidad total de treinta y nueve mil novecientos noventa y nueve euros y ochenta y cinco céntimos (39.999,85 €).

12. Conclusiones

En este proyecto se ha realizado el diseño e implementación de un sistema para diagnósticos médicos a través de análisis clínicos basado en árboles de decisión.

Se ha considerado relevante llevar a cabo este proyecto debido al actual cambio en la mentalidad de las personas en lo referente al cuidado personal de la salud, la preocupación actual de varios gobiernos de países occidentales por el futuro de la atención y el gasto sanitarios, y la búsqueda de un paciente más empoderado, de forma que se agilice su seguimiento y recuperación, reduciendo el gasto en salud pública. El desarrollo del sistema se ha dividido en 6 módulos principales que son los siguientes:

- *Módulo de almacenamiento de datos:* este módulo consigue, por una parte, almacenar los datos referentes a usuarios y sus analíticas y, por otra, almacenar toda la información necesaria para la realización de los diagnósticos.
- *Módulo de recolección automatizada de los datos:* este módulo consigue proporcionar una interfaz para la subida de ficheros JSON en formato FHIR, permitiendo almacenarlos y procesarlos cuando sea necesario.
- *Módulo de tratamiento de datos:* este módulo consigue realizar diagnósticos médicos, basándose en los resultados de un análisis clínico.
- *Módulo de gestión de usuarios y generación de gráficas y estadísticos:* este módulo consigue gestionar el registro de los usuarios en el sistema y les proporciona una interfaz para visualizar sus analíticas y diagnósticos.
- *Módulo de sistema de cuestionarios:* este módulo ha sido diseñado para ajustar el resultado de los diagnósticos en base a unas preguntas realizadas al usuario.

- *Módulo de interfaz de usuario*: este módulo proporciona una interfaz gráfica sencilla e intuitiva, permitiendo navegar a cualquier tipo de perfil de usuario por el sistema.

Cabe mencionar que este proyecto ha logrado cumplir con todas las expectativas establecidas en su inicio. La herramienta diseñada e implementada es capaz de realizar un diagnóstico en base a los resultados de una analítica y permite que usuarios se registren y mantengan un histórico de sus análisis.

El sistema ha sido diseñado para que acepte parámetros de cualquier tipo de analítica. Es decir, en la implementación actual los parámetros que se utilizan son principalmente los existentes en la sangre, pero en futuras ampliaciones del sistema se puede añadir cualquier tipo de parámetro (análisis de orina, médula, etc.) sin la necesidad de modificar ninguno de los módulos existentes.

13. Bibliografía

- [1]<https://ec.europa.eu/digital-single-market/en/eu-policy-ehealth>
- [2]http://www.msssi.gob.es/organizacion/sns/planCalidadSNS/pdf/ESTRATEGIA_ABORDAJE_CRONICIDAD.pdf
- [3]<https://www.hl7.org/fhir/>
- [4]<http://www.hl7.org/>
- [5]<http://loinc.org/>
- [6]https://www.agpd.es/portalwebAGPD/canaldocumentacion/informes_juridicos/datos_esp_protegidos/index-ides-idphp.php
- [7] <http://www.informatica-medica.com/gesmed/>
- [8] <https://medlineplus.gov/spanish/>
- [9] <http://www.symcat.com/>
- [10] <https://www.mediktor.com/es>
- [11] <http://www.asp.net/>
- [12] <https://secure.php.net/>
- [13]<http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>
- [14] <http://docs.oracle.com/javaee/5/tutorial/doc/bnajo.html>
- [15] <https://msdn.microsoft.com/es-es/library/bb545450.aspx>
- [16] <https://dev.mysql.com/doc/>
- [17] <https://www.postgresql.org/docs/>
- [18] <https://confluence.ihtsdotools.org/display/DOC/Release+Documents>
- [19]<http://www.msssi.gob.es/estadEstudios/estadisticas/normalizacion/>
- [20][http://www.itnuevolaredo.edu.mx/takeyas/apuntes/Inteligencia%20Artificial/Apuntes/tareas_alumnos/C4.5/C4.5\(2005-II-B\).pdf](http://www.itnuevolaredo.edu.mx/takeyas/apuntes/Inteligencia%20Artificial/Apuntes/tareas_alumnos/C4.5/C4.5(2005-II-B).pdf)

[21]<http://www.itnuevolaredo.edu.mx/takeyas/apuntes/Inteligencia%20Artificial/Apuntes/IA/ID3.pdf>

[22]http://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Magerit.html#.V3JtxzX0fwa

Hemograma – Manual de Interpretación: Renato Failace y colaboradores

Manual Práctico de Hematología Clínica: Miguel, A. Sanz Alonso, Enric Cabrerias I Pons

Anexo I: Normativa aplicable

En este anexo se muestran, por una parte, la normativa aplicable con respecto a la ley de protección de datos, mediante una extracción de dicha ley referente al tratamiento de datos de salud, y por otra, un resumen de la estructura JSON del estándar FHIR.

- **Extracción de la LOPD referente al tratamiento de datos de salud**



La consulta plantea si la recogida de datos de salud requiere el consentimiento escrito de los pacientes de acuerdo con las previsiones contenidas en la Ley Orgánica 11/1999, de 13 de diciembre de Protección de Datos de Carácter personal.

I

Como regla general, el artículo 7.3 de la Ley Orgánica 15/1999 dispone que “los datos de carácter personal que hagan referencia al origen racial, a la salud y a la vida sexual, sólo podrán ser recabados, tratados y cedidos cuando por razones de interés general, así lo disponga una Ley o el afectado consienta expresamente”. Este artículo determina el contenido esencial del derecho fundamental a la protección de datos de carácter personal debido a su carácter orgánico.

La especial protección conferida a los datos relacionados con la salud de las personas no es arbitraria, sino que resulta de lo dispuesto en las normas Internacionales y Comunitarias reguladoras del tratamiento automatizado de datos de carácter personal. En este contexto, tanto el artículo 8 de la Directiva 95/46/CE del Parlamento y del Consejo, así como el artículo 6 del Convenio 108 del Consejo de Europa para la protección de las personas con respecto al tratamiento automatizado de datos de carácter personal, hecho en Estrasburgo el 28 de enero de 1981, ratificado por España en fecha 27 de enero de 1984, hacen referencia a los datos de salud como sujetos a un régimen especial de protección.

En este sentido, el artículo 8 de la Directiva 95/46/CE limita el tratamiento de datos a supuestos y finalidades concretos en los que será preciso el consentimiento, que además deberá ser expreso, del afectado o la necesidad del tratamiento con fines de asistencia sanitaria o atención de un interés vital del afectado. Esta cuestión ha sido especialmente analizada por el Grupo de Autoridades de Protección de Datos creado por el artículo 29 de la citada Directiva en su Documento de trabajo sobre el tratamiento de datos personales relativos a la salud en los historiales

médicos electrónicos (Documento EP131), en el que se indica expresamente que “todos los datos contenidos en documentos médicos, en historiales médicos electrónicos y en sistemas de HME son “datos personales sensibles”. Por tanto, no sólo están sujetos a todas las normas generales sobre protección de datos personales de la Directiva, sino también a las normas sobre protección de datos especiales que rigen el tratamiento de la información sensible, contenidas en el artículo 8 de la Directiva”.

II

Establecida ya por el artículo 7.3 la regla general del consentimiento para el tratamiento de los datos de salud, el artículo 7.6 establece en su párrafo primero que “podrán ser objeto de tratamiento los datos de carácter personal a que se refieren los apartados 2 y 3 de este artículo, cuando dicho tratamiento resulte necesario para la prevención o para el diagnóstico médicos, la prestación de asistencia sanitaria o tratamientos médicos o la gestión de servicios sanitarios, siempre que dicho tratamiento de datos se realice por un profesional sanitario sujeto al secreto profesional o por otra persona sujeta asimismo a una obligación equivalente de secreto.”

Igualmente, conforme al párrafo segundo del propio artículo 7.6, “También podrán ser objeto de tratamiento los datos a que se refiere el párrafo anterior cuando el tratamiento sea necesario para salvaguardar el interés vital del afectado o de otra persona, en el supuesto de que el afectado esté física o jurídicamente incapacitado para dar su consentimiento”.

Respecto de lo dispuesto en el artículo 7.6, el documento WP131 del Grupo creado por el artículo 29 de la Directiva 95/46/CE recuerda que “Puesto que el artículo 8, apartado 3, de la Directiva (que transpone el citado artículo 7.6)) constituye una excepción a la prohibición general de tratar datos sensibles, esta excepción deberá interpretarse de forma restrictiva”. De este modo, señala que:

“Esta excepción cubre solamente el tratamiento de datos personales para el propósito específico de proporcionar servicios relativos a la salud de carácter preventivo, de diagnóstico, terapéutico o de convalecencia, y a efectos de la gestión de estos servicios sanitarios, como por ejemplo facturación, contabilidad o estadísticas.

No se cubre el tratamiento posterior que no sea necesario para la prestación directa de tales servicios. Ahora bien, esta excepción legal del artículo 7.6, así como la del artículo 8 de la Ley Orgánica 15/1999, se refieren al tratamiento de los datos, no a su cesión.

Por otra parte, en el marco de la asistencia sanitaria añade el artículo 8 que “sin perjuicio de lo que se dispone en el artículo 11 respecto de la cesión, las instituciones y los centros sanitarios públicos y privados y los profesionales correspondientes podrán proceder al tratamiento de los datos de carácter personal relativos a la salud de las personas que a ellos acudan o hayan de ser tratados en los mismos, de acuerdo con lo dispuesto en la legislación estatal o autonómica sobre sanidad”. En este sentido, también se establece en el artículo 10. 5 del Real Decreto 1720/2007, de 21 de diciembre, por el que se aprueba el Reglamento de Desarrollo de la Ley Orgánica 15/1999 que “Los datos especialmente protegidos podrán tratarse y cederse en los términos previstos en los artículos 7 y 8 de la Ley Orgánica 15/1999, de 13 de diciembre.”

En particular, no será necesario el consentimiento del interesado en la comunicación de datos personales sobre la salud, incluso a través de medios electrónicos, entre organismos, centros y servicios del Sistema Nacional de Salud cuando se realice para la atención sanitaria de las personas, conforme a lo dispuesto en el Capítulo V de la Ley 16/2003, de 28 de mayo, de cohesión y calidad del Sistema Nacional de Salud.”

En conclusión, la regla general para la recogida y tratamiento de los datos de salud es la que exige el consentimiento libre, inequívoco, informado y expreso de los afectados, sin perjuicio de lo señalado en la normativa estatal y autonómica respectivamente en cuanto a las historias clínicas.

III

En cuanto a la normativa estatal, la única ley que resulta aplicable a su tratamiento y/o cesión es la Ley 41/2002, de 14 de noviembre, básica reguladora de la Autonomía del Paciente y de Derechos y Obligaciones en materia de Información y Documentación Clínica. En la misma se establece un régimen específico que delimita la aplicación de la Ley 15/1999 en este caso, de manera que los accesos a la historia clínica deberán ser los previstos en su artículo 16 contemplado como aplicación de la excepción al consentimiento prevista en el artículo 7.3 de la Ley Orgánica 15/1999.

El artículo 16 de la Ley 41/2002 limita los supuestos de cesión de datos de la historia clínica a terceros ajenos a la asistencia sanitaria a dos supuestos y en los siguientes términos:

“3. El acceso a la historia clínica con fines judiciales, epidemiológicos, de salud pública, de investigación o de docencia, se rige por lo dispuesto en la Ley Orgánica 15/1999, de Protección de Datos de Carácter Personal, y en la Ley 14/1986, General de Sanidad, y demás normas de aplicación en cada caso. El acceso a la historia clínica con estos fines obliga a preservar los datos de identificación personal del paciente, separados de los de carácter clínico-asistencial, de manera que como regla general quede asegurado el anonimato, salvo que el propio paciente haya dado su consentimiento para no separarlos. Se exceptúan los supuestos de investigación de la autoridad judicial en los que se considere imprescindible la unificación de los datos identificativos con los clínico-asistenciales, en los cuales se estará a lo que dispongan los jueces y tribunales en el proceso correspondiente. El acceso a los datos y documentos de la historia clínica queda limitado estrictamente a los fines específicos de cada caso”.

“5. El personal sanitario debidamente acreditado que ejerza funciones de inspección, evaluación, acreditación y planificación, tiene acceso a las historias clínicas en el cumplimiento de sus funciones de comprobación de la calidad de la asistencia, el respeto de los derechos del paciente o cualquier otra obligación del centro en relación con los pacientes y usuarios o la propia Administración sanitaria”.

En cuanto a la legislación autonómica, la Ley de Salud de Aragón, Ley 6/2002, de 15 de abril, establece en su artículo 18 la regulación de la historia clínica en los siguientes términos: 1.” El Departamento responsable de Salud determinará reglamentariamente, en relación con la historia clínica:

- a) Los datos y documentos que la componen.
- b) La gestión, utilización, acceso y conservación de la misma.

c) El tiempo durante el que deberá conservarse.

En cualquier caso, todo el personal que acceda, en el uso de sus competencias, a cualquier dato de la historia clínica quedará sujeto al deber de guardar secreto sobre los datos de la misma.”

IV

En términos generales, debe atenderse en la recogida y tratamiento de los datos, a los principios generales que informan la protección de datos recogidos en el artículo 4 de la Ley Orgánica 15/1999 que dispone en su número 1” Los datos de carácter personal sólo se podrán recoger para su tratamiento, así como someterlos a dicho tratamiento, cuando sean adecuados, pertinentes y no excesivos en relación con el ámbito y finalidades determinadas, explícitas y legítimas para las que se hayan obtenido.

Y en su número 2 señala que “Los datos de carácter personal objeto de tratamiento no podrán usarse para finalidades incompatibles con aquellas para las que los datos hubieran sido recogidos.”

Así mismo, en la recogida y tratamiento de datos de salud, deberán observarse las medidas de seguridad, según dispone el artículo 9. 3 de la Ley Orgánica 15/1999, que deberán ser las de nivel alto, además de las de nivel medio y básico, conforme señala el artículo 81.3 a) del Real Decreto 1720/2007 que concreta las mismas en sus artículos 89 y siguientes.

Todo ello en aplicación de la Ley Orgánica 15/1999, de 13 de diciembre, de Protección de datos de Carácter Personal.

● Resumen de la estructura JSON del estándar FHIR



1.18.2 JSON Representation of Resources

Implementable Technology Specifications Work Group	Maturity Level: 5	Ballot Status: DSTU 2
--	-------------------	-----------------------

The JSON representation for a resource is described using this format:

```
{
  "resourceType": "Element",
  // from Source: element #1
  "property1": "<[primitive]>", // short description
  "property2": { [Data Type] }, // short description
  "property3": { // Short Description
    "propertyA": { CodeableConcept }, // Short Description (example)
  },
  "property4": [ { // Short Description
    "propertyB": { Reference(ResourceType) } // R! Short Description
  } ]
}
```

Using this format:

- To build a valid JSON instance of a resource, replace the contents of the property values with valid content as described by the type rules and content description found in the property value for each element
- In this example:
 1. `property1` has a primitive data type; the value of the property will be as described for the stated type
 2. `property2` has a complex data type; the value of the property is an object that has the content as described for the stated type
 3. `property3` is an object property that contains additional properties (e.g. `propertyA`; the allowable properties are listed (but also include extensions as appropriate)
 4. `property4` is an array property that contains items which are objects themselves. The items may have any of the types already encountered in points 1-3
 5. `propertyA` has a binding to a value set - the Short description is a link to the value set. In addition, the binding strength is shown
 6. `propertyB` is a reference to a particular kind of resource
- Property names are case-sensitive (though duplicates that differ only in case are never defined)
- Properties can appear in any order

- XHTML is represented as an escaped string
- Objects are never empty. If an element is present in the resource, it SHALL have properties as defined for its type, or 1 or more [extensions](#)
- String property values can never be empty. Either the property is absent, or it is present with at least one character of content
- The **R!** denotes that an element is mandatory - it must be present (or in an array, at least one item must be present)
- In this format, `//` is used for comments but these can't be in the JSON instances
- The character encoding is always UTF-8
- The MIME-type for this format is `application/json+fhir`.

Given the way [extensions](#) work, applications reading JSON resources will never encounter unknown properties. However once an application starts trading with other applications that conform to later versions of this specification, unknown properties may be encountered. Applications MAY choose to ignore unknown properties in order to foster forwards compatibility in this regard, but may also choose not to. Applications declare their behavior with regard to unknown elements using [Conformance.acceptUnknown](#).

1.18.2.1 Comparison with XML

The JSON format is similar to the XML format:

- The names for the JSON object members are the same as the names of the elements and attributes in XML, including elements that may repeat. Property names are case sensitive
- Just as in XML, JSON objects and arrays are never empty, and properties never have null values (except for a special case documented below). Omit a property if it is empty
- JSON whitespace is not part of the contents of a resource. Applications MAY preserve the whitespace when handling resources, but are not required to do so. Note that digital signatures may depend on the whitespace

There are differences to XML:

- There are no namespaces in the JSON representation
- The type of the resource is represented differently in JSON - instead of being the name of the base object (there is none in JSON), it is carried as the property `resourceType`
- The order of properties of an object is not significant in the JSON representation, though order within an array SHALL be maintained
- JSON does not have a notion of attributes versus elements, so attributes (e.g. `id`, `value`) are handled differently (see below)
- JSON has the array notation, which is used to represent repeating elements. Note that arrays are used when the item might repeat, even if it does not repeat in a specific instance
- The XHTML `<div>` element in the [Narrative](#) datatype is represented as a single escaped string of XHTML. This is to avoid problems in JSON with mixed content, etc. The XHTML SHALL still conform to the rules described for the [Narrative](#)
- There is no inherent support in JSON for a comment syntax. As a convention, content that would be comments in an XML representation is represented in a property with the name `fhir_comments`, which is an array of strings, which can appear on any JSON object. This is heavily used in example instances, e.g. in this specification, but not usually used in production systems (and production systems may choose to reject resources with comments in them)

The JSON format for the resources follows the standard XML format closely to make interconversion easy, and so that XPath queries can easily be mapped to query the JSON structures. However the differences - particularly the repeating element one, which cannot be avoided - mean that generic XML -> JSON converters are not able to perform correctly. The [reference platforms](#) provide XML -> JSON conversion functionality that accommodates these FHIR-specific characteristics.

1.18.2.2 JSON Representation for repeating elements 🌐

An element that has a maximum cardinality of >1 (e.g. x..* in the definitions) may occur more than once in the instance. In XML, this is simply done by repeating the XML element multiple times. In JSON, this is done by using an array type. Note that:

- The name of the array is singular - the same as the XML element
- An item that may repeat is represented as an array even in the case that it doesn't repeat so that the process of parsing the resource is the same either way

```
<coding>
  <system value="http://snomed.info/sct"/>
  <code value="104934005"/>
</coding>
<coding>
  <system value="http://loinc.org"/>
  <code value="2947-0"/>
</coding>
```

is represented in JSON like this:

```
"coding": [
  {
    "system": "http://snomed.info/sct",
    "code": "104934005"
  },
  {
    "system": "http://loinc.org",
    "code": "2947-0"
  }
]
```

1.18.2.3 JSON representation of primitive elements 🌐

FHIR elements with primitive data types are represented in two parts:

- A JSON property with the name of the element, which has a JSON type of number, boolean, or string
- a JSON property with `_` prepended to the name of the element, which, if present, contains the value's id and/or extensions (and comments, if there are any)

The FHIR types `integer` and `decimal` are represented as a JSON number, the FHIR type `boolean` as a JSON boolean, and all other types are represented as a JSON string which has the same content as that specified for the relevant data type. Whitespace is always significant (i.e. no leading and trailing spaces for non-strings).

```
<code value="abc"/> <!-- code -->
<date value="1972-11-30"/> <!-- dateTime -->
<deceased value="false" /> <!-- boolean -->
<count value="23" /> <!-- integer -->
```

is represented in JSON as

```
"code": "abc",
"date": "1972-11-30",
"deceased": false,
"count": 23
```

When using a JavaScript `JSON.parse()` implementation, note that JavaScript natively supports only one numeric datatype, which is a floating point number. This can cause loss of precision for FHIR numbers. In particular, trailing 0s after a decimal point will be lost e.g. 2.00 will be converted to 2. The FHIR decimal data type is defined such that precision, including trailing zeros, is preserved for presentation purposes, and this is widely regarded as critical for correct presentation of clinical measurements. Implementations should consider using a custom parser and big number library (e.g. <https://github.com/jtobey/javascript-bignum>) to meet these requirements.

If the value has an id attribute, or extensions, then this is represented as follows:

```
<birthDate id="314159" value="1970-03-30" >
  <extension url="http://example.org/fhir/StructureDefinition/text">
    <valueString value="Easter 1970"/>
  </extension>
</birthDate>
```

is represented in JSON as:

```
"birthDate": "1970-03-30",
"_birthDate": {
  "id": "314159",
  "extension": [ {
    "url": "http://example.org/fhir/StructureDefinition/text",
    "valueString": "Easter 1970"
  } ]
}
```

Note: If the primitive has an id attribute or extension, but no value, only the property with the `_` is rendered.

In the case where the primitive element may repeat, it is represented in two arrays. JSON null values are used to fill out both arrays so that the id and/or extension are aligned with the matching value in the first array, as demonstrated in this example:

```
<code value="a1"/>
<code value="a2"/>
  <extension url="http://hl7.org/fhir/StructureDefinition/display">
    <valueString value="New Zealand a.k.a Kiwiland"/>
  </extension>
</code>
```

is represented in JSON as:

```

"code": [ "au", "nz" ],
"_code": [
  null,
  {
    "extension": [ {
      "url": "http://hl7.org/fhir/StructureDefinition/display",
      "valuestring": "New Zealand a.k.a Kiwiland"
    }
  ]
}
]
    
```

Note: when one of the repeating elements has no value, it is represented in the first array using a null. When an element has a value but no extension/id, the second array will have a null at the position of that element.

Design Note: The representation of primitive data types has been split into two parts like this in order to simplify the representation of simple primitive values without id or extensions. This does have the cost of making the representation of the id attribute and extensions more ungainly, but these are both rarely used with primitive data types.

1.18.2.4 JSON representation of Elements, and Complex Data types

Elements, and complex *datatypes* (types that contain named elements of other types) are represented using a JSON object, containing a member for each element in the datatype. Composites can have id attributes, which are converted to JSON member values, in the same manner as described for primitives. For example:

```

<person>
  <text>
    <status value="generated" />
    <div xmlns="http://www.w3.org/1999/xhtml"><p>...</p></div>
  </text>
  <name>
    <use value="official" />
    <given value="Karen" />
    <family id="a2" value="Van" />
  </name>
</person>
    
```

is represented in JSON as:

```

{
  "person": {
    "name": {
      "use": "official",
      "given": [ "Karen" ],
      "family": [ "Van" ]
    },
    "family": [ { "id": "a2" } ]
  },
  "text": {
    "status": "generated",
    "div": "<div xmlns='http://www.w3.org/1999/xhtml'><p>...</p></div>"
  }
}
    
```

Things to note here are:

- Both given and family are repeating XML elements, so they are serialised as an Array whether or not they repeat in this instance
- In the family part of `name`, the `id` is added represented in `_family` as described above
- The XHTML content in the `div` element which is in the Narrative element `text` is represented as an escaped string in the value property in JSON. The `xhtml`'s root element needs to be a `<div>` in the `xhtml` namespace

1.18.2.5 JSON representation of Resources

A resource is a JSON object with a property `resourceType` which informs the parser which resource type this is:

```

{
  "resourceType": "Patient",
  "text": {
    "status": "generated",
    "div": "<div xmlns='http://www.w3.org/1999/xhtml'><p>...</p></div>"
  }
  etc...
}
    
```

Note that parsers cannot assume that the `resourceType` property will come first.

Design Note: This is a problem for several JSON -> Object serialisers that assume that the `resourceType` property does come first, including [Json.NET](#). However some JSON generators do not give the authoring application control of the order of the property values, and so these implementations cannot inter-operate with implementations that make assumptions about order. Given that JSON says that the property values are an unordered map of name/value pairs, this specification cannot require that properties come in any particular order, though implementers may choose to fix the property order if they are able (and the reference platforms provided with this specification do so).

There is a [sample file](#) with many edge cases to help test JSON parsers.

1.18.2.6 Canonical JSON

Resources and/or Bundles may be digitally signed (see [Bundle](#) and [Provenance](#)).

This specification defines the following method for canonicalizing FHIR resources, when represented as JSON:

- No whitespace other than single spaces in property values and in the `xhtml` in the `Narrative`
- Order properties alphabetically
- Omit all properties that have a default value, if a default value is defined
- Omit all comments (`fhir_comments` property as defined above)

This canonicalization method is identified by the URL <http://hl7.org/fhir/canonicalization/json>. The following additional canonicalization URLs are also defined:

http://hl7.org/fhir/canonicalization/json#data	The narrative (<code>Resource.text</code>) is omitted prior to signing (note the deletion is at <code>Resource.text</code> , not <code>Resource.text.div</code>)
http://hl7.org/fhir/canonicalization/json#static	In addition to narrative (<code>Resource.text</code>), the <code>Resource.meta</code> element is removed. This makes the signature robust as the content is moved from server to server, or workflow and access tags are added or removed
http://hl7.org/fhir/canonicalization/json#narrative	The method only covers the <code>Resource.id</code> and Narrative is retained

These canonicalization methods allow system the flexibility to sign the various portions of the resource that matter for the workflow the signature serves.

Note: One consequence of signing the document is that URLs, identifiers and internal references are frozen and cannot be changed. This might be a desired feature, but it may also cripple interoperability between closed ecosystems where [re-identification](#) frequently occurs. For this reason, it is recommended that systems consider carefully the impact of any signature processes. The impact of signatures on [Document bundles](#) and their related processes is the most well understood use of digital signatures.