

GRADO EN INGENIERÍA INFORMÁTICA DE  
GESTIÓN Y SISTEMAS DE INFORMACIÓN

# TRABAJO FIN DE GRADO

## ***FUENTE ESTADÍSTICA DE INFORMES MÉDICOS (FEIM)***

### ***MEMORIA***

**Alumno/Alumna:** Álvaro Ortiz, Leire

**Director/Directora (1):** Atutxa Salazar, Aitziber

**Director/Directora (2):** Gojenola Gallettebeitia, Koldobika

**Curso:** 2017-2018

**Fecha:** Bilbao, 24 de julio de 2018

## Resumen

La investigación en el sector de la medicina se ha convertido en algo fundamental para la toma de decisiones clínicas.

El personal sanitario muchas veces quisiera recurrir a datos estadísticos para poder encontrar por ejemplo, la frecuencia de incidencia de una enfermedad para estudiar su estacionalidad o su prevalencia, o la frecuencia en la que se ha suministrado una cierta medicación para una determinada enfermedad, para poder determinar por ejemplo si esa medicación se ha probado que puede provocar algún efecto adverso, saber si el problema puede ser potencialmente generalizado o muy localizado. Sin embargo esto necesita mucha dedicación ya que se deben encontrar propiedades en común en multitud de informes médicos.

Actualmente no existen apenas herramientas que faciliten esa gestión ya que la mayoría de ellas están orientadas únicamente a las historias clínicas de los pacientes y no a la obtención de información de todos ellos.

Es por esto, que se ha decidido realizar este Trabajo de Fin de Grado (TFG). En él, se presentará una aplicación web que automatice la gestión de los informes médicos de los pacientes y los procese de tal manera que se pueda recopilar y mostrar a los usuarios estadísticas temporales en base a búsquedas concretas y poder acelerar así el proceso de investigación.

En cuanto a las herramientas utilizadas para su desarrollo, se ha optado por utilizar una combinación de HTML, JavaScript, Java y Spring Framework. Para el almacenamiento de datos se ha utilizado MySQL, y para la visualización de los gráficos la librería ChartJS.

## Palabras clave

Frecuencia, Estadísticas temporales, Spring Framework, ChartJS.

## Laburpena

Medikuntza sektoreko ikerketak funtsezkoak bihurtu dira erabaki klinikoak hartzeko. Askotan osasun arloko langileek datu estatistikoetara jo nahiko lukete, adibidez, gaixotasun baten eragina ezagutzeko, bere prebalentzia ikasteko, edo gaixotasun jakin baterako zehaztu diren botiken maiztasuna aurkitzeko, eragin kaltegarria izan dezakeela frogatu ahal izateko, eta arazoa hedatu edo oso lokalizatua izan daitekeen jakiteko.

Hala eta guztiz ere, dedikazio handia behar du lan honek, mediku-txosten ugariren informazioa bildubeharko luketeelako.

Gaur egun, lan hori errazteko tresnarik ez dago. Gehienak bakarrik pazienteen historia klinikoetara bideratuta daude eta ez haien guztien informazio orokorraren lorpena.

Horregatik, Gradu Amaierako Lan hau egitea erabaki da. Bertan, aplikazio web bat aurkeztuko da, pazienteen txosten medikoen kudeaketa automatizatzen duena eta bilaketan arabera denborako estatistikak erakusten duena, ikerketen prozesua azkartzeko.

Bere garapenean erabilitako tresnei dagokienez, Java eta Spring Framework konbinazioak JavaScript eta HTMLrekin batera aukeratu dira. Datuen biltegitratzerako MySQL erabili da eta, grafikoak ikusteko, ChartJS liburutegia.

## Hitz nagusiak

Maiztasuna, Estatistikak, Spring Framework, ChartJS.

## Abstract

Research in the medical sector has become a fundamental part of clinical decision-making. Many times clinicians would like to resort to statistical data to be able to find, for example, the frequency of incidence of a disease, to study its seasonality or its prevalence, or the frequency in which a certain medication has been provided for a certain disease, to be able to determine for example if that medication has been proven to cause some adverse effect, knowing if the problem can be potentially widespread or very localized.

However, this needs a lot of effort since common properties must be found in a multitude of medical reports.

Currently, there are hardly any tools to facilitate this management for health workers, since most of them are focused only on the patient's medical histories and not on the information gathering of all of them in general.

This is the reason why it has been decided to carry out this Final Degree Project (FDP). In it, a web application will be presented that automates the management of patients' medical reports and processes them in such a way that temporary statistics can be collected and displayed to users based on specific searches to accelerate the research process.

Regarding the tools used for its development, a combination of Java and Spring Framework together with JavaScript and HTML was used. For the storage of data, MySQL, and for the visualization of the graphics, the ChartJS library.

## Keyword

Frequency, Temporary statistics, Spring Framework, ChartJS

# Índice

1. INTRODUCCIÓN.....	11
2. PLANTEAMIENTO INICIAL .....	13
2.1. Objetivos.....	13
2.2. Arquitectura.....	13
2.3. Alcance.....	14
2.3.1. Tareas del proyecto .....	17
2.4. Planificación temporal .....	24
2.5. Herramientas .....	26
2.6. Gestión de riesgos.....	27
2.7. Evaluación económica.....	29
3. ANTECEDENTES .....	31
3.1. Descripción situación actual.....	31
3.2. Análisis de alternativas .....	34
3.2.1. Lenguaje de desarrollo (Servidor) .....	35
3.2.2. Sistema gestor de bases de datos .....	36
3.2.3. Herramientas para visualización de gráficos .....	37
3.2.4. Elección de herramientas.....	38
4. CAPTURA DE REQUISITOS.....	41
4.1. Origen de los datos .....	41
4.2. Casos de uso .....	43
4.2.1. Jerarquía de actores.....	43
4.2.2. Diagrama de casos de uso .....	44
4.3. Modelo de dominio.....	44
5. ANÁLISIS Y DISEÑO.....	47
5.1. Arquitectura.....	47
5.1.1. Introducción a Spring Framework .....	47
5.2. Diagramas de secuencia .....	50
5.3. Diagrama de clases .....	50

5.3.1.	Model .....	51
5.3.2.	Controller .....	51
5.3.3.	Services.....	52
5.3.4.	Dao .....	52
5.4.	Base de datos .....	53
6.	DESARROLLO.....	55
6.1.	Herramientas adicionales .....	55
6.1.1.	JQuery .....	55
6.1.2.	Bootstrap.....	56
6.2.	Estructura del proyecto .....	56
6.3.	Configuración de Spring.....	57
6.4.	Desarrollo funciones principales.....	59
6.4.1.	Carga de informes.....	59
6.4.2.	Búsqueda de estadísticas .....	63
7.	VERIFICACIÓN Y EVALUACIÓN .....	69
7.1	Primer prototipo.....	70
7.2	Segundo prototipo .....	71
7.3	Tercer prototipo .....	72
7.4	Errores y soluciones.....	73
8.	CONCLUSIONES Y TRABAJO FUTURO.....	75
8.1.	Valoración de objetivos del proyecto.....	75
8.2.	Valoración de planificación del proyecto.....	76
8.3.	Trabajo futuro .....	76
9.	BIBLIOGRAFÍA .....	77
	ANEXO I - CASOS DE USO EXTENDIDOS.....	79
	ANEXO II – DIAGRAMAS DE SECUENCIA.....	89
	ANEXO III – MANUAL DE INSTALACIÓN .....	99
	ANEXO IV – MANUAL DE USUARIO .....	105

## Índice de Ilustraciones

<i>Ilustración 1: Paso de multitud de datos a conocimiento</i> .....	11
<i>Ilustración 2: Arquitectura Cliente/Servidor</i> .....	14
<i>Ilustración 3: Modelo incremental</i> .....	14
<i>Ilustración 4: Diseño primer prototipo</i> .....	15
<i>Ilustración 5: Diseño segundo prototipo</i> .....	16
<i>Ilustración 6: Diseño tercer prototipo</i> .....	17
<i>Ilustración 7: EDT</i> .....	18
<i>Ilustración 8: Diagrama de Gantt</i> .....	25
<i>Ilustración 9: Elección del tema estadístico</i> .....	32
<i>Ilustración 10: Estadísticas OMS</i> .....	32
<i>Ilustración 11: Patientslikeme ejemplo de búsqueda</i> .....	33
<i>Ilustración 12: Patientslikeme ejemplo de estadísticas</i> .....	34
<i>Ilustración 13: Esquema tecnologías</i> .....	39
<i>Ilustración 14: BRAT ejemplo con anotaciones</i> .....	42
<i>Ilustración 15: BRAT ejemplo fichero formato .txt</i> .....	43
<i>Ilustración 16: BRAT ejemplo fichero formato .ann</i> .....	43
<i>Ilustración 17: Diagrama de casos de uso</i> .....	44
<i>Ilustración 18: Modelo de dominio prototipo I</i> .....	45
<i>Ilustración 19: Modelo de domino prototipos II y III</i> .....	46
<i>Ilustración 20: Arquitectura MVC</i> .....	47
<i>Ilustración 21: Módulos de Spring</i> .....	48
<i>Ilustración 22: Spring MVC</i> .....	49
<i>Ilustración 23: Diagrama de clases (Model)</i> .....	51
<i>Ilustración 24: Diagrama de clases (Controller)</i> .....	51
<i>Ilustración 25: Diagrama de clases (Service)</i> .....	52
<i>Ilustración 26: Diagrama de clases (Dao)</i> .....	53
<i>Ilustración 27: Diseño de base de datos</i> .....	53
<i>Ilustración 28: Estructura del proyecto</i> .....	56
<i>Ilustración 29: Configuración DispatcherServlet</i> .....	57
<i>Ilustración 30: Contexto de Spring</i> .....	58
<i>Ilustración 31: Configuración Tiles.xml</i> .....	59
<i>Ilustración 32: Formato nombre de fichero</i> .....	60
<i>Ilustración 33: Contenido fichero ANN</i> .....	60
<i>Ilustración 34: Línea especial de fichero</i> .....	61
<i>Ilustración 35: Referencia de línea especial fichero ANN</i> .....	61
<i>Ilustración 36: Definición controlador</i> .....	62
<i>Ilustración 37: Carga Informes</i> .....	62
<i>Ilustración 38: Carga informes incorrecta</i> .....	63
<i>Ilustración 39: Carga informes correcta</i> .....	63
<i>Ilustración 40: Componente Autocomplete jQueryUI</i> .....	64

<i>Ilustración 41: Componente Datepicker jQueryUI</i>	64
<i>Ilustración 42: Llamada filtro básico</i>	65
<i>Ilustración 43: Llamada filtro avanzado</i>	65
<i>Ilustración 44: Consulta sql dinámica</i>	65
<i>Ilustración 45: Ejemplo Chart.js</i>	66
<i>Ilustración 46: Ejemplo visualización gráfico avanzado</i>	67
<i>Ilustración 47: Diagrama de casos de uso</i>	79
<i>Ilustración 48: Casos de uso "Carga informes"</i>	79
<i>Ilustración 49: Menú principal</i>	81
<i>Ilustración 50: Interfaz carga de informes</i>	81
<i>Ilustración 51: Carga de informes correcta</i>	81
<i>Ilustración 52: Carga de informes errónea</i>	82
<i>Ilustración 53: Caso de uso "Obtener estadísticas básicas"</i>	82
<i>Ilustración 54: Menú información estadística</i>	83
<i>Ilustración 55: Interfaz búsqueda básica</i>	83
<i>Ilustración 56: Gráfico búsqueda</i>	84
<i>Ilustración 57: Caso de uso "Obtener estadísticas avanzadas"</i>	84
<i>Ilustración 58: Interfaz búsqueda avanzada</i>	85
<i>Ilustración 59: Caso de uso "Ver datos en gráfico lineal"</i>	86
<i>Ilustración 60: Filtro de la interfaz</i>	87
<i>Ilustración 61: Gráfico de barras</i>	87
<i>Ilustración 62: Caso de uso "Ver datos en gráfico de barras"</i>	87
<i>Ilustración 63: Filtro interfaz gráfica</i>	88
<i>Ilustración 64: Carga de informes</i>	91
<i>Ilustración 65: Añadir relación medicamento</i>	91
<i>Ilustración 66: Añadir relación enfermedad</i>	92
<i>Ilustración 67: Procesar enfermedades causadas por medicamentos</i>	93
<i>Ilustración 68: Diagrama estadísticas básicas</i>	94
<i>Ilustración 69: Diagrama estadísticas avanzadas</i>	96
<i>Ilustración 70: Instalar MySQL</i>	99
<i>Ilustración 71: Instalar MySQL Workbench</i>	100
<i>Ilustración 72: Acceder a Mysql Workbench</i>	100
<i>Ilustración 73: Iniciar conexión MySQL</i>	101
<i>Ilustración 74: Crear nuevo esquema</i>	101
<i>Ilustración 75: Importar base de datos</i>	102
<i>Ilustración 76: Esquema creado</i>	102
<i>Ilustración 77: Instalar Apache Tomcat</i>	102
<i>Ilustración 78: Importar archivo .war</i>	102
<i>Ilustración 79: Acceso a la aplicación</i>	103
<i>Ilustración 80: Carga de informes</i>	105
<i>Ilustración 81: Formato de carga de informes</i>	106
<i>Ilustración 82: Carga de informes errónea</i>	106
<i>Ilustración 83: Carga de informes correcta</i>	106

<i>Ilustración 84: Información estadística básica .....</i>	<i>107</i>
<i>Ilustración 85: Gráfico búsqueda básica .....</i>	<i>108</i>
<i>Ilustración 86: Información estadística avanzada .....</i>	<i>108</i>
<i>Ilustración 87: Búsqueda avanzada en gráfico de barras .....</i>	<i>109</i>

## Índice de tablas

<i>Tabla 1: Definición de objetivos</i> .....	19
<i>Tabla 2: Análisis y diseño (Prototipo I)</i> .....	19
<i>Tabla 3: Análisis y diseño (Prototipo II)</i> .....	20
<i>Tabla 4: Análisis y diseño (Prototipo III)</i> .....	20
<i>Tabla 5: Configuración entorno del proyecto</i> .....	20
<i>Tabla 6: Implementación (Prototipo I)</i> .....	21
<i>Tabla 7: Implementación (Prototipo II)</i> .....	21
<i>Tabla 8: Implementación (Prototipo III)</i> .....	22
<i>Tabla 9: Documentación (Memoria del proyecto)</i> .....	22
<i>Tabla 10: Documentación (Manual de usuario)</i> .....	23
<i>Tabla 11: Resumen horas estimadas</i> .....	23
<i>Tabla 12: Medidas de impacto</i> .....	27
<i>Tabla 13: Rotura de equipo</i> .....	27
<i>Tabla 14: Infección de virus en el equipo</i> .....	28
<i>Tabla 15: Baja del desarrollador</i> .....	28
<i>Tabla 16: Bloqueo en el desarrollo</i> .....	28
<i>Tabla 17: Análisis y estimación errónea</i> .....	29
<i>Tabla 18: Coste Software</i> .....	29
<i>Tabla 19: Coste hardware</i> .....	30
<i>Tabla 20: Coste mano de obra</i> .....	30
<i>Tabla 21: coste total proyecto</i> .....	30
<i>Tabla 22: Lenguaje de desarrollo (Servidor)</i> .....	35
<i>Tabla 23: Sistema gestor de base de datos</i> .....	36
<i>Tabla 24: Herramientas visualización de gráficos</i> .....	37
<i>Tabla 25: Pruebas prototipo I</i> .....	70
<i>Tabla 26: Pruebas prototipo II</i> .....	71
<i>Tabla 27: Pruebas prototipo III</i> .....	72
<i>Tabla 28: Errores y soluciones</i> .....	73

# 1. INTRODUCCIÓN

El mundo de la tecnología cada vez es más imprescindible en el ámbito de la medicina. Tanto es así, que hoy en día se haría muy complicado desprenderse de las grandes ventajas que ha supuesto utilizarla. En este sector, la investigación es un factor muy importante, sin embargo, el volumen de datos recopilados a lo largo de los años es cada vez mayor y eso puede dificultar bastante a la hora de encontrar datos útiles en esa información.

Por ese motivo es importante el uso de las estadísticas. Son algo más que números, ya que permiten abordar los problemas de salud y priorizar el uso de recursos sanitarios muy valiosos.

Un ejemplo de esto, puede ser que actualmente se han descubierto nuevos casos en los que ciertos medicamentos han provocado cáncer. Encontrar qué medicamento o cual ha sido la causa de la enfermedad conociendo únicamente a que pacientes ha afectado es complicado, ya que habría que mirar todas sus historias clínicas y ver que tienen en común. Sin embargo esto puede llevar mucho tiempo, y eso, es algo demasiado valioso en este sector.

Por este motivo, en este proyecto se buscará el paso de multitud de datos a conocimiento.



*Ilustración 1: Paso de multitud de datos a conocimiento*

Se tratará de acortar ese tiempo de investigación necesario, ofreciendo una herramienta capaz de extraer información de enfermedades, medicamentos y relaciones entre sí, de multitud de informes médicos. A su vez, ofrecerá la

posibilidad de visualizar estadísticas temporales basadas en búsquedas concretas. De esta manera, se podría visualizar de manera rápida qué medicamentos pueden estar causando una determinada enfermedad.

Además de servir de apoyo para la toma de decisiones clínicas, también puede serlo para aspectos económicos relacionados con este sector. Esas estadísticas, podrán servir como guía de qué medicamentos se utilizan más a menudo, y en función de ellas establecer el capital que se desea invertir en su compra.

## 2. PLANTEAMIENTO INICIAL

### 2.1. Objetivos

Los objetivos principales del proyecto son los siguientes:

- **Automatizar el proceso de obtención de datos de informes médicos:** Se procesarán multitud de ficheros, y se extraerá de ellos información relativa a las enfermedades y medicamentos que aparecen, así como de las relaciones que estos puedan tener.
- **Obtención de estadísticas médicas:** Se desarrollará un sistema capaz de mostrar al usuario estadísticas temporales en base a búsquedas que se realizarán en los informes médicos que se carguen en el sistema.
- **Aprendizaje personal:** En la aplicación se utilizarán nuevas tecnologías, por lo que se necesitará invertir tiempo en su aprendizaje, y se obtendrán de esta manera nuevos conocimientos.

### 2.2. Arquitectura

Para el desarrollo del proyecto, se ha optado por emplear una arquitectura Cliente/Servidor, que constará de dos partes:

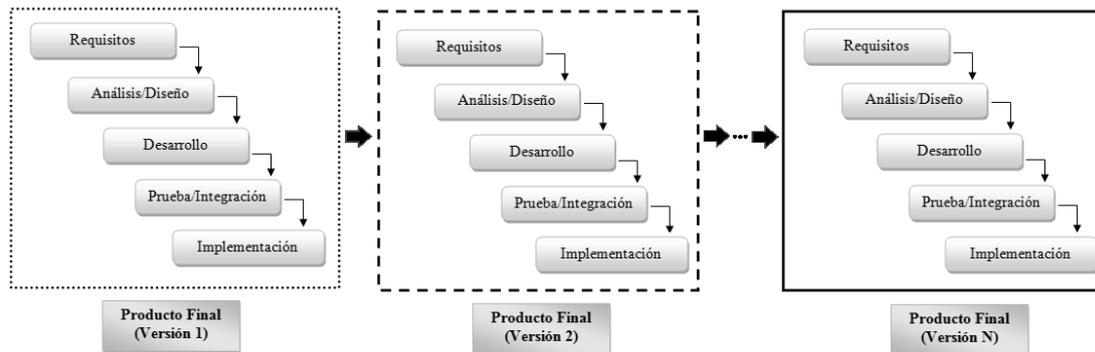
- El cliente será el navegador que utilicen los usuarios para acceder a la interfaz gráfica de la aplicación, y será quien realice las solicitudes necesarias al servidor así como la parte que muestre los datos obtenidos al usuario.
- El servidor será quien se encargue de procesar las solicitudes enviadas por los clientes, conectarse con el sistema gestor de base de datos para obtener la información necesaria y devolver el resultado correspondiente.



*Ilustración 2: Arquitectura Cliente/Servidor*

### 2.3. Alcance

Teniendo en cuenta que las funcionalidades de la aplicación podrían tratarse de manera independiente, se ha podido contemplar que el modelo que más se adecúa es la utilización de un modelo incremental, es decir basado en distintos prototipos. Cada una de las iteraciones generará un prototipo que será utilizado en la siguiente etapa. De esta manera se obtendrá una versión cada vez más completa, ampliando las funcionalidades progresivamente y que finalmente darán lugar a un prototipo final que tendrá todas las funcionalidades definidas inicialmente.

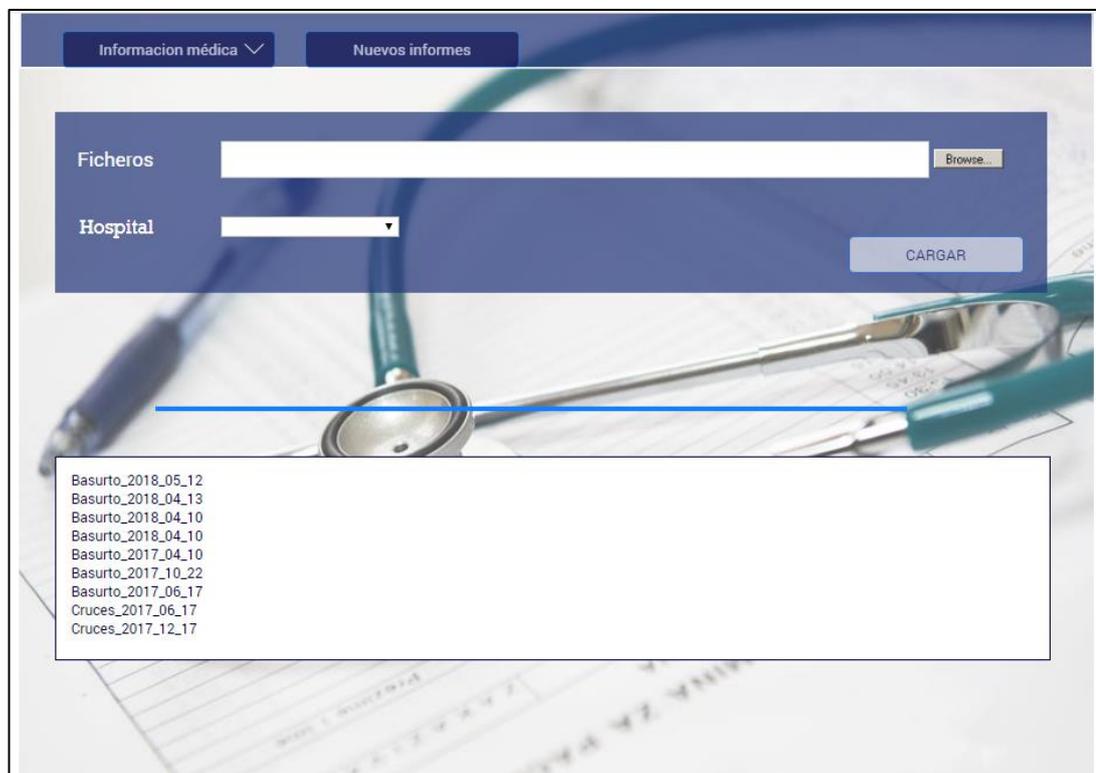


*Ilustración 3: Modelo incremental*

Se creará una aplicación basada en un diseño de 3 prototipos.

- **Primer prototipo**

La interfaz tendrá una página que permitirá cargar informes médicos, los procesará y se almacenará en la base de datos toda la información obtenida de ellos.



*Ilustración 4: Diseño primer prototipo*

- **Segundo prototipo**

Se añadirá una nueva pantalla que permitirá obtener estadísticas de forma visual de la información guardada en el sistema, y mostrará cuantos casos médicos se han producido, basándose en un filtro básico elegido por el usuario, de enfermedades o medicamentos.



*Ilustración 5: Diseño segundo prototipo*

- **Tercer prototipo**

Se ampliará la pantalla del segundo prototipo. Esta, tendrá un filtro dinámico más avanzado, que permitirá obtener los casos de enfermedades causadas por distintos medicamentos.



Ilustración 6: Diseño tercer prototipo

### 2.3.1. Tareas del proyecto

En la siguiente figura, se muestra la estructura de descomposición del trabajo (EDT). En ella se indica cada una de las tareas en las que se ha dividido el desarrollo del proyecto.

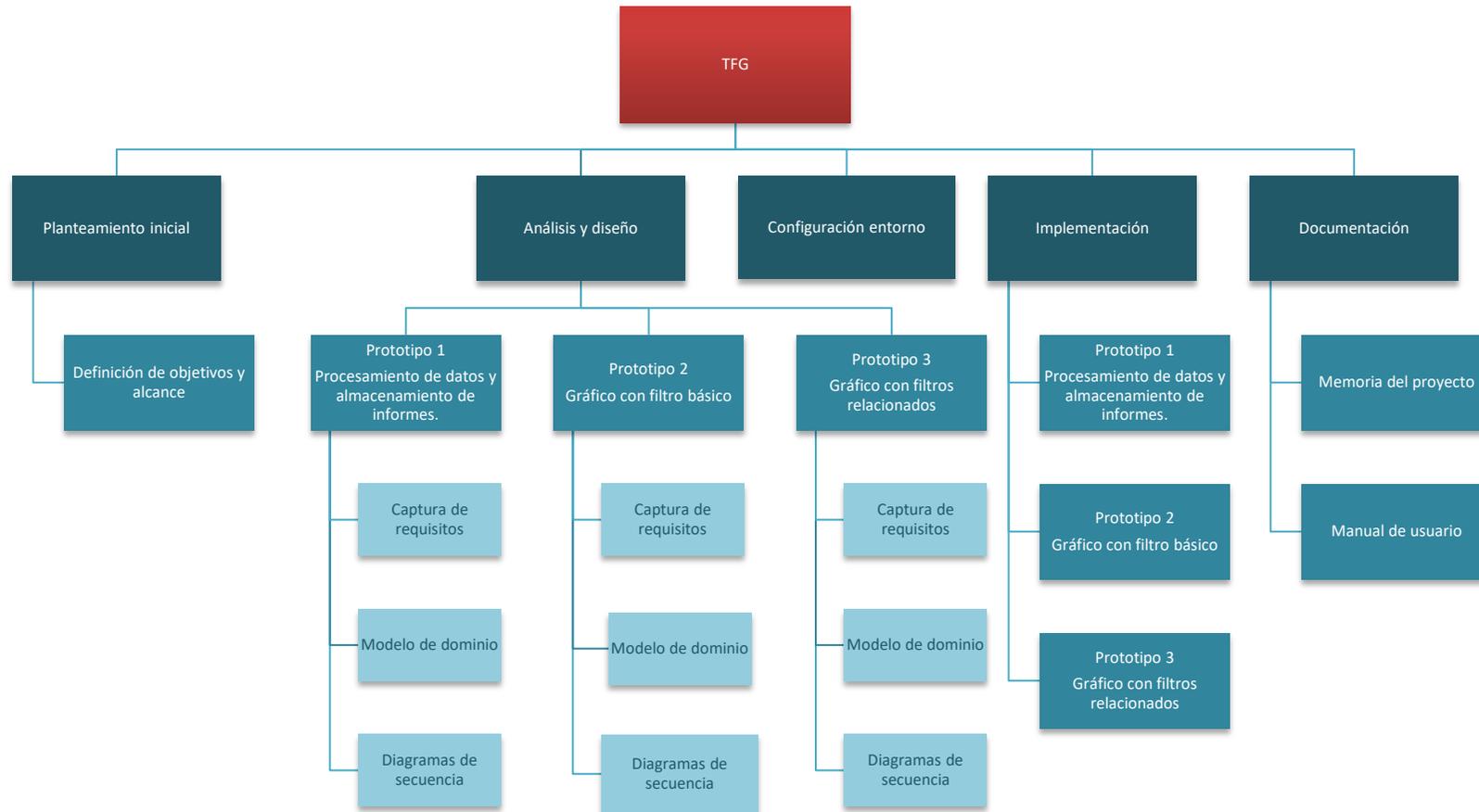


Ilustración 7: EDT

A continuación se muestra la descripción de las tareas, así como el tiempo que se estima de desarrollo para cada una de ellas.

Definición de objetivos	
<b>Descripción:</b>	Definición completa de los objetivos del proyecto y planificación temporal basada en el desglose de tareas.
<b>Entradas:</b>	
<b>Salidas/Entregables:</b>	Documento explicativo de los objetivos del proyecto y su planificación.
<b>Recursos utilizados:</b>	Equipo informático y editor de texto de Microsoft Office.
<b>Duración estimada:</b>	7h

*Tabla 1: Definición de objetivos*

Análisis y diseño (Prototipo I)	
<b>Descripción:</b>	Captura de requisitos, modelo de dominio y diagramas de secuencia del primer prototipo.
<b>Entradas:</b>	
<b>Salidas/Entregables:</b>	Documento de diseño del primer prototipo.
<b>Recursos utilizados:</b>	Equipo informático y Cacao
<b>Duración estimada:</b>	25h

*Tabla 2: Análisis y diseño (Prototipo I)*

<b>Análisis y diseño (Prototipo II)</b>	
<b>Descripción:</b>	Captura de requisitos, modelo de dominio y diagramas de secuencia del segundo prototipo.
<b>Entradas:</b>	
<b>Salidas/Entregables:</b>	Documento de diseño del segundo prototipo
<b>Recursos utilizados:</b>	Equipo informático y Cacao
<b>Duración estimada:</b>	30h

*Tabla 3: Análisis y diseño (Prototipo II)*

<b>Análisis y diseño (Prototipo III)</b>	
<b>Descripción:</b>	Captura de requisitos, modelo de dominio y diagramas de secuencia del tercer prototipo.
<b>Entradas:</b>	
<b>Salidas/Entregables:</b>	Documento de diseño del tercer prototipo.
<b>Recursos utilizados:</b>	Equipo informático y Cacao
<b>Duración estimada:</b>	30h

*Tabla 4: Análisis y diseño (Prototipo III)*

<b>Configuración entorno del proyecto</b>	
<b>Descripción:</b>	Preparación y configuración del entorno para la implementación del proyecto.
<b>Entradas:</b>	
<b>Salidas/Entregables:</b>	Entorno de la aplicación
<b>Recursos utilizados:</b>	Eclipse
<b>Duración estimada:</b>	15h

*Tabla 5: Configuración entorno del proyecto*

Implementación (Prototipo I)	
<b>Descripción:</b>	Desarrollo del software que procesará los informes médicos y los almacenará en la base de datos.
<b>Entradas:</b>	
<b>Salidas/Entregables:</b>	Primer prototipo
<b>Recursos utilizados:</b>	Equipo informático, entorno de desarrollo de Eclipse y base de datos MySQL.
<b>Duración estimada:</b>	80h

*Tabla 6: Implementación (Prototipo I)*

Implementación (Prototipo II)	
<b>Descripción:</b>	Añadirá una nueva pantalla al primer prototipo, la cual generará un histórico de los informes médicos procesados anteriormente, y permitirá obtenerlo filtrando por enfermedades o medicamentos.
<b>Entradas:</b>	Primer prototipo
<b>Salidas/Entregables:</b>	Segundo prototipo
<b>Recursos utilizados:</b>	Equipo informático, entorno de desarrollo de Eclipse y base de datos MySQL.
<b>Duración estimada:</b>	90h

*Tabla 7: Implementación (Prototipo II)*

### Implementación (Prototipo III)

<b>Descripción:</b>	Ampliará la pantalla del segundo prototipo y permitirá generar el histórico de datos basándose en filtros de hospitales o fechas de informes, así como filtros combinados.
<b>Entradas</b>	Segundo prototipo
<b>Salidas/Entregables:</b>	Tercer prototipo
<b>Recursos utilizados:</b>	Equipo informático, entorno de desarrollo de Eclipse y base de datos MySQL.
<b>Duración estimada:</b>	75h

*Tabla 8: Implementación (Prototipo III)*

### Documentación (Memoria del proyecto)

<b>Descripción:</b>	Desarrollo de la documentación completa del proyecto.
<b>Entradas:</b>	
<b>Salidas/Entregables:</b>	Documentación del proyecto
<b>Recursos utilizados:</b>	Equipo informático y editor de texto de Microsoft Office.
<b>Duración estimada:</b>	60h

*Tabla 9: Documentación (Memoria del proyecto)*

<b>Documentación (Manual de usuario)</b>	
<b>Descripción:</b>	Desarrollo de un documento que explique el funcionamiento completo de la aplicación.
<b>Entradas:</b>	
<b>Salidas/Entregables:</b>	Manual de usuario
<b>Recursos utilizados:</b>	Equipo informático y editor de texto de Microsoft Office.
<b>Duración estimada:</b>	18h

*Tabla 10: Documentación (Manual de usuario)*

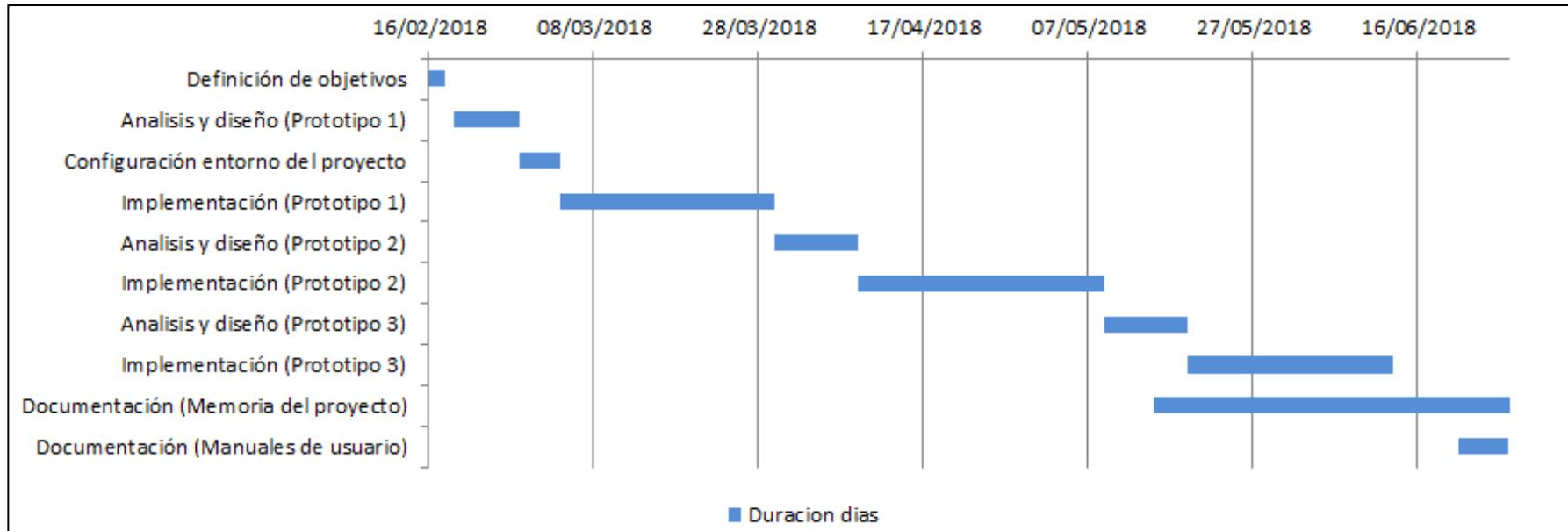
A continuación se muestra una tabla con el resumen de las horas necesarias para las tareas del proyecto.

<b>Tarea</b>	<b>Horas estimadas</b>
Definición de objetivos	7h
Análisis y diseño (Prototipo I)	25h
Análisis y diseño (Prototipo II)	30h
Análisis y diseño (Prototipo III)	30h
Configuración entorno del proyecto	15h
Implementación (Prototipo I)	80h
Implementación (Prototipo II)	90h
Implementación (Prototipo III)	75h
Documentación (Memoria del proyecto)	60h
Documentación (Manual de usuario)	18h

*Tabla 11: Resumen horas estimadas*

## 2.4. Planificación temporal

Una vez realizada la estimación de las tareas, se muestra en el diagrama de Gantt la dedicación temporal que se estima para el proyecto. La estimación inicial es de 430h. Por motivos de trabajo, se ha estimado que en un principio, se dedicarán 3h diarias a la semana, por lo que teniendo en cuenta la duración de cada tarea definida, el proyecto empezaría el 16 de enero y finalizaría el 27 de junio de 2018.



*Ilustración 8: Diagrama de Gantt*

## 2.5. Herramientas

A continuación se explican las herramientas utilizadas para el desarrollo del trabajo:

- **Eclipse:** Es un entorno de desarrollo compuesto por un conjunto de herramientas, de código abierto y multiplataforma. Es una plataforma de desarrollo y compilación de elementos como sitios web, programas en C++ o Java. [1]
- **Apache Tomcat:** Es un software de código abierto desarrollado con Java, que funciona como contenedor de servlets y JSPs. [2]
- **Chrome:** Es un navegador web creado por Google. Ha sido utilizado para la búsqueda de recursos necesarios para la aplicación, tanto herramientas utilizadas, como manuales de estas. [3]
- **Cacoo:** Herramienta online y gratuita, que permite la creación de diagramas de todo tipo, y contiene plantillas de ayuda para su diseño, así como gran variedad de paletas de objetos para los diagramas. [4]
- **Justinmind:** Herramienta de prototipado, para páginas web y aplicaciones móviles. Permite utilizar distintas plantillas para los distintos dispositivos, y personalizarlos. Además es posible añadirle funcionalidad con elementos clickables y links de interacción. [5]
- **Microsoft Office:** Es un paquete de programas informáticos desarrollado para realizar tareas ofimáticas. Ha sido utilizado para realizar la documentación necesaria del proyecto. [6]
- **GitHub:** Es una plataforma de desarrollo colaborativo de software que permite almacenar copias de seguridad de los proyectos utilizando el sistema de control de versiones. Ha sido utilizado para realizar las copias de seguridad semanales del desarrollo de la aplicación. [7]

## 2.6. Gestión de riesgos

Es importante plantear un análisis de los riesgos que puedan ocasionar pérdidas potenciales durante el desarrollo del proyecto, y así poder prevenir al máximo las consecuencias que pudieran tener.

Para ello, se crearán unas medidas orientativas que permitan valorar cada uno de los riesgos posibles y su impacto en caso de que ocurriera.

Impacto	Retraso en el proyecto
<b>Leve</b>	1 día
<b>Medio</b>	3 - 4 días
<b>Grande</b>	5 - 7 días

*Tabla 12: Medidas de impacto*

A continuación se detallará cada uno de los riesgos y su plan de contingencia, así como el impacto que ocasionaría.

- **Riesgos de hardware y software**

<b>Rotura del equipo</b>	
<b>Plan de contingencia</b>	Restaurar las copias de seguridad realizadas semanalmente a lo largo del desarrollo
<b>Impacto</b>	Leve
<b>Probabilidad</b>	10%

*Tabla 13: Rotura de equipo*

<b>Infección de virus en el equipo</b>	
<b>Plan de contingencia</b>	Eliminar las amenazas mediante una herramienta de antivirus, y restaurar la última copia de seguridad del proyecto.
<b>Impacto</b>	Leve
<b>Probabilidad</b>	10%

*Tabla 14: Infección de virus en el equipo*

- **Riesgos del desarrollador**

<b>Baja del desarrollador</b>	
<b>Plan de contingencia</b>	Reorganizar la planificación del proyecto para evitar el retraso del mismo lo máximo posible.
<b>Impacto</b>	Leve
<b>Probabilidad</b>	10%

*Tabla 15: Baja del desarrollador*

<b>Bloqueo en el desarrollo por el uso de nuevas tecnologías</b>	
<b>Plan de contingencia</b>	En la planificación del proyecto se ha tenido en cuenta un margen de horas para esta situación.
<b>Impacto</b>	Medio
<b>Probabilidad</b>	55%

*Tabla 16: Bloqueo en el desarrollo*

<b>Análisis y estimación errónea</b>	
<b>Plan de contingencia</b>	Reajustar la planificación temporal o el alcance del proyecto.
<b>Impacto</b>	Grande
<b>Probabilidad</b>	55%

*Tabla 17: Análisis y estimación errónea*

## 2.7. Evaluación económica

En este apartado se definirán los costes del desarrollo de la aplicación. Para ello se expondrán por una parte, las herramientas utilizadas y su coste, y por otra la mano de obra del proyecto.

Para el desarrollo se utilizarán únicamente herramientas libres, esto es, que no necesitarán de la realización de ningún pago para la obtención de las respectivas licencias. La licencia de Microsoft Office estará incluida en el precio del ordenador, por lo tanto de Software no será necesario, de momento, ningún gasto.

<b>Software</b>	<b>Coste</b>
<b>Eclipse</b>	0,00 €
<b>Microsoft Office</b>	0,00 €
<b>GitHub</b>	0,00 €
<b>Chrome</b>	0,00 €
<b>Cacao</b>	0,00 €

*Tabla 18: Coste Software*

También será necesario tener en cuenta el coste del equipo portátil que se utilizará.

Hardware	Coste
<b>Acer Aspire E5-571-39GE Intel Core i3</b>	450€

*Tabla 19: Coste hardware*

Basándose en el Convenio Colectivo de Oficinas y Despachos de Bizkaia, se estimará el coste de la mano de obra de un total de 430 horas de proyecto. [8]

Mano de obra	Coste Anual	Coste por hora	Coste Total
<b>1 Analista Programador</b>	23.783,55 €	13,64 €	5.865,20 €

*Tabla 20: Coste mano de obra*

A los gastos obtenidos anteriormente, habría que añadir el mantenimiento, es decir, gastos de electricidad y agua. Para ello se considerará un 7% de los gastos totales del proyecto.

A continuación se muestra el presupuesto total:

Gastos del proyecto	Coste
<b>Gastos de software</b>	0,00 €
<b>Gastos de hardware</b>	450 €
<b>Gastos de mano de obra</b>	5.865,20 €
<b>Gastos de mantenimiento</b>	454,47 €
<b>TOTAL : 6.769,67 €</b>	

*Tabla 21: coste total proyecto*

## 3. ANTECEDENTES

### 3.1. Descripción situación actual

A la hora de comenzar un proyecto, es necesario hacer un análisis de la situación actual respecto al tema que se va a tratar.

Actualmente existe gran variedad de herramientas y aplicaciones creadas que facilitan mucho la gestión de información médica. La gran mayoría están dedicadas a la gestión de pacientes de hospitales, con el fin de facilitar la búsqueda de sus informes y tener siempre la información actualizada de todos ellos. Sin embargo, todas ellas están creadas a nivel personal, es decir, que el usuario puede ver informes individuales pero no obtener información a nivel general de varios de ellos.

La mayor parte, contienen toda la información extraída en informes estadísticos que se realizan anualmente y en los cuales puede ser costoso encontrar exactamente lo que se necesita en cada momento.

Uno de los ejemplos que hace uso de estas estadísticas y que puede que sea la más utilizada por su importancia, es la página web de la Organización Mundial de la Salud (OMS). [9]

En ella, podemos encontrar estadísticas mundiales recopiladas de múltiples fuentes, como pueden ser las encuestas realizadas, los censos de población, y los informes médicos. Esta información es accesible para cualquier persona que la necesite, tanto pacientes como personal sanitario.

Para acceder a ella es necesario elegir qué estadísticas se quieren mostrar.



Ilustración 9: Elección del tema estadístico

Como se puede observar en la Ilustración 10, una vez elegido el tema, permite filtrar por continentes, y muestra los datos de todos los países, sin embargo no es posible filtrar por ninguna característica más.

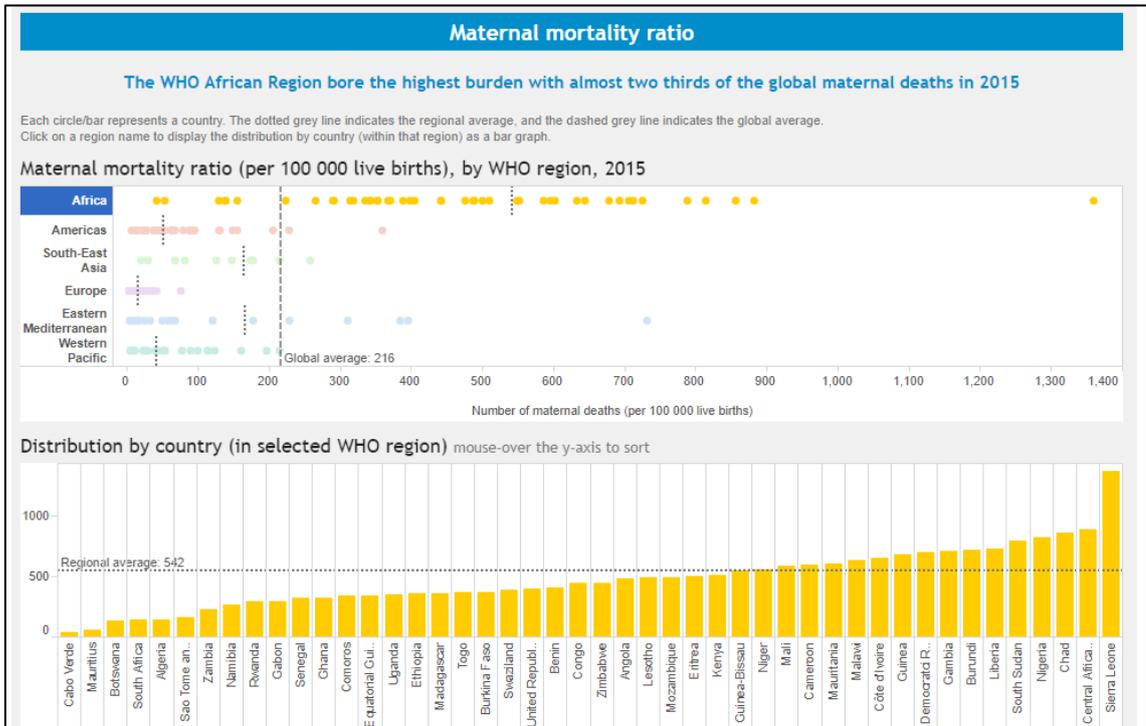
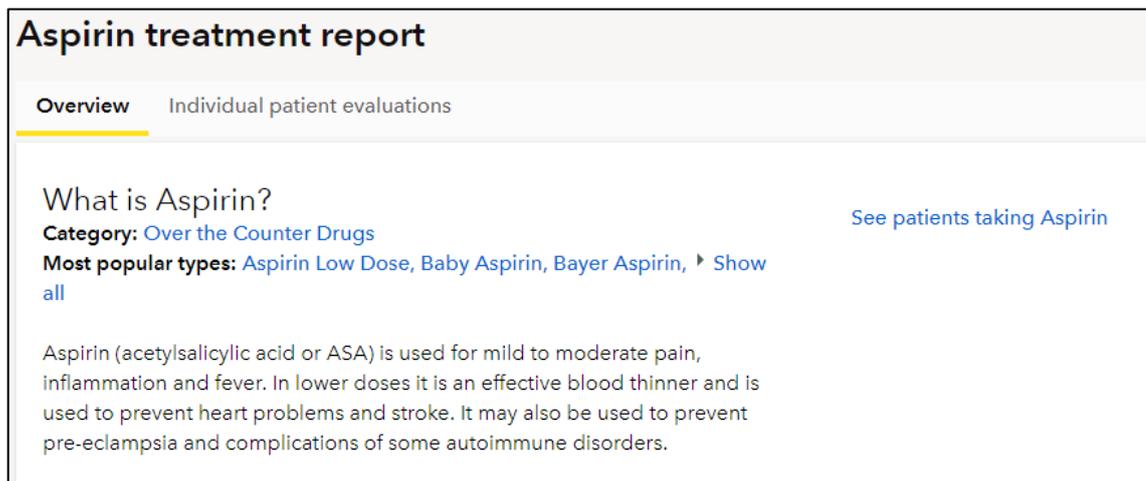


Ilustración 10: Estadísticas OMS

Otra herramienta similar es *Patientslikeme*. Es una página accesible para cualquier usuario ya sean pacientes, personal de investigación, o cualquier persona que necesite obtener información. Esta herramienta permite a los usuarios compartir tratamientos y síntomas de enfermedades que padecen. Además también les permite interactuar con más gente y ver sus resultados a través de mensajes, lo cual les ayuda a empatizar con otras personas, y poder obtener así datos más concretos sobre lo que buscan. [10]

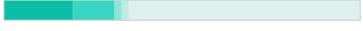
En la siguiente ilustración se muestra un ejemplo de las estadísticas que se muestran al realizar la búsqueda de un medicamento.



The screenshot shows a web interface for an Aspirin treatment report. At the top, there is a title "Aspirin treatment report" and two tabs: "Overview" (which is selected and highlighted with a yellow underline) and "Individual patient evaluations". Below the tabs, the main content area starts with the question "What is Aspirin?". To the right of this question is a link that says "See patients taking Aspirin". Below the question, there is a "Category:" field with the value "Over the Counter Drugs". Underneath that, there is a "Most popular types:" field with the value "Aspirin Low Dose, Baby Aspirin, Bayer Aspirin," followed by a "Show all" link. At the bottom of the content area, there is a paragraph of text describing Aspirin: "Aspirin (acetylsalicylic acid or ASA) is used for mild to moderate pain, inflammation and fever. In lower doses it is an effective blood thinner and is used to prevent heart problems and stroke. It may also be used to prevent pre-eclampsia and complications of some autoimmune disorders."

*Ilustración 11: Patientslikeme ejemplo de búsqueda*

En primer lugar se muestra la categoría y los tipos más populares del medicamento así como una breve descripción del mismo. Seguido de esto y como se observa en la *Ilustración 12*, se muestran las estadísticas de los pacientes que lo han utilizado, así como su efectividad para cada enfermedad en la que se haya incluido como tratamiento.

Reported purpose & perceived effectiveness			
Purpose	Patients	Evaluations	Perceived Effectiveness
General health	693	182	
Coronary artery disease	330	126	
Migraine	326	82	
Prevent blood clotting	183	44	
Heart attack (myocardial infarction)	130	31	
Support heart health	119	86	

*Ilustración 12: Patientslikeme ejemplo de estadísticas*

En esta página web, se puede obtener mucha información, sin embargo uno de los inconvenientes que tiene, es que cualquier persona puede incluir los datos en ella, por lo que puede que las estadísticas no sean del todo veraces.

Partiendo de herramientas como las mostradas anteriormente, se pretende crear una nueva aplicación web que pueda complementar la funcionalidad de estas, y optimizar sus recursos, permitiendo la interacción del usuario en todo momento para la búsqueda de estadísticas y no tener que visualizar así, múltiples documentos en busca de la información que se necesite obtener.

## 3.2. Análisis de alternativas

La aplicación que se propone desarrollar es posible realizarla con múltiples herramientas, y por ello, es necesario realizar un estudio de las opciones para elegir así la más adecuada para el cumplimiento de las necesidades del proyecto. Se valorarán alternativas para los siguientes aspectos ya que se han considerado los más importantes:

- Lenguaje de desarrollo (Servidor)
- Herramienta para la visualización de gráficos
- Sistema gestor de bases de datos

A continuación se muestra una comparativa de cada uno de los aspectos.

### 3.2.1. Lenguaje de desarrollo (Servidor)

	Ventajas	Desventajas
<b>PHP</b>	<ul style="list-style-type: none"> <li>- Rápido</li> <li>- Libre</li> <li>- Multiplataforma</li> <li>- Amplia documentación</li> <li>- Soporta gran cantidad de base de datos</li> <li>- No requiere manejo detallado del bajo nivel, ni definición de tipos de variable</li> </ul>	<ul style="list-style-type: none"> <li>- Dificulta la modularización</li> <li>- Programación orientada a objetos deficiente para grandes aplicaciones</li> <li>- Dificulta la organización de las capas de la aplicación</li> </ul>
<b>Java (JSP)</b>	<ul style="list-style-type: none"> <li>- Integridad con módulos de Java</li> <li>- Apoyo de librerías</li> <li>- Multiplataforma</li> <li>- Amplia documentación</li> </ul>	<ul style="list-style-type: none"> <li>- Limitada portabilidad</li> <li>- Complejidad de aprendizaje</li> </ul>
<b>ASP .NET</b>	<ul style="list-style-type: none"> <li>- Orientado a objetos</li> <li>- Facilita el mantenimiento de grandes aplicaciones</li> <li>- Mayor seguridad</li> <li>- División entre capa de diseño y capa de código</li> </ul>	<ul style="list-style-type: none"> <li>- Consumo alto de recursos</li> <li>- Plataforma de pago</li> <li>- Requiere servidor IIS</li> </ul>

Tabla 22: Lenguaje de desarrollo (Servidor)

### 3.2.2. Sistema gestor de bases de datos

En una aplicación como esta en la que se podrá albergar una gran cantidad de informes médicos, es importante valorar como se guardará esa información. Para ello será necesario elegir la mejor alternativa para la utilización del gestor de base de datos, de manera que se pueda mantener un control seguro de toda la información y se pueda acceder a ella de la forma más rápida y sencilla posible.

	Ventajas	Desventajas
<b>ORACLE</b>	<ul style="list-style-type: none"> <li>- Más utilizado</li> <li>- Multiplataforma</li> <li>- Soporta varios sistemas operativos</li> <li>- Permite el uso de particiones para mejora de eficiencia</li> </ul>	<ul style="list-style-type: none"> <li>- Propietario</li> <li>- No permite ejecutar procesamientos recursivos</li> </ul>
<b>MySQL</b>	<ul style="list-style-type: none"> <li>- Open Source</li> <li>- Soporta varios Sistemas operativo</li> <li>- Sin limitaciones para conexiones simultáneas</li> <li>- Portabilidad sencilla</li> </ul>	<ul style="list-style-type: none"> <li>- Gran porcentaje de utilidades sin documentar</li> </ul>

*Tabla 23: Sistema gestor de base de datos*

### 3.2.3. Herramientas para visualización de gráficos

Para el usuario será importante la forma en la que obtiene la información, ya que no es lo mismo ver los datos escritos de cualquier forma en una interfaz, que verlos gráficamente. De esta manera, podrá ver más fácilmente y a primera vista lo que necesita y se le ofrecerá una mejor experiencia.

Para este aspecto es importante valorar la herramienta con la que se van a realizar los gráficos de las estadísticas. Existen multitud de librerías que facilitan este desarrollo pero se ha optado por valorar dos de las más conocidas.

Característica	ChartJS	D3js
<b>Curva de aprendizaje</b>	- Baja	- Elevada
<b>Compatibilidad</b>	- Compatible con la mayoría de navegadores	- Compatible con los navegadores más modernos
<b>Documentación</b>	- Amplia	- Amplia
<b>Diseño</b>	- Responsivo	- No responsivo

*Tabla 24: Herramientas visualización de gráficos*

### 3.2.4. Elección de herramientas

Una vez valoradas todas las herramientas, es hora de tomar una decisión y elegir las más adecuadas teniendo en cuenta cada uno de los aspectos positivos y negativos de cada una de ellas.

Comenzando por el lenguaje de la parte del servidor, se ha decidido desarrollar con JSP. La razón principal de esa elección ha sido el conocimiento previo del lenguaje, que en comparación con los otros dos, es bastante más alto.

Las 3 alternativas expuestas tienen multitud de ventajas, sin embargo ASP .NET además de ser de pago a diferencia de las dos restantes, necesita de un servidor concreto para su funcionamiento lo que dificulta la compatibilidad con el resto de componentes que serán necesarios para la aplicación. PHP podría ser otra de las alternativas válidas, sin embargo no ofrece la misma organización entre las capas de la aplicación. [11][12]

Una vez elegido el lenguaje Java, se ha decidido incluir la utilización de Spring Framework, para profundizar y entender mejor las ventajas que puede llegar a ofrecer.

Por otra parte se elige MySQL como sistema gestor de base de datos. Principalmente se ha elegido por el conocimiento previo de su funcionamiento gracias a proyectos realizados con anterioridad. Sin embargo, con Oracle no se ha llegado a profundizar de la misma manera, y valorando el tamaño de la aplicación que se desea realizar no será necesario recurrir a este gestor, ya que es más adecuado para empresas o proyectos más amplios y complejos. [13][14]

Por último, respecto a la herramienta para la visualización de gráficos se ha elegido ChartJS. Aunque la librería D3js ofrezca multitud de tipos de gráficos y amplia documentación para todos ellos, el tiempo dedicado para su aprendizaje es más elevado. Además es importante valorar también otros aspectos como la posibilidad de gráficos responsivos, ya que la librería D3js no lo ofrece y es algo

que habría que tener en cuenta puesto que hoy en día los usuarios acceden a las páginas web principalmente desde dispositivos móviles. [15][16]

A continuación se muestra un esquema de las tecnologías que se utilizarán para el desarrollo.

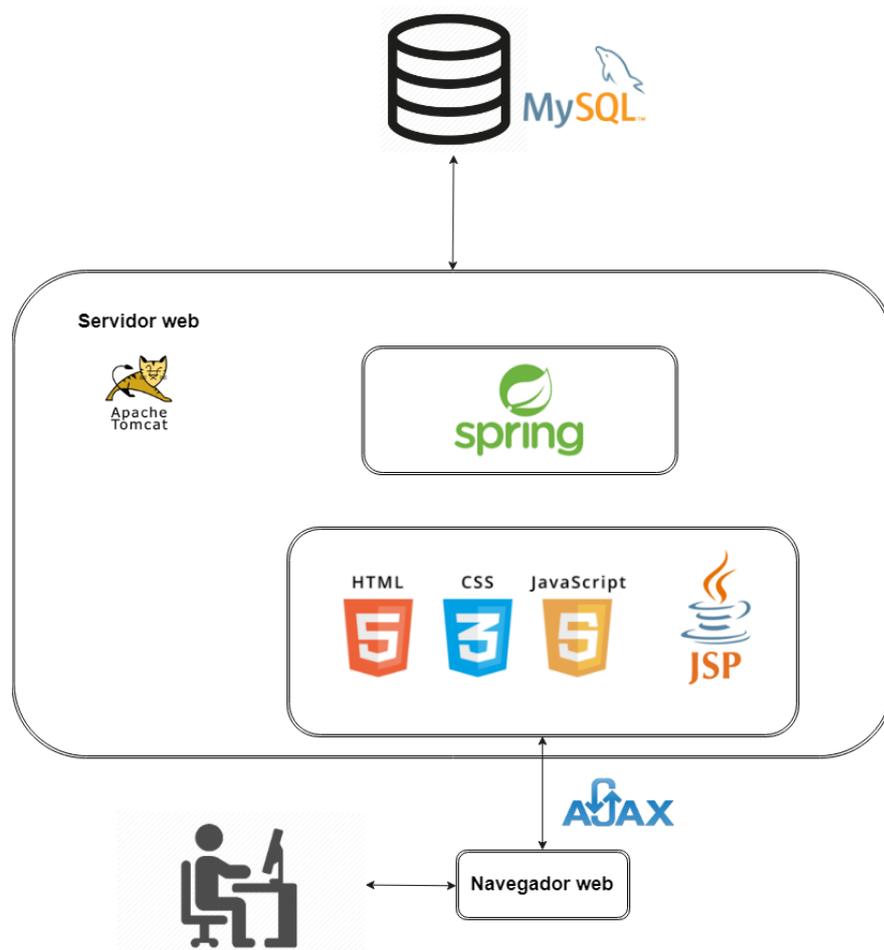


Ilustración 13: Esquema tecnologías



## 4. CAPTURA DE REQUISITOS

### 4.1. Origen de los datos

La aplicación propuesta, permitirá cargar informes médicos pertenecientes a distintos hospitales. Para un mejor tratamiento de los datos, y poder así guardar de manera más correcta la información en la base de datos, se ha partido de datos preprocesados con la herramienta BRAT (Rapid Annotation Tool). [17]

Esta herramienta web ofrece las siguientes características:

- Permite agregar notas a documentos de texto fácilmente, e incluso conectarlas entre sí.
- Es posible crear anotaciones en cualquier idioma.
- Integra una funcionalidad para poder comparar varios conjuntos de anotaciones automáticamente, marcando las diferencias entre si y pudiendo así por ejemplo, evaluar sistemas automáticos comparando resultados con anotaciones creadas por los usuarios para identificar errores con mayor rapidez.
- Integración con recursos externos.

A continuación se muestra un ejemplo de un informe médico en el que se han incluido las anotaciones.

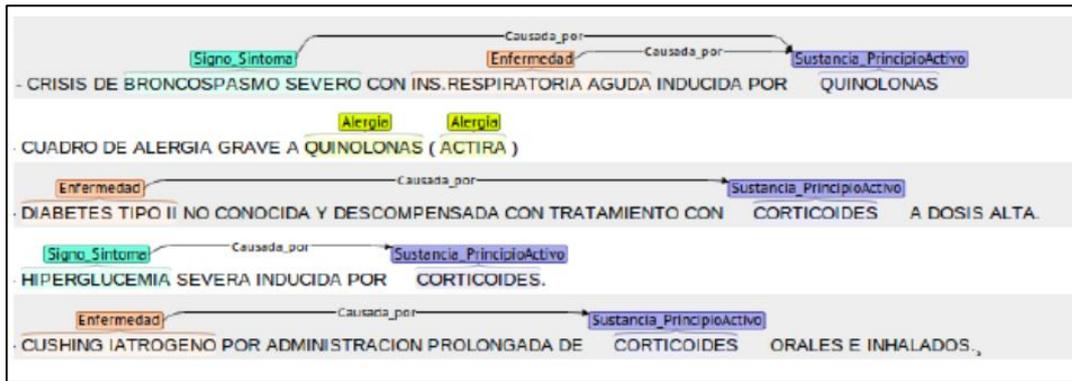


Ilustración 14: BRAT ejemplo con anotaciones

Como se puede observar en la *Ilustración 14*, hay distintas categorías, diferenciadas por colores: síntomas, enfermedades, e incluso la relación entre los medicamentos y las enfermedades que han causado.

Cada fichero al que se le añaden anotaciones, tendrá asociado un fichero con extensión .ann. Este tipo de fichero será el que se utilice como entrada de datos para la aplicación que se va a crear, ya que permite leer la información de manera más sencilla gracias a como está estructurada en el fichero. A continuación se muestra un ejemplo.

Como se puede observar en la *Ilustración 16*, existen distintos tipos de identificadores para las anotaciones, compuestas por el carácter único que identifica la anotación, junto con un número:

- T: Anotación con texto
- R: Relación
- E: Evento
- A: Atributo
- #: Nota

La descripción que aparece después de estos identificadores, hace referencia al tipo de evento, en este caso diferencia entre medicamentos (*Grp\_Medicamento*), enfermedades (*Grp\_Enfermedad*), y en el caso de que existiera una enfermedad causada por un medicamento con la etiqueta *Causado\_por*.

A continuación de esta descripción, se define el *offset*, esto es, la posición desde y hasta la cual se encuentra el texto de la propiedad en el fichero. Por último, seguido del offset, se mostrará la descripción de dicha propiedad.

```
LA CRISIS DE BRONCOSPASMO SEVERO CON INS.RESPIRATORIA AGUDA  
INDUCIDA POR QUINOLONAS...
```

*Ilustración 15: BRAT ejemplo fichero formato .txt*

```
T1 Grp_Enfermedad 34 56 INS.RESPIRATORIA AGUDA  
T2 Grp_Medicamento 70 79 QUINOLONAS  
T3 Grp_Enfermedad:T1 Causada_por: T2  
E1 Grp_Enfermedad:T18 Tratamiento:T13  
#1 AnnotatorNotes T9 sigla: diabetes mellitus (contexto), densitometría, dermatomiositis, duramadre
```

*Ilustración 16: BRAT ejemplo fichero formato .ann*

## 4.2. Casos de uso

A continuación se muestran la jerarquía de actores y los casos de uso del sistema.

### 4.2.1. Jerarquía de actores

La aplicación únicamente tendrá un actor, ya que la aplicación está diseñada para cualquier persona que quiera utilizar el sistema.

- **Usuario:** Este rol define al usuario que tendrá la posibilidad de obtener estadísticas de los informes médicos del sistema, así como cargar nuevos archivos.

## 4.2.2. Diagrama de casos de uso

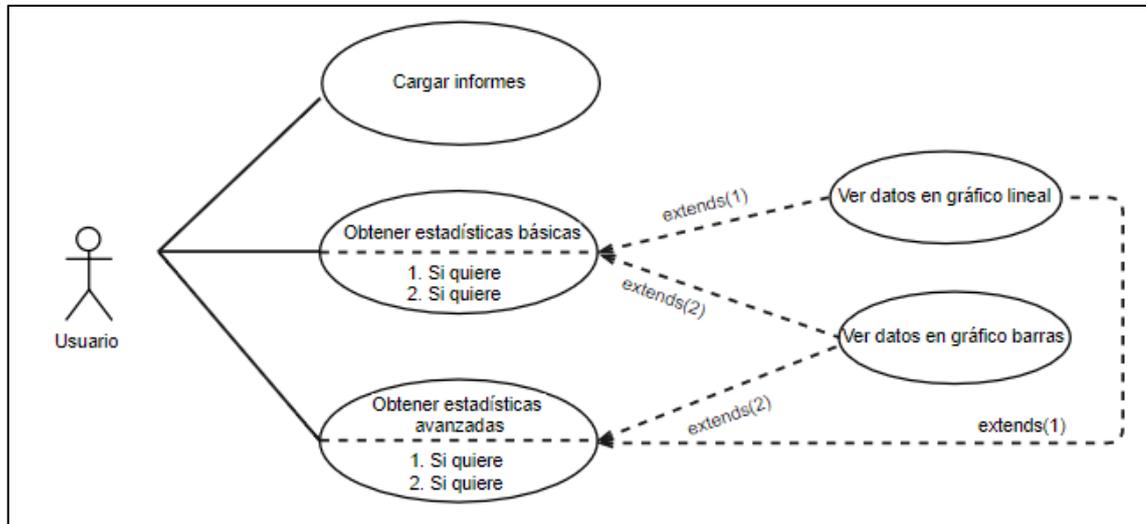


Ilustración 17: Diagrama de casos de uso

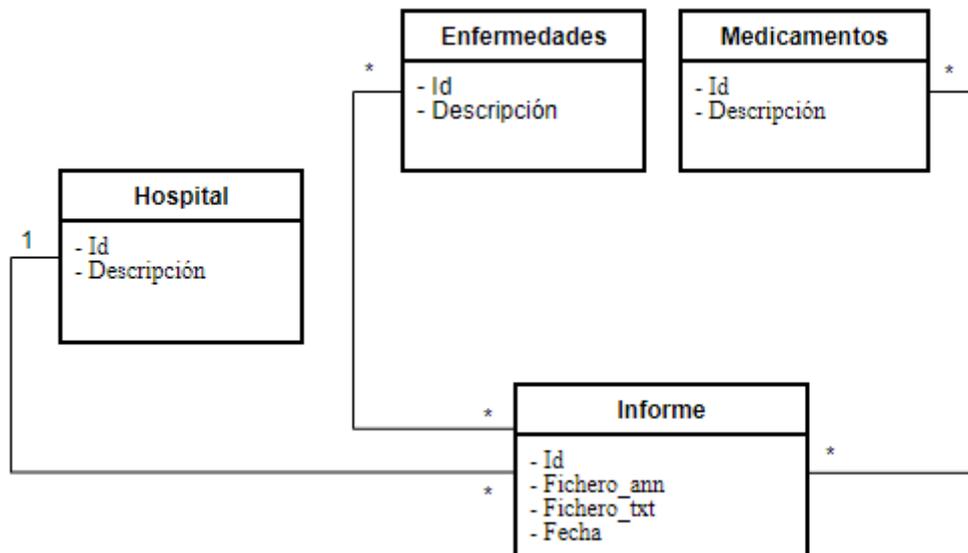
El diagrama consta de 3 casos de uso referentes a las 3 funciones principales del sistema, carga de informes, y obtención de estadísticas tanto básicas como avanzadas. Asimismo, se le añaden 2 subcasos que hacen referencia a las funciones que el usuario podrá realizar una vez obtenidas las estadísticas deseadas.

Cada uno de los casos de uso, así como los subcasos, se detallarán en profundidad en el *Anexo I Casos de uso*.

## 4.3. Modelo de dominio

En el primer y segundo prototipo del proyecto, será necesario tratar con los mismos datos, por lo tanto se utilizará el mismo modelo. Éste, tendrá que cumplir las siguientes condiciones para adecuarse a las necesidades de la aplicación:

- Los usuarios tratarán con datos de informes médicos de los cuales se almacenará su contenido tanto en formato *.txt*, como el formato *.ann*, y su fecha.
- En los informes aparecerán numerosas enfermedades y medicamentos.
- Se podrán almacenar tantos informes médicos como quiera el usuario, pero cada uno de ellos únicamente pertenecerá a un hospital.



*Ilustración 18: Modelo de dominio prototipo 1*

El modelo del tercer prototipo, únicamente deberá ampliar el modelo anterior con una nueva condición.

- De los informes médicos será necesario guardar también, las enfermedades que han sido causadas por medicamentos, así como la descripción de estos.

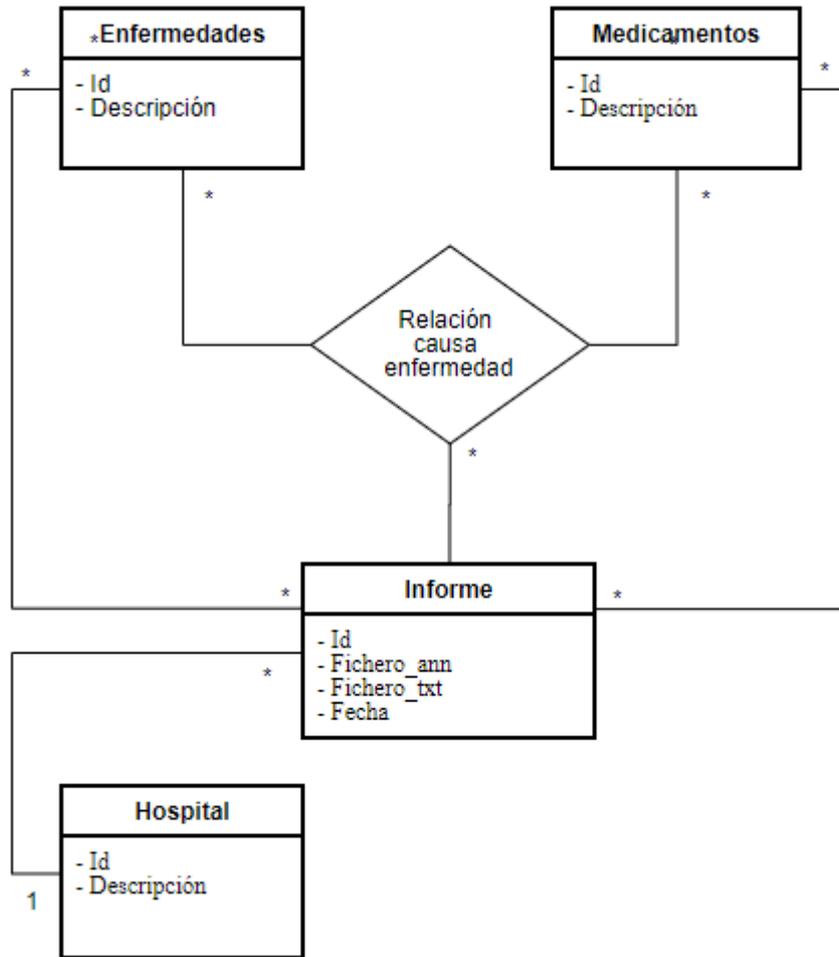


Ilustración 19: Modelo de domino prototipos II y III

## 5. ANÁLISIS Y DISEÑO

### 5.1. Arquitectura

Para el desarrollo se utilizará una arquitectura MVC, con el que se obtendrán tres estructuras independientes.

Por una parte el Modelo, que gestiona el acceso a la información almacenada en la base de datos, por otra, la Vista, que permitirá la interacción del usuario, y por último el Controlador, que será quien una las dos estructuras mencionadas gestionando las peticiones entre sí.

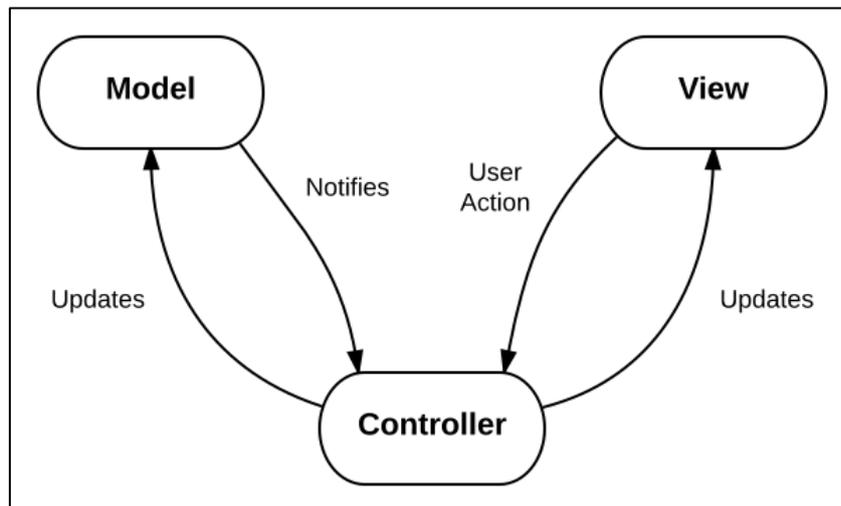


Ilustración 20: Arquitectura MVC

#### 5.1.1. Introducción a Spring Framework

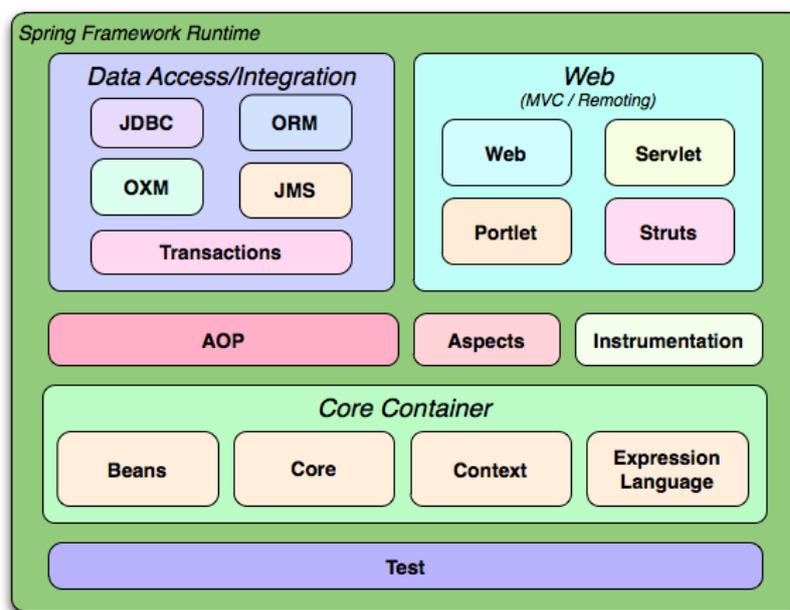
Spring permite el desarrollo de aplicaciones de manera eficaz y rápida, evitando tareas repetitivas y ahorrando tiempo y líneas de código.

Una de sus principales características es la inyección de dependencias.

Es un tipo de inversión de control, es decir, Spring toma el control del manejo de las propiedades de un objeto que son inyectadas a través de un constructor, o un

servicio, creando de esta manera un comportamiento más adecuado de la aplicación. Con esta característica, los objetos de Spring (beans), no serán responsables de encontrar o crear el resto de objetos que necesiten para llevar a cabo distintas funcionalidades, sino que, se les asignarán referencias a los objetos con los que tienen que colaborar.

Es una funcionalidad orientada a la creación de grandes aplicaciones, más complejas y con funcionamientos más encapsulados. Spring Framework está compuesto por varios módulos, que se muestran a continuación.[18]



*Ilustración 21: Módulos de Spring*

La modularidad de Spring permite utilizar solo las partes del framework que sean necesarias. Y que se ajusten mejor a la aplicación. Para el desarrollo de la aplicación, se han utilizado principalmente el módulo Web y el Data Access.

**Web:** Es el módulo que permite implementar el patrón Modelo-Vista-Controlador de manera sencilla y separando de manera limpia la lógica de negocio del acceso a los datos así como de la presentación de los mismos. Su funcionamiento es el siguiente:

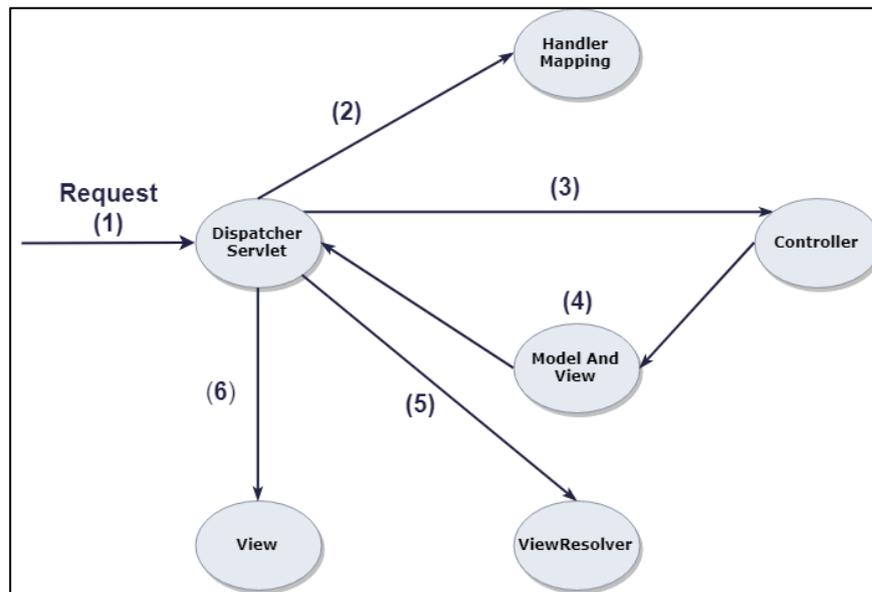


Ilustración 22: Spring MVC

En cada petición se realizarán las siguientes funciones:

- (1) Todas las peticiones HTTP llegarán al DispatcherServlet, que no es más que un Servlet cuya implementación es propia de Spring.
- (2) Mediante la fase de HandlerMapping, se encontrará el controlador asociado a la url de la petición. Éste será quien albergue la lógica de negocio de la aplicación.
- (3) El controlador retornará un objeto de tipo ModelAndView. Este objeto, será la unión del nombre de la vista de la cual se quiere mostrar la información (View) y los datos que se devolverán (Model).
- (4) El objeto ModelAndView se enviará de vuelta al DispatcherServlet, que será quien lo asocie a una vista concreta.
- (5) Finalmente el DispatcherServlet pasará el objeto Model a la vista con la que se ha asociado.

**Data Access:** Está formado por distintos submódulos, que permiten simplificar el acceso a los datos, y dar soporte para utilizar bases de datos relacionales (JDBC) entre otras funcionalidades.



## 5.2. Diagramas de secuencia

Con el fin de entender mejor el funcionamiento de la aplicación, se detallarán en el *Anexo II Diagramas de Secuencia*, los diagramas del sistema, correspondientes a las principales funcionalidades, es decir, la carga de informes, así como la obtención de estadísticas tanto del filtro básico como del avanzado.

## 5.3. Diagrama de clases

En este apartado se mostrará el diagrama de clases de la aplicación. Únicamente se mostrarán las clases referentes a la lógica de negocio, es decir, las clases que implementan las interfaces gráficas como los archivos *.js* o *.jsp*, se obviarán.

Los diagramas se dividirán en 4 grupos.

- **Model:** Referente a los beans creados para almacenar los datos.
- **Controllers:** Referente a la lógica de negocio que une las interfaces con el modelo de datos.
- **Service:** Referente a las clases que unirán la funcionalidad de los controladores con las funcionalidades que realizarán las llamadas a la base de datos.
- **Dao:** Referente a las clases que realizan las llamadas correspondientes a la base de datos.

### 5.3.1. Model

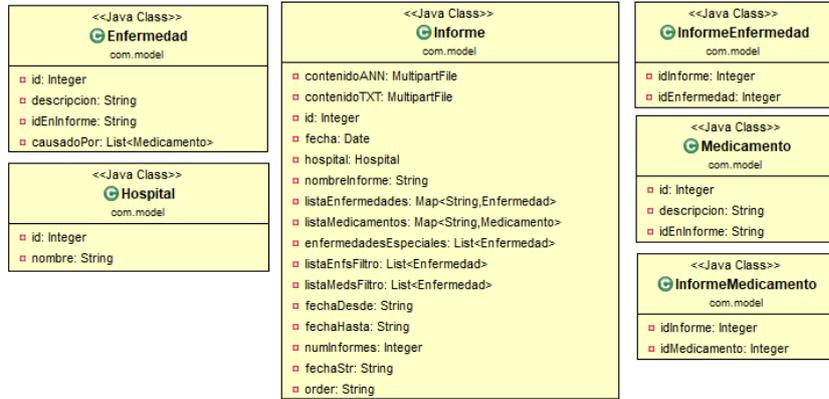


Ilustración 23: Diagrama de clases (Model)

### 5.3.2. Controller

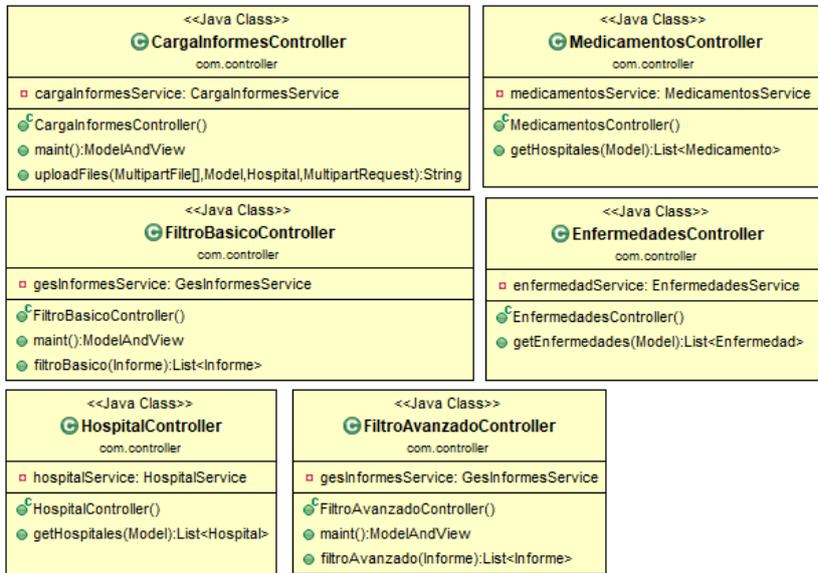


Ilustración 24: Diagrama de clases (Controller)

### 5.3.3. Services

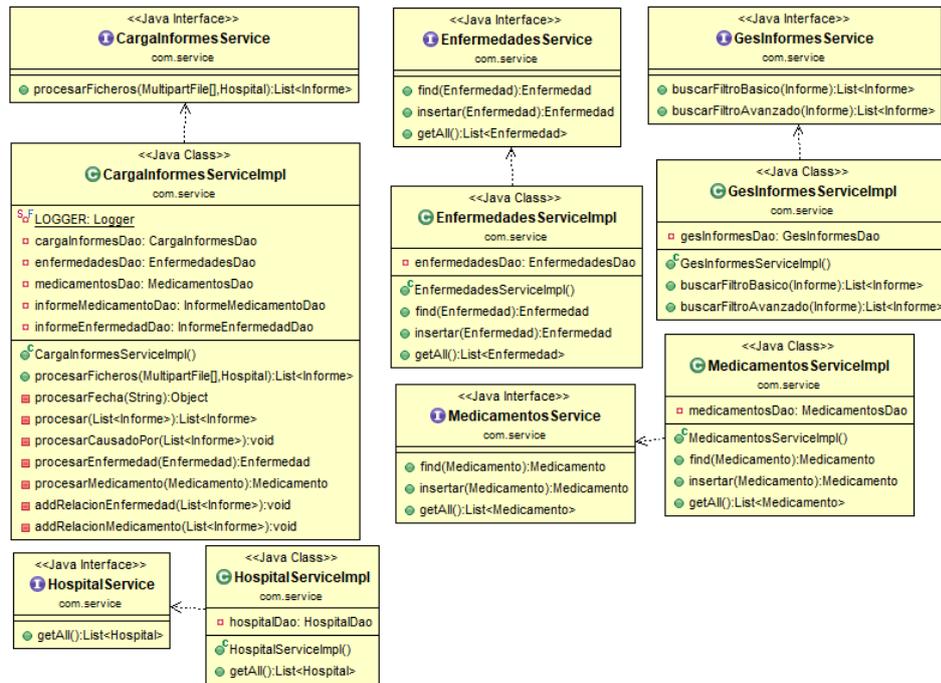
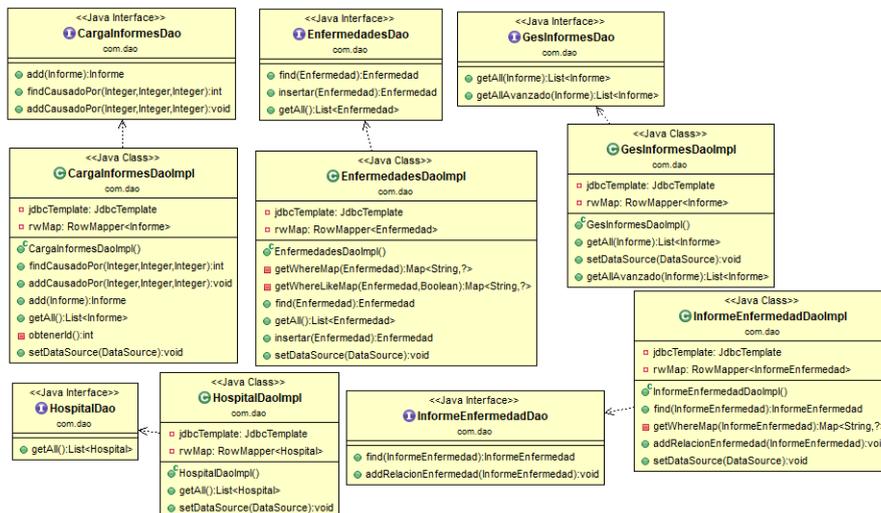


Ilustración 25: Diagrama de clases (Service)

### 5.3.4. Dao



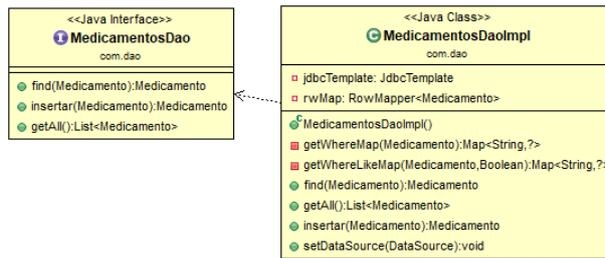


Ilustración 26: Diagrama de clases (Dao)

## 5.4. Base de datos

Teniendo en cuenta el análisis realizado en los modelos de dominio indicados anteriormente, se ha definido el siguiente modelo de base de datos.

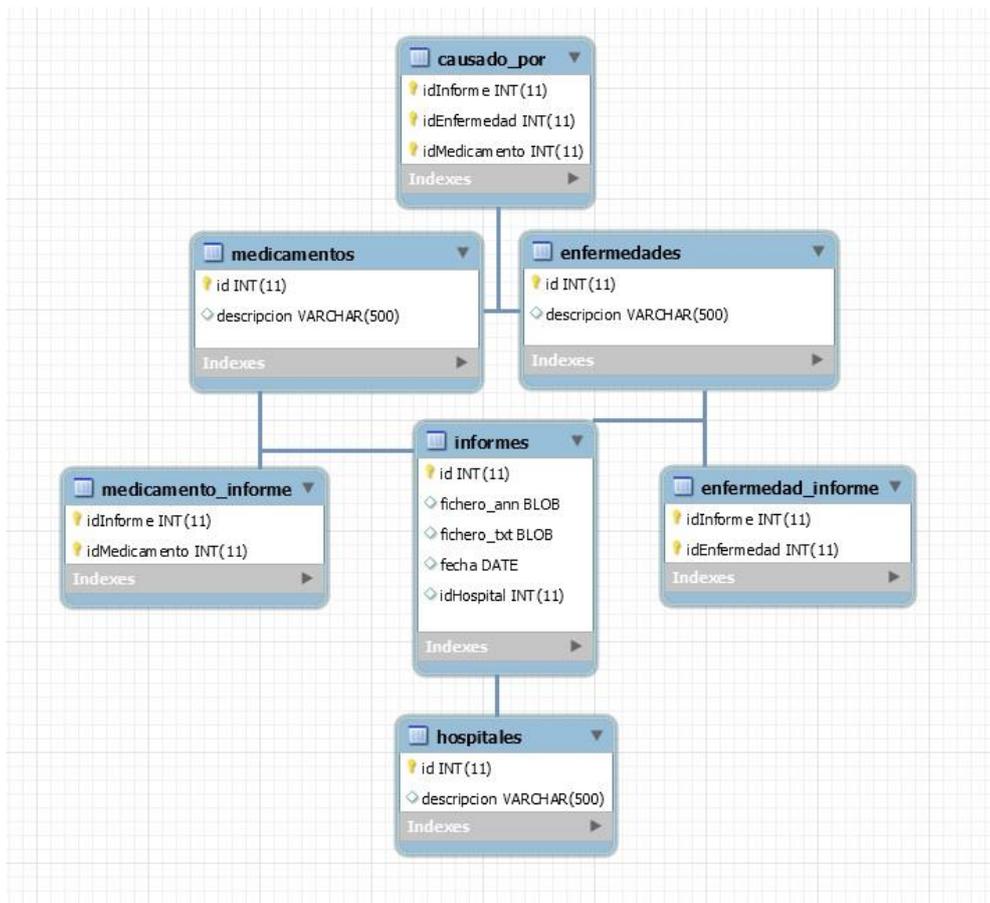


Ilustración 27: Diseño de base de datos

- **Hospitales, enfermedades, medicamentos:** Se almacenará la descripción del hospital y su clave primaria será un id autoincremental ya que podría haber varios con el mismo nombre.
- **Informes:** Se identificarán mediante un id autoincremental, y se almacenará el id del hospital al que pertenecen, así como la fecha del informe. También se guardarán los ficheros *.txt* y *.ann* en la base de datos con formato BLOB, por si se necesitaría para futuras funcionalidades.
- **Enfermedad\_informe, Medicamento\_informe:** Se almacenará qué medicamentos o qué enfermedades aparecen en cada informe.
- **Causado\_por:** Esta tabla almacenará los id de los medicamentos causantes de qué enfermedades, así como el informe en el que aparecen.

## 6. DESARROLLO

A continuación se detallará el desarrollo de la aplicación, explicando cada uno de los componentes que se han utilizado, así como herramientas adicionales que se han decidido incluir para la implementación de algunas funcionalidades.

### 6.1. Herramientas adicionales

#### 6.1.1. JQuery

JQuery es una librería de JavaScript de código abierto. Esta herramienta ofrece una serie de funcionalidades que simplifican la interacción con los documentos HTML, y la forma de manejar los eventos de la aplicación que requerirían de bastantes líneas de código de no ser por el uso de la librería. Además de estas características, es compatible con la mayoría de los navegadores y es la biblioteca más utilizada actualmente. [19]

Adicionalmente, se han añadido bibliotecas para el desarrollo de algunas de las funcionalidades:

- **jQueryUI:** Es una librería que ofrece múltiples componentes de interacción, efectos visuales y widgets que facilitan el desarrollo web. [20]
- **jQuery BlockUI:** Este complemento permite simular el comportamiento síncrono de llamadas Ajax. Cuando se activa, evita la interacción del usuario con la página hasta que sea desactivado. Esto se utilizará, por ejemplo, para mostrar un mensaje al usuario mientras se realiza el procesamiento de los ficheros. [21]

## 6.1.2. Bootstrap

Para el diseño de la aplicación, se ha utilizado Bootstrap. Es un conjunto de herramientas de código abierto, que ofrece la posibilidad de desarrollar con HTML, CSS Y JS, simplificando bastante el proceso de maquetación, gracias a algunas plantillas de diseño que ofrece, por ejemplo para el diseño de botones, formularios, menús de navegación etc. Es una herramienta sencilla de utilizar y que ofrece muchas posibilidades, además de eso, es compatible con la mayoría de navegadores web. [22]

## 6.2. Estructura del proyecto

En la imagen inferior se puede ver la estructura completa del proyecto.

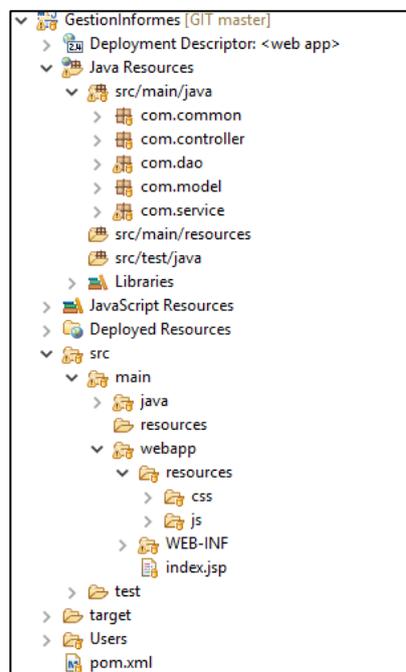


Ilustración 28: Estructura del proyecto

Por una parte, en `src/main/java`, estarán todas las clases relativas a la lógica de negocio de la aplicación.

Bajo la carpeta de `src/main/webapp/resources` se almacenará todo lo relativo a archivos estáticos, es decir, los ficheros `.js` y lo referente a todos los estilos de la aplicación.

Los archivos correspondientes a las interfaces y los archivos necesarios para la configuración de Spring permanecerán bajo la ruta `src/main/webapp/web-inf`.

### 6.3. Configuración de Spring

Para empezar a configurar Spring es necesario comenzar por el `DispatcherServlet`, que como se ha mencionado anteriormente será quien reciba la petición que se enviará al controlador. Para ello se configurará en el archivo `web.xml` de la aplicación, ya que debe cargarse al inicio de ésta.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">

  <servlet>
    <servlet-name>spring</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>

  <servlet-mapping>
    <servlet-name>spring</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>

</web-app>
```

*Ilustración 29: Configuración DispatcherServlet*

También será necesario configurar un archivo de contexto de la aplicación como el siguiente:

```

<context:component-scan
    base-package="com.controller,com.dao,com.service" />

<bean id="viewResolver"
    class="org.springframework.web.servlet.view.UrlBasedViewResolver">
    <property name="viewClass">
        <value>
            org.springframework.web.servlet.view.tiles3.TilesView
        </value>
    </property>
</bean>
<bean id="dataSource"
    class="org.springframework.jdbc.datasource.DriverManagerDataSource">

    <property name="driverClassName" value="com.mysql.jdbc.Driver" />
    <property name="url" value="jdbc:mysql://localhost:3306/proyecto?useUnicode=true&characterEncoding=UTF-8" />
    <property name="username" value="root" />
    <property name="password" value="root" />
</bean>
<bean id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
    <property name="maxUploadSize" value="20848820" />
</bean>
<bean id="tilesConfigurer"
    class="org.springframework.web.servlet.view.tiles3.TilesConfigurer">
    <property name="definitions">
        <list>
            <value>/WEB-INF/tiles.xml</value>
        </list>
    </property>
</bean>
<mvc:resources mapping="/resources/**" location="/resources/" />

<mvc:annotation-driven />
    
```

*Ilustración 30: Contexto de Spring*

En él se configurarán varias partes de la aplicación.

- **Etiqueta <context:component-scan>**

Se definirá el paquete donde se incluirán los controladores y los servicios para poder acceder a ellos mediante anotaciones.
- **Etiqueta <bean id = "dataSource">**

En esta parte del archivo se configurará lo referente a Spring JDBC. Mediante la etiqueta <property> se definirán cada una de las propiedades para poder acceder a la base de datos, esto es, el driver, la url, el nombre de usuario y el password.
- **Etiqueta <mvc:resources>**

Se indicará la ruta del contenido estático de la aplicación

- **Etiqueta <bean id = "tilesConfigurer">**

Se creará un archivo tiles.xml que será el encargado de definir cada una de las vistas de la aplicación a las que se accederá desde el controlador de la siguiente forma.

```
<definition name="gesInformesAvanzado" extends="base.definition">
  <put-attribute name="title" value="Informacion estadística" />
  <put-attribute name="body" value="/WEB-INF/jsp/gesInformesAvanzado/gesInformesAvanzado.jsp" />
  <put-attribute name="includes" value="/WEB-INF/jsp/gesInformesAvanzado/gesInformesAvanzado-includes.jsp" />
</definition>
</tiles-definitions>
```

*Ilustración 31: Configuración Tiles.xml*

En una vista que se utilizará como plantilla, se introducirán las propiedades definidas en este fichero.

- **Title:** Almacenará el título de la vista correspondiente.
- **Body:** Tendrá todo el contenido para crear la vista que se quiere mostrar al usuario.
- **Includes:** En él se definirán los archivos .js necesarios para el funcionamiento de la vista.

## 6.4. Desarrollo funciones principales

A continuación se explicarán las funciones principales de la aplicación.

### 6.4.1. Carga de informes

Esta parte procesará los informes médicos que se introduzcan y almacenará su información en la base de datos. Para ello, será necesario tener archivos con

extensión .ann procesados previamente con la herramienta BRAT explicada anteriormente.

A la hora de cargar los informes, se deberán subir dos archivos. El informe original en formato .txt, y su correspondiente fichero .ann. Ambos deberán llamarse igual, y tendrán el mismo formato de nombre, el identificador, seguido de la fecha de informe, y todo ello separado por guiones bajos. En la imagen inferior se muestra un ejemplo.

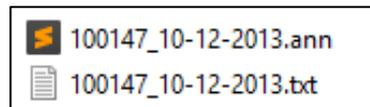


Ilustración 32: Formato nombre de fichero

A continuación se muestra un ejemplo del contenido que los ficheros .ann deberán tener.

```
1 T6 Grp_Enfermedad 304 315 anovulación
2 T7 Grp_Enfermedad 317 342 hiperandrogenismo clínico
3 T8 Grp_Enfermedad 361 380 ovario poliquístico
4 T9 Grp_Enfermedad 393 401 síndrome
5 T10 Grp_Enfermedad 465 489 diabetes mellitus tipo 2
6 T11 Grp_Enfermedad 631 650 síndrome metabólico
7 T14 Grp_Enfermedad 710 729 síndrome metabólico
8 T15 Grp_Enfermedad 839 858 síndrome metabólico
9 T16 Grp_Enfermedad 912 930 obesidad abdominal
10 T18 Grp_Enfermedad 1014 1022 obesidad
11 T19 Grp_Enfermedad 1150 1160 hirsutismo
```

Ilustración 33: Contenido fichero ANN

Como se puede observar, las enfermedades y medicamentos estarán definidos con distintas etiquetas.

El identificador T seguido de un número junto con la etiqueta *Grp\_Enfermedad* define la posición en la que se encuentra del fichero .txt correspondiente y la descripción de la enfermedad.

Los medicamentos se definirán de la misma forma, cambiando la etiqueta por *Grp\_Medicamento*.

Otro aspecto importante para la aplicación son las enfermedades causadas por distintos medicamentos. Estas se definirán de la siguiente forma:

```
E3 Grp_Enfermedad:T3 Causada_por:E2 Causada_por2:E1
```

*Ilustración 34: Línea especial de fichero*

Como se ve en la ilustración anterior, el identificador de la línea es distinto a los anteriores y ahora es definido con una *E*. Esto nos permitirá identificar líneas especiales que no sean referentes únicamente a enfermedades o medicamentos a la hora de procesar el fichero. Mediante la etiqueta *Causado\_por* se definirán los medicamentos causantes de la enfermedad, pudiendo hacer referencia como en este caso a otra línea del fichero que a su vez, hará referencia a la línea que contiene la descripción del medicamento.

```
16 E1 Grp_Enfermedad:T1
17 T2 Grp_Enfermedad 37 40 SOP
18 E2 Grp_Enfermedad:T2
19 T3 Grp_Enfermedad 140 168 infertilidad por anovulación
```

*Ilustración 35: Referencia de línea especial fichero ANN*

Para la subida de ficheros, se utilizará el controlador *CargaInformesController*. Esta clase, así como el resto de controladores, estarán dentro del paquete que se ha definido en el fichero xml del contexto de Spring como los controladores de la aplicación, explicado anteriormente. Todos los controladores deberán definirse de la misma forma, tal y como se muestra en la *Ilustración 36*.

```
@Controller
@RequestMapping("/cargaInformes")
public class CargaInformesController {
    @Autowired()
    private CargaInformesService cargaInformesService;
```

Ilustración 36: Definición controlador

La anotación `@Controller` indica que la clase es un bean del tipo controlador y con la etiqueta `@RequestMapping` se le asociará la url a la cual llegarán las llamadas HTTP. Para las consultas que se realizarán a la base de datos será necesario el uso de objetos de negocio, por lo que se instanciarán como beans de Spring mediante la etiqueta `@Autowired`.

Una vez definido el controlador, se realizará la siguiente llamada.

```
@RequestMapping(value = "/uploadFiles", method = RequestMethod.POST)
public String uploadFiles(@RequestParam("ficheros") MultipartFile[] files
    , Model model
    , Hospital hospital,
    MultipartRequest request) {
```

Ilustración 37: Carga Informes

Este método, recibirá los archivos que el usuario haya introducido y el hospital al que pertenecen, y realizará una llamada a la clase `CargaInformesService` que será quien procese el fichero.

El método `procesarFicheros()` de esta clase será quien examine cada línea del informe y realice las llamadas correspondientes a la base de datos para insertar la información de los medicamentos y enfermedades, así como sus relaciones. Este funcionamiento se encuentra detallado en el correspondiente diagrama de secuencia del *Anexo II: Diagramas de secuencia*.

Una vez procesados los informes, si estos fueran erróneos o los nombres no fueran los adecuados, se le informaría al usuario, y se mostraría en una tabla los archivos erróneos que no se han podido guardar.

The screenshot shows a web interface with a light blue background. At the top, there is a section titled "Informes". Below it, there is a text input field with the placeholder "Elegir archivos" and the text "Ningún archivo seleccionado". Below that, there is a dropdown menu labeled "Hospital" with the selected option "Basurto". At the bottom right of this section, there is a button labeled "Cargar ficheros". Below the form, there is a yellow warning banner with the text: "Cuidado! Los siguientes informes no han podido ser cargados. Revise que el nombre del archivo sea correcto, que se ha adjuntado su correspondiente fichero.txt y cárguelos nuevamente." Below the banner, there is a table with the following data:

#	Informe
1	100147.ann
2	100159.ann
3	103581.ann

*Ilustración 38: Carga informes incorrecta*

En caso contrario, se le indica al usuario que los archivos y la información incluida en ellos se ha cargado en el sistema correctamente.

The screenshot shows a web interface with a light blue background. At the top, there is a green success banner with the text: "Correcto! Todos los informes se han cargado correctamente." Below the banner, there is a section titled "Informes". Below it, there is a text input field with the placeholder "Elegir archivos" and the text "Ningún archivo seleccionado". Below that, there is a dropdown menu labeled "Hospital" with the selected option "Basurto". At the bottom right of this section, there is a button labeled "Cargar ficheros".

*Ilustración 39: Carga informes correcta*

## 6.4.2. Búsqueda de estadísticas

Estas pantallas contienen varios filtros con los que el usuario puede interactuar.

- Input enfermedad
- Input medicamento
- Combo hospital
- Input fecha desde

- Input fecha hasta

Para que el usuario pueda realizar búsquedas de medicamentos y enfermedades de forma más sencilla, los input correspondientes serán implementados mediante el componente *Autocomplete* de jQueryUI, que irá mostrando las descripciones a medida que el usuario introduzca los caracteres.

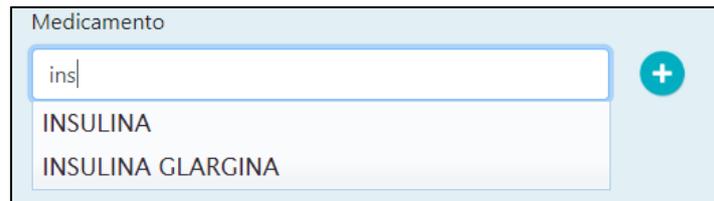


Ilustración 40: Componente Autocomplete jQueryUI

Además se utilizará también el componente *Datepicker* que permitirá al usuario elegir una fecha de un calendario desplegable, en lugar de tener que introducirla a mano con el formato indicado.

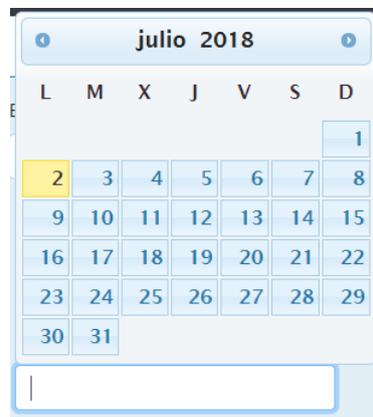


Ilustración 41: Componente Datepicker jQueryUI

Tanto para la búsqueda básica de estadísticas, como para la avanzada, se accederá con llamadas similares a distintos controladores para separar las consultas que ejecutarán y tener acceso a ellas más fácilmente. Las llamadas se muestran a continuación.

```

@RequestMapping(value = "/filtroBasico", method = RequestMethod.POST)
public @ResponseBody() List<Informe> filtroBasico(@RequestBody() Informe informe)
  
```

*Ilustración 42: Llamada filtro básico*

```

@RequestMapping(value = "/filtroAvanzado", method = RequestMethod.POST)
public @ResponseBody() List<Informe> filtroAvanzado(@RequestBody() Informe informe)
  
```

*Ilustración 43: Llamada filtro avanzado*

Por parámetro recibirán un Informe, que tendrá almacenados todos los datos del filtro gracias a las dependencias explicadas anteriormente del framework Spring, y con los cuales se construirá la consulta de base de datos correspondiente. Se hará de la siguiente forma.

Se definirán las partes SELECT y FROM de cada consulta y se añadirá la parte WHERE con el filtro dinámicamente añadiendo las condiciones correspondientes a los campos que el usuario haya rellenado. Un ejemplo será la obtención de las fechas de los informes.

```

if (informe.getFechaDesde() != null && !"".equals(informe.getFechaDesde())) {
    where.append("AND T1.FECHA >= ?");
    params.add(Funciones.tratarFecha(informe.getFechaDesde()));
}
if (informe.getFechaHasta() != null && !"".equals(informe.getFechaHasta())) {
    where.append(" AND T1.FECHA <= ?");
    params.add(Funciones.tratarFecha(informe.getFechaHasta()));
}
  
```

*Ilustración 44: Consulta sql dinámica*

El funcionamiento de las consultas, se encuentra detallado en el correspondiente diagrama de secuencia del *Anexo II: Diagramas de secuencia*.

Una vez obtenida la información de la base de datos, es decir la fecha y el número de informes que cumplen las condiciones del filtro, se le enviará de nuevo al controlador, ya que será éste quien envíe la información a la vista para mostrarla.

En la vista, al recibir los datos, se creará un gráfico con la librería Chart.js, por defecto será lineal pero será posible cambiarlo a un gráfico de barras mediante un

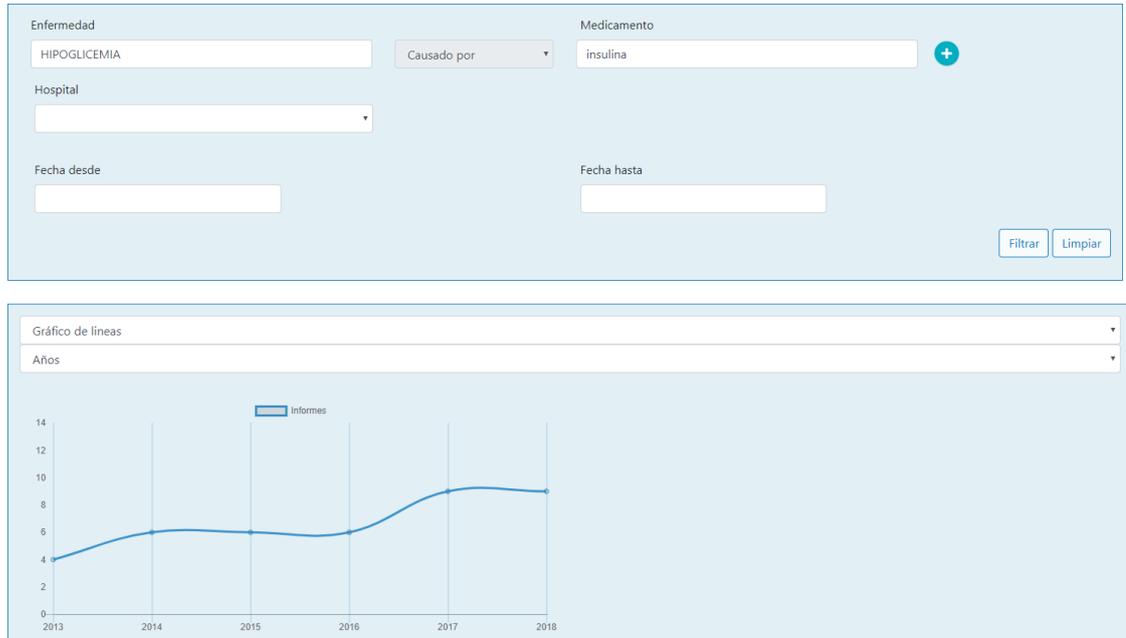
combo definido en la pantalla. Además permitirá que el usuario elija el formato de fechas que desea ver, ya sea por años, por meses y años, así como por fechas concretas de los informes. En la *Ilustración 45* se puede ver un ejemplo de su creación.

```
grafico = new Chart(ctx, {
  type: 'line',
  data: {
    labels: fechas,
    datasets: [{
      data: numInformes,
      label: "Informes",
      borderColor: "#3e95cd",
      fill: false
    }
  ]
},
options: {
  title: {
    display: true,
    text: ''
  },
},
scales: {
  yAxes: [{
    gridLines: {
      color: "rgba(0, 0, 0, 0)",
    },
    ticks: {
      beginAtZero: true,
      max: Math.max.apply(Math, numInformes)+ 5,
      userCallback: function(label, index, labels) {
        if (Math.floor(label) === label) {
          return label;
        }
      }
    },
  }
},
}],
});
```

*Ilustración 45: Ejemplo Chart.js*

En la *Ilustración 46*, se puede observar un ejemplo de los resultados obtenidos para un filtro avanzado, en el cual se deseaba obtener las estadísticas de los casos en los que la insulina hubiese causado hipoglucemia.

El filtro introducido por el usuario sería el siguiente.



*Ilustración 46: Ejemplo visualización gráfico avanzado*

Con este ejemplo, se puede ver la utilidad de los gráficos que se le muestran al usuario, ya que en este caso, se podría predecir solo con ver la imagen, que la insulina podría ser uno de los motivos causantes de enfermedades como la hipoglucemia, y no sería necesario mirar todos los informes de cada uno de los pacientes que padecen la enfermedad.



## 7. VERIFICACIÓN Y EVALUACIÓN

Para un buen funcionamiento de la aplicación, es necesario crear un plan de pruebas adecuado, teniendo en cuenta todas y cada una de las posibles funciones que esta pueda tener. Las pruebas es un factor muy importante debido a que la aplicación puede ser utilizada por usuarios sin conocimientos previos informáticos. Es por ello, que es necesario contemplar cada función que se pudiera hacer sobre el sistema, ya que en este caso, los datos con los que se trata tienen riesgo debido a que son informes médicos de pacientes, y es necesario tener en cuenta la seguridad a la hora de realizar cualquier acción con ellos. En este apartado se describirán las pruebas realizadas a la aplicación.

## 7.1 Primer prototipo

Id	Descripción	Resultado Esperado	Resultado Obtenido
1.1	Cargar ficheros .ann o .txt sin fecha	Aparecen como ficheros incorrectos en la tabla inferior.	✓
1.2	Cargar ficheros .ann y .txt de la misma fecha, y en distinto formato	Aparece error validando la extensión de los ficheros.	✓
1.3	Cargar ficheros .ann y .txt con mismo nombre	Los ficheros se cargan correctamente.	✓
1.4	Cargar fichero .ann y no su correspondiente fichero .txt	Aparece como fichero erróneo en la tabla.	✓
1.5	Cargar fichero .txt y no su correspondiente fichero .ann	Aparece como fichero erróneo en la tabla.	✗
1.6	Cargar fichero .txt y su correspondiente .ann pero de contenido vacío	No se insertan nuevos datos en la BD respecto a enfermedades o medicamentos.	✓
1.7	Cargar ficheros que no sean del tipo .ann o .txt	Aparece error validando la extensión de los ficheros.	✓
1.8	Cargar información extraída de los informes en la base de datos	Se carga la información en la BD	✗

Tabla 25: Pruebas prototipo I

## 7.2 Segundo prototipo

Id	Descripción	Resultado Esperado	Resultado Obtenido
2.1	Añadir medicamento	Se añade un nuevo criterio de búsqueda de medicamento.	✓
2.2	Borrar medicamento	Se borra el criterio actual de búsqueda de medicamento.	✓
2.3	Añadir enfermedad	Se añade un nuevo criterio de búsqueda de enfermedad.	✓
2.4	Borrar enfermedad	Se borra el criterio actual de búsqueda de enfermedad.	✓
2.5	Formato de fechas incorrecto	Se valida que las fechas sean correctas y no lanza la búsqueda.	✓
2.6	Filtrar sin introducir ningún filtro en el formulario	Se lanza la búsqueda sin filtrar por ningún criterio y se muestra el gráfico.	✓
2.7	Mostrar gráfico al filtrar	Si los criterios de filtrado son válidos, se lanza la búsqueda y se muestra el gráfico con los datos.	✓
2.8	Cambiar tipo de gráfico	Al elegir el tipo de gráfico, desaparece el actual y se muestra el elegido.	✗
2.9	Datos correctos en el gráfico	Se comprueba que los datos que se recogen de la base de datos son correctos.	✗
2.10	Limpiar formulario	Se limpian todos los criterios de búsqueda introducidos.	✓

Tabla 26: Pruebas prototipo II

### 7.3 Tercer prototipo

Id	Descripción	Resultado Esperado	Resultado Obtenido
3.1	Añadir fila de enfermedad causada por medicamento	Se añade una nueva fila en el filtro	✓
3.2	Borrar fila de enfermedad causada por medicamento	Se borra la fila actual del filtro	✓
3.3	Mostrar grafico al filtrar	Si los criterios de filtrado son válidos, se lanza la búsqueda y se muestra el gráfico con los datos.	✓
3.4	Filtrar sin introducir ningún filtro en el formulario	Se lanza la búsqueda sin filtrar por ningún criterio y se muestra el gráfico.	✓
3.5	Introducir formato de fechas incorrecto y filtrar	Se valida que las fechas sean correctas y no lanza la búsqueda.	✓
3.6	Limpiar filtro	Se limpian todos los criterios de búsqueda introducidos.	✓
3.7	Datos correctos en el grafico	Se comprueba que los datos que se recogen de la base de datos son correctos.	✓
3.8	Cambiar tipo de gráfico	Al elegir el tipo de gráfico, desaparece el actual y se muestra el elegido.	✓

Tabla 27: Pruebas prototipo III

## 7.4 Errores y soluciones

Id	Error	Solución
1.5	Solo se comprobaba si eran erróneos en el caso de que hubiera un fichero ann.	Se añadió una comprobación anterior para validar en el caso de que solo se hubiesen cargado ficheros .txt
1.8	Al procesar el fichero, y guardar la relación de una enfermedad causada por un medicamento, no se estaba teniendo en cuenta que en el informe había enfermedades causadas por enfermedades, y se estaban procesando de la misma manera.	Se añadió una comprobación para evitar este caso y que solo procesara las enfermedades causadas por medicamentos.
2.8	Al pasar el ratón por el gráfico aparecían gráficos con datos de búsquedas anteriores.	Para evitar este caso, se llama a la función <i>update</i> que incluye la librería Chart.js que actualiza el gráfico con los nuevos datos.
2.9	La query que se lanzaba al filtrar era incorrecta, y salían datos repetidos.	Se arregló para que se agrupara por fecha y por el id del informe.

Tabla 28: Errores y soluciones



## 8. CONCLUSIONES Y TRABAJO FUTURO

Valorando el trabajo una vez finalizado el proyecto, la parte más complicada ha sido el hecho de realizarlo de forma individual, ya que los proyectos más extensos o parecidos a este realizados con anterioridad en la carrera, se han hecho siempre en grupo. Esto ha supuesto tener que aprender a definir mejor las tareas que se debían realizar para poder desarrollarlo, así como a tener una mayor organización en el tiempo de trabajo para poder llegar a cumplir los objetivos propuestos, ya que por motivos de trabajo las horas diarias para la dedicación del desarrollo eran limitadas.

### 8.1. Valoración de objetivos del proyecto

Para valorar mejor el trabajo realizado, es importante recordar los objetivos que se definieron al inicio del proyecto, y valorar su cumplimiento.

- **Automatizar el proceso de obtención de datos de informes médicos:** Este objetivo se ha cumplido, ya que mediante la pantalla de “Carga de Informes” se procesan los informes de forma automática obteniendo y guardando toda la información de éstos en la base de datos.
- **Obtención de estadísticas médicas:** Este objetivo también ha sido cumplido. Con los informes procesados guardados anteriormente, y en base a la búsqueda que el usuario desee se le muestra las estadísticas temporales correspondientes.
- **Aprendizaje personal:** Este objetivo se puede dar por cumplido. Se definió para comprobar si la adaptación al uso de herramientas no utilizadas anteriormente podría realizarse con éxito. Aunque en algún momento el uso de las nuevas tecnologías haya supuesto un bloqueo en el desarrollo, finalmente se ha conseguido avanzar, por lo que ha supuesto obtener grandes conocimientos y satisfacción personal.

## 8.2. Valoración de planificación del proyecto

A continuación se reflexionará sobre el cumplimiento de la planificación definida al comienzo del proyecto.

La planificación inicial estaba estimada en un total de 430h, sin embargo finalmente, se ha llegado a superar con un total de 460h.

En un principio solo se pudieron dedicar 3 horas diarias para el desarrollo por temas laborales ajenos al proyecto. Pasados dos meses, esta dedicación se pudo ampliar. Sin embargo, no fue suficiente ya que varias de las tareas se estimaron en un tiempo menor del que finalmente se necesitó.

Por una parte, la tarea de la configuración del entorno bloqueó el desarrollo inicial debido al desconocimiento de las nuevas tecnologías con las que se iba a trabajar, como Spring Framework, ya que nunca habían sido utilizadas en proyectos realizados anteriormente en el grado.

Por otra parte, documentar el proyecto completo llevo más tiempo del estimado inicialmente, ya que nunca antes se habían realizado estimaciones completas de proyectos de este tipo.

Estos problemas de estimación sirven de aprendizaje y serán tenidos en cuenta para futuros desarrollos.

## 8.3. Trabajo futuro

Con vistas de futuro, se podrían plantear nuevas mejoras para el proyecto.

Un aspecto interesante, podría ser ampliar la aplicación con una nueva pantalla la cual tratara con los datos de enfermedades y tratamientos obtenidos de los informes y que en base a ellos mostrara posibles tratamientos para distintas enfermedades que se propusieran.

Además se podría dedicar más tiempo al diseño de las interfaces y poder así ofrecer una mejor experiencia al usuario.

## 9. BIBLIOGRAFÍA

- [1] **Eclipse**, <https://www.eclipse.org/>
- [2] **Apache**, <http://www.apache.org/>
- [3] **Chrome**, [https://www.google.com/intl/es\\_ALL/chrome/](https://www.google.com/intl/es_ALL/chrome/)
- [4] **Cacoo**, <https://nulab-inc.com/products/#cacoo>
- [5] **Justinmind**, <https://www.justinmind.com/>
- [6] **Microsoft Office**, <https://products.office.com/es-es/home>
- [7] **Github**, <https://github.com/>
- [8] **Convenio oficinas y despachos**, [https://www.ccoo-servicios.es/archivos/OyDBizkaia0912\\_Castellano.pdf](https://www.ccoo-servicios.es/archivos/OyDBizkaia0912_Castellano.pdf)
- [9] **Organización Mundial de la Salud**, <http://www.who.int/es>
- [10] **Patientslikeme**, <https://www.patientslikeme.com/>
- [11] **ASP .NET**, <https://www.asp.net/>
- [12] **PHP**, <http://www.php.net/>
- [13] **MySQL**, <https://www.mysql.com/>
- [14] **Oracle**, <https://www.oracle.com/es/index.html>
- [15] **Chartjs**, <https://www.chartjs.org/>
- [16] **D3js**, <https://d3js.org/>
- [17] **BRAT Annotation tool**, <http://brat.nlplab.org/>
- [18] **Spring Framework**, <https://spring.io/>
- [19] **jQuery**, <https://jquery.com/>
- [20] **jQueryUI**, <https://jqueryui.com/>
- [21] **jQuery BlockUI**, <http://malsup.com/jquery/block/>
- [22] **Bootstrap**, <https://getbootstrap.com/>



# ANEXO I - CASOS DE USO EXTENDIDOS

En ese apartado se detallarán los casos de uso de la aplicación.

En la siguiente figura se muestra el esquema general.

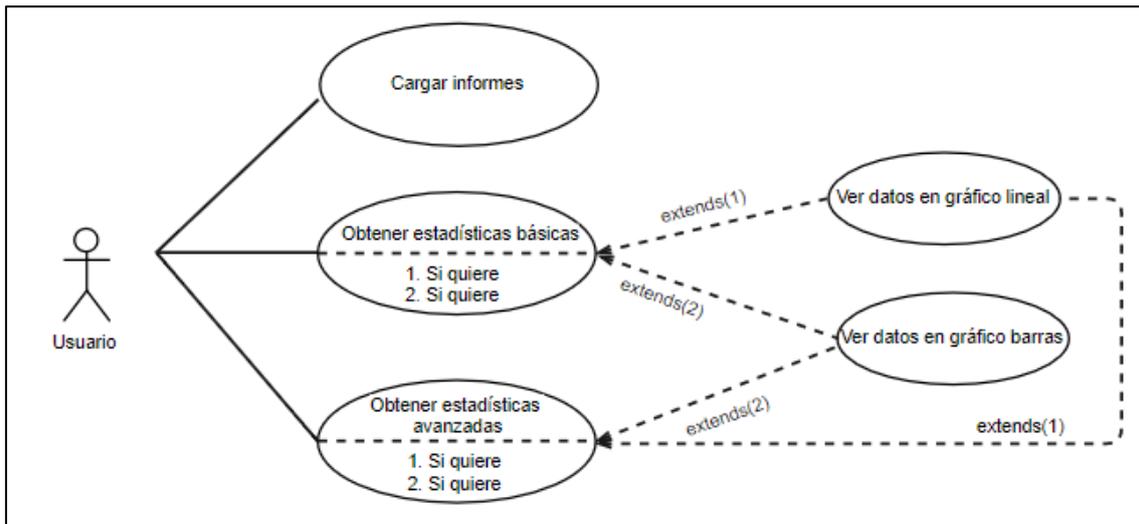


Ilustración 47: Diagrama de casos de uso

## 1. Casos de uso generales

### 1.1 Carga informes

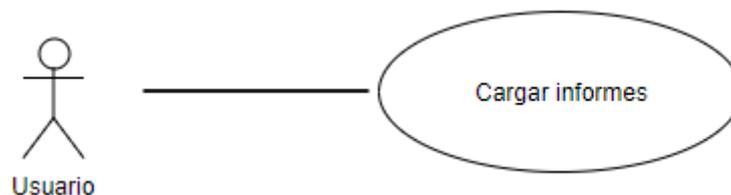


Ilustración 48: Casos de uso "Carga informes"

**Nombre:** Carga informes

---

**Descripción:** Permitirá al usuario cargar la información de informes médicos en el sistema.

---

**Actores:** Usuario

---

**Precondiciones :** -

---

**Requisitos no funcionales:** -

---

**Flujo de eventos:**

1. El usuario accede a la pantalla “Carga de informes” desde el menú principal.  
(*Ilustración 49*)
2. El usuario pulsará en el botón “Elegir archivos” y elegirá los ficheros que quiera cargar en el sistema (*Ilustración 50*)
3. El usuario elegirá en el combo inferior el hospital al que pertenecen los informes. (*Ilustración 50*)
4. El usuario pulsará en el botón “Cargar” y se almacenarán los datos de los informes en el sistema.

[Si los informes son válidos]

5.1 Se le muestra al usuario un mensaje indicando que la carga de informes se ha realizado correctamente. (*Ilustración 51*)

[Si los informes no son válidos]

5.2 Se le muestra al usuario un aviso indicando que revise los informes que ha decido cargar, junto con una tabla indicando cuales no han podido ser cargados en el sistema. (*Ilustración 52*)

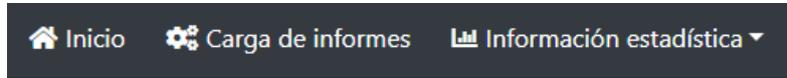
---

---

**Postcondiciones:** Se ha guardado la información de los informes en el sistema.

---

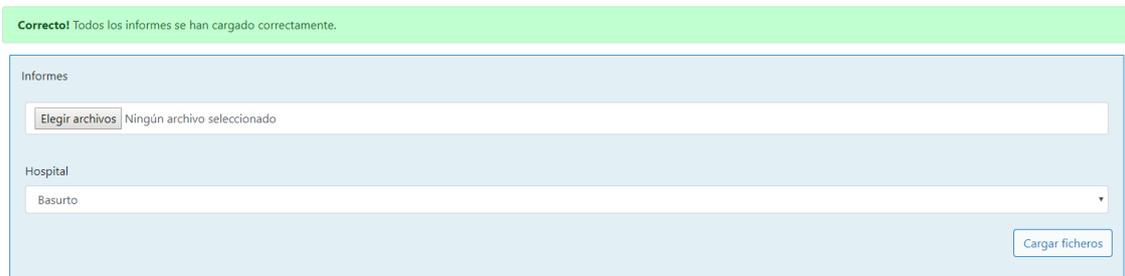
**Interfaz gráfica:**



*Ilustración 49: Menú principal*



*Ilustración 50: Interfaz carga de informes*



*Ilustración 51: Carga de informes correcta*

Informes

Elegir archivos Ningún archivo seleccionado

Hospital

Basurto

Cargar ficheros

**Cuidado!** Los siguientes informes no han podido ser cargados. Revise que el nombre del archivo sea correcto, que se ha adjuntado su correspondiente fichero.txt y cárguelos nuevamente.

#	Informe
1	100147.ann
2	100159.ann
3	103581.ann

Ilustración 52: Carga de informes errónea

## 1.2. Obtener estadísticas básicas

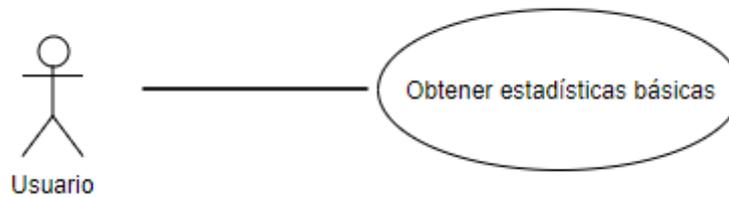


Ilustración 53: Caso de uso "Obtener estadísticas básicas"

**Nombre:** Obtener estadísticas básicas

**Descripción:** Permitirá al usuario obtener estadísticas de los informes médicos basadas en la búsqueda de medicamentos o enfermedades, así como por fechas de informes o por hospital al que pertenecen.

**Actores:** Usuario

**Precondiciones :** -

**Requisitos no funcionales:** -

---

### Flujo de eventos:

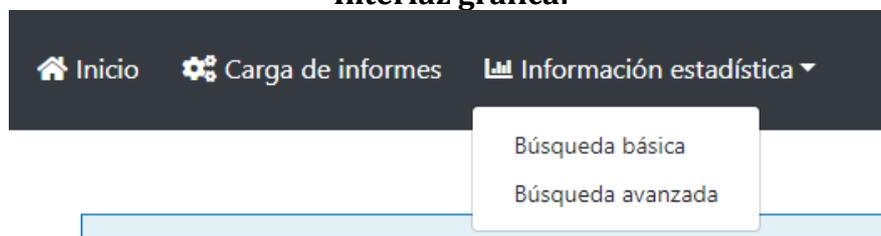
1. El usuario accede a la pantalla “Información estadística→Búsqueda básica” desde el menú principal. (*Ilustración 54*)
2. El usuario indicará los datos que desee para obtener las estadísticas en base a ellos y pulsará el botón “Filtrar”( *Ilustración 55*)
3. Se mostrará un gráfico con las estadísticas obtenidas de la base de datos.(*Ilustración 56*)

---

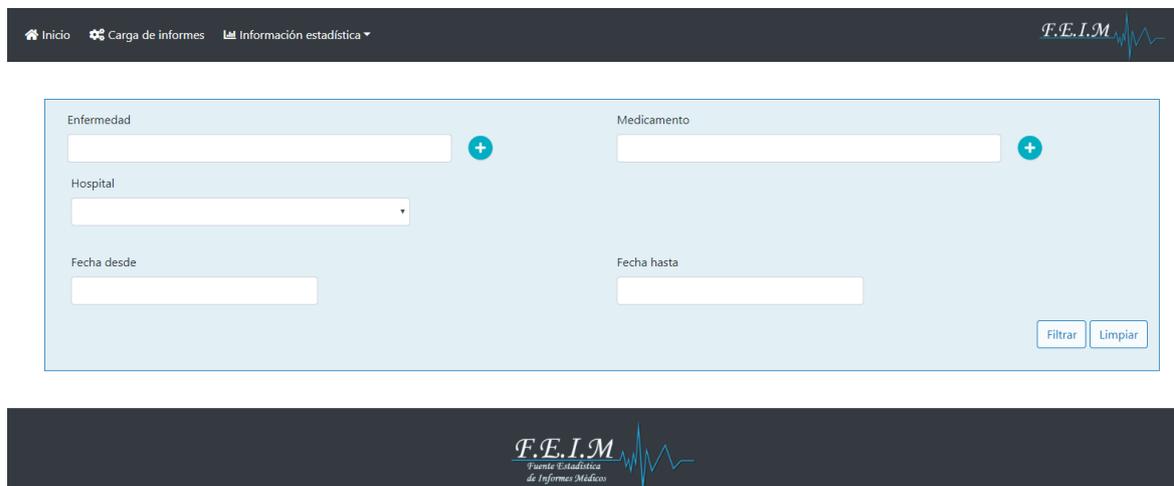
### Postcondiciones:

---

#### Interfaz gráfica:



*Ilustración 54: Menú información estadística*



*Ilustración 55: Interfaz búsqueda básica*

---

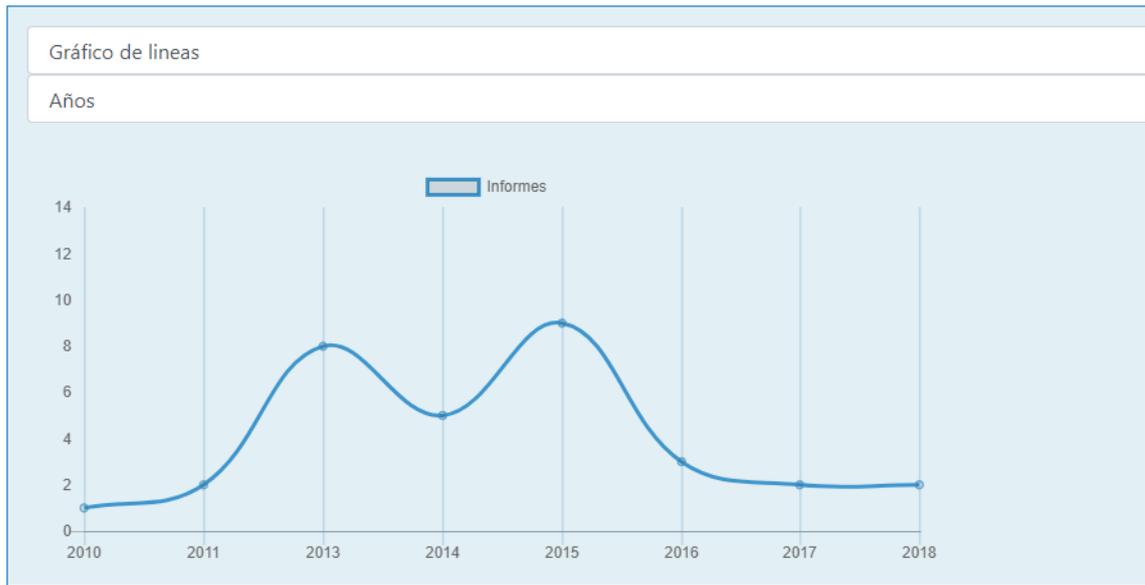


Ilustración 56: Gráfico búsqueda

### 1.3. Obtener estadísticas avanzadas

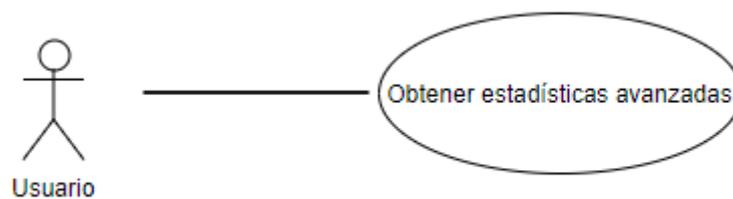


Ilustración 57: Caso de uso "Obtener estadísticas avanzadas"

**Nombre:** Obtener estadísticas avanzadas

**Descripción:** Permitirá al usuario obtener estadísticas de los informes médicos basadas en la búsqueda de enfermedades causadas por medicamentos, así como por fechas de informe o por el hospital al que pertenecen.

**Actores:** Usuario

---

## Precondiciones : -

---

## Requisitos no funcionales: -

---

### Flujo de eventos:

1. El usuario accede a la pantalla “Información estadística→Búsqueda avanzada” desde el menú principal. (*Ilustración 54*)
  2. El usuario indicará los datos que desee para obtener las estadísticas en base a ellos y pulsará el botón “Filtrar” (
  3. *Ilustración 58*)
  4. Se mostrará un gráfico con las estadísticas obtenidas de la base de datos.(*Ilustración 56*)
- 

### Postcondiciones:

---

### Interfaz gráfica:

The screenshot shows the 'Información estadística' menu item selected in the top navigation bar. The search interface includes the following elements:

- Enfermedad:** A text input field.
- Causado por:** A dropdown menu.
- Medicamento:** A text input field with a plus icon to its right.
- Hospital:** A dropdown menu.
- Fecha desde:** A date input field.
- Fecha hasta:** A date input field.
- Buttons:** 'Filtrar' and 'Limpiar' buttons at the bottom right.

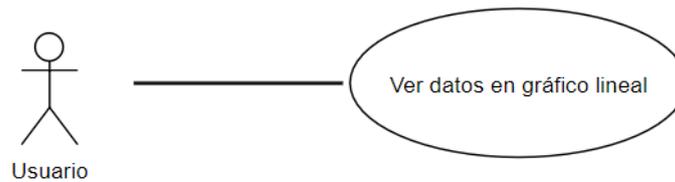
The F.E.I.M. logo is located in the top right corner of the navigation bar and the bottom center of the page.

*Ilustración 58: Interfaz búsqueda avanzada*

---

## 2. Subcasos de uso

### 2.1. Ver datos en grafico lineal



*Ilustración 59: Caso de uso "Ver datos en gráfico lineal"*

---

**Nombre:** Ver datos en grafico lineal

---

**Descripción:** Permitirá al usuario visualizar las estadísticas en un gráfico lineal.

---

**Actores:** Usuario

---

**Precondiciones :** -

---

**Requisitos no funcionales:** -

---

**Flujo de eventos:**

1. El usuario elige los datos del filtro y pulsa en el botón "Filtrar" (*Ilustración 60*)
2. Se muestran las estadísticas en un gráfico lineal. (*Ilustración 56*)  
[Si el usuario ha seleccionado previamente la opción de gráfico de barras y desea verlos en un gráfico lineal]
  - 3.1 El usuario selecciona la opción de "Gráfico lineal" en el combo. (*Ilustración 61*)

---

**Postcondiciones:**

---

## Interfaz gráfica:

Hospital

Fecha desde

Fecha hasta

Filtrar Limpiar

Ilustración 60: Filtro de la interfaz



Ilustración 61: Gráfico de barras

## 2.2. Ver datos en gráfico de barras

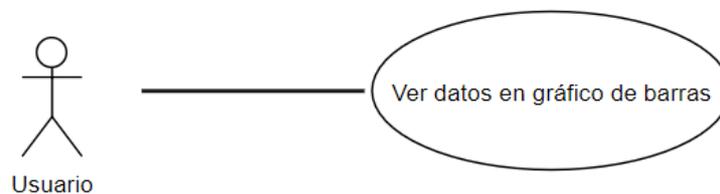


Ilustración 62: Caso de uso "Ver datos en gráfico de barras"

**Nombre:** Ver datos en grafico de barras

---

**Descripción:** Permitirá al usuario visualizar las estadísticas en un gráfico de barras.

---

**Actores:** Usuario

---

**Precondiciones :** Se ha pulsado el botón de filtrar.

---

**Requisitos no funcionales:** -

---

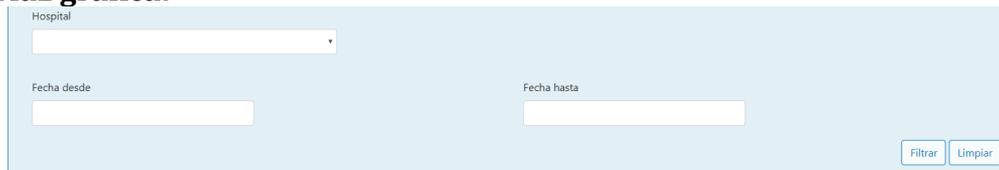
**Flujo de eventos:**

1. El usuario elige los datos del filtro y pulsa en el botón “Filtrar” (*Ilustración 63*)
  2. Se muestran las estadísticas en un gráfico lineal. (*Ilustración 56*)  
[Si el usuario quiere]
    - 4.1 El usuario selecciona la opción de “Gráfico de barras” en el combo.  
(*Ilustración 61*)
- 

**Postcondiciones:**

---

**Interfaz gráfica:**



The image shows a light blue filter interface. At the top, there is a dropdown menu labeled 'Hospital'. Below it, there are two date input fields: 'Fecha desde' on the left and 'Fecha hasta' on the right. At the bottom right, there are two buttons: 'Filtrar' and 'Limpiar'.

*Ilustración 63: Filtro interfaz gráfica*

---

## ANEXO II – DIAGRAMAS DE SECUENCIA

En ese apartado se detallarán los diagramas de secuencia de las funciones principales de la aplicación.

Por cada uno de ellos se indicarán las llamadas realizadas entre las distintas clases, con el fin de entender mejor el procedimiento que se sigue. Además se mostrarán las consultas de base de datos realizadas para obtener la información necesaria con la que se dibujarán los gráficos de la aplicación.

### 1. Carga de informes

A continuación se muestra el proceso para la carga de nuevos informes en el sistema.

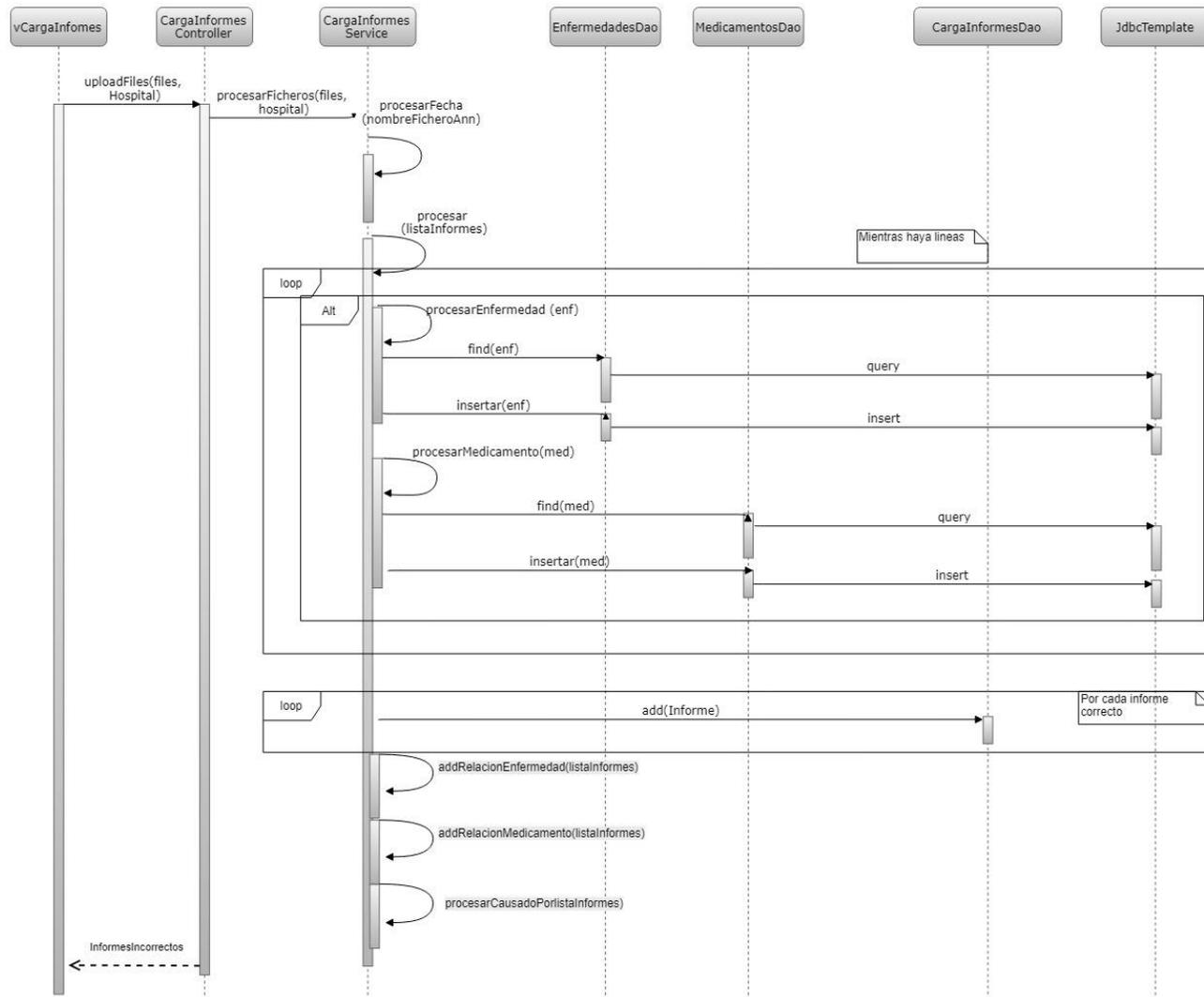


Ilustración 64: Carga de informes

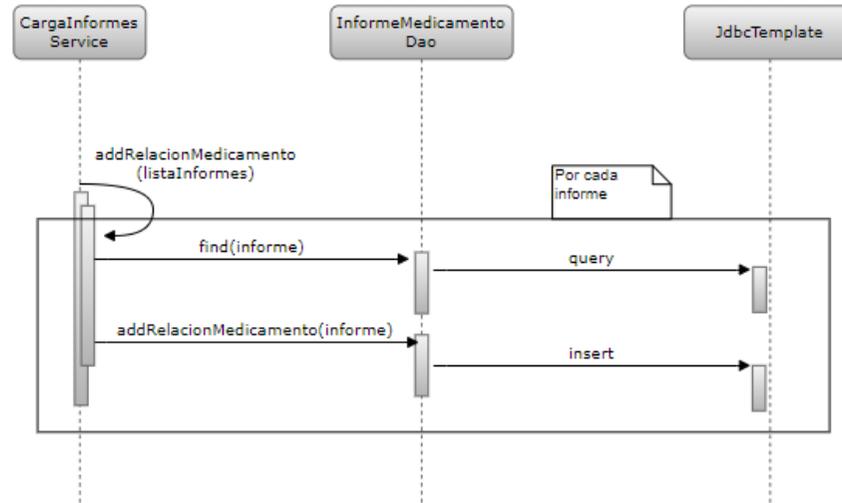


Ilustración 65: Añadir relación medicamento

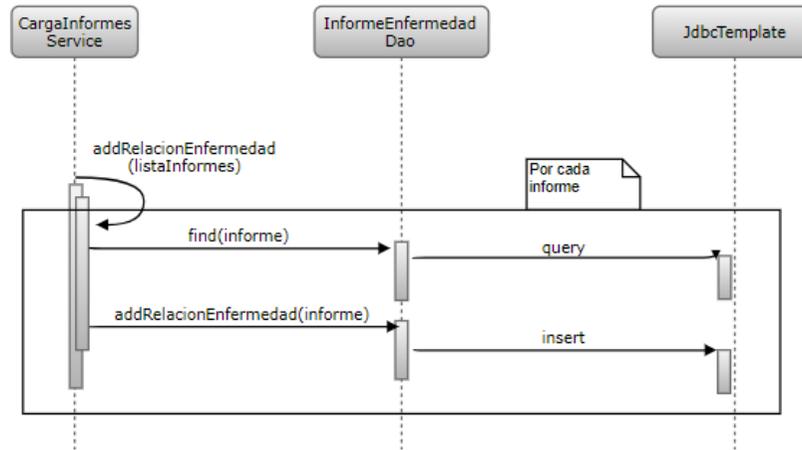


Ilustración 66: Añadir relación enfermedad

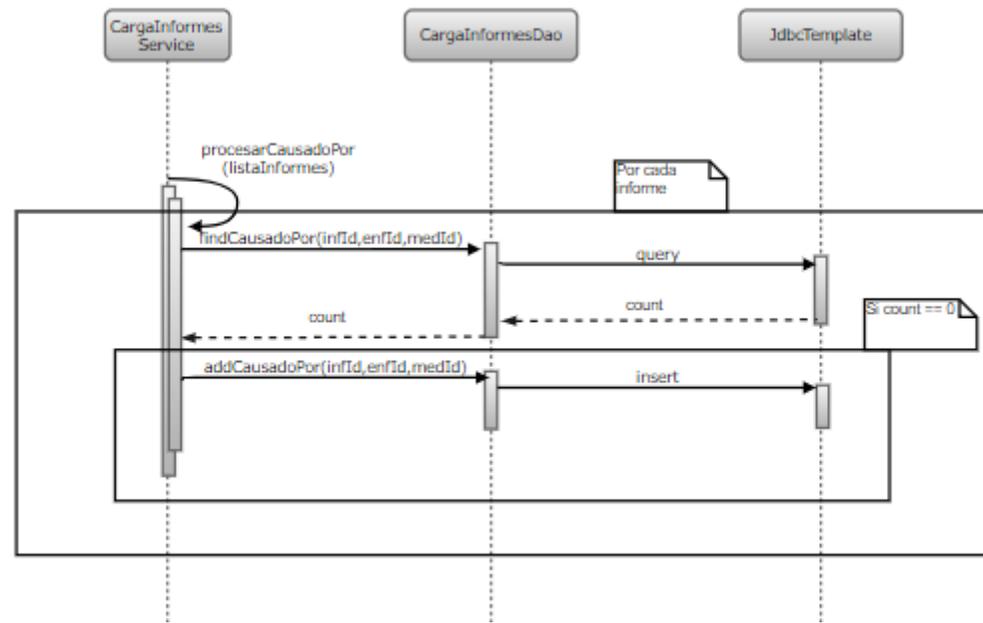


Ilustración 67: Procesar enfermedades causadas por medicamentos

## 2. Estadísticas básicas

A continuación se muestra el proceso para la obtención de estadísticas desde el filtro básico. Para poder dibujarlo posteriormente con la librería de gráficos, se ha decido realizar una única consulta que recogiera todos los datos necesarios.

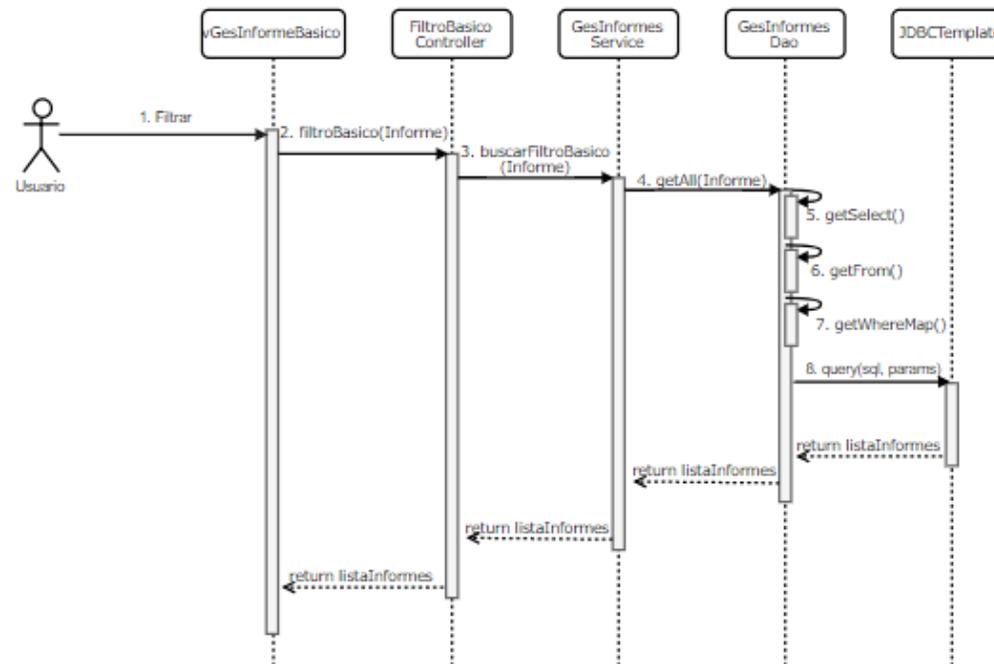


Ilustración 68: Diagrama estadísticas básicas

Los acciones 5, 6 y 7 corresponden al montaje de la consulta dinámica de función de los filtros de la pantalla. La consulta principal que se lanzaría en el paso 8 sería la siguiente:

```
SELECT COUNT(ID) NUMINFORMES, DATE_FORMAT(FECHA,'%d/%m/%Y') as FECHA FROM ( SELECT T1.ID, T1.FECHA FROM
INFORMES T1 JOIN HOSPITALES T2 ON T1.IDHOSPITAL = T2.ID LEFT JOIN ENFERMEDAD_INFORME T3 ON T1.ID = T3.IDINFORME
LEFT JOIN ENFERMEDADES T5 ON T3.IDENFERMEDAD = T5.ID LEFT JOIN MEDICAMENTO_INFORME T4 ON T1.ID = T4.IDINFORME
LEFT JOIN MEDICAMENTOS T6 ON T4.IDMEDICAMENTO = T6.ID WHERE 1=1 GROUP BY T1.ID ORDER BY FECHA ASC ) AS T GROUP
BY YEAR(FECHA) ORDER BY YEAR(FECHA) ASC
```

En el caso de que el usuario hubiera introducido algún filtro en la pantalla. Se añadirían en la parte WHERE mas condiciones como las siguientes.

- AND T6.DESCRIPCION IN (?)
- AND T5.DESCRIPCION IN (?)
- AND T1.FECHA >= ?
- AND T1.FECHA <= ?
- AND T2.ID = ?

## 2.1. Estadísticas avanzadas

A continuación se muestra el proceso para la obtención de estadísticas desde el filtro avanzado.

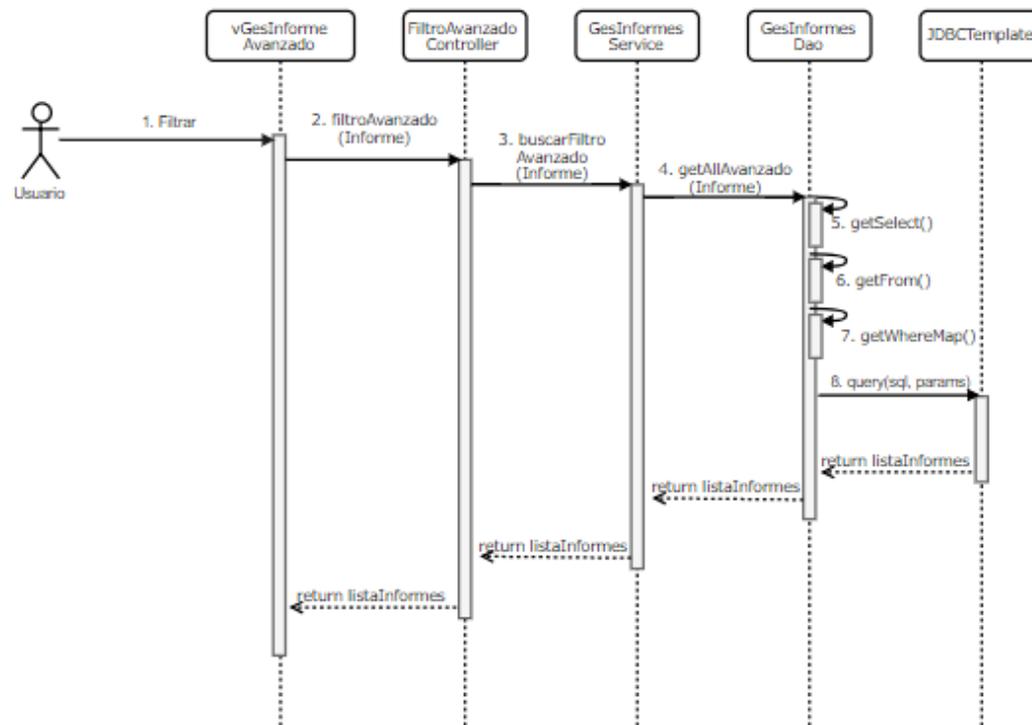


Ilustración 69: Diagrama estadísticas avanzadas

El proceso para la obtención de las estadísticas avanzadas se hace de forma similar al de la búsqueda básica. La consulta que se realiza sería la siguiente:

```
SELECT COUNT(ID) NUMINFORMES, DATE_FORMAT(FECHA,'%d/%m/%Y') AS FECHA FROM ( SELECT T1.ID, T1.FECHA FROM
INFORMES T1 JOIN HOSPITALES T2 ON T1.IDHOSPITAL = T2.ID LEFT JOIN ENFERMEDAD_INFORME T3 ON T1.ID = T3.IDINFORME
LEFT JOIN ENFERMEDADES T5 ON T3.IDENFERMEDAD = T5.ID LEFT JOIN MEDICAMENTO_INFORME T4 ON T1.ID = T4.IDINFORME
LEFT JOIN MEDICAMENTOS T6 ON T4.IDMEDICAMENTO = T6.ID LEFT JOIN CAUSADO_POR T10 ON T10.IDINFORME = T1.ID WHERE
1=1 GROUP BY T1.ID ORDER BY FECHA ASC ) AS T GROUP BY YEAR(FECHA) ORDER BY YEAR(FECHA) ASC
```

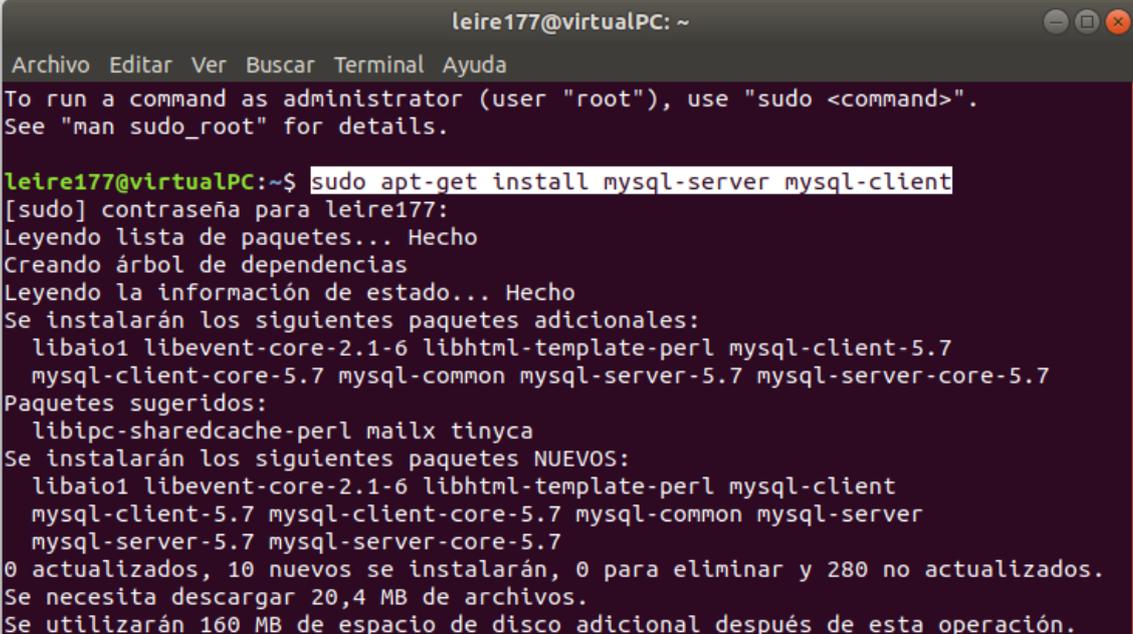
La consulta es parecida a la anterior, ya que se realiza la misma búsqueda en base a descripciones de enfermedades y medicamentos, pero esta vez en la tabla de relaciones que hace referencia a las enfermedades causadas por medicamentos. Para añadir las condiciones WHERE se reutilizará el mismo método ya que los filtros a introducir por el usuario serían los mismos.



## ANEXO III – MANUAL DE INSTALACIÓN

A continuación se detallarán los pasos a seguir para poner en funcionamiento la aplicación en Ubuntu 18.04, explicando la instalación de las herramientas necesarias.

1. Se instalará MySQL desde la línea de comandos con la instrucción `sudo apt-get install mysql-server mysql-client` tal y como se muestra en la *Ilustración 70*.



```
leire177@virtualPC: ~
Archivo Editar Ver Buscar Terminal Ayuda
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

leire177@virtualPC:~$ sudo apt-get install mysql-server mysql-client
[sudo] contraseña para leire177:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
 libaio1 libevent-core-2.1-6 libhtml-template-perl mysql-client-5.7
 mysql-client-core-5.7 mysql-common mysql-server-5.7 mysql-server-core-5.7
Paquetes sugeridos:
 libipc-sharedcache-perl mailx tinyca
Se instalarán los siguientes paquetes NUEVOS:
 libaio1 libevent-core-2.1-6 libhtml-template-perl mysql-client
 mysql-client-5.7 mysql-client-core-5.7 mysql-common mysql-server
 mysql-server-5.7 mysql-server-core-5.7
0 actualizados, 10 nuevos se instalarán, 0 para eliminar y 280 no actualizados.
Se necesita descargar 20,4 MB de archivos.
Se utilizarán 160 MB de espacio de disco adicional después de esta operación.
```

*Ilustración 70: Instalar MySQL*

2. Se instalará MySQL Workbench para poder importar la base de datos. Se hará mediante la instrucción `sudo apt-get install mysql-workbench` tal y como se muestra en la *Ilustración 71*.

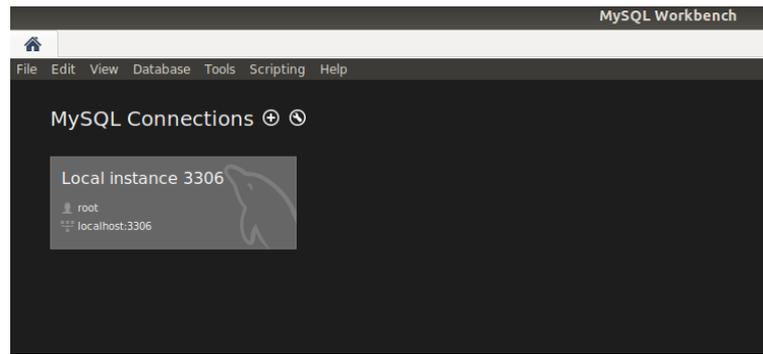
```
leire177@virtualPC: ~
Archivo Editar Ver Buscar Terminal Ayuda
cnf (my.cnf) en modo automático
Renaming removed key_buffer and myisam-recover options (if present)
Created symlink /etc/systemd/system/multi-user.target.wants/mysql.service → /lib
/systemd/system/mysql.service.
Configurando mysql-server (5.7.22-0ubuntu18.04.1) ...
Procesando disparadores para libc-bin (2.27-3ubuntu1) ...
Procesando disparadores para ureadahead (0.100.0-20) ...
Procesando disparadores para systemd (237-3ubuntu10) ...
leire177@virtualPC:~$ sudo apt-get install mysql-workbench
[sudo] contraseña para leire177:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
gdal-data libaec0 libarmadillo8 libarpack2 libatkmm-1.6-1v5 libblas3
libcairomm-1.0-1v5 libctemplate3 libdap25 libdapclient6v5 libepsilon1
libfreexl1 libfyba0 libgdal20 libgeos-3.6.2 libgeos-c1v5 libgeotiff2
libgfortran4 libgif7 libglibmm-2.4-1v5 libgnome-keyring-common
libgnome-keyring0 libgtkmm-2.4-1v5 libhdf4-0-alt libhdf5-100 libkmlbase1
libkmldom1 libkmlengine1 liblapack3 libminizip1 libmysqlclient20
libmysqlcppconn7v5 libnetcdf13 libodbc1 libogdi3.2 libopenjp2-7
libpangomm-1.4-1v5 libpcrecpp0v5 libpq5 libproj12 libpython-stdlib libqhull7
libquadmath0 libsigc++-2.0-0v5 libspatialite7 libsperl5 libsz2
libtinysql2.6.2v5 liburiparser1 libvsqllitepp3v5 libxerces-c3.2 libzip4
```

*Ilustración 71: Instalar MySQL Workbench*

3. Se accederá a la aplicación Mysql Workbench y se seleccionará la conexión tal y como se muestra en la *Ilustración 73*.



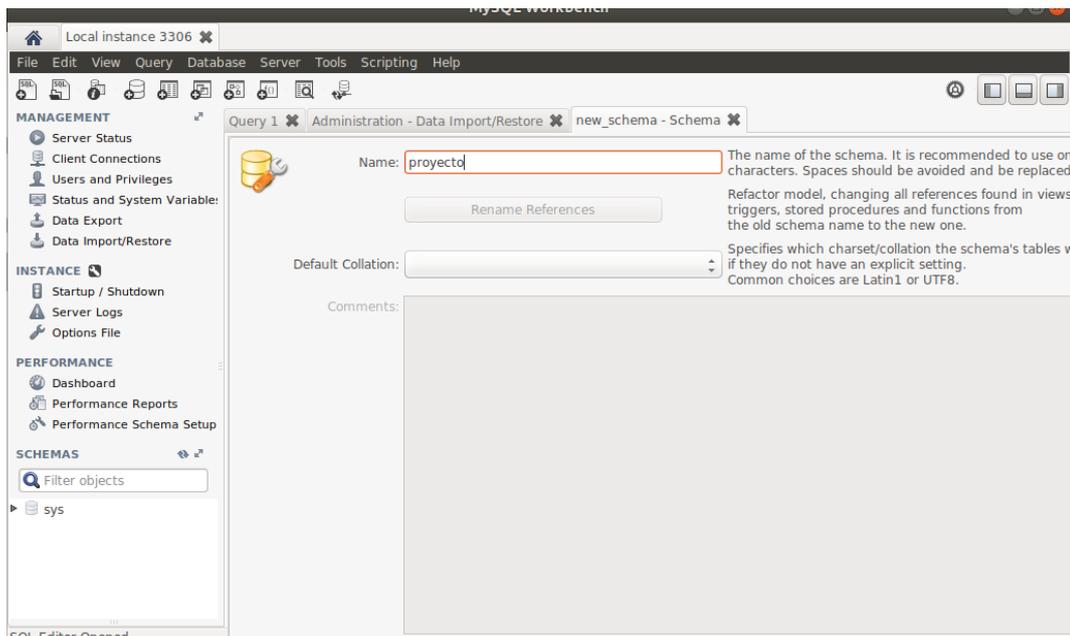
*Ilustración 72: Acceder a Mysql Workbench*



*Ilustración 73: Iniciar conexión MySQL*

4. Se deberá crear un nuevo esquema llamado “proyecto” haciendo click con el botón derecho sobre el apartado *schemas* situado abajo a la izquierda.

Schemas → Create Schema

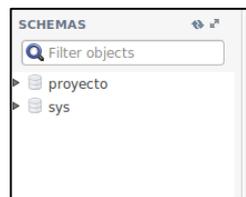


*Ilustración 74: Crear nuevo esquema*

5. Seguidamente, se importarán los datos, con la instrucción que se muestra en la *Ilustración 75*. Una vez realizado deberán aparecer en MySQL Workbench las tablas correspondientes a la aplicación bajo el esquema llamado “Proyecto”

```
leire177@virtualPC:~$ mysql -u root -p proyecto < /home/leire177/Descargas/TFG.sql
```

*Ilustración 75: Importar base de datos*



*Ilustración 76: Esquema creado*

6. A continuación se procederá a instalar Apache Tomcat 8.0. Se hará mediante la instrucción que se muestra en la siguiente figura.

```
leire177@virtualPC:~$ sudo apt-get install tomcat8
```

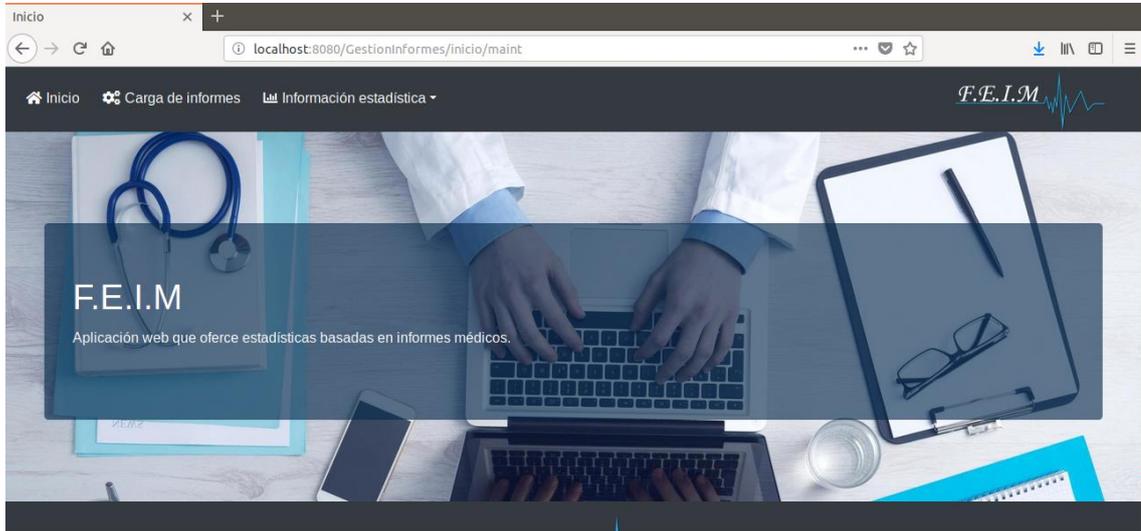
*Ilustración 77: Instalar Apache Tomcat*

7. Una vez instalado, será necesario copiar el archivo “GestionInformes.war” en la ruta `/var/lib/tomcat8/webapps` para poder desplegar la aplicación correctamente.

```
leire177@virtualPC:~$ sudo cp /home/leire177/Descargas/GestionInformes.war /var/lib/tomcat8/webapps
```

*Ilustración 78: Importar archivo .war*

8. Por último, se iniciará el navegador, y se podrá acceder a la aplicación mediante la ruta " *localhost:8080/GestionInformes/inicio/maint*" tal y como se muestra en la *Ilustración 79*.



*Ilustración 79: Acceso a la aplicación*



# ANEXO IV – MANUAL DE USUARIO

En este apartado se detallarán las instrucciones a seguir para poder acceder a la aplicación.

El sistema consta de 3 funcionalidades principales, la carga de nuevos informes, y la obtención de estadísticas tanto de forma básica como avanzada.

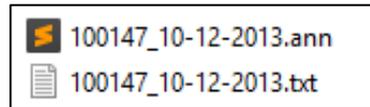
## 1. Carga de informes

Se accederá a la pantalla desde el menú principal “Carga de informes”.



Ilustración 80: Carga de informes

En el filtro que se muestra por pantalla, pulsando en el botón “Elegir archivos”, se seleccionarán los informes que se desean subir. Cada uno de estos ficheros deberá cargarse en dos extensiones distintas, una en formato `.txt` y otra en formato `.ann`. Los dos ficheros deberán seguir el mismo formato para su nombre: el identificador, seguido de la fecha del informe separado por guiones, y todo ello separado por un guion bajo. A continuación se muestra un ejemplo.



*Ilustración 81: Formato de carga de informes*

Una vez elegidos los ficheros, se selecciona en el combo inferior el hospital al que pertenecen y se pulsa en el botón “Cargar”.

Si los ficheros tuvieran errores o el formato de su nombre fuera incorrecto, se mostraría el siguiente aviso, seguido de los ficheros que no han podido cargarse en el sistema.

A screenshot of a web interface for uploading reports. At the top, there is a section titled 'Informes' with a file selection button and a dropdown menu for 'Hospital' set to 'Basurto'. A 'Cargar ficheros' button is on the right. Below this is a yellow warning banner: 'Cuidado! Los siguientes informes no han podido ser cargados. Revise que el nombre del archivo sea correcto, que se ha adjuntado su correspondiente fichero.txt y cárguelos nuevamente.' Below the banner is a table with the following data:

#	Informe
1	100147.ann
2	100159.ann
3	103581.ann

*Ilustración 82: Carga de informes errónea*

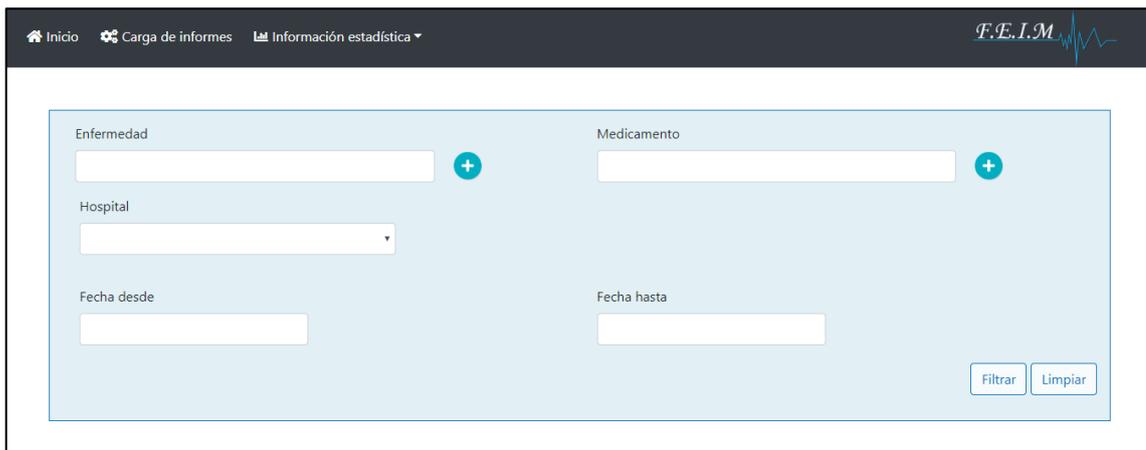
En caso contrario, se mostrará un mensaje indicando que la carga de informes se ha realizado correctamente.

A screenshot of the same web interface as in Illustration 82, but with a green success banner at the top: 'Correcto! Todos los informes se han cargado correctamente.' The rest of the interface, including the file selection, hospital dropdown, and 'Cargar ficheros' button, is identical to the previous screenshot.

*Ilustración 83: Carga de informes correcta*

## 2. Información estadística en búsqueda básica

Se accederá a la pantalla desde el menú principal “Información estadística → Búsqueda básica”.



*Ilustración 84: Información estadística básica*

En el filtro que se muestra por pantalla se podrán introducir nombres de medicamentos así como de enfermedades, pudiendo añadir más pulsando en el icono + situado a la derecha de cada campo. Asimismo, una vez añadidos, si se desean borrar bastará con pulsar en el botón X de cada campo.

También existe la posibilidad de filtrar por la fecha de los informes seleccionándola desde el calendario que se desplegará al situarse en ese mismo campo.

Una vez elegidas las características en base a las cuales se desea realizar la búsqueda, se pulsa en el botón “Filtrar”.

Como se observa en la *Ilustración 85* en la parte inferior de la pantalla se mostrará un gráfico con las estadísticas de los informes que cumplen con las características del filtro superior.

Por defecto la información se mostrará en un gráfico lineal y representado por los años de los informes, sin embargo, existe la posibilidad de mostrar la información en un gráfico de barras así como por otros periodos de tiempo, eligiendo las opciones correspondientes en los combos situados encima del gráfico.

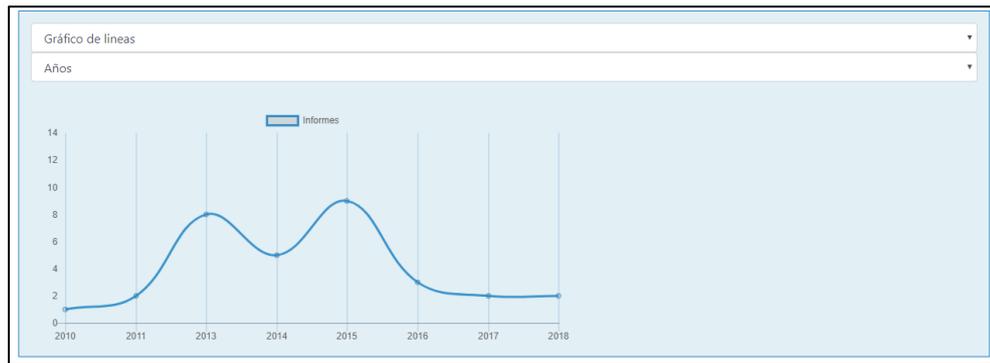


Ilustración 85: Gráfico búsqueda básica

### 3. Información estadística en búsqueda avanzada

Se accederá a la pantalla desde el menú principal “Información estadística → Búsqueda avanzada”.

Inicio Carga de informes Información estadística F.E.I.M.

Enfermedad  Causado por  +

Hospital

Fecha desde  Fecha hasta

Filtrar Limpiar

F.E.I.M. Fuente Estadística de Informes Médicos

Ilustración 86: Información estadística avanzada

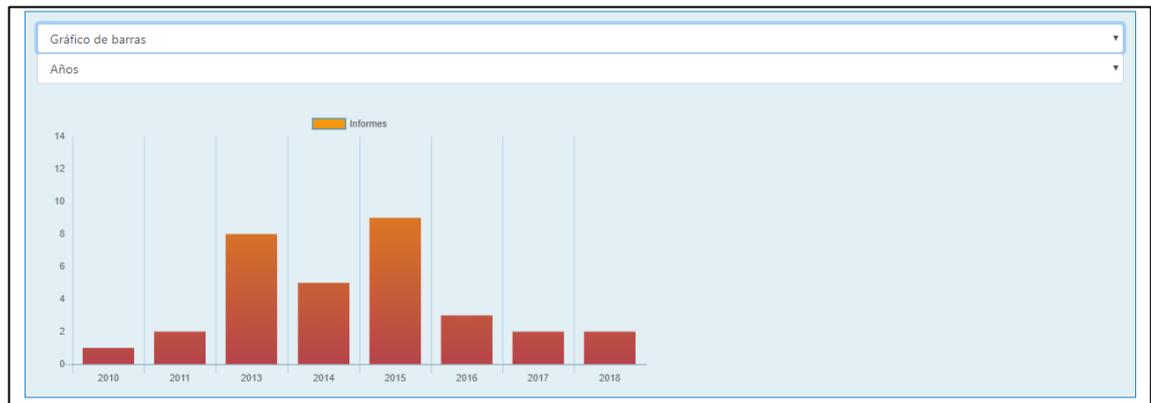
En el filtro que se muestra por pantalla se podrán introducir nombres de enfermedades y el medicamento causante de ellas. Existe la posibilidad de añadir más pulsando en el icono + situado a la derecha del campo medicamento, que añadirá otra línea completa igual que la anterior para poder introducir otra enfermedad causada por otro medicamento.

Asimismo, una vez añadidos, si se desean borrar bastará con pulsar en el botón X que aparecerá a la derecha del botón +.

También existe la posibilidad de filtrar por la fecha de los informes seleccionándola desde el calendario que se desplegará al situarse en ese mismo campo.

Una vez elegidas las características en base a las cuales se desea realizar la búsqueda, se pulsa en el botón “*Filtrar*”.

Como se observa en la *Ilustración 87* en la parte inferior de la pantalla se mostrará un gráfico, que al igual que en la pantalla de *Búsqueda básica*, se podrá visualizar tanto de forma lineal como en gráfico de barras, y podrá obtenerse en distintos periodos de tiempo seleccionando las opciones en los combos correspondientes.



*Ilustración 87: Búsqueda avanzada en gráfico de barras*