

# Facultad de Informática

## Grado de Ingeniería Informática

▪ Trabajo Fin de Grado ▪

Ingeniería del Software

Creación de un REgistro NAcional de pacientes  
intervenidos de Cáncer de Esófago (RENACE) para la  
Asociación Española de Cirujanos.

---

Jon Díez Barrios

Junio 2020

Dirigido por: Rosa Arruabarrena



# Agradecimientos

A los cirujanos que han contribuido aportando los datos de sus pacientes y el *feedback* necesario para que este registro pueda coger cuerpo.

A la Sección de Cirugía esófago-gástrica, en particular a su coordinador, por la idea y el apoyo incondicional.

A la Asociación Española de Cirujanos, por creer que este registro puede ser un punto de partida para un proyecto más amplio en el futuro.

A Luis Puertas, de iMedia Comunicación, por facilitar la utilización del servidor de la AEC.

A mi directora, Rosa, por su ayuda, motivación y confianza, además de sus consejos.

A todos y todas las compañeras con las que he compartido este grado, por hacer de esta etapa una experiencia inolvidable.



# Resumen

## Introducción

La elaboración de registros multicéntricos de actividad, particularmente relativos a enfermedades malignas, representa uno de los principales objetivos de la Asociación Española de Cirujanos. Poder disponer de bases de datos amplias representa una oportunidad de mejorar la práctica clínica, al conocer la realidad global de un problema concreto más que las experiencias particulares de uno u otro equipo.

La propuesta de crear una interfaz ágil, sencilla, intuitiva y que genere automáticamente informes globales y gráficos demostrativos supuso la base para realizar este TFG.

## Objetivos

Objetivo principal: crear un REgistro NAcional de Cirugía Esofágica (RENACE) que permita la recopilación de datos para su posterior análisis, facilitando la toma de decisiones.

Objetivos secundarios:

- Importación fácil de pacientes desde otras bases de datos vigentes.
- Creación automática de informes basados en los datos almacenados.
- Visualización de los datos de manera gráfica.
- Preparación del software desarrollado para su extensión a otras secciones de la AEC.

Para ello, se intentarán aprovechar al máximo las herramientas y recursos aprendidos durante el grado. Igualmente, es fundamental que se respeten los principios de calidad, exhaustividad y protección de datos.

## Metodología y desarrollo del proyecto

Se ha utilizado la metodología Agile, que mejora la satisfacción del cliente y la implicación del equipo, ahorra tiempo y costes, divide el trabajo en entregas y permite un mayor control de problemas.

Se ha creado una base de datos relacional MySQL, que facilita el trabajo en conjunto con PHP y permite la conexión entre las diferentes variables de cada paciente, cuya gestión se lleva a cabo a través de phpMyAdmin.

Para la interfaz se ha hecho uso de HMTL, en conjunto con PHP, que permite la comunicación con el servidor que almacena los datos. La librería CanvasJS facilita la visualización de los resultados mediante gráficos.

Se ha insistido especialmente en evitar duplicar el esfuerzo de aquellos equipos que ya formaban parte de otros registros, facilitando la importación desde estos, con la ayuda de la librería PhpSpreadsheet.

## Resultados

La evaluación realizada por los 9 hospitales implicados en el proyecto destaca el carácter intuitivo, la facilidad de inserción de variables, su interés para la práctica clínica y para posibles publicaciones de estudios multicéntricos. Se enfatiza el valor de los indicadores generados en la sección de Informes, su comparación con los datos globales y la posibilidad de hacerlo con los

estándares internacionales. Los gráficos permiten la visualización de las diferencias entre equipos y el análisis de resultados según el tipo de técnica aplicada.

### **Perspectivas de futuro**

El registro RENACE, una vez aceptado, puede instaurarse a corto plazo para su uso generalizado dentro de la Sección de Cirugía esófago-gástrica y, con ciertas adaptaciones, constituir la base de un eventual Registro Nacional de Cirugía. Integrado con alguna plataforma de base estadística, permitirá añadir datos de supervivencia a largo plazo, aspecto relevante dentro de la evaluación de resultados del tratamiento multidisciplinar del cáncer.

# Índice

<b>1.</b>	<b>Antecedentes .....</b>	<b>1</b>
1.1.	Introducción .....	1
1.2.	Interés por el área .....	2
1.3.	Antecedentes del proyecto .....	2
<b>2.</b>	<b>Planificación del proyecto .....</b>	<b>5</b>
2.1.	Objetivos .....	5
2.1.1.	Objetivos del alumno .....	5
2.1.2.	Objetivos del trabajo .....	5
2.1.2.1.	Requisitos mínimos .....	5
2.1.2.2.	Objetivos opcionales .....	6
2.1.3.	Objetivos del interesado .....	6
2.2.	Alcance .....	6
2.3.	Interesados.....	6
2.4.	Metodología de trabajo .....	7
2.4.1.	Metodología Agile .....	7
2.4.2.	Características .....	7
2.4.3.	Aplicación en el trabajo.....	8
2.5.	Estructura de descomposición del trabajo (EDT).....	8
2.6.	Diagrama de Gantt .....	12
2.7.	Comunicación y sistemas de información.....	12
2.8.	Gestión de riesgos .....	13
2.9.	Recursos .....	14
2.10.	Entregables.....	14
2.10.1.	Entregables relacionados con la gestión .....	14
2.10.2.	Entregables relacionados con el desarrollo .....	14
2.10.3.	Otros entregables.....	15
2.11.	Plan de calidad .....	15
2.11.1.	Certificadores de calidad.....	15
2.11.2.	Control de calidad .....	16
2.11.3.	Aspectos éticos.....	16
<b>3.</b>	<b>Análisis y diseño .....</b>	<b>17</b>
3.1.	Arquitectura de la aplicación .....	17
3.2.	Interacción con el usuario .....	17
3.2.1.	Perfil de usuario .....	17
3.2.2.	Circunstancias de uso .....	17
3.3.	Modelo de datos .....	18
3.3.1.	Base de datos .....	18
3.3.2.	Sistema de ficheros de la aplicación .....	21
3.4.	Objetivos del trabajo.....	22
3.4.1.	Inserción y edición de los pacientes.....	22
3.4.2.	Importación desde Eureka.....	22
3.4.3.	Creación de informes .....	22
3.4.4.	Visualización de gráficos .....	22
3.4.5.	Anonimidad de los datos.....	23

3.4.6. Preparación del software desarrollado para su extensión a otras secciones de la AEC  
23

<b>4.</b>	Casos de uso.....	25
4.1.	Usuario no registrado.....	25
4.1.1.	Solicitud de registro .....	25
4.2.	Usuario registrado.....	25
4.2.1.	Inicio de sesión .....	26
4.2.2.	Añadir paciente .....	26
4.2.3.	Visualización/Edición de los datos de un paciente .....	27
4.2.4.	Visualización de informes.....	27
4.2.5.	Visualización de gráficos .....	28
4.3.	Data manager.....	29
4.3.1.	Creación de usuarios .....	29
4.3.2.	Borrado de pacientes .....	29
<b>5.</b>	Tecnologías y entorno de desarrollo.....	31
5.1.	Tecnologías.....	31
5.1.1.	HTML .....	31
5.1.2.	PHP .....	31
5.1.3.	Javascript.....	31
5.1.4.	JQuery.....	32
5.1.5.	MySQL .....	32
5.2.	Herramientas de software .....	32
5.2.1.	GitHub Desktop .....	32
5.2.2.	Visual Studio Code.....	33
5.2.3.	PhpMyAdmin.....	33
5.2.4.	FileZilla.....	33
5.2.5.	Xampp .....	33
<b>6.</b>	Implementación .....	35
6.1.	Configuración de herramientas.....	35
6.2.	Decisiones globales .....	36
6.3.	Registro RENACE .....	37
6.4.	Pruebas.....	51
6.4.1.	Pruebas Entrega parcial 1.....	52
6.4.2.	Pruebas Entrega parcial 2.....	52
6.4.3.	Pruebas Entrega parcial 3.....	53
6.4.4.	Pruebas Entrega parcial 4.....	53
6.4.5.	Pruebas Entrega parcial 5.....	54
6.4.6.	Pruebas con usuarios .....	54
<b>7.</b>	Gestión del proyecto .....	57
7.1.	Gestión de la planificación .....	57
7.2.	Gestión del alcance .....	58
7.3.	Gestión de los sistemas de información .....	59
<b>8.</b>	Conclusiones.....	61
8.1.	Conclusión general .....	61



8.2. Tecnologías y metodologías utilizadas .....	61
8.3. Respecto al grado .....	61
8.4. Futuro de la aplicación .....	62
Bibliografía .....	63
Anexos .....	65
1. Informe favorable del CEImE a la realización de este proyecto.....	66
2. Diagrama de Gantt detallado .....	67
3. Formulario completo “Añadir nuevo paciente” .....	68
4. Modelo de datos íntegro de la base de datos de RENACE.....	70
5. Gráficos .....	71



## Índice de figuras

Figura 1- Diagrama EDT.....	9
Figura 2- Diagrama Gantt.....	12
Figura 3- Tabla Usuarios (cirujanos) de Renace.....	18
Figura 4- Tabla Pacientes.....	19
Figura 5- Tablas Variables simples.....	19
Figura 6- Tablas Variables múltiples.....	19
Figura 7- Tabla eurecca_import_variables.....	20
Figura 8- Tabla eurecca_import_traducciones.....	21
Figura 9- Sistema de ficheros de la memoria.....	21
Figura 10- Sistema ficheros de la implementación.....	22
Figura 11- Solicitud de registro.....	25
Figura 12- Login.....	26
Figura 13- Añadir nuevo paciente.....	26
Figura 14- Ver/Editar paciente propio.....	27
Figura 15- Informes.....	28
Figura 16- Gráficos.....	28
Figura 17- Crear usuario.....	29
Figura 18- Borrar paciente.....	30
Figura 19- Panel de control Xampp.....	35
Figura 20- Inicio del registro.....	37
Figura 21. Codificación de la disposición de los elementos principales.....	38
Figura 22- Llamada a la cabecera principal.....	38
Figura 23- Hoja de estilos de la cabecera y el menú de navegación.....	39
Figura 24- Llamada al archivo de configuración de la base de datos.....	39
Figura 25- Conexión y consulta a la base de datos.....	40
Figura 26- Cifrado de la contraseña.....	40
Figura 27- Verificación de la contraseña.....	40
Figura 28- Tabla pacientes.....	41
Figura 29- Script buscar paciente por código.....	41
Figura 30- Buscar paciente por código.....	42
Figura 31- Consulta buscar por fecha.....	42
Figura 32- Buscar paciente por fecha.....	42
Figura 33- Fragmento añadir nuevo paciente.....	43
Figura 34- Ejemplo del input de tipo numérico.....	43
Figura 35- Comparación de fechas en diferentes buscadores.....	44
Figura 36- Fragmento del input Fechas.....	44
Figura 37- Ejemplo dropdown lists.....	44
Figura 38- Llamada a la generación de código automático.....	45
Figura 39- Código switches.....	45
Figura 40- Ejemplo visual de variables de dos valores.....	45
Figura 41- String creada por la edición de las variables.....	47
Figura 42- Sentencia update que utiliza la string generada.....	47
Figura 43- Switch para el tratamiento de las diferentes variables.....	48
Figura 44- Informes.....	49
Figura 45- Indicadores de calidad de la resección.....	50
Figura 46- Fracasos según el tipo de reconstrucción.....	50

Figura 47- Estructura de datos exigida.....	51
Figura 48- Creación array de datos .....	51
Figura 49- Informe favorable del CEImE a la realización de este proyecto .....	66
Figura 50- Inform favoblea CEIM .....	66
Figura 51- Diagrama Gantt detallado.....	67
Figura 52- Parte superior del formulario completo “Añadir nuevo paciente” .....	68
Figura 53- Parte inferior del formulario completo “Añadir nuevo paciente” .....	69
Figura 54- Estructura de la base de datos.....	70
Figura 55- Página de visualización general de Gráficos .....	71

## Índice de tablas

Tabla 1- Planificación de tareas con horas estimadas .....	11
Tabla 2- Pruebas Entrega parcial 1.....	52
Tabla 3- Pruebas Entrega parcial 2.....	52
Tabla 4- Pruebas Entrega parcial 3.....	53
Tabla 5- Pruebas Entrega parcial 4.....	54
Tabla 6- Pruebas Entrega parcial 5.....	54
Tabla 7- Resultados de la encuesta de satisfacción .....	55



# 1.

## Antecedentes

En este capítulo se explican los antecedentes del proyecto, aspecto importante para situar al lector y comprender las motivaciones y razones por las que se ha llevado a cabo esta idea.

### 1.1. Introducción

La práctica quirúrgica se basa con frecuencia en unas pocas publicaciones científicas con elevado nivel de evidencia, derivada más bien del tipo de diseño (ensayos clínicos o metaanálisis) que del volumen de pacientes estudiados. Cada profesional, inmerso en las rutinas de su entorno habitual y observando los resultados de su limitada experiencia, va introduciendo pequeñas modificaciones que van convirtiéndose en ley, de forma que la respuesta a un problema en un determinado Servicio de Cirugía se fundamenta en la costumbre, en la repetición de una serie de actos y circuitos y hasta en anécdotas clínicas.

En la actualidad, la Cirugía general está dividida en subespecialidades (esófago-gástrica, colon y recto, hepatobiliar, pared abdominal, endocrina), de forma que cada profesional se convierte en un gran “experto” en tratar unas pocas patologías, y nadie puede ratificar o negar sus propuestas de cambio, sus recursos técnicos o sus protocolos de actuación. Sin embargo, en un hospital o comunidad autónoma diferente, las indicaciones de tratamiento, el proceso diagnóstico, las demoras, el tipo y orden de los tratamientos recibidos o la técnica quirúrgica pueden variar, sin que en realidad pueda saberse quién hace bien y quién no.

Un registro multicéntrico de actividad puede permitir introducir datos de un número importante de pacientes, enfocados de diferentes maneras, y poder comparar los resultados. A mayor número de pacientes introducidos, habrá mayor potencia en los datos, mayor validez en las conclusiones y más razones para defender uno u otro abordaje con el que dar respuesta a un problema común. En una patología concreta, podremos encontrar una relación entre el tipo de técnica quirúrgica y las complicaciones postoperatorias o entre el tipo de tratamiento recibido y la supervivencia a largo plazo.

Este tipo de argumentación es especialmente válido cuando se trata de enfermedades poco frecuentes y complejas, en las que la experiencia acumulada por cada equipo es dudosa de interpretar y las líneas de trabajo difíciles de estandarizar.

La creación de registros multicéntricos es a veces difícil de aceptar, porque compara a cada equipo quirúrgico con los demás y muestra sus fracasos y miserias, porque exige un esfuerzo de sistematización y transparencia. Pero es probablemente el único camino para difundir el conocimiento y mejorar la práctica clínica en beneficio del paciente.

En la Asociación Española de Cirujanos (AEC) es un tema de debate actual. Parece que ha llegado el momento de aceptar ese cambio de cultura que permita progresar hacia la toma de decisiones basándose en la experiencia de muchos, más que en la de unos pocos. La cirugía del cáncer de esófago permite ejercer una acción “piloto” al tratarse de una patología rara, en la que introducir los pacientes en un registro no supone un gran trabajo añadido a las responsabilidades cotidianas de un cirujano. Si la experiencia es positiva, la introducción de datos suficientemente intuitiva y visual, el *feedback* inmediato y las conclusiones evidentes, su extensión a otras patologías y secciones estará justificada y garantizada.

Sobre esta base, se ha propuesto la creación de una base de datos accesible desde la página Web de la Asociación Española de Cirujanos (AEC), anonimizada, fácil de cumplimentar, que simplifique la exportación de datos desde otras bases existentes y que proporcione informes dinámicos de los resultados. Para ello, se ha buscado la colaboración entre la Sección de Cirugía esófago-gástrica y Facultad de Informática de la UPV/EHU, a través de este trabajo de fin de grado (TFG), con la intención de consolidar este proyecto y que sirva eventualmente como modelo de instauración progresiva.

## 1.2. Interés por el área

Este proyecto cubre dos áreas principales: el desarrollo de páginas web y la medicina, por lo que se justificará el interés por ambas.

En el tercer curso del grado en Ingeniería Informática el autor cursó la asignatura de Sistemas Web, donde entre otras cosas aprendió las bases sobre el desarrollo de páginas web, y donde desarrolló un pequeño proyecto de creación de una de ellas. Hasta el momento, fue la asignatura que más interés y curiosidad le produjo, y estando cerca de acabar el grado, manifiesta que ha sido la que más le gustó, por lo que la idea de hacer un trabajo de fin de grado en torno a esta asignatura ha estado presente en él desde entonces.

En lo que se refiere a la parte médica de esta idea, en la familia del autor siempre ha habido un ambiente que incitaba al interés por esta área, ya que su padre es médico y desde pequeño ha estado rodeado de grabaciones de operaciones y llamadas telefónicas de casos médicos. Además, desde hace dos años, tanto su hermana como su hermano han cursado los estudios de medicina, por lo que ha sido imposible perder el interés que tenía en torno a esta.

Teniendo en cuenta esto, cuando surgió la posibilidad de hacer el trabajo de fin de grado combinando estos dos temas, al autor le fue imposible rechazar la idea y se empezó directamente a analizar, diseñar y desarrollar lo que ha acabado siendo este trabajo.

## 1.3. Antecedentes del proyecto

Antes de empezar explicando los antecedentes que han llevado al desarrollo de este trabajo, es preciso aclarar que el proyecto le fue propuesto por Ismael Díez del Val, coordinador de la Sección de cirugía esófago-gástrica de la AEC y cliente *stakeholder* de este proyecto. La idea de formar este registro forma parte de los objetivos de la sección durante el periodo 2019-2020.

El uso de Registros Nacionales de actividad es una práctica habitual en varios países europeos [1-3], donde los y las cirujanas tienen la obligación no sólo de coordinar la intervención con otras especialidades, acortar demoras y responder adecuadamente a las complicaciones, sino de recoger datos, compararlos con otros equipos y evaluarlos periódicamente.



En el Estado español, esta práctica ha sido desarrollada en algunos casos a nivel autonómico, como en la Comunidad Valenciana, o desde iniciativas privadas, como el Registro Eurecca, adaptado desde su versión europea [4, 5], extensiones de experiencias europeas que han ido sumando de forma progresiva a varias CCAA (Cataluña, Navarra, Euskadi, La Rioja y, desde enero de 2020, Castilla-la Mancha), pero que se encuentran aún lejos de representar al conjunto de la población.

La Sección de Cirugía esófago-gástrica de la AEC, desde la creación de su Junta actual en enero de 2019, se ha planteado como objetivo la creación de un REgistro NAcional de Cirugía Esofágica (RENACE), que incluya al conjunto de pacientes intervenidos por cáncer de esófago. Esta patología tiene varias “ventajas” para servir como modelo de registro y poder abrir en el futuro las puertas a otras patologías: baja incidencia, equipos dedicados específicamente a ella, varias especialidades implicadas, pocos hospitales e importante interacción entre los profesionales. Además de la “facilidad” de registrar por los pocos casos (apenas 2 hospitales en España intervienen 30 o más pacientes al año), existe el claro interés de conocer los resultados, dado que se admite un porcentaje de complicaciones alto incluso en entornos de alto volumen de actividad (hasta el 50%).



# 2.

## Planificación del proyecto

En este capítulo se tratan, como bien dice el propio título, los apartados relacionados con la planificación llevada a cabo antes de comenzar a desarrollar el registro. Es decir, la definición del trabajo a realizar, la estructura de éste, de qué manera se va a realizar, prever cuáles son los peligros que podrían sorprendernos, cómo se va a asegurar que el producto a realizar sea de calidad, etc.

### 2.1. Objetivos

A continuación, se describen los objetivos del trabajo, divididos en tres apartados, dependiendo quién es el beneficiado del cumplimiento de estos objetivos.

El objetivo principal es la creación de un REgistro NAcional de Cirugía Esofágica (RENACE) que permita la recopilación de datos para su posterior análisis y, de esta manera, poder facilitar la toma de decisiones y conclusiones para la mejora de este tipo de cirugía.

#### 2.1.1. Objetivos del alumno

El objetivo del alumno es poner en práctica las herramientas disponibles para la realización óptima de una base de datos y su interfaz correspondiente que refleje la realidad, permita interpretarla y mejorarla en el futuro, logrando la satisfacción del interesado como de los usuarios que hagan uso del registro creado.

Además, como objetivo más personal, se puede remarcar el hecho de llevar a cabo un proyecto con influencia en la vida real, teniendo que planificar todo el progreso, analizar las especificaciones y objetivos del registro, diseñar la solución para cumplirlas en la mayor medida de lo posible, y por último desarrollar una solución acorde a todo lo planeado anteriormente.

Esto supone también un reto personal, ya que se aleja de los trabajos realizados durante los cuatro años de universidad y da un paso hacia la vida laboral que le espera en un futuro no tan lejano.

#### 2.1.2. Objetivos del trabajo

En este apartado se especificarán los objetivos que se han establecido para asegurar el buen funcionamiento del registro.

##### 2.1.2.1. Requisitos mínimos

Los requisitos mínimos previstos son los siguientes:

- Importación de pacientes desde el registro Eureka.
- Añadir, editar y visualizar pacientes.
- Creación de informes basados en los datos almacenados.
- Visualización de los datos de manera gráfica.

- Anonimidad de los datos.

#### 2.1.2.2. *Objetivos opcionales*

Si el tiempo de desarrollo del TFG así lo permitiera, sería interesante realizar:

- Preparación del software desarrollado para su extensión a otras secciones de la AEC.

#### 2.1.3. *Objetivos del interesado*

Por último, estos son los objetivos que el interesado desea cumplir con la creación de este registro.

En primer lugar, y como principal objetivo, dotar a la AEC de un instrumento que le permita conocer los resultados de la cirugía del cáncer de esófago en nuestro país.

Además, una vez habiendo cumplido con el objetivo principal:

- Elaboración de propuestas globales respecto al tratamiento del cáncer de esófago basándose en la explotación de los resultados.
- Extensión del registro RENACE a otras secciones de la AEC.
- Preparación del camino hacia la acreditación de Unidades.

## 2.2. Alcance

En este punto se especificará cuál es el alcance que tendrá el registro, acordado entre el interesado y el desarrollador.

Este registro será integrado en la AEC como precursor de futuros registros en otras secciones, por lo que el desarrollo no será un mero prototipo y su implementación y funcionalidades deben estar a la altura de un registro que se usará a nivel estatal por diversos hospitales.

La aplicación deberá ser fácil de usar para cualquier persona, ya que está dirigido a cirujanos con diferentes niveles de conocimientos tecnológicos.

Por último, se deben cumplir todos los requisitos mínimos acordados y en la medida de lo posible los requisitos optativos. En caso de no poder cumplir estos últimos, sería de gran ayuda para el futuro de la aplicación que la implementación se llevase a cabo de manera que facilite en la medida de lo posible el cumplimiento de los objetivos opcionales.

## 2.3. Interesados

En este apartado se nombrarán los principales interesados de que este trabajo de fin de grado salga adelante y se lleve a cabo cumpliendo todos los objetivos que se han establecido.

El interesado más directo es el Dr. Díez del Val, coordinador de la Sección de cirugía esófago-gástrica de la AEC, que fue el que desarrolló esta idea y propuso el proyecto al alumno en cuestión.

Además, como interesados indirectos se encuentra la AEC misma, que actualmente no cuenta con un registro como el que se está desarrollando en este trabajo, por lo que le servirá para poder analizar los datos y poder tomar decisiones con una buena perspectiva de los datos. Y, por otro lado, los médicos y hospitales que formarán parte de este registro, ya que el objetivo

de éste es mejorar en la medida de lo posible las prácticas actuales en cuanto a cirugía se refiere, y como profesionales de esta sección, también es de su interés el progreso de la misma.

Por otro lado, cabe destacar a la tutora del trabajo de fin de grado, siendo una de las principales interesadas en que este desarrollo se complete de manera satisfactoria, ya que, al aceptar la propuesta del alumno, se convierte en un punto de apoyo para la toma de decisiones y una pieza clave a la hora encaminar al alumno al correcto desarrollo del trabajo a realizar.

Por último, el principal interesado en este proyecto es el propio alumno, quien se encargará de la gran parte del análisis, diseño e implementación del registro, y a quien afectará directamente la calidad final del producto a crear, ya que el desarrollo de este trabajo será el último hito a completar para terminar el grado en ingeniería informática.

## 2.4. Metodología de trabajo

En este apartado se especificará cuál ha sido la metodología de trabajo utilizada para el desarrollo del registro, cuáles han sido las razones de su elección, las ventajas que ha aportado y cómo se ha estructurado el trabajo para seguir esta metodología.

### 2.4.1. Metodología Agile

La metodología empleada ha sido Agile [6], que como se puede intuir por su nombre, entra dentro del grupo que engloba las tecnologías ágiles. Como la mayoría de metodologías, está enfocada al desarrollo del proyecto por parte de una empresa, con sus diferentes grupos de trabajos y sus múltiples desarrolladores, pero como en este trabajo solo se cuenta con un interesado, una tutora y un alumno que ejerce de desarrollador, se han acomodado las bases a las características de este trabajo. Además, como explicaremos a continuación, es una tecnología que seguirá siendo muy útil y encajará a la perfección en el caso de que este proyecto se expanda en un futuro y sea necesario seguir trabajando en la extensión e implementación de este registro.

### 2.4.2. Características

El objetivo principal de las metodologías ágiles es entregar los productos o servicios con una mayor calidad y con unos costes y tiempos más reducidos. Estas permiten adaptar la forma de trabajo a las exigencias del proyecto, consiguiendo con esta flexibilidad mayor eficacia y productividad.

A continuación, se enumeran las ventajas más destacables de este conjunto de metodologías:

- **Mejoran la satisfacción del cliente:** dado que lo involucran y lo comprometen a lo largo de todo el proyecto. En cada etapa se informará al cliente del progreso de esta, permitiéndole aportar su conocimiento con el fin de mejorar las características del producto final.
- **Mejora la implicación del equipo:** al estar todos los miembros que forman parte del proyecto involucrados, los compromisos son negociados y aceptados por todos.
- **Ahorra tiempo y costes:** por el hecho de ser una metodología ágil, se trabaja de manera más eficiente y se cumplen de manera estrictas los plazos pactados.
- **División del trabajo en entregas:** una de las bases de la aplicación de esta metodología es que se trabaja a través de entregas parciales del producto, lo que hace que en un intervalo mucho más corto se pueda entregar una versión del producto funcional.

Además, gracias a estar entregando parciales y del seguimiento cercano que permiten, se podrá eliminar cualquier característica que se vea innecesaria por parte del cliente.

- **Mayor control de problemas:** unido a la característica anterior, al tratarse de entregas parciales y al tener siempre la aportación de los demás miembros, es posible alertar de forma rápida sobre los errores o problemas que surjan, y buscar una solución o alternativa de manera inmediata.

### 2.4.3. Aplicación en el trabajo

Ahora se explicará qué características de nuestro trabajo hacen que esta metodología sea útil durante el desarrollo del producto y mejore su calidad final.

Como se ha explicado en una de las características listadas en el apartado anterior, en esta metodología se divide el trabajo en pequeñas entregas. Aplicando este principio al trabajo a realizar, se creará un plazo de entrega por cada objetivo a realizar. Esto permitirá ir probando la calidad y el funcionamiento de cada objetivo realizado, permitiendo corregirlos al momento y de esta manera dejarlos preparados para la posible publicación del registro. Además, el crear productos funcionales por cada entrega, es decir, por cada objetivo, dará más margen de tiempo para que los usuarios ingresen sus pacientes en el registro mientras se sigue trabajando en otros aspectos del proyecto, sin tener que esperar a que el trabajo esté totalmente terminado.

Otro aspecto a tener en cuenta es el hecho de que al ser un grupo de trabajo tan reducido y con tanta cercanía entre el desarrollador y el interesado, la implicación del equipo y el constante contacto para conocer el estado del desarrollo va a estar asegurada. Esto ayudará a que el interesado pueda dar su continua opinión sobre el trabajo que se está realizando aportando su previo conocimiento y experiencia en registros médicos.

## 2.5. Estructura de descomposición del trabajo (EDT)

Una EDT es una herramienta fundamental que consiste en la descomposición jerárquica, orientada al entregable, del trabajo a ser ejecutado por el equipo de proyecto, para cumplir con los objetivos de éste y crear los entregables requeridos, donde cada nivel descendente de la EDT representa una definición con un detalle incrementado del trabajo del proyecto.

El propósito de una EDT es organizar y definir el alcance total y sirve como base para la planificación del proyecto.

En concreto, la Figura 1 muestra el EDT con los paquetes de trabajos identificados que forman parte de este proyecto. A continuación, se detallan cada una de las tareas, con sus respectivas horas estimadas.

### Gestión:

- **Presentación del trabajo y definición de objetivos:** Primer entregable presentado por el interesado, donde se explica la idea principal del trabajo a desarrollar. Basándose en este documento se han fijado los objetivos que se van a llevar a cabo durante el proyecto, tanto los imprescindibles, que serán necesarios para completar el trabajo, como los complementarios, que serán los que añadan un punto de calidad al trabajo realizado.

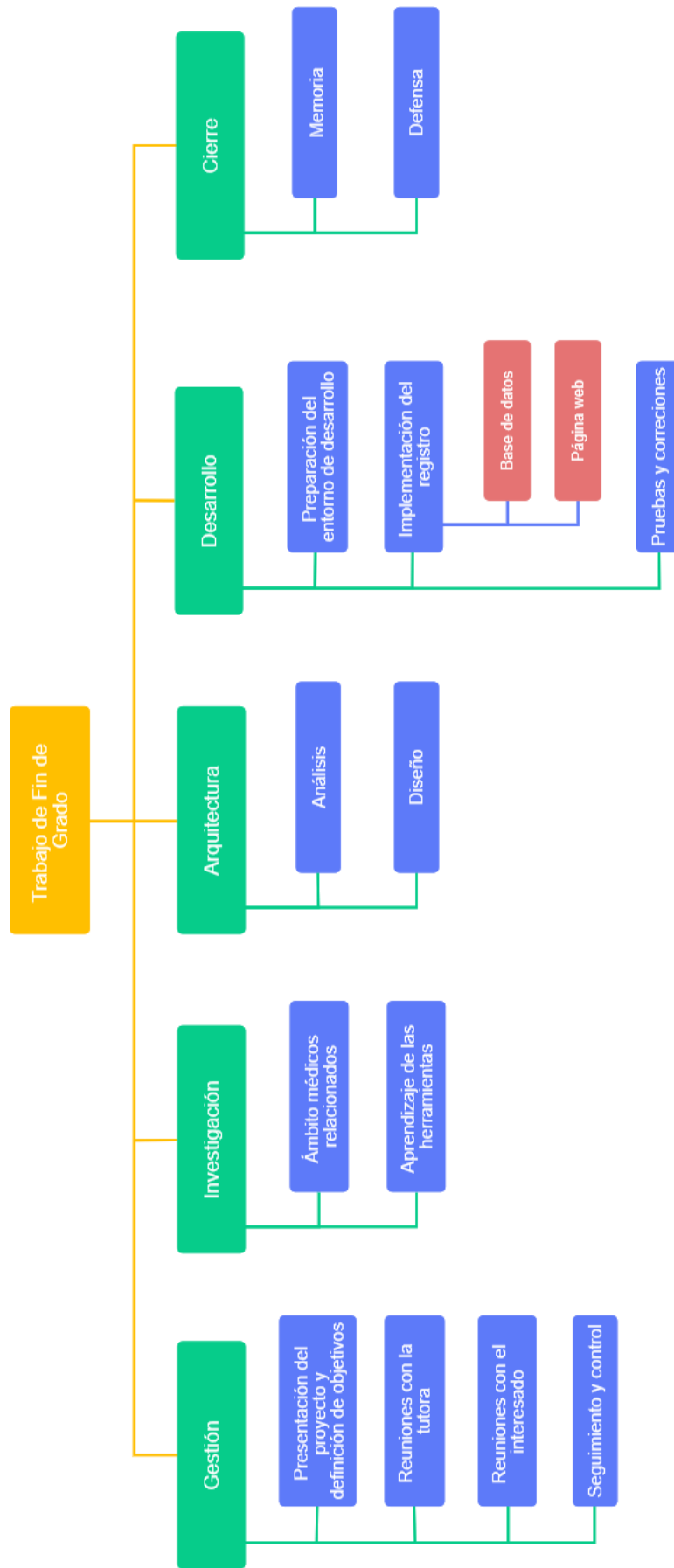


Figura 1- Diagrama EDT

- **Reuniones con la tutora:** A pesar de que la dirección que tomará el proyecto se enfoque más en las decisiones tomadas entre el alumno y el interesado, la participación de la tutora ha sido y será **imprescindible** para el correcto avance del trabajo. Se realizarán reuniones eventualmente para solucionar dudas en el diseño y de la implementación y para discutir diferentes aspectos del trabajo.
- **Reuniones con el interesado:** Se han realizado reuniones con el interesado para definir los aspectos esenciales del trabajo y decidir el orden y proceso de desarrollo. Además, a medida que el proyecto avance se espera realizar más reuniones para especificar y discutir decisiones entorno a la implementación y al funcionamiento del registro.
- **Seguimiento y control:** Realizar un seguimiento del progreso del trabajo, de principio a fin, de manera que se controle por parte de la tutora y del interesado que se cumplen los tiempos y objetivos establecidos.

#### Formación:

- **Ámbitos médicos relacionados:** Al ser un registro médico, hay muchos conceptos del ámbito de la medicina que se desconocen, por lo que se hará un estudio previo para mejorar la comprensión del registro y poder así tomar decisiones acordes a lo exigido.
- **Aprendizaje de las herramientas:** A pesar de que la mayoría de las herramientas se han utilizado anteriormente, habrá que dedicar tiempo a las nuevas tecnologías y herramientas que se utilizarán a lo largo del proyecto.

#### Arquitectura:

- **Análisis:** Se realizará un análisis de los objetivos y del plan establecido, analizando su viabilidad y los aspectos importantes de este trabajo.
- **Diseño:** Se hará un primer diseño de cómo se espera que sean los rasgos generales del registro. Lo más probable es que a lo largo de la implementación este diseño sufra pequeñas variaciones, por lo que este paquete estará presente en gran parte del desarrollo del trabajo.

#### Desarrollo:

- **Preparación del entorno de desarrollo:** Preparación de las herramientas que se van a utilizar en el desarrollo del proyecto, como GitHub o Xampp entre otras. Además, será necesario configurar el entorno local, el de la web, y la conexión entre ambas.
- **Implementación del registro:** la implementación de este registro se puede dividir en dos grandes apartados, la base de datos que almacenará los datos y la página web que se mostrará a los usuarios.
  - **Base de datos:** elaboración de una base de datos que responda de la manera más eficiente posible a los objetivos establecidos y a las necesidades que se presenten durante el proyecto.
  - **Página web:** implementación de la interfaz con la que interactuará el usuario y de la lógica detrás de esta, que se encargará del manejo de los datos. Este paquete se dividirá en cuatro más específicos.
    - **Entrega parcial 1:** implementación de la interfaz base de la página web, junto con el desarrollo de la primera sección del registro, donde se mostrarán los datos generales y los hospitales participantes.
    - **Entrega parcial 2:** implementación del primer objetivo, Inserción y edición de los pacientes.<sup>22</sup>



- **Entrega parcial 3:** implementación del segundo objetivo, Importación desde Eureka.
- **Entrega parcial 4:** implementación del tercer objetivo, Creación de informes.
- **Entrega parcial 5:** implementación del cuarto y último objetivo, Visualización de gráficos.
- **Pruebas y correcciones:** realización de pruebas con los primeros usuarios, de manera que se reciba un *feedback* con el fin de hacer pequeñas correcciones que respondan a los errores o desajustes recibidos.

**Cierre:**

- **Memoria:** se recogerá en un documento el trabajo realizado en diferentes funcionalidades y la documentación generada para el desarrollo del trabajo de fin de grado.
- **Póster:** se hará un póster que resuma el contenido del TFG y sirva como presentación del mismo.
- **Defensa:** realización y preparación de la defensa del proyecto a presentar frente al tribunal.

La Tabla 1 muestra de forma agrupada todas y cada una de las tareas principales identificadas junto con las respectivas horas estimadas que requerirá el desarrollo de este TFG.

*Tabla 1- Planificación de tareas con horas estimadas*

<b>TAREAS</b>	<b>DURACIÓN ESTIMADA</b>
<b>Gestión</b>	<b>30</b>
Reuniones con la tutora	10
Reuniones con el interesado	10
Definición de los objetivos	5
Seguimiento y control	5
<b>Formación</b>	<b>12</b>
Ámbitos médicos relacionados	2
Aprendizaje de las herramientas	10
<b>Arquitectura</b>	<b>25</b>
Análisis	5
Diseño	20
<b>Desarrollo</b>	<b>180</b>
Preparación del entorno de desarrollo	5
Implementación del registro	160
Base de datos	15
Página web	125
Entrega parcial 1	20
Entrega parcial 2	50
Entrega parcial 3	30
Entrega parcial 4	10
Entrega parcial 5	15
Pruebas y correcciones	15
<b>Cierre</b>	<b>120</b>
Memoria	100
Póster	5
Defensa	20
<b>En total (TFG):</b>	<b>352</b>

## 2.6. Diagrama de Gantt

En este apartado se presentará el diagrama [7] acordado para el desarrollo del registro, donde se mostrarán las fechas límite de los eventos importantes y los tiempos y plazos establecidos al desarrollo de cada parte del trabajo.

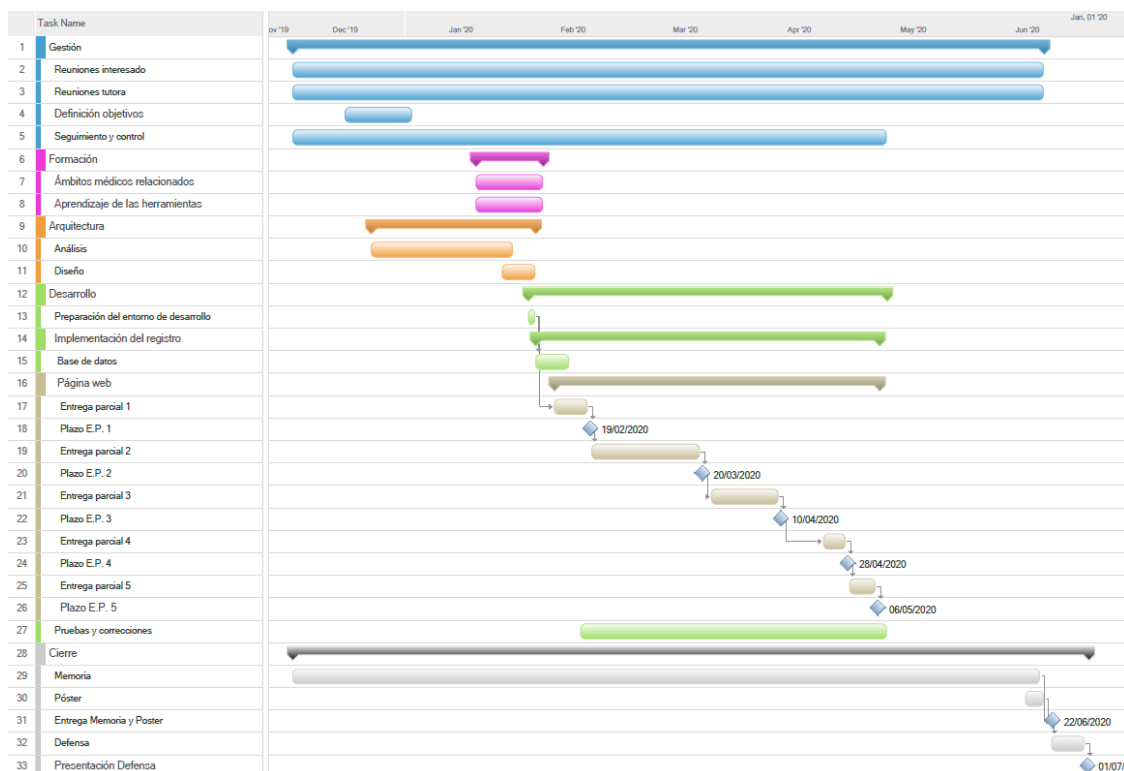


Figura 2- Diagrama Gantt

Para obtener una versión más detallada del diagrama Gantt véase en Anexos la Figura 51.

## 2.7. Comunicación y sistemas de información

Para facilitar la fluidez del intercambio de información entre las diferentes personas que forman el grupo de trabajo, es importante definir que canales de comunicación se utilizarán.

La comunicación alumno-tutora se realizará a través del mail y de tutorías. Esta primera se utilizará para dudas puntuales, de solución sencilla y directa, y para concretar las fechas de las reuniones. Las tutorías se utilizarán para tratar problemas que requieran un mayor desarrollo y para tomar decisiones importantes en cuanto al diseño se refiere. Como se prevé escribir la memoria utilizando Word Office, se utilizará el sistema de revisión y control de cambios que aporta esta herramienta para llevar a cabo esta fase final del trabajo.

Por otro lado, la comunicación con el interesado en el producto se realizará a través de la aplicación WhatsApp, para dudas más puntuales, y del mail, para compartir los archivos necesarios. Además, para casos en los que sea necesaria una interacción más continua, se prevé realizar reuniones bien presenciales bien telemáticas mediante BlackBoard Collaborate o alguna otra plataforma similar.

## 2.8. Gestión de riesgos

En este apartado se explicarán cuáles son los riesgos que pueden influir en el correcto desarrollo del trabajo de fin de grado.

1. **Problemas personales:** desde problemas de salud, que requieran una pausa en el trabajo, hasta problemas anímicos que pueden surgir a lo largo del curso por presión y estrés en las épocas más difíciles pueden suponer un riesgo para los plazos previamente establecidos.
2. **Disponibilidad de la tutora y de los interesados:** ya que no es un trabajo totalmente individual, sino que se cuenta con la ayuda de una tutora y las peticiones de los interesados, hay que tener en cuenta que estas personas tienen más tareas en su vida laboral y no podrán prestarnos el 100% de su atención todo el tiempo.
3. **Mala planificación:** una mala planificación puede acarrear problemas con el cumplimiento de los plazos y conllevar a retrasos en las entregas.
4. **Cambios durante el desarrollo:** puede suceder que durante el desarrollo del trabajo haya cambios en las peticiones de los interesados o que ciertas cosas no puedan llevarse a cabo como se había planeado. Esto necesitaría una reorganización que supondría un retraso respecto a la planificación inicial.
5. **Falta de conocimiento sobre ciertas tecnologías o herramientas:** a pesar de que la mayoría de las herramientas a utilizar son conocidas, la poca experiencia con algunas de ellas puede ralentizar ciertas fases del desarrollo o incluso provocar errores que retrasen el trabajo.
6. **Pérdida de información:** puede suceder que por problemas con el equipo o con otros elementos externos se produzca la pérdida de información o de las copias de seguridad.

Una vez identificados que factores pueden suponer un riesgo para el cumplimiento de los plazos y para la finalización de este proyecto, a continuación, se explicará qué medidas se tomarán tanto para evitar como para solucionar los problemas anteriores.

1. **Problemas personales:** prever los problemas relacionados con la salud es imposible, por lo que se tendrá que tener esta posibilidad abierta y planificar los plazos de tal manera que se pueda estar unos días sin trabajar sin afectar muy gravemente a los plazos establecidos.
2. **Disponibilidad del director y de los interesados:** las reuniones se planificarán con la mayor antelación posible para que no se paralice el desarrollo a la espera de una decisión que tiene que ser tomada en una reunión. En el caso en el que una reunión sea muy urgente se podrá recurrir a las reuniones online, pero sin abusar de esta práctica ya que hay que respetar el horario de trabajo y la disponibilidad de la tutora y los interesados.
3. **Mala planificación:** en el proceso de planificación se buscará el mejor planteamiento posible con el fin de que el origen no sea este. Sin embargo, si se detecta un fallo en la planificación, habrá que dedicar tiempo a replanificar el desarrollo.
4. **Cambios durante el desarrollo:** para evitar este problema será necesario concretar de la mejor manera posible los objetivos del interesado, evitando cualquier duda a la hora de desarrollar el producto. Aun así, es inevitable que algunas cambien a lo largo que el trabajo avanza y va cogiendo forma, para prevenir que se haga trabajo en vano, se mantendrá un mínimo contacto continuo para asegurar que lo implementado está al nivel de lo establecido.

5. **Falta de conocimiento sobre ciertas tecnologías o herramientas:** el primer contacto con aplicaciones que no se han usado durante mucho tiempo o con las que no se está muy familiarizado siempre reduce la productividad, por lo que se tendrá que tener en cuenta que el comienzo del proyecto pueda ser más lento, y esto se tendrá que plasmar en la planificación.
6. **Pérdida de información:** este riesgo debería estar controlado gracias a la utilización de GitHub, donde se irán guardando las nuevas versiones del trabajo. Además, muchos programas tienen su propio sistema de recuperación de archivos en caso de que se suceda algo que no permita realizar el guardado o la copia de seguridad de los documentos. Sin embargo, si esto sucediese habría que volver atrás en el desarrollo y supondría un claro retraso respecto a lo planificado.

## 2.9. Recursos

En este apartado se nombrarán brevemente los recursos utilizados para llevar a cabo este trabajo.

Como recursos principales se han utilizado dos ordenadores; un ordenador fijo con el que se ha trabajado desde casa, y al ser este el más potente se ha intentado priorizar su uso en la medida de lo posible. Y por otro lado un ordenador portátil, que se ha utilizado en las reuniones o en los casos en los que el alumno tenía que viajar.

Para facilitar la utilización de estos dos ordenadores, GitHub ha sido imprescindible, gracias a la facilidad que ofrece para actualizar el código en el portátil momentos antes de su utilización.

En lo que ha software se refiere, los recursos más significativos que se han utilizado han sido Xampp, MSOffice y Visual Studio, por su previo conocimiento y porque cumplen con creces las necesidades para las tareas que se van a llevar a cabo.

## 2.10. Entregables

En esta sección se explicarán los entregables relacionados con el desarrollo y la gestión del producto.

### 2.10.1. Entregables relacionados con la gestión

Relacionado con la gestión solo hay un entregable que merezca la pena comentar. Se trata del documento desarrollado por el interesado para presentar el trabajo al alumno, donde se especifican los primeros plazos estimados y los objetivos a realizar en el trabajo. Este documento se revisó por parte de ambos y se definieron las fechas de referencia a seguir durante el desarrollo.

Una vez acordadas las fechas, se hizo una segunda versión dirigida a los usuarios que realizarán las pruebas para informarles de las fechas en las que se pediría su aportación y que pudieran así planificarse y pedir con antelación el certificado a los Comités Éticos de Investigación Clínica (CEIC), para poder así introducir sus pacientes en el registro siguiendo todas las normativas necesarias.

### 2.10.2. Entregables relacionados con el desarrollo

- **Memoria:** Este propio documento es un entregable imprescindible exigido por las bases del trabajo de fin de grado en la facultad de ingeniería informática de la UPV/EHU,

donde se recogen principalmente los aspectos relacionados con el análisis, gestión y desarrollo del proyecto. Fecha de entrega límite el 21 de junio del 2020.

- **Póster:** Junto con la memoria ya nombrada, también se exige la entrega de un póster, de diseño libre, que sirva para identificar y representar el trabajo de fin de grado. Fecha de entrega límite el 21 de junio del 2020.
- **Presentación:** Por último, se preparará una presentación que ayudará a la defensa del trabajo frente al tribunal. Fecha de entrega límite entre el 29 de junio de 2020 y el 10 de julio del 2020.

### 2.10.3. Otros entregables

Durante el desarrollo del registro se han creado tres entregables que no estaban previstos desde un principio.

- Por un lado, se tuvo que preparar una solicitud al CEIC (véase en Anexos la Figura 49- Informe favorable del CEImE a la realización de este proyecto), para que permitiese al registro el manejo de pacientes reales, y poder así contar con usuarios que añadiesen sus casos al registro. Este entregable ha tenido una gran importancia ya que ha permitido que el registro tratase desde el primer momento con datos reales, y acercarse así al verdadero objetivo de formar parte de la AEC y no quedarse tan solo en un prototipo.
- Por otro lado, una vez acabada la implementación, se preparó un pequeño tutorial donde se explicaban los pasos para la importación de datos desde el registro Eurecca. Éste fue entregado a los usuarios para que seguidamente realizaran las pruebas.
- Por último, una vez se publicó la primera versión del registro, donde se permitía la inserción y edición de los pacientes a la base de datos, el Hospital Jiménez Díaz se puso en contacto con el alumno con el fin de obtener el script utilizado para el cálculo de la variable *Índice de comorbilidad de Charlson* [8].  
Tras discutir esta decisión con el Dr. Díez del Val, se decidió crear un entregable donde se explicase la estructura del formulario y los scripts utilizados para el cálculo de esa variable. Este documento se le facilitó al interesado vía mail el 24 de mayo de 2020.

## 2.11. Plan de calidad

El análisis de la calidad de un proyecto y la planificación para poder asegurarla son muy importantes hoy en día, ya que el mercado está es muy competitivo y hay que asegurar que el producto creado sea igual o mejor que los ya existentes. A continuación, se presentan los criterios que nos ayudarán a medir la calidad del registro RENACE.

### 2.11.1. Certificadores de calidad

Hay muchas maneras de comprobar la calidad de un trabajo realizado, pero estos son los que hemos considerado más significativos teniendo en cuenta las características del trabajo.

En cuanto a la gestión, el cumplimiento de los plazos establecidos asegura que el trabajo se está desarrollando de manera correcta y que el tiempo dedicado a éste está siendo de provecho, es decir, de calidad, y que no se está malgastando en distracciones.

Por el lado del diseño del proyecto y de la arquitectura del registro, es importante que la interacción entre las diferentes partes que forman el proyecto, tanto por el lado técnico como humano, sea fácil y fluida. Ya que esto simplificará y agilizará el trabajo a realizar.

Por último, en cuanto a la implementación se refiere, el código debe ser claro y limpio, que esté documentado y que no haya restos de código de viejas versiones que ya no se utilizan. A su vez, debe facilitar la extensión del registro, ya que, como se ha comentado, se prevé que el registro pueda sufrir cambios en un futuro, y debe facilitar en la medida de lo posible la edición del mismo por parte de otro programador, ya que no se sabe cuál va a ser el destino de este registro. Y unido a esto, la calidad de la memoria asociada a este trabajo también facilitará la comprensión del código, por aquel que lo retome, por lo que es una parte que no se puede pasar por alto.

### 2.11.2. Control de calidad

Los controles de calidad son indispensables para asegurar que el producto en el que se está trabajando realmente tiene la calidad esperada, y la ayuda externa puede traer puntos de vista muy útiles en cuanto a correcciones se refiere. Para realizar estos controles se contará de la ayuda de dos criterios diferentes.

En primer lugar, a medida que se cumplen los objetivos establecidos, se dejará al cliente comprobar que el producto cumple las expectativas y se realizarán pequeños ajustes que puedan mejorar lo ya desarrollado.

En segundo lugar, una vez el producto esté terminado y pueda ser desplegado y probado por unos pocos futuros usuarios del registro, se les pasará un cuestionario donde expresen su grado de satisfacción con el producto, lo evalúen y presenten posibles mejoras para éste.

### 2.11.3. Aspectos éticos

La base de datos a utilizar está anonimizada.

El proyecto es respetuoso con la confidencialidad y cumple con las normas y protocolos respecto de la protección de datos de carácter personal de los hospitales que participan en el registro, adecuadamente avalados por los respectivos Comités de Ética y de Investigaciones Clínicas.

El estudio ha sido elaborado respetando los principios establecidos en la declaración de Helsinki (1964) última versión Fortaleza, Brasil 2013, en el Convenio del Consejo de Europa relativo a los derechos humanos y la biomedicina (1997), y en la normativa sobre investigación biomédica, protección de datos de carácter personal. Ley 14/2007 de Investigación Biomédica.

El estudio cumple lo establecido en el REGLAMENTO (UE) 2016/679 DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 27 de abril de 2016 relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos. No se cederán datos a terceros, salvo obligación legal. El paciente será informado de que tiene derecho de acceso, rectificación, supresión de sus datos, y la limitación u oposición a su tratamiento.

# 3.

## Análisis y diseño

En este capítulo se analizarán diferentes aspectos sobre los que se han tenido que tomar decisiones y se explicará la influencia de estas en el posterior diseño del registro.

### 3.1. Arquitectura de la aplicación

Al ser un proyecto para la AEC, el objetivo era y es que el registro esté integrado en la medida de lo posible dentro de la página web de la asociación. Para ello se contactó con la empresa responsable de la gestión y el mantenimiento de la página web de la AEC, iMedia Producciones, quien aportó todas las facilidades para llevar a cabo la idea principal.

Por un lado, se habilitó un hosting y se puso a disposición del alumno la dirección <https://registrorenace.aecirujanos.es>, que cuenta con la versión 7.1.29 de PHP.

Por otro lado, se permitió el acceso a un servidor con el que poder gestionar una base de datos MySQL, y se habilitó el acceso a mediante el gestor phpMyAdmin. La versión de éste es 5.5.62-Oubuntu0.14.04.1-log.

### 3.2. Interacción con el usuario

En este apartado se analizará el perfil de usuario que hará uso de este registro y cuáles son las circunstancias en las que utilizará la aplicación. De esta manera, podremos tomar decisiones a la hora de diseñar diferentes aspectos basándonos en el análisis previo.

#### 3.2.1. Perfil de usuario

El perfil de usuario que tendrá acceso a este registro RENACE se limitará a cirujanos, probablemente con poco conocimiento tecnológico, y, sobre todo, con intención de pasar el menor tiempo posible en él.

Teniendo esto en mente, la aplicación debe limitarse a mostrar la información necesaria para el usuario, sin sobrecargar la interfaz de datos inútiles, de manera que agilice y haga menos pesada la experiencia del cirujano. Además, los pasos a seguir para realizar cada acción deben ser claros y sin dejar lugar a dudas, para que el usuario tenga una experiencia fluida y sin errores que le hagan perder el tiempo.

Para la selección de los hospitales que han participado en el registro, se han seguido dos criterios: una cierta representatividad geográfica (hay hospitales de 7 comunidades autónomas diferentes) y un número similar de centro pertenecientes o no al registro Eurecca, para poder testar tanto la importación desde otra base de datos como la introducción directa de pacientes.

#### 3.2.2. Circunstancias de uso

Los momentos en los que el usuario haga uso de RENACE son muy limitados. Generalmente será desde el ordenador del hospital, cuando se vayan a insertar nuevos pacientes en el registro, y

puntualmente será desde el domicilio del usuario, cuando se quieran hacer cambios como ediciones o consultas de los informes o los gráficos.

En respuesta a esos casos en los que se utilizará el registro, se puede concluir que no será necesario implementar una interfaz *responsive* que se adecue a su uso en un teléfono móvil o en un dispositivo de tamaño reducido, como pudiera ser una Tablet. Este punto se explica más a fondo en el capítulo de implementación, en el apartado 6.2. Por otro lado, como probablemente el usuario inserte los pacientes en grupos, la interfaz debe ayudar a su uso repetitivo, sin hacer esta introducción pesada y tratando de responder siempre de la misma manera a acciones similares.

### 3.3. Modelo de datos

El modelo de datos de este registro se centra en la descripción de la información almacenada en la base de datos y del sistema de información utilizado para organizar los archivos utilizados.

#### 3.3.1. Base de datos

En este proyecto se ha utilizado una base de datos MySQL [9], principalmente porque es la que nos ha facilitado iMedia Producciones. A pesar de eso, es una base relacional la que se necesita para almacenar y organizar los datos de los pacientes de la presente aplicación, ya que se distribuirán en diferentes tablas.

A continuación, se muestra la estructura de la base de datos, organizadas en cinco grupos principales, datos de usuarios (cirujanos), datos de pacientes, datos de las variables simples, datos de las variables múltiples y datos para la importación desde Eurecca.

**Datos de los usuarios:** la tabla *Usuarios* (véase Figura 3- Tabla Usuarios) guardará la información de los usuarios que pueden acceder al registro. Los tres campos relevantes de esta tabla son; *User*, que almacenará el nombre de usuario; *Password*, que almacenará la contraseña encriptada; *Hospital*, que indicará a que hospital pertenece el usuario.

Como para iniciar sesión y acceder al registro se utilizará el nombre de usuario, éste será asignado como clave única, pero la clave primaria será el nombre del hospital, ya que de esta manera los pacientes podrán hacer referencia al hospital al que pertenecen, que es una información más visual que la del nombre de usuario.

Usuarios	
PK	Hospital
	User
	Password
	Ciudad

Figura 3- Tabla Usuarios (cirujanos) de Renace

**Datos de los pacientes:** la tabla *Pacientes* (véase Figura 4) almacenará todos los valores de las variables simples, además de otros datos como el código o la fecha de creación que nos ayudarán en la organización de los pacientes. Por otro lado, crearemos una tabla por cada variable múltiple, que tendrá el nombre de *pacientes\_{categoría}\_mult\_{variable}*, por ejemplo, *pacientes\_dp\_mult\_comorbilidad*. Estas tablas harán referencia al código de la tabla *Pacientes* y guardaran un valor booleano que dirá si la casilla ha sido marcada o no (nótese que las variables se visualizan como un conjunto de *checkbox*-es).



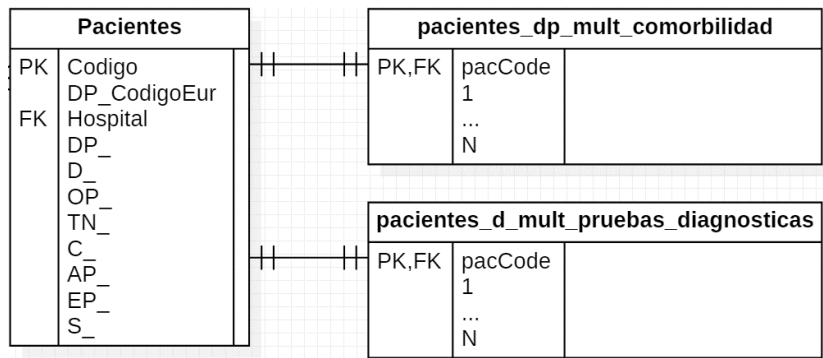


Figura 4- Tabla Pacientes

**Datos de las variables simples:** con el fin de simplificar el manejo de las variables a la hora de crear informes y generar gráficos, los datos que se almacenan en la tabla *Pacientes* se representan con números. Esto obliga a crear una tabla para cada variable (véase Figura 5) donde se especifica qué palabra está asociada a cada número identificativo. Esta palabra o frases, en casos puntuales, serán las que se mostrarán en el formulario. La estructura de estas tablas es muy sencilla e igual para todas. Constan de dos columnas: la primera, almacena el código numérico, y la segunda, la palabra o frase correspondiente. El nombre de estas tablas seguirá un patrón de manera que coincida con los nombres de las columnas de la tabla *pacientes* y que facilite su utilización en las consultas.

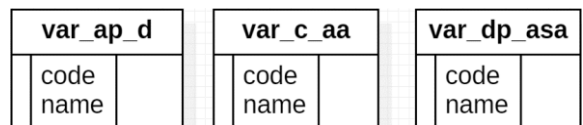


Figura 5- Tablas Variables simples

**Datos de variables múltiples:** al igual que las tablas que guardan los datos de las variables simples, se necesitan tablas para guardar la información correspondiente a las variables múltiples. Sin embargo, estas tablas son ligeramente diferentes, diseñadas para facilitar el proceso de importación desde el registro Eurecca.

Estas tablas están divididas en cuatro columnas (véase Figura 6). Las dos primeras, *code* y *name* cumplen la misma función que en las tablas de las variables simples, es decir, guardan la relación entre el código y la palabra que se mostrará en el formulario. La tercera columna, *val*, se ha implementado en todas las tablas de las variables múltiples, con el fin de mantener una misma estructura y facilitar su uso, pero hasta la fecha solo tiene valor en la tabla correspondiente a la comorbilidad. Por último, se tiene la columna *nameEur*, donde se almacena el nombre que tiene esa variable en la base de datos de Eurecca. Esta columna se ha añadido expresamente para facilitar el proceso de importación desde el registro Eurecca.

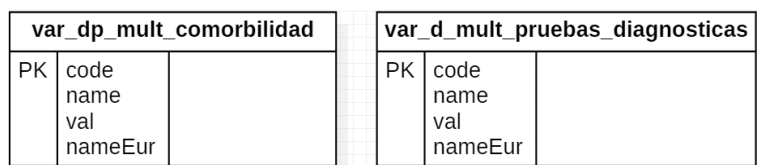


Figura 6- Tablas Variables múltiples

El último grupo de tablas, dedicadas a la importación de pacientes desde Eurecca, está formado por dos tablas: *eurecca\_import\_variables* y *eurecca\_import\_traduccion*.

La tabla correspondiente a las variables a importar (véase Figura 7 y Figura 8) ayudará a identificar qué datos de Eurecca se quieren importar, a qué variable de RENACE corresponden y qué método de traducción se utilizará, ya que habrá que aplicar ciertos cambios a los datos del archivo Excel para que encajen en la base de datos.

Esta tabla cuenta con cuatro columnas. La columna *orden* se utilizará como llave principal y permitirá mantener el orden correcto de las variables a importar, que coincidirá con el que se halla en el archivo Excel. De esta manera, se acelera el proceso de importación, ya que no se tendrá que recorrer toda la lista por cada variable a importar. La segunda columna, *name*, indicará cuál es el nombre de la variable en el archivo Excel de Eurecca. Esta ayudará a identificar si el dato ha de ser o no importado. La siguiente columna, *traducción*, guardará cuál es la columna de RENACE correspondiente a la variable de Eurecca, es decir, en qué columna se almacenará el dato importado. Por último, la columna *trad\_val* indicará qué cambios hay que aplicar a la variable importada para que coincida con el formato de almacenamiento. Estos cambios se indican mediante un número, que tienen el siguiente significado.

0. Estas variables no necesitan ser editadas. Se utilizará en casos de datos numéricos como cantidad de albúmina, hemoglobina o de ganglios extraídos.
1. Estas variables necesitarán una traducción que las convierta de frases a números. Esta conversión se hará basándose en los datos de la tabla *eurecca\_import\_traduccion*.
2. Estas variables contienen fechas, por lo que precisarán de la aplicación de un filtro que extraiga solo la parte deseada, es decir, que obtenga solo los datos del día, mes y año.
3. Caso especial para las variables a las que haya que quitarles el último carácter, como las variables que especifican la talla del paciente, que en Eurecca se exportan con el formato 1.80m.
4. Caso especial para las variables a las que haya que quitarles los dos últimos caracteres, como las variables que especifican el peso del paciente entre otras, que en Eurecca se exportan con el formato 80kg.
5. Este número indica que los datos pertenecen a una de las variables RENACE definidas como múltiples, por lo que recibirán un trato especial para importar la lista de palabras en tablas de variables múltiples de RENACE.

eurecca_import_variables	
PK	orden name traduccion trad_val

Figura 7- Tabla *eurecca\_import\_variables*

Por último, está la tabla correspondiente a la traducción de variables (véase Figura 8). La función de esta es relacionar todas las frases que se puedan encontrar en el registro Eurecca con el valor numérico que tendrá en la base de datos de RENACE. Para cumplir esta función solo se necesitan dos columnas: en la primera, *Eurecca*, se guardará la frase que se puede encontrar en el registro Eurecca y, en la segunda, *renace*, se guardará el valor numérico que le corresponde en el registro RENACE. Esta tabla se utilizará en el caso de traducción número 1, nombrado en la lista numérica anterior.

eurecca_import_traducciones		
PK	key	
	eurecca	
	renace	

Figura 8- Tabla eurecca\_import\_traducciones

### 3.3.2. Sistema de ficheros de la aplicación

Tan importante es mantener los datos bien estructurados como mantener un sistema de organización de los archivos con los que se está trabajando, para agilizar así su búsqueda y evitar la pérdida de información.

Por un lado, se han almacenado los archivos que tienen que ver con la memoria, como pueden ser los archivos utilizados para generar gráficos, las imágenes que insertaremos, los documentos del interesado y la memoria misma.

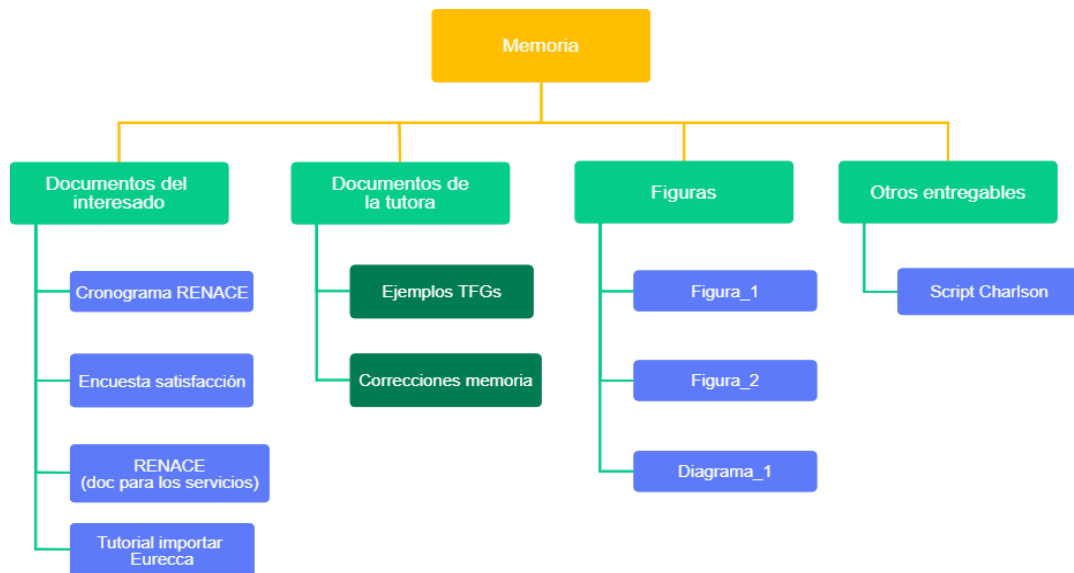


Figura 9- Sistema de ficheros de la memoria

Por otro lado, como se explicará más adelante, todos los archivos relacionados con la implementación del trabajo se han guardado dentro de una carpeta a la que se puede acceder a través de Xampp y de GitHub, para poder así probar el registro en el *localhost* y a su vez mantener el control de versiones. Dentro de esta carpeta los archivos se han organizado en subcarpetas dependiendo de su papel en el registro, como pueden ser los archivos JavaScript, los de estilos o los de PHP.

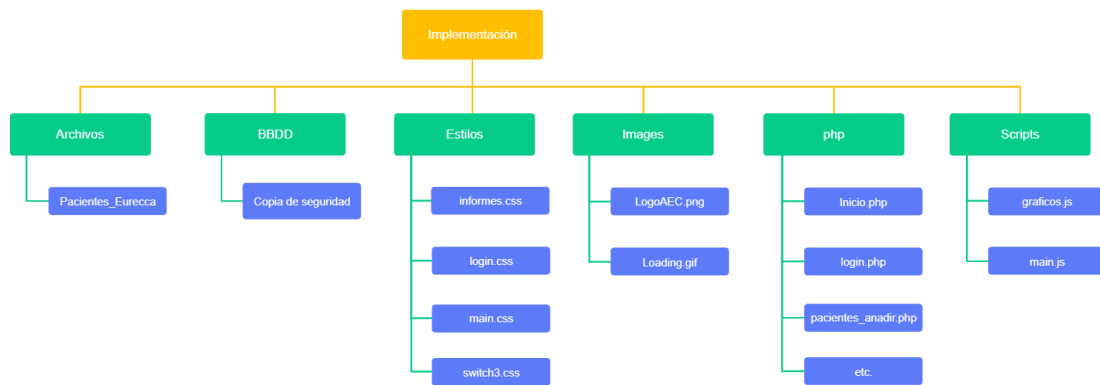


Figura 10- Sistema ficheros de la implementación

### 3.4. Objetivos del trabajo

En este apartado se extiende la información dada en el punto 2.1.2 para comprender mejor la funcionalidad de cada objetivo de este trabajo.

#### 3.4.1. Inserción y edición de los pacientes

Como base para que el registro funcione correctamente, la inserción de pacientes deberá ser sencilla y deberá prever la opción de que las variables médicas del formulario aumenten en un futuro. Además, ya que meses después de realizar la operación y de haber ingresado los datos en el registro, es posible que se le haga una revisión al paciente, los datos de este deberán poder modificarse.

#### 3.4.2. Importación desde Eurecca

Con el fin de no desperdiciar el trabajo realizado por los y las médicos al introducir los casos de sus pacientes en el registro Eurecca, se permitirá la importación de esos pacientes al registro RENACE.

#### 3.4.3. Creación de informes

Creación de informes para mostrar una vista rápida del estado de cada hospital confrontando sus datos con los generales de todos los hospitales del registro RENACE.

#### 3.4.4. Visualización de gráficos

Creación de dos tipos de gráficos, los que muestran datos por hospital y los que muestran los datos generales. En los gráficos pertenecientes al primer grupo se podrá comparar los datos de todos los hospitales, de manera anónima, donde el usuario solo sabrá qué datos le pertenecen. En el segundo grupo se generarán los gráficos que reúnen datos generales de los pacientes, sin importar de que hospital provengan.

Estos gráficos deberán actualizarse automáticamente a medida que se inserten pacientes en el registro o se modifiquen sus datos, de manera que siempre muestren el estado actual de la base de datos.

#### 3.4.5. Anonimidad de los datos

Se debe mantener la anonimidad de los datos que se muestren en las secciones de informes y gráficos, por lo que el usuario solo puede ser capaz de identificar los datos que le pertenecen y los de ningún hospital más. Asimismo, cada usuario solo puede tener acceso a sus pacientes, y se debe eliminar cualquier opción que dé acceso a los pacientes de otro hospital.

#### 3.4.6. Preparación del software desarrollado para su extensión a otras Secciones de la AEC

Tras confirmar el correcto funcionamiento del proyecto, habiendo hecho las correcciones que se vean pertinentes después de recibir el *feedback* por parte de los usuarios que probarán el registro, se prevé presentar el proyecto a la AEC con el fin de que esta muestre su interés por la extensión del registro a otras secciones de la AEC.



# 4.

## Casos de uso

En este apartado se describen los casos de uso de los tres actores que pueden tener interacción con el registro.

### 4.1. Usuario no registrado

En esta sección se explicará el caso de uso que puede ser llevado a cabo por un usuario no registrado.

#### 4.1.1. Solicitud de registro

Al ser un registro orientado a un público muy específico, no se permite la opción de registrarse como se haría en una aplicación común. Por ello, en caso de que un o una médico estuviese interesado en formar parte de este registro, tendría que contactar con la AEC para recibir tal permiso. Si su solicitud es aceptada, el *data manager* crearía su usuario y dotaría al nuevo usuario de sus datos de acceso.

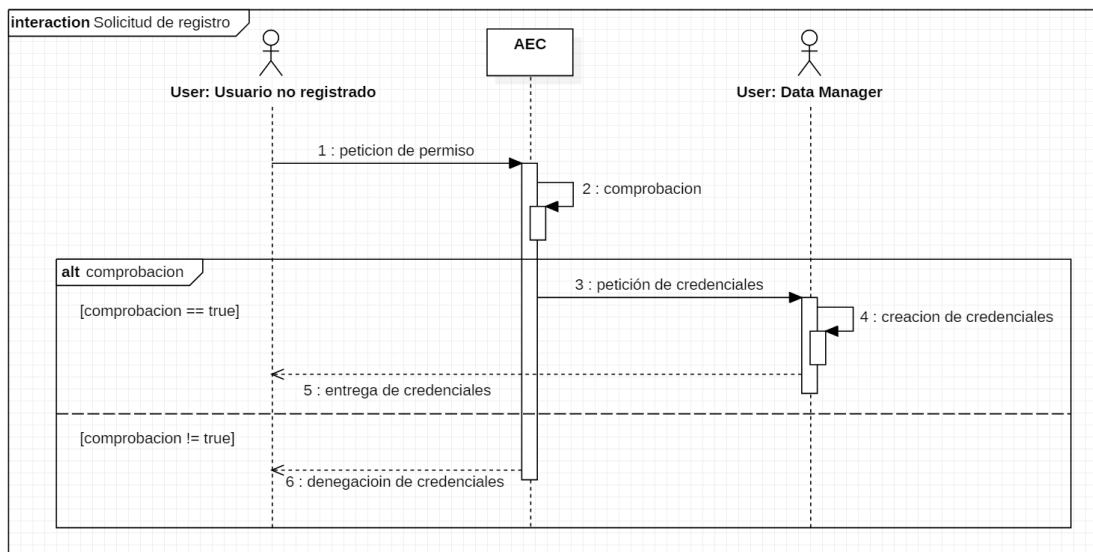


Figura 11- Solicitud de registro

### 4.2. Usuario registrado

Ahora se explicarán los diferentes casos de uso que pueden ser realizados por un usuario registrado.

#### 4.2.1. Inicio de sesión

El usuario registrado podrá introducir sus credenciales en la página de inicio de sesión. El sistema verificará si los datos son correctos, y en caso de que así sea, se le redirigirá a la página principal del registro.

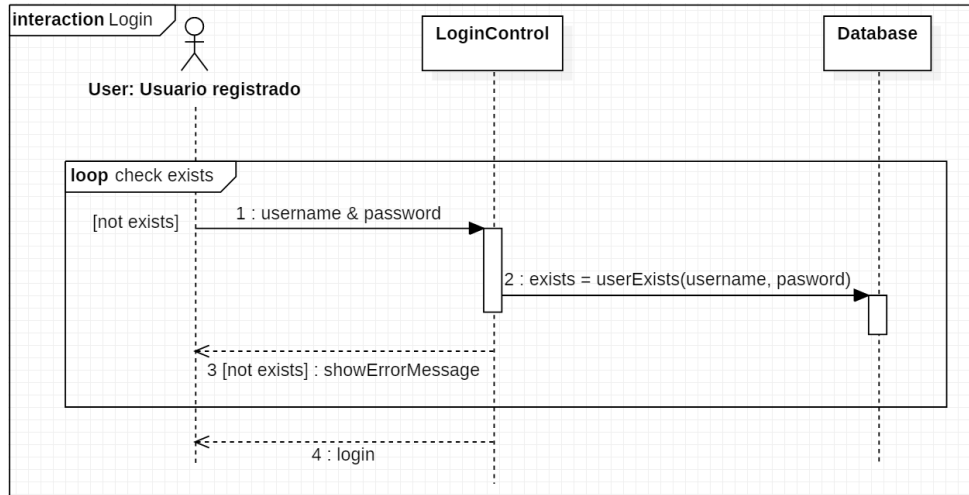


Figura 12- Login

#### 4.2.2. Añadir paciente

Este caso de uso puede ser llevado a cabo por cualquier usuario registrado. Tras acceder a la pestaña *Pacientes* deberá hacer *click* en el botón *Añadir pacientes* y será redirigido al formulario correspondiente. Una vez rellenado el formulario, el usuario enviará los datos y el sistema lo redirigirá de vuelta a la pestaña *Pacientes* mostrándole un mensaje de confirmación.

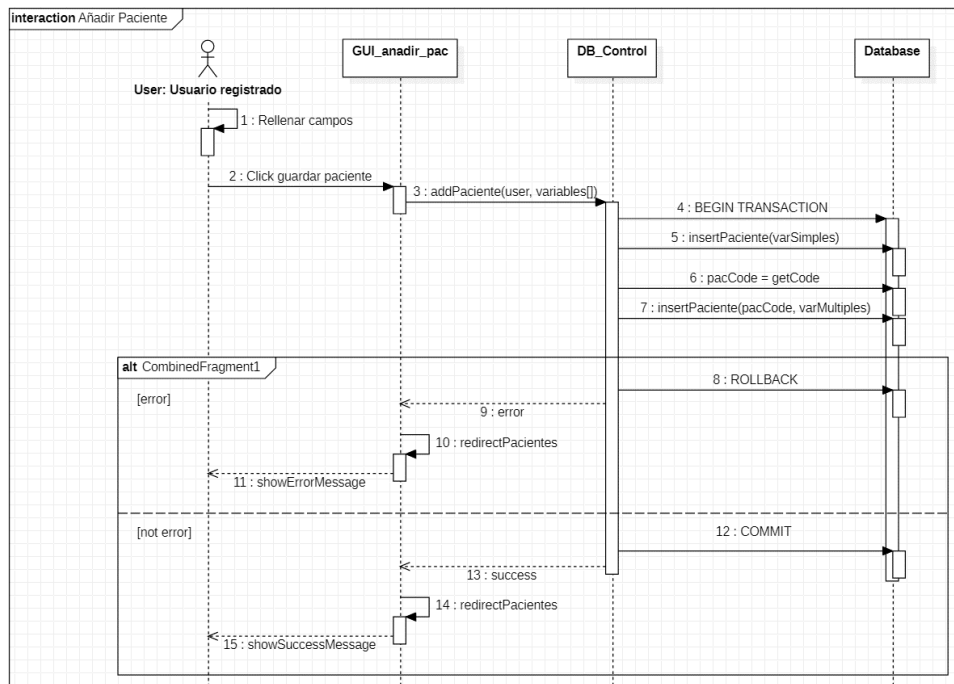


Figura 13- Añadir nuevo paciente



### 4.2.3. Visualización/Edición de los datos de un paciente

Un usuario podrá ver y editar los datos de los pacientes. Para ello debe acceder a la pestaña *Pacientes* y en esta el sistema le mostrará una tabla con el código y la fecha de creación de todos los pacientes que le pertenezcan.

Haciendo *click* en el código deseado podrá ver el formulario relleno del paciente. Aquí puede tanto ver los datos introducidos como editar o añadir nuevos datos. En caso realizar algún cambio, deberá enviar los datos haciendo *click* en el botón *Guardar cambios* que se encuentra al final de la página. Si los datos guardados no presentan ningún error, el sistema lo redirigirá a la pestaña *Pacientes* y le mostrará un mensaje confirmando que los datos han sido guardados correctamente.

Nótese que solo se muestran los códigos de los pacientes del médico usuario registrado.

En este punto, indicar que la edición de los datos del paciente permite que haya ya variables que almacenan datos relacionados con la evolución posterior a la intervención realizada.

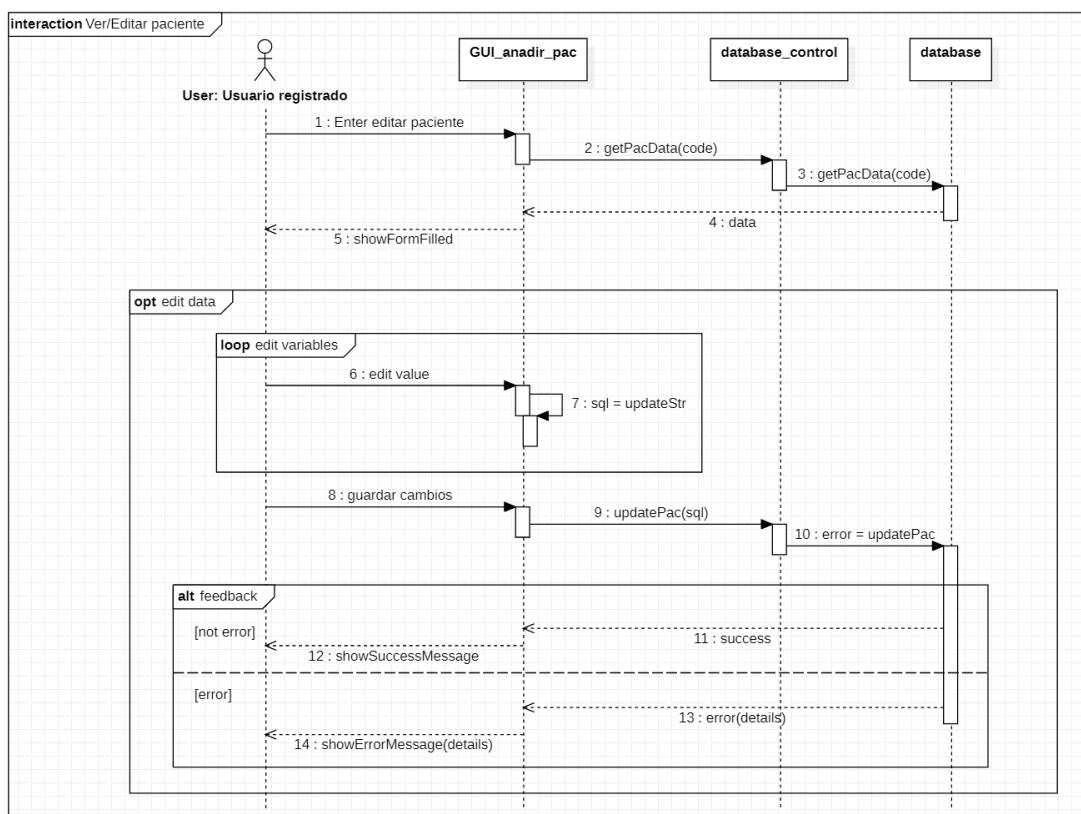


Figura 14- Ver/Editar paciente propio

### 4.2.4. Visualización de informes

Los usuarios tendrán la posibilidad de ver informes accediendo a la pestaña *Informes*, donde podrán acceder a un resumen de los datos y estadísticas más relevantes. Estos informes muestran a cada usuario los datos personales, que incluyen solo los pacientes pertenecientes a su hospital, y los datos generales, que muestran las estadísticas globales del registro.

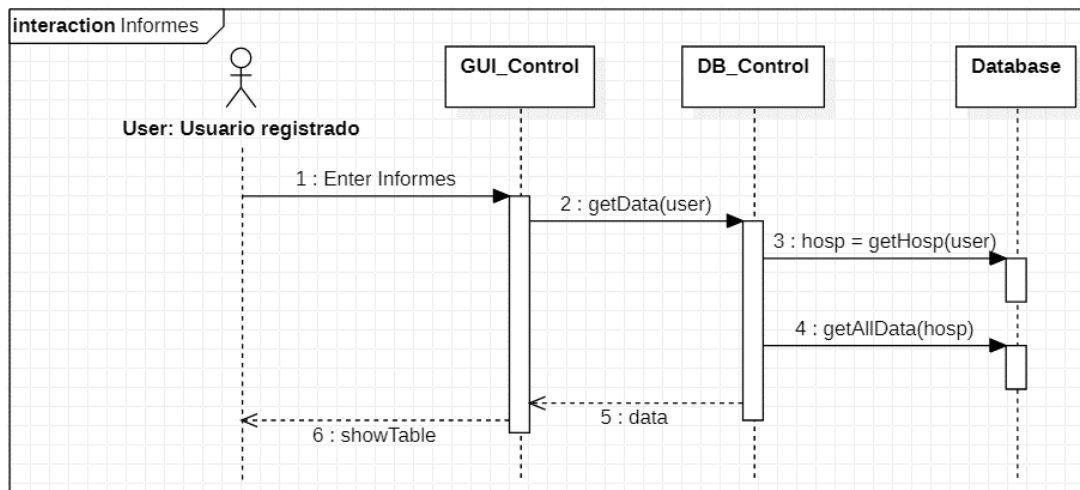


Figura 15- Informes

#### 4.2.5. Visualización de gráficos

Los usuarios tendrán la posibilidad de ver gráficos accediendo a la pestaña *Gráficos*. Aquí podrán ver de manera visual los datos pertenecientes a diferentes estadísticas. Estos gráficos están divididos en dos grupos: por un lado, se encuentran los que comparan los datos de todos los hospitales; y por otro lado se encuentran los que muestran, como en el caso de los informes, estadísticas basadas en la totalidad de los pacientes del registro.

Si el usuario quisiera ver un gráfico de manera más clara, podrá hacer *click* sobre el título del mismo y el gráfico se mostrará en pantalla completa.

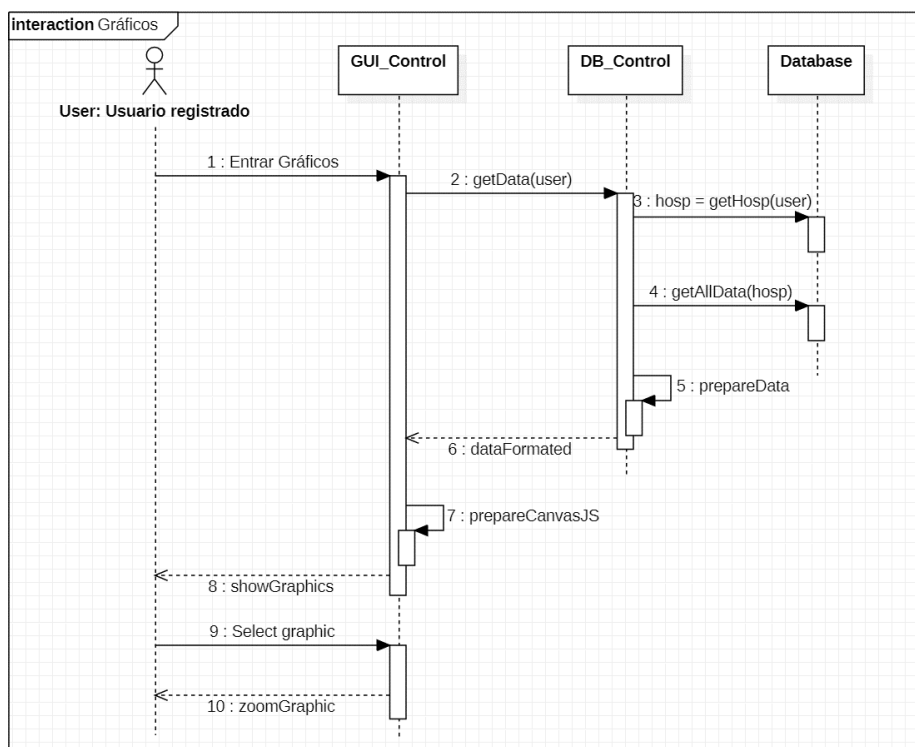


Figura 16- Gráficos

### 4.3. Data manager

#### 4.3.1. Creación de usuarios

Como se ha explicado anteriormente, los usuarios solo podrán ser creados a través del *data manager*. El nombre de usuario se creará siguiendo el patrón utilizado con los usuarios ya registrados. Para la creación de la contraseña se utilizará un script propio de creación de contraseñas encriptadas. En este se insertará la contraseña deseada y se recibirá de vuelta la contraseña cifrada. A continuación, se confirmará que la contraseña original y la cifrada coinciden y se procederá a insertar los datos del nuevo usuario en la base de datos.

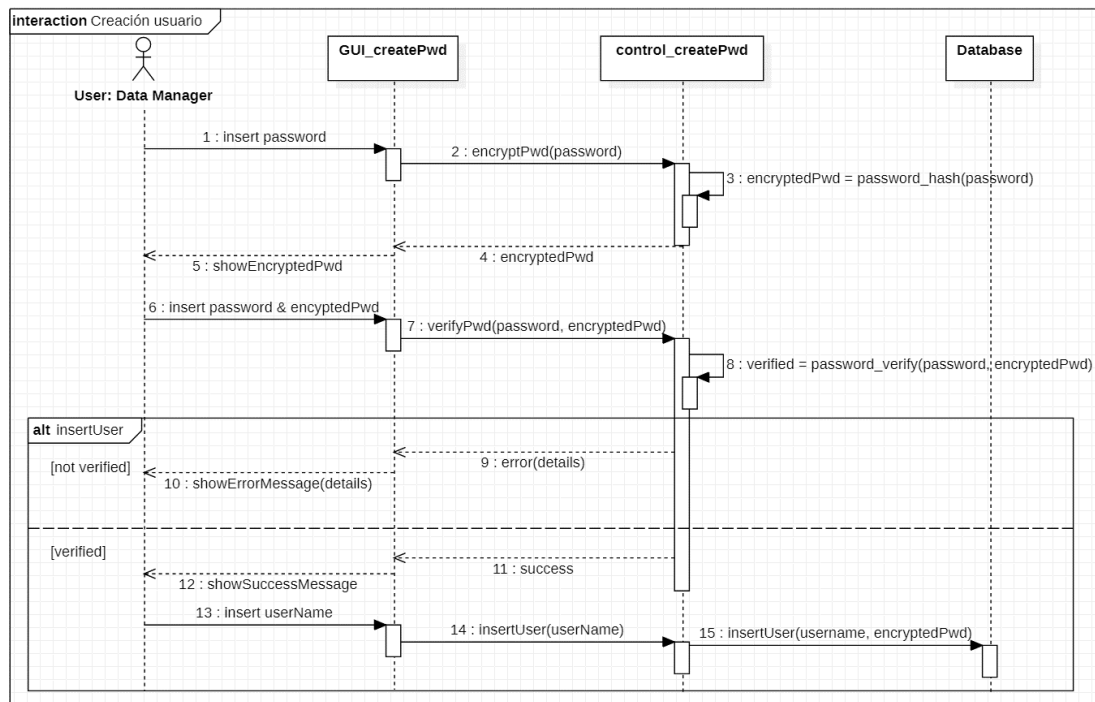


Figura 17- Crear usuario

#### 4.3.2. Borrado de pacientes

El borrado de pacientes no está al alcance de los usuarios con el fin de evitar el borrado involuntario o el borrado por motivos estadísticos. Por lo tanto, para borrar un paciente el usuario deberá informar al *data manager* de qué paciente quiere borrar y cuáles son los motivos. Si las razones son aceptables el *data manager* procederá a eliminar el paciente de la base de datos.

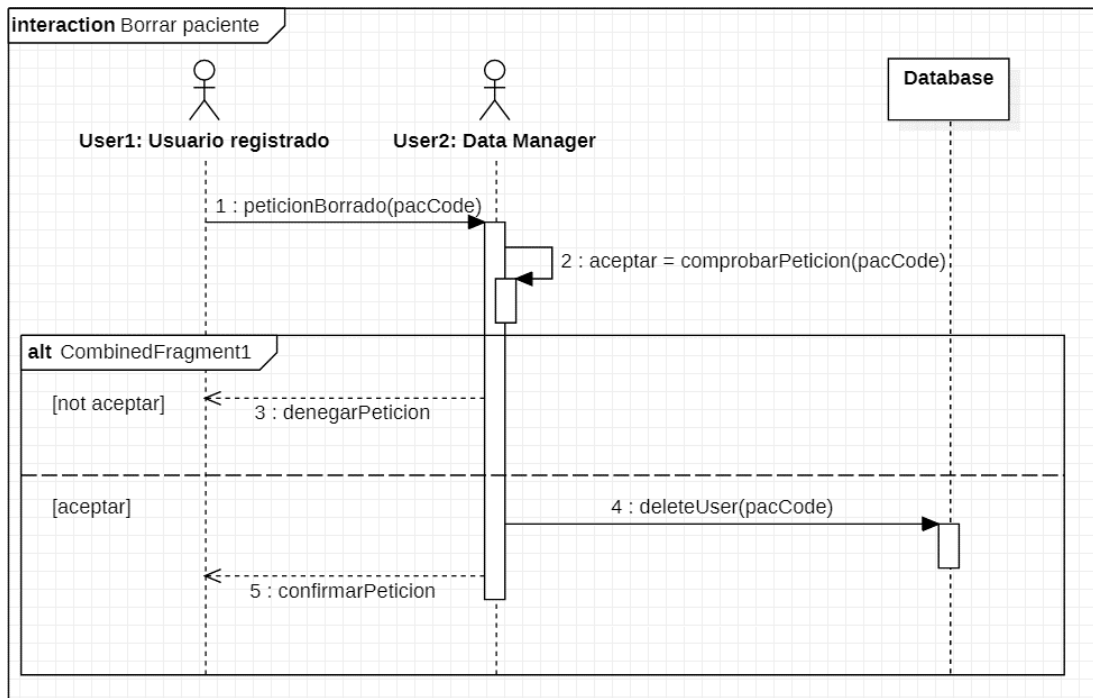


Figura 18- Borrado paciente

# 5.

## Tecnologías y entorno de desarrollo

En este capítulo se listarán las diferentes tecnologías que se han utilizado durante todo el proyecto. De esta manera se describirá el entorno de desarrollo en el que se ha desarrollado este registro.

### 5.1. Tecnologías

En este apartado se dará una pequeña introducción a cada una de las tecnologías utilizadas y se explicará cual ha sido su papel en este trabajo en concreto.

#### 5.1.1. HTML

Hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones. Define una estructura básica y un código para la definición de contenido de una página web, como texto, imágenes, videos o juegos, entre otros.

En este caso se ha utilizado como tecnología principal para la definición del contenido de la interfaz que muestra los datos de la base de datos.

#### 5.1.2. PHP

PHP, acrónimo recursivo en inglés de PHP: Hypertext Preprocessor (preprocesador de hipertexto), es un lenguaje de programación de código del lado del servidor que con el tiempo se ha aplicado al desarrollo web de contenido dinámico, dando un paso evolutivo en el concepto de aplicación en línea, por su carácter de servicio.

Gracias a ese manejo del lado del servidor, y la facilidad que aporta a la hora de desarrollar una página web dinámica, su utilización en el desarrollo de este registro ha sido enorme e imprescindible. Su principal uso ha sido el de conectar con la base de datos y permitir mostrar los datos en pantalla, gracias a su generador de texto plano reconocido por el estándar HTML.

#### 5.1.3. Javascript

JavaScript (JS) es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web.

Se ha utilizado únicamente en su forma del lado del cliente, gracias a que, implementado como parte de un navegador web, permite mejoras en la interfaz de usuario y páginas web dinámicas. De hecho, su principal papel ha sido el de crear interacciones entre diferentes elementos del formulario: controlar cuando tiene que aparecer o desaparecer un campo, calcular variables en el momento como el Índice de Masa Corporal (IMC) o el Índice de Comorbilidad de Charlson.

#### 5.1.4. JQuery

JQuery es una librería de JavaScript que facilita la manipulación de los documentos HTML, el control de eventos, las animaciones y las llamadas Ajax gracias a su API que funciona en la mayoría de los buscadores más utilizados hoy en día [10].

En nuestro se ha utilizado mayormente para simplificar y mejorar la legibilidad de las funciones de JavaScript que se encargan de manipular el documento y controlar los eventos que lanzan los diferentes elementos que forman la interfaz, ya que, como dice su eslogan *“write less, do more”*, esta librería disminuye en gran medida la longitud de los diferentes comandos de JavaScript. Además, para los casos en los que se ha hecho uso de Ajax la utilización de JQuery es prácticamente obligatoria, gracias a la gran variedad de opciones que aporta.

#### 5.1.5. MySQL

MySQL es una base de datos relacional habitualmente usada en páginas web que trabajan en conjunto con PHP [9]. Relacional hace referencia al hecho de que las diferentes tablas de la base de datos pues hacerse referencia unas a otras y las siglas SQL viene de las palabras en inglés *“Structured Query Language”* que es el lenguaje estándar utilizado para interactuar con la base de datos.

En este proyecto era necesaria la referencia entre ciertas tablas, por lo que era necesario el uso de MySQL. Además, gracias al amplio uso de MySQL, su integración con PHP es muy avanzada y facilita mucho la interacción entre ambas. Esto, unido al previo conocimiento y manejo de las bases de datos MySQL, han hecho indudable su elección para el desarrollo de este trabajo.

MySQL cuenta con diferentes maneras de interacción con PHP, pero los detalles del método elegido se especifican en el apartado de implementación.

## 5.2. Herramientas de software

En este apartado se explicarán las herramientas de software que han acompañado a la implementación a lo largo de todo el desarrollo. Como en el caso de las tecnologías, se hará una pequeña descripción de la utilidad de cada una y luego se definirá cual ha sido su papel en este trabajo en concreto.

### 5.2.1. GitHub Desktop

Git es un sistema libre y de código abierto de control de versiones. Basándose en este sistema se desarrolló GitHub, un servicio de servidores de repositorios Git, que, entre otras muchas facilidades para llevar a cabo las acciones más comunes, añade una interfaz gráfica a Git, que es una herramienta basada en la línea de comandos.

Basado en GitHub se desarrolló GitHub Desktop, una aplicación de escritorio, que por preferencias del desarrollador de este trabajo ha sido el modo elegido para llevar el control de versiones. Las principales ventajas en las que se ha basado esta decisión son las siguientes: olvidarse completamente de la línea de comandos, ya que todas las acciones se realizan a través de la interfaz; el hecho de que sea una aplicación y no una página web; y por último, las experiencias no del todo satisfactorias del alumno con las versiones GitHub en la web.

### 5.2.2. Visual Studio Code

Visual Studio Code es un potente editor de código fuente desarrollada por Microsoft. Ofrece soporte para depuración, control integrado de Git, resaltado de sintaxis y finalización inteligente de código, entre otras funciones. Además, contiene una inmensa cantidad de extensiones que facilitan el uso de otros lenguajes.

Han sido dos las principales razones por las que se ha elegido este editor para desarrollar el trabajo; el control integrado de Git, que, aunque no se haya hecho uso de todas sus ventajas, se ha utilizado para llevar un mejor control de que archivos están siendo editados y cuales ya están guardados en el repositorio. Para las demás funciones se ha utilizado el anteriormente nombrado GitHub Desktop; y, por otro lado, gracias a la variedad de extensiones, permite con solo un editor trabajar cómodamente con todos los tipos de archivos que se han tratado, como son: .php, .css, .js y .html.

#### Extensiones utilizadas:

Aquí se nombrarán y se explicará brevemente las extensiones utilizadas para mejorar la experiencia de Visual Studio Code.

- **HTML CSS Support:** ya que Visual Studio Code por sí solo no ofrece soporte para los archivos .css, se ha hecho uso de esta extensión para tener un soporte que estructure estos archivos de manera correcta y que aporte la finalización inteligente de código.
- **PHP Intelephense:** extensión para los archivos PHP que ofrece todas las funcionalidades esenciales a la hora de programar con este lenguaje. Además, ofrece una conexión con el manual oficial de PHP, que agiliza la búsqueda de información para las funciones desconocidas.

### 5.2.3. PhpMyAdmin

phpMyAdmin es una herramienta de software libre que ayuda al manejo de la administración de MySQL. Su interfaz permite realizar las operaciones más comunes como crear tablas, columnas, relaciones, permisos, etc. A la vez que mantiene la opción de ejecutar sentencias SQL que ayudan a realizar pruebas directamente desde la propia herramienta.

Se ha elegido esta herramienta ya que se ha utilizado numerosas veces durante el grado, siendo de gran ayuda y sin traer grandes problemas.

### 5.2.4. FileZilla

FileZilla es una aplicación de escritorio que facilita la conexión con el servidor mediante el FTP (File Transfer Protocol) [11].

Esta herramienta se ha utilizado para facilitar el movimiento de los archivos de las carpetas locales al servidor aportado por la AEC.

### 5.2.5. Xampp

Xampp es un paquete de software libre distribuido por Apache, que actúa como servidor web. Consiste principalmente en el servidor de Apache, el sistema de gestión de bases de datos MySQL y los intérpretes para los lenguajes PHP y Perl.

Gracias a Xampp se ha podido ejecutar la página web de manera local, permitirla interpretar las páginas basadas en PHP y dotarla de una base de datos. Con esto, se han podido realizar las pruebas necesarias antes de lanzar la página al servidor dotado por la AEC.



# 6.

## Implementación

En este capítulo se le dará un repaso a la implementación que se ha llevado a cabo durante el desarrollo, destacando los puntos más relevantes y los que más problemas han causado. Además, se explicarán las diferentes pruebas hechas a lo largo de la implementación para asegurar que todo funcionaba tal y como se esperaba.

### 6.1. Configuración de herramientas

Lo primero que se debe hacer a la hora de implementar una página web es configurar tanto GitHub, para mantener un control de las versiones del trabajo, como Xampp, que permitirá hacer pruebas en el localhost antes de publicar las versiones estables. Además, se deben preparar el entorno de desarrollo a utilizar, como ha sido Visual Studio Code en este trabajo.

#### Xampp

Una vez descargado e instalado Xampp, se creará una carpeta con el nombre *xampp* en la dirección especificada. Dentro de esta carpeta se encontrará otra con el nombre *htdocs*, aquí es donde se deben almacenar los documentos relacionados al trabajo. Una vez esté implementada parte de la página web, solo habrá que activar la opción de *Apache* y la de *MySQL* (véase Figura 19- Panel de control Xampp).

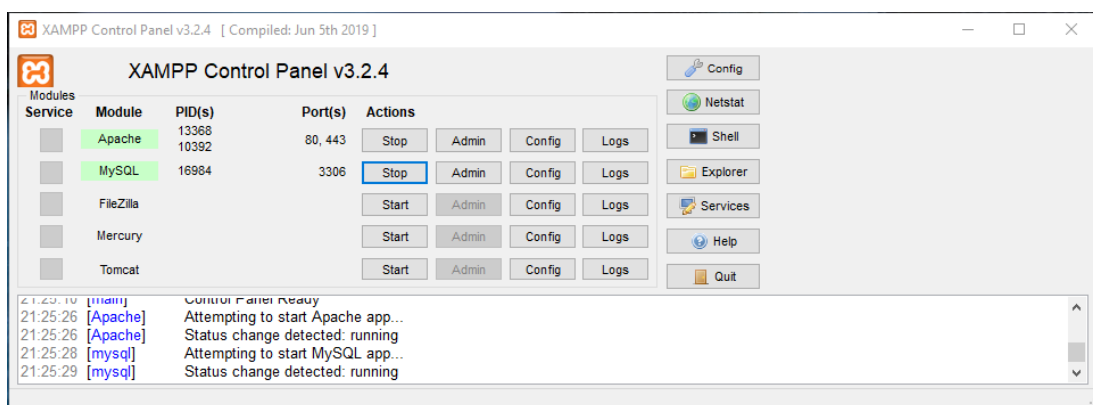


Figura 19- Panel de control Xampp

Cuando ambos se activen, se podrá acceder a la página web poniendo en el buscador *localhost/* seguido del *path* al archivo principal.

#### GitHub

En este trabajo se ha utilizado la aplicación para escritorio que ofrece GitHub, GitHub Desktop, por preferencias del alumno y ya que ofrece unas pequeñas ventajas a la hora de acceder a los ficheros con los que se está trabajando y facilidades a la hora de llevar a cabo acciones como los *commit* o los *fetch*.

Para preparar esta aplicación para su uso, primero se debe descargar e instalar. Si se cuenta con una cuenta de GitHub como era el caso, se iniciará sesión y de esta manera se podrá acceder a todos los repositorios con los que se quiera trabajar. Una vez configurada la cuenta, habrá que crear un nuevo repositorio desde *File->New Repository* y se seleccionará la carpeta principal creada anteriormente en *Xampp*.

Una vez configuradas estas dos herramientas, se podrá acceder a los archivos a través del *localhost* sin perder la capacidad del control de versiones.

### *Visual Studio Code*

Como ya se ha explicado, esta herramienta tiene una gran compatibilidad con GitHub, y ayudará con el control de cambios marcando el estado de cada archivo: nuevo, editado, borrado o intacto. Para maximizar la interacción entre estas dos herramientas existe la posibilidad de instalar una extensión de GitHub que permitiría hacer los *commits* desde el propio editor de texto, pero esta opción ha sido descartada.

## 6.2. Decisiones globales

Antes de comenzar con la explicación de la implementación llevada a cabo se aclararán ciertos aspectos globales que influyen al desarrollo en su totalidad. Estas decisiones han sido tomadas mirando al posible futuro de la aplicación, pero a su vez teniendo en cuenta los plazos y el tiempo límite del que se dispone para el trabajo de fin de grado.

La primera decisión que se tuvo que tomar era si hacer una página web *responsive* o no. Las ventajas que aporta una web con esta característica tienen mucho peso, ya que permitiría un uso cómodo desde una gran variedad de dispositivos, desde ordenadores hasta móviles, pasando por tabletas de tamaño intermedio. Sin embargo, al ser una aplicación orientada a un usuario específico, se ha llegado a la conclusión de que el principal uso de este servicio se hará desde un ordenador y que el tiempo que requieren el diseño y la implementación añadida no aportarán una mejora de calidad proporcional al trabajo realizado. Por ello, sí que se han utilizado las clases *row* y *col* que nos facilita *Bootstrap* [12], para aligerar su implementación en un futuro, pero no se ha utilizado para hacer que la web sea *responsive*.

En segundo lugar, mirando a un posible futuro de la aplicación, y a su posible extensión tanto en cuanto a variables se refiere como a secciones que abarque el registro, se ha implementado el código de tal manera que su escalabilidad sea sencilla y requiera los mismos cambios posibles en el código. Para ello, como se explicará más detalladamente a continuación, las acciones a realizar con las variables se han implementado de manera genérica, permitiendo así que sean compatibles con las nuevas variables que puedan formar parte del registro en un futuro.

Por último, al ser un proyecto con muchas líneas de código y una gran interacción entre el código HTML y los archivos PHP, se ha modulado el código en la medida de lo posible, dividiendo en diferentes las diferentes funcionalidades. De esta manera se ha conseguido que el código sea más legible y que a la hora de buscar qué se quiere editar, sea más fácil encontrarlo buscando por los nombres de los archivos. Esta decisión ha ayudado a la hora de desarrollar el trabajo de fin de grado, pero también supondrá una mejora en un futuro, cuando sea necesario realizar cambios en el código.

### 6.3. Registro RENACE

En este apartado se explicará el proceso de implementación del trabajo realizado, haciendo énfasis en las tomas de decisiones más importantes y en los puntos en los que se crea necesario dar una explicación más específica. Para facilitar la comprensión del proceso, se dividirá en pasos ordenados por orden cronológico, aun así, el orden no reflejará al 100% el desarrollo real ya que en este se han mezclado diferentes aspectos que presentaremos por separados a continuación.

#### Creación de la interfaz

El comienzo del desarrollo de esta página web se encuentra en el archivo *header\_navbar.php* donde se implementan tanto la cabecera como la barra de navegación principal y se prepara el espacio donde se cargará el contenido de las diferentes secciones.

La cabecera se divide en tres columnas, una para el logotipo de la AEC, la central para el nombre del registro, y la tercera para mostrar el nombre del usuario conectado. Esta división se ha hecho mediante Bootstrap.

La barra de navegación se ha implementado siguiendo el modelo que ofrece Bootstrap. Esta barra ocupará al completo la vertical de la página y no se verá afectada por el *scroll* que se haga al resto de los elementos.

En la Figura 20 que se presenta a continuación se puede ver: en rojo, el encabezado, con su división en tres columnas; en azul, la barra de navegación; y en morado, el espacio reservado para mostrar el contenido de la página.

Hospital	Ciudad
Hospital Clínico Universitario de Santiago de Compostela	Santiago de Compostela
Hospital Universitari Bellvitge	Barcelona
Hospital Universitari Germans Trias i Pujol	Badalona
Hospital Universitario Araba	Vitoria-Gasteiz
Hospital Universitario Basurto	Bilbao
Hospital Universitario Central de Asturias	Oviedo
Hospital Universitario Fundación Jiménez Díaz	Madrid
Hospital Universitario Río Hortega	Valladolid
Hospital Universitario Virgen de la Salud	Toledo

Figura 20- Inicio del registro

```

<!-- HEADER -->
<header id="main-header" class="header">
  <div class="container-fluid">
    <div class="row">
      <div class="col-2"> ... </div>
      <div class="col-lg-8"> ... </div>
      <div class="col-lg-2"> ... </div>
    </div>
  </div>
</header>
<!-- PAGE -->
<div id="main-page" class="container-fluid">
  <div class="row">
    <!-- NAV -->
    <div id="nav" class="col-2 sidebar-sticky sidebar-shadow-right">
      <nav class="nav-rnc">
        <ul class="nav flex-column">
          <li>Inicio</li>
          <li>Pacientes</li>
          <li>Gráficos</li>
          <li>Informes</li>
        </ul>
      </nav>
    </div>
    <!-- CONTENT -->
    <div id="content" class="col-10 main">

```

Figura 21. Codificación de la disposición de los elementos principales

Para que este diseño funcione, cada elemento de la lista de navegación tendrá un link a su archivo correspondiente, que contendrá la información a mostrar. Además, estos archivos deberán incluir en la primera línea la siguiente sentencia:

```
<?php include "header_navbar.php"; ?>
```

Figura 22- Llamada a la cabecera principal

Visto cómo funciona esta estructura, se remarcarán las principales ventajas del método en el que se han implementado esta parte del trabajo.

- **Escalabilidad:** en caso de querer implementar otro apartado, solo se tendrá que añadir un nuevo elemento a la lista de navegación e insertar la sentencia nombrada anteriormente en el archivo correspondiente.
- **Fácil edición:** tener el encabezado y la navegación en un archivo por separado ahorra tiempo a la hora de realizar cambios si se compara con el caso en el que se tuviese ese código repetido en cada archivo.
- **Mayor legibilidad:** programar por módulos facilita la comprensión del código ya que cada sección esta implementada en un archivo diferente y no se mezclan contenidos que no tienen relación entre ellos.

Por último, en lo relacionado a esta parte del desarrollo, cabe destacar la hoja de estilos utilizada. Para conseguir el efecto de que ambos elementos no se muevan al hacer *scroll* se le ha asignado a la propiedad *position* el valor *sticky*. De esta manera, la cabecera se mantendrá fija en la parte superior de la página y la barra de navegación tendrá como limite la cabecera misma.

<pre>.header {   position: -webkit-sticky;   position: sticky;   top: 0; }</pre>	<pre>.sidebar-sticky {   position: -webkit-sticky;   position: sticky;   top: 100px;   height: calc(100vh - 100px); }</pre>
--	---

Figura 23- Hoja de estilos de la cabecera y el menú de navegación

### Creación del Inicio

La página de inicio muestra los datos generales relacionados con el registro, como puede ser el número total de pacientes o los hospitales que apoyan esta iniciativa.

Si bien este apartado cuenta con poca profundidad y complejidad en cuanto a implementación se refiere, cabe destacar que es el primer sitio en el que se ha utilizado la conexión a la base de datos, por lo que se explicará el mecanismo utilizado y se pondrá como ejemplo este sencillo caso.

Para guardar los datos tanto de la base de datos local como de la del servidor, se utilizará un archivo al que se ha llamado *dbConfig.php*. En éste se guardarán el nombre del servidor, usuario, contraseña y nombre de la base de datos. De esta manera, solo se tendrá que hacer la llamada que se muestra en la Figura 24 antes de comenzar las transacciones con la base de datos y estas se ejecutarán correctamente tanto de manera local como en el servidor.

```
<?php include "dbConfig.php"; ?>
```

Figura 24- Llamada al archivo de configuración de la base de datos

A la hora de elegir que extensión utilizar para interactuar con la base de datos, la página principal de PHP ofrecía tres opciones: MySQL, mysqli y PDO. Tras analizar lo que aportaba cada una de ellas, se decidió utilizar mysqli, que, entre otras ventajas, aporta una interfaz orientada a objetos, tiene soporte para ejecutar múltiples declaraciones, soporte para transacciones y, además, es la recomendada por el manual oficial de PHP [9]. Unido a lo anterior, notar que al contar con un servidor con la versión de PHP 7.1.29, esta extensión de mysqli ya viene incluida.

Como se ha explicado, el primer uso de esta herramienta sirvió para mostrar los hospitales en la página de inicio. A continuación de muestran las sentencias necesarias para llevar a cabo esa consulta.

Primero se creará la conexión, utilizando las variables guardadas en *dbConfig.php*. Seguidamente, hay que asegurarse de que se está utilizando el *charset* correcto y se creará la sentencia deseada. Por último, se ejecutará la sentencia y se recorrerán los resultados, en este caso, para crear la tabla deseada.

```

<?php
$conexion = new mysqli($servidor, $usuario, $contrasena, $db);

$conexion->set_charset("utf8")

$sqlHosp = "SELECT Hospital, Ciudad FROM usuarios";
$resHosp = $conexion->query($sqlHosp);

while ($rowHosp = $resHosp->fetch_assoc()) {...}
?>

```

Figura 25- Conexión y consulta a la base de datos

#### Creación del inicio de sesión

Como se ha explicado anteriormente, el acceso al registro está limitado a un número pequeño de personas y, además, los datos que se muestran cambiarán para cada usuario, por lo tanto, es imprescindible implementar un sistema de inicio de sesión.

Este inicio de sesión es muy sencillo, cuenta con dos entradas, una para el nombre de usuario y otro para la contraseña. Una vez el usuario ha introducido los datos, se comprobarán si estos concuerdan con los almacenados en la base de datos. De ser así, se le redirigirá a la página de inicio; y en caso de que los datos no concuerden, se le mostrará un mensaje para que vuelva a insertar sus credenciales.

En este proceso de verificación de los datos del usuario cabe destacar la utilización del cifrado de la contraseña, que es el mismo que se utilizará a la hora de insertar el usuario en la base de datos, haciendo uso de la función `password_hash()`.

```

<?php
password_hash($clave, PASSWORD_DEFAULT, array("cost" => 15));
?>

```

Figura 26- Cifrado de la contraseña

Al estar la contraseña cifrada en la base de datos, a la hora de comprobar esta coincide con la que el usuario a logearse a insertado, se debe hacer uso de la función `password_verify()`.

```

<?php
$sql = "SELECT * FROM usuarios WHERE User = '" . $_POST['user'] . "'";

password_verify($_POST['pwd'], $row['Password']);
?>

```

Figura 27- Verificación de la contraseña

#### Pacientes

La siguiente sección desarrollada fue la que permite la gestión de los pacientes. Esta página contiene, por un lado, una tabla que muestra los pacientes del usuario conectado (véase Figura 28- Tabla pacientes) y, por otro lado, una barra lateral que da acceso a diferentes acciones a realizar en torno a los pacientes.

La tabla principal sigue una implementación similar a la utilizada para los hospitales en la página de inicio, con la única diferencia de que las filas de esta cuentan con el atributo *tabindex*, que será útil cuando se explique la implementación de la funcionalidad del buscador de pacientes.

The screenshot shows a web application interface. On the left, there is a table titled "Pacientes" with three columns: "Código", "Hospital", and "Fecha de cirugía". The table contains 14 rows of patient data. On the right, there is a green sidebar with three main sections: "Añadir" (Add) with a button "Añadir nuevo paciente"; "Buscar" (Search) with a dropdown menu showing "Código" and "Fecha", a search input field, and a "Buscar" button; and "Importar" (Import) with a section "Elige el archivo a importar:" containing a "Seleccionar archivo" button and the text "Sin archivos.", and an "Importar" button.

Código	Hospital	Fecha de cirugía
01013	Hospital Universitario de Salamanca	23-10-2019
01014	Hospital Universitario de Salamanca	07-10-2019
01015	Hospital Universitario de Salamanca	01-10-2019
01016	Hospital Universitario de Salamanca	24-09-2019
01017	Hospital Universitario de Salamanca	18-09-2019
01018	Hospital Universitario de Salamanca	10-09-2019
01019	Hospital Universitario de Salamanca	30-07-2019
01020	Hospital Universitario de Salamanca	16-07-2019
01021	Hospital Universitario de Salamanca	02-07-2019
01022	Hospital Universitario de Salamanca	25-06-2019
01023	Hospital Universitario de Salamanca	18-06-2019
01024	Hospital Universitario de Salamanca	07-05-2019
01025	Hospital Universitario de Salamanca	29-04-2019
01026	Hospital Universitario de Salamanca	01-04-2019

Figura 28- Tabla pacientes

Acompañando a esta tabla contamos con tres funcionalidades: buscar, añadir e importar pacientes, de las cuales las dos últimas han supuesto la gran parte del desarrollo de este trabajo.

### Buscar

Con el fin de facilitar la búsqueda de pacientes en los usuarios que cuentan con un gran número de casos insertados, se ha implementado un sistema de búsqueda que facilita la búsqueda del paciente a través de su código o estableciendo una fecha de inicio y otra de fin.

Para implementar la búsqueda por código se ha hecho uso de la combinación de jQuery con CSS. Gracias a jQuery se puede coger el *item* de la lista a través del atributo *tabindex* que se ha nombrado anteriormente, y combinandolo con CSS ser capaces de aplicar *focus* a este *item*. De esta manera, se hará *scroll* automático y se remarcará el paciente que se ha buscado.

```

$("#btnBuscarCodigo").click(function() {
    $("tr[tabindex=" + parseInt($("#codigoBuscarInput").val()) +
    "]").focus();
});

```

Figura 29- Script buscar paciente por código

En la Figura 30 se puede ver en el lado derecho el código del paciente a buscar, y en la tabla ese mismo paciente resaltado.

01018	[Redacted]	10-09-2019
01019	[Redacted]	30-07-2019
01020	[Redacted]	16-07-2019
01021	[Redacted]	02-07-2019
01022	[Redacted]	25-06-2019

**Buscar**

Código
Fecha

Buscar

Figura 30- Buscar paciente por código

Por el otro lado, la búsqueda por fechas se basa en una consulta a la base de datos que devuelve todos los pacientes en los que su fecha de cirugía se encuentre entre las fechas de inicio y fin dadas. Esta consulta se puede ver en la Figura 31.

```

<?php
$sqlPacientes = "SELECT Codigo, Hospital, C_F
FROM pacientes
WHERE Hospital = '" . $rowHosp["Hospital"] . "'
AND C_F BETWEEN '" . $_POST['fInicio'] . "' AND
'" . $_POST['fFinal'] . "'";
?>
```

Figura 31- Consulta buscar por fecha

Haciendo uso de esa consulta, si se inserta en la fecha de inicio el 01/10/2019 y en la fecha de fin el 12/06/2020 se puede ver como se obtienen tres pacientes cuya fecha de cirugía está dentro de los parámetros dados.

Pacientes		
Código	Hospital	Fecha de cirugía
01013	[Redacted]	2019-10-23
01014	[Redacted]	2019-10-07
01015	[Redacted]	2019-10-01

**Añadir**

Añadir nuevo paciente

**Buscar**

Código
Fecha

Buscar

Figura 32- Buscar paciente por fecha

### Añadir/Ver/Editar paciente (interfaz)

Como todo registro, debe existir una manera de que el usuario inserte sus pacientes en la base de datos. En este apartado explicaremos en detalle las decisiones importantes tomadas en lo que es la parte más importante del desarrollo del trabajo.

Antes de comenzar con la implementación, se debe explicar la estructura del formulario y su importancia en su futura extensión a otras secciones. Este cuenta con cinco apartados, cada uno de ellos dedicado a un aspecto diferente de la cirugía realizada, como pueden ser los datos del paciente, los datos de la enfermedad o el seguimiento realizado después de la operación. La mayoría de estos son comunes para cualquier tipo de cirugía, por lo que serán reutilizados una vez incorporadas nuevas secciones, sin embargo, otros apartados más específicos como el “Cirugía” deberán ser implementados de nuevo.



Para añadir los pacientes, se ha habilitado un formulario al que se accede mediante el botón *Añadir nuevo paciente* situado en la barra lateral derecha de la página *Pacientes*. Este formulario cuenta con cinco tipos de entrada de datos; numéricos, fechas, *dropdown lists*, *switches* y *checkboxes*. En la Figura 33 se puede ver una pequeña parte del formulario, y en la sección de Anexos se puede consultar el formulario completo, que a causa de su extensión se presenta fraccionado en dos (Figura 52 y Figura 53) para una visualización más adecuada.

## Añadir nuevo paciente

Diagnóstico y estudio de extensión (datos de la enfermedad)

Fecha del diagnóstico dd/mm/aaaa	Tipo histológico Seleccione una opción	<b>Pruebas diagnósticas</b> <input type="checkbox"/> Gastroscopia <input type="checkbox"/> TC <input type="checkbox"/> Ecoendoscopia <input type="checkbox"/> Resección mucosa/submucosa <input type="checkbox"/> Broncoscopia <input type="checkbox"/> Laparoscopia exploradora <input type="checkbox"/> PET-TC <input type="checkbox"/> Gammagrafía ósea <input type="checkbox"/> RM <input type="checkbox"/> Ecografía-biopsia
Localización del tumor Seleccione una opción		
<b>Hemoglobina</b>		
Al diagnóstico gr/dl	Próximo a la cirugía gr/dl	
Albúmina preoperatoria gr/dl	Ca 19.9 U/ml	
<b>Estadificación clínica (cTNM)</b>		
cT Seleccione una opción	cN Negativo Positivo	cM M0 Indet. M1

Figura 33- Fragmento añadir nuevo paciente

### Numéricos

Estas entradas servirán para que el usuario inserte datos como la edad, el peso o la talla del paciente, entre otros. Dependiendo de la variable se limitará su rango y la cantidad de decimales aceptados.

En la implementación y el diseño de este tipo de *input* caben destacar dos cosas. Por un lado, el atributo *name* siempre coincidirá con el nombre del campo correspondiente en la base de datos. Y, por último, puesto que se utiliza la misma base de código para añadir, visualizar y editar los pacientes, se debe comprobar qué caso se está tratando antes de imprimir un valor. Para ello, se utilizará la función *isset()*, que determina si una variable está definida y no es *NULL*, y solo en el caso afirmativo se mostrará su valor. Si no se realiza esta comprobación previa, al entrar al formulario con la intención de añadir un nuevo paciente, el sistema nos mostrará un *warning* por cada *input*.

```
<p>Edad</p>
<input type="number" id="edad" name="DP_edad" min="0"
value="<?php if (isset($pac['DP_Edad'])) echo $pac['DP_Edad']; ?>">
```

Figura 34- Ejemplo del input de tipo numérico

### Fechas

Esta entrada es similar a la numérica, y se utilizará para que el usuario inserte la fecha de diferentes eventos correspondientes a un paciente como la fecha de diagnóstico o la fecha de la cirugía.

Gracias a la utilización del tipo fecha para estos tipos de *input*, el manejo de las fechas no ha creado mayor problema. Aun así, esta facilidad no está implementada en todos los buscadores, por lo que se han tenido que indagar soluciones para que todos los usuarios pudiesen insertar las fechas en el formato correcto. A día de hoy, Internet Explorer y Safari son dos buscadores que no soportan este tipo de entrada en los formularios, y para facilitar el uso correcto a los usuarios, se ha utilizado el atributo *placeholder*, de esta manera el usuario podrá ver cuál es el formato de fecha permitido.

En la figura que se presenta a continuación se puede ver la diferencia entre los buscadores que sí (izquierda) y que no (derecha) aceptan el tipo fecha:

<p><b>Fecha inicio</b></p> <input type="text" value="dd/mm/aaaa"/>	<p><b>Fecha inicio</b></p> <input type="text" value="aaaa-mm-dd"/>
--	--

Figura 35- Comparación de fechas en diferentes buscadores

En el fragmento de código de la Figura 36 se puede ver el valor dado al atributo *placeholder* y la utilización de *isset()* que ya se ha explicado en el tipo de dato numérico.

```
<p>Fecha del diagnóstico</p>
<input type="date" name="D_FD" placeholder="aaaa-mm-dd"
value="<?php if (isset($pac['D_FD'])) echo $pac['D_FD']; ?>">
```

Figura 36- Fragmento del input Fechas

#### Dropdown lists variables

Esta entrada es la más usada en el formulario, y permite al usuario elegir una opción de las que se le presentan en la lista desplegable.

Para facilitar y agilizar la programación de estas listas y a su vez el proceso de ampliación de la base de datos para un futuro, se ha automatizado, en la medida de lo posible, la creación de este tipo de entradas. Además, gracias a este diseño, el código es más limpio y compacto, esto es, con un volumen de líneas mucho más reducido.

La primera opción que vendría a la cabeza al tener que programar estas listas, sería escribir a mano la estructura de una lista desplegable, con sus etiquetas, atributos y contenido, que generalmente solo consta de unas pocas palabras, pero otras veces, pueden llegar a ocupar varias líneas.

```
<p>Tipo de tratamiento</p>
<select id="trat-neo-tipo" name="TN_T">
  <option value="" selected hidden="">Seleccione una opción</option>
  <option value="1">No</option>
  <option value="2">Radioterapia</option>
  <option value="3">Quimioterapia</option>
  <option value="4">Radio-quimioterapia</option>
</select>
```

Figura 37- Ejemplo dropdown lists

Sin embargo, al ver que esto llevaría a una gran cantidad de código prácticamente idéntico, un largo tiempo de programación, muchas dificultades a la hora de ampliar el formulario y potenciales errores en el código, se decidió crear un archivo PHP que generase estas listas de manera automática. Este archivo genera la lista desplegable basándose en la base de datos, y funciona tanto para el caso en el que se esté añadiendo como se esté visualizando un paciente.

Para crear estas listas se necesitarán dos sentencias PHP. En la primera se indicará cual es el nombre de la variable a mostrar, que deberá coincidir con su nombre en la base de datos, y en la segunda se hará un *include* al archivo especificado. Seguidamente, se muestra un ejemplo de cómo sería la llamada para general el código del ejemplo anterior.

```
<?php
$genVal = "TN_T";
include "dobb_get_generic.php";
?>
```

Figura 38- Llamada a la generación de código automático

Como se ha dicho, este diseño depende de que las variables estén almacenadas en la base de datos. Aun así, el proceso de creación de las tablas en la base de datos ha sido mucho más sencillo y cómodo que el de programar a mano todas las listas desplegables. Y, por otra parte, si se considera la posible extensión tanto del número de variables médicas como de la cantidad de opciones dentro de una lista desplegable, la opción de que todo esté almacenado en la base de datos hará este proceso más sencillo y reducirá el riesgo a cometer errores a la hora de programar.

#### Switches

Para las variables que solo disponen de dos valores posibles, se ha utilizado un *switch* de tres posiciones donde los laterales presentan los valores a elegir.

Estos *switches* son un conjunto de tres *inputs* de tipo *radio*, con sus respectivos *labels* que nos permiten darles un nombre. Para hacer de este grupo de botones, el *switch* que se muestra en la Figura 40, se asignarán las clases *switch-toggle*, *switch-3* y *switch-candy*, que basadas en *css* creando esta apariencia de interruptor.

```
<div class="switch-toggle switch-3 switch-candy">
  <input id="..." name="..." type="radio">
  <label for="..." onclick="">No</label>
  <input id="..." name="..." type="radio" value="2">
  <label for="..." onclick="">&nbsp;</label>
  <input id="..." name="..." type="radio" value="3">
  <label for="..." onclick="">Si</label>
</div>
```

Figura 39- Código switches



Figura 40- Ejemplo visual de variables de dos valores

Este *switch* no ha sido creación propia del autor, sino que se ha inspirado en Colceriu [13]. Al igual que con las listas desplegables, se ha creado un archivo PHP para la creación del código correspondiente a cada *switch*. Su funcionamiento es idéntico; solo se necesita especificar el nombre de la variable, que debe coincidir con el nombre que contenga en la base de datos, y hacer un *include* al archivo generador, que en este caso se llama *dadb\_get\_genericSwitch.php*. Al igual que en el caso anterior, los valores se extraen de la base de datos, por lo que en caso de querer editar el valor de alguna de estas variables o crear una nueva variable de este tipo se deberá hacer a través de la base de datos.

#### *Checkbox*

Este tipo de entrada se utilizará para las variables múltiples, es decir, las variables que pueden tener más de una respuesta posible prevista. Ya se ha hablado de ellas al explicar los tipos de tablas de la base de datos, por lo que en este apartado solo se explicará su papel en el formulario.

Al igual que los dos tipos de entradas anteriores, se ha creado un archivo PHP que genera estas variables múltiples, por las mismas razones con las que se ha razonado el uso de este archivo en las listas desplegables y en los *switch*-es. Para llamar a este archivo, se deberá asignar a la variable *varGenMC* el nombre de la tabla de la base de datos que contenga la variable actual y después hacer un *include* al archivo *dadb\_mult\_checkbox\_generic.php*.

#### Añadir pacientes (base de datos)

Tras haber visto cómo se crea la interfaz de este formulario, se analizará cuál es el procedimiento implementado para enviar esta información a la base de datos.

La primera división clara que se puede hacer a la hora de insertar los datos en la base de datos es la del tipo de variable, simple o múltiple. De hecho, las variables simples están almacenadas en una única tabla, y las variables múltiples cuentan cada una con su propia tabla.

Para almacenar las variables se ha creado una sentencia *insert* por cada tabla. Sin embargo, como las tablas de las variables múltiples hacen referencia a la tabla principal del paciente, el proceso de ejecución de las sentencias se ha modificado, para adecuarlo a las necesidades de este caso en particular.

Para empezar, se ha iniciado una transacción, de tal forma que se puedan deshacer todos los cambios en caso de encontrarse un error en alguna sentencia. A continuación, se ha ejecutado la sentencia que incluye las variables simples. Una vez hecho esto, se puede recuperar el código asignado de manera automático al paciente recién creado, de tal manera que se asegure que las variables múltiples tengan ese mismo código. Una vez ejecutada la sentencia múltiple que contiene los datos de todas las variables múltiples, se hará *commit*, de tal manera que todos los cambios queden guardados en la base de datos. En caso de que alguna sentencia lance un error, se ejecutará la sentencia *rollback*, deshaciendo las sentencias previas y no dejando rastro en la base de datos.

#### Editar pacientes (base de datos)

Cuando se abre el formulario haciendo *click* sobre el código de un paciente, se mostrarán sus datos asociados y estos podrán ser actualizados.

Se plantearon varias opciones de actualizar estos datos y enviar la información a la base de datos. A continuación, se explica el funcionamiento del programa finalmente implementado. Como se ha hecho hasta ahora, se tratarán de diferente manera las variables simples y múltiples.

En este caso se empezará explicando el tratamiento de las múltiples, por ser este caso el más sencillo. Al contar con muy pocas variables múltiples, se ha decidido que las sentencias de actualizarlas se ejecuten siempre, de tal manera que no se tenga que llevar control de sus cambios.

Sin embargo, teniendo en cuenta que las variables simples son las más comunes en el formulario ocupando el mayor espacio asignado al formulario, sí se ha implementado un método con el fin de aligerar las sentencias. Para poder llevar a cabo esta implementación, se ha creado una función que se ejecuta cada vez que un *input* cambia de valor. Esta función guarda en una *string* el atributo *name* del *input* editado, que debe coincidir con el nombre de su campo correspondiente en la base de datos, y el nuevo valor asignado (véase Figura 41). De esta manera, se inserta es *string* en una sentencia *update* (véase Figura 42) y los valores se actualizan correctamente en la base de datos.

```
<?php
$editedValues = C_AA='1',C_AbT='4',C_AbT_P='1'
?>
```

Figura 41- String creada por la edición de las variables

```
<?php
$sqlEdit = "UPDATE pacientes SET $editedValues WHERE Codigo=$codigo";
?>
```

Figura 42- Sentencia update que utiliza la string generada

#### Importación desde Eureka

El proceso que más diseño ha requerido y que más tiempo ha llevado en la implementación ha sido la importación de los datos del registro EURECCA al RENACE.

La primera complicación surgió del hecho de que los datos de EURECCA se exportasen a un archivo .xlsx (Excel). Para recorrer este archivo y poder así manejar los datos de manera cómoda, dentro de lo posible, se ha utilizado PhpSpreadsheet [14], una librería de PHP que permite la lectura y escritura en archivos como Excel. Gracias a esta librería, se ha podido recorrer este documento por filas, de tal manera que se recojan todos los datos de un paciente por cada iteración.

Cada vez que se visita una casilla, se coge el valor de la primera celda de esa columna, que indicará el nombre de la variable. Con la ayuda de la tabla *eureka\_import\_variables* se podrá saber qué tipo de modificación necesita cada variable y a qué variable de RENACE corresponde. Como se ha explicado en el diseño de la base de datos, el tipo de modificación se especifica a través de un número que va del 0 al 5. Las traducciones que se vayan obteniendo se almacenarán en la *string sqlIns* y esta se utilizará en una consulta de inserción.

En el fragmento de código que se presenta en la Figura 43, se muestra el *switch-case* utilizado para tratar cada modificación.

```

<?php
switch ($tradCase) {
case 0: // No hace falta editar
    $sqlIns .= $separator . "" . $value . "";
    break;
case 1: // Traducción de frase a número
    $sqlIns .= $separator . "" . ($arrayCombined[$value] ?? null)
    break;
case 2: // Fecha
    $sqlIns .= $separator . "" . createDateFormat($value) . "";
    break;
case 3: // Talla (Quitar el último carácter)
    $sqlIns .= $separator . "" . floatval(substr($value, 0, -1))
    * 100 . "";
    break;
case 4: // Quitar los dos últimos caracteres
    $sqlIns .= $separator . "" . substr($value, 0, -2) . "";
    break;
case 5: // múltiple
    tratarMultiple($colName, $value);
    break;
}
?>

```

Figura 43- Switch para el tratamiento de las diferentes variables

Cada vez que se termina de recorrer una fila, es decir, un paciente, se volcarán los datos a la base de datos. Como en el caso de añadir un paciente, se ejecutará primero la sentencia que corresponde a las variables simples, y con el código obtenido se añadirán las variables múltiples.

Para dar un *feedback* de cómo ha ido la importación, una vez tratados todos los pacientes, se le redirigirá a la sección pacientes y se le mostrará un mensaje diciendo cuantos pacientes se encontraron y cuantos se añadieron correctamente.

### Informes

La creación de informes basados en los pacientes almacenados en la base de datos ha sido el primer sistema utilizado para mostrar los datos y las estadísticas del registro. Se han seleccionado una serie de indicadores de calidad de la cirugía reconocidos internacionalmente.

En esta sección se muestran los datos a través de una sencilla tabla que consta de tres columnas. La primera contiene el nombre de la estadística, la segunda mostrará los datos pertenecientes al hospital del usuario conectado, y la tercera, los datos generales del registro, es decir, los que involucran a todos los hospitales. En la Figura 44- Informes, se muestra un ejemplo concreto. Se ha valorado la pertinencia o no de incluir una cuarta columna con los estándares de calidad reconocidos actualmente [15,16], cuestión que será decidida en el futuro.

La implementación de este apartado ha sido generalmente sencilla y rápida de hacer. Sin embargo, las consultas a la base de datos han requerido más tiempo del esperado debido al problema que se muestra a continuación.

La mayor parte de la tabla consta de porcentajes, que a pesar de que *mysql* no cuente con una función directa que los calcule, es un dato estadístico sencillo y que no conlleva a una gran complejidad a la hora de formular las consultas. Sin embargo, la tabla requiere el cálculo de la

mediana en varias de sus variables, y esta sí ha requerido la búsqueda e investigación de su solución para poder implementarla de manera correcta.

The screenshot shows a web application interface. On the left is a green sidebar with navigation icons and labels: 'Inicio', 'Pacientes', 'Gráficos', and 'Informes'. The main content area is titled 'Informes' and contains a table with three columns: 'INDICADORES GENERALES', 'Hospital Universitario Basurto', and 'Total'. The table lists various medical indicators with their respective counts and percentages for the hospital and the total.

INDICADORES GENERALES	Hospital Universitario Basurto	Total
% R0 (Radicalidad de la resección)	35 / 35 (100,00%)	212 / 222 (95,50%)
Mediana de ganglios resecaados	33,71	28,54
% pacientes con $\geq 15$ ganglios resecaados	34 / 35 (97,14%)	194 / 222 (87,39%)
% complicaciones	24 / 35 (68,57%)	156 / 222 (70,27%)
% complicaciones leves	14 / 35 (40,00%)	83 / 222 (37,39%)
% complicaciones graves	9 / 35 (25,71%)	65 / 222 (29,28%)
% mortalidad	1 / 35 (2,86%)	8 / 222 (3,60%)
% complicaciones anastomóticas (fistula)	8 / 35 (22,86%)	64 / 222 (28,83%)
% complicaciones respiratorias	10 / 35 (28,57%)	93 / 222 (41,89%)
% failure-to-rescue	1 / 10 (10,00%)	8 / 73 (10,96%)
Estancia mediana	14,00	14,00

Figura 44- Informes

### Gráficos

Por último, y probablemente uno de los puntos más fuertes de este registro, se ha reservado un apartado para mostrar gráficamente los datos más significativos obtenidos de la base de datos.

Tras una búsqueda para encontrar la herramienta que mejor cubriese las necesidades exigidas a la hora de diseñar los gráficos a mostrar, se dio con la librería *CanvasJS* [17].

CanvasJS es una sencilla API que genera gráficos basados en JavaScript y facilita la integración con tecnologías como PHP. Esto último es esencial para nuestro registro ya que todos los datos a mostrar en estos gráficos proceden de la base de datos, a la que accedemos a través de PHP. Además de cumplir estos requisitos mínimos que se han nombrado, cuenta con una gran variedad de ejemplos que han sido de gran ayuda a la hora de empezar a crear los primeros gráficos, ya que es una librería que no se había utilizado anteriormente, y ha hecho que el primer contacto sea muy gratificante e incite a su uso. Por otro lado, cuenta con un manual muy extenso donde poder aprender sobre cada atributo que acompaña a los gráficos y que ayuda a personalizar al gusto del desarrollador cada detalle.

El registro cuenta con dos grupos de gráficos. Los primeros muestran las estadísticas de cada hospital, ordenados por volumen de actividad, sin mostrar qué dato pertenece a qué hospital. El objetivo de estos es ver la situación de cada hospital en cada indicador, sin la intención de impulsar las comparaciones, pero sí con el objetivo de que cada hospital sepa en qué aspectos es más débil y se centre así en su mejora. En la Figura 45 se muestra un ejemplo.

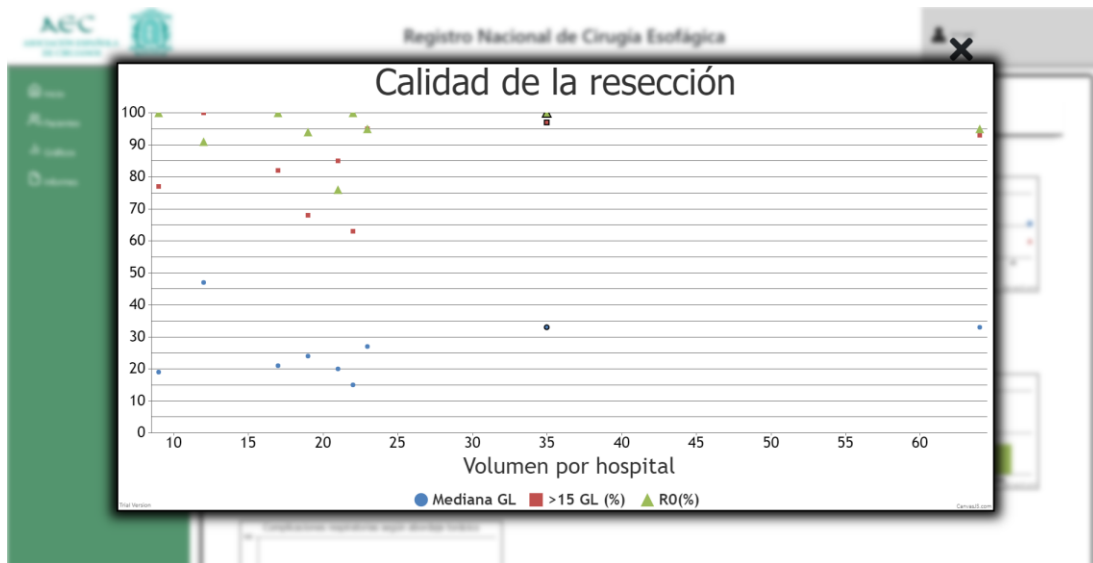


Figura 45- Indicadores de calidad de la resección

El segundo grupo muestra los datos del registro al completo, y estos se utilizarán para analizar las técnicas y métodos empleados en diferentes aspectos de la cirugía esofágica, ya que hay muchas características que comparten todos los hospitales. La Figura 46 compara los resultados de los diferentes tipos de reconstrucción del tránsito digestivo tras la resección del esófago.

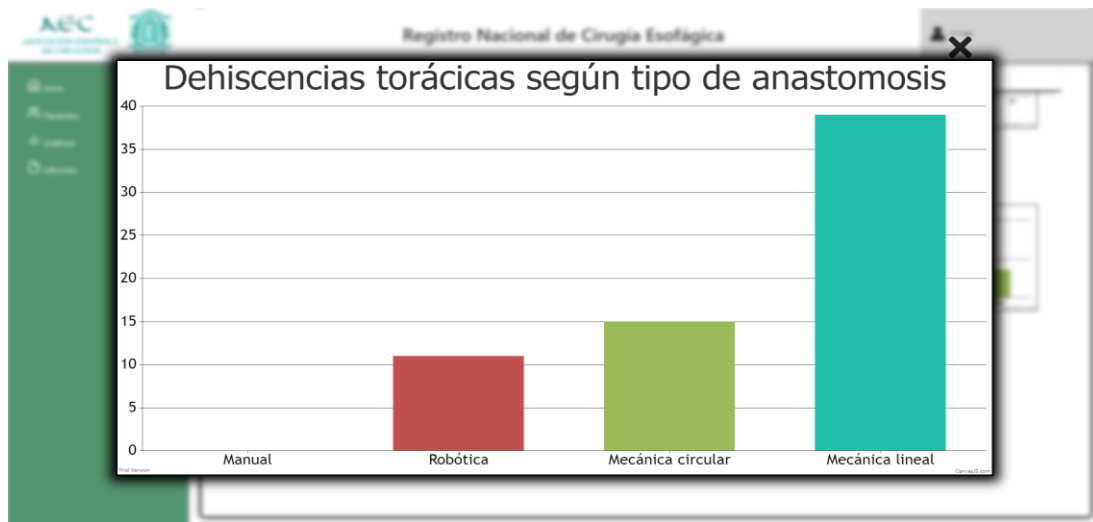


Figura 46- Fracasos según el tipo de reconstrucción

Para obtener una vista general de la sección que contiene todos los gráficos generados, véase en Anexos la Figura 55.

Volviendo a la implementación, cabe destacar que estos gráficos muestran los datos desde una estructura específica (véase Figura 47), por lo que se ha tenido que generar un método que convierta los datos obtenidos de la base de datos en el formato requerido.



```

<?php
$dataPoints = array(
    array("x"=> 10, "y"=> 41),
    array("x"=> 20, "y"=> 35),
    array("x"=> 30, "y"=> 50),
    array("x"=> 40, "y"=> 45),
    array("x"=> 50, "y"=> 52),
);
?>

```

Figura 47- Estructura de datos exigida

Esta generación de la estructura exigida se ha dividido en dos partes. Primero se prepara y ejecuta la consulta, obteniendo así un *array* con los datos que se desean mostrar, y, después, se filtran estos datos creando el *array* final deseado. Las consultas se han preparado y ejecutado como en todos los casos anteriores, por lo que no merece la pena enfocarse en ese aspecto, sin embargo, la generación del *array* sí que se explicará brevemente.

Como se puede observar en la Figura 47, la estructura exigida consiste de un *array* formado de *arrays*. Éstos últimos constarán de una llave y de un valor; la llave hará referencia a un atributo del gráfico, como puede ser el eje X, el eje Y, el nombre que se le dará a ese punto, su tamaño, etc; y el valor, como bien dice su nombre, guardará el valor de ese atributo.

En la Figura 48 que se presenta a continuación, se puede ver a grandes rasgos el método utilizado para obtener el formato correcto de estos datos. Para empezar, se creará un *loop* en el que se entrará tantas veces como hospitales haya en el registro en cada iteración se accederá a todos los *arrays* utilizando el índice *\$i*, de manera que se guarden todos los datos correspondientes a un hospital. Por último, se creará una condición que servirá para remarcar los datos del gráfico pertenecientes al usuario conectado, para que pueda saber que datos le corresponden, pero no tenga la capacidad de identificar a qué hospital pertenecen los demás datos.

```

<?php
for ($i = 0; $i < intval($countHospitales); $i++) {
    if ($consultaBBDD[$i]['Hospital'] == $hospitalUser) {
        array_push($arrayDeDatos,
            array("x"=>$i, "y"=>calcEstad($i), "borderThickness" => 2));
    } else {
        array_push($arrayDeDatos,
            array("x"=>$i, "y"=>calcEstad($i)));
    }
}
?>

```

Figura 48- Creación array de datos

## 6.4. Pruebas

En este capítulo se detallan las pruebas realizadas a lo largo del desarrollo del trabajo. Gracias a la ejecución de estas pruebas se pueden identificar comportamientos no deseados y corregirlos antes de comenzar con la implementación de la siguiente fase.

Para asegurar el correcto funcionamiento de todas las funcionalidades, las pruebas se han realizado primero por parte del desarrollador, con la ayuda del interesado, y una vez publicada la entrega, por parte de los usuarios.

En las pruebas que se presentan a continuación se describirán las pruebas llevadas a cabo por el alumno, y en un apartado final se resumirán las pruebas hechas por los usuarios.

#### 6.4.1. Pruebas Entrega parcial 1

Recordar que en la primera entrega parcial se ha desarrollado la interfaz base que contendrá las demás secciones a realizar.

Tabla 2- Pruebas Entrega parcial 1

#	Descripción	Resultado esperado	Observaciones <sup>1</sup>
1	Comprobar que al hacer <i>scroll</i> todos los elementos se comportan de manera lógica.	El encabezado y el menú de navegación se mantienen en su lugar y los demás elementos se mueven respetando el <i>scroll</i> .	El comportamiento es correcto.
2	Comprobar que el atributo <i>z-index</i> está asignado correctamente, es decir, que los elementos están superpuestos manteniendo un orden lógico.	El encabezado se sobrepone al menú de navegación dejando a la vista la <i>shadow-box</i> y se sobrepone al contenido cuando se hace <i>scroll</i> .	El comportamiento es correcto.
3	Los elementos que crean una acción al hacer <i>click</i> muestran un correcto <i>feedback</i> al pasar el ratón por encima.	Todos estos elementos deben hacer que el ratón pase de mostrar una flecha a mostrar una mano. Los links también cambiarán el color de su letra.	El comportamiento es correcto.

#### 6.4.2. Pruebas Entrega parcial 2

Recordar que en la segunda entrega parcial se ha desarrollado la inserción y edición de los pacientes.

Tabla 3- Pruebas Entrega parcial 2

#	Descripción	Resultado esperado	Observaciones
1	Se respetan las diferencias entre insertar y editar un paciente.	Cuando se edita un paciente se debe mostrar su código junto al título, los inputs deben mostrar el valor de la base de datos, y el botón mostrado al final debe ser el de "guardar".	El comportamiento es correcto.
2	El usuario no debe ser capaz de cambiar su hospital.	La entrada en la que se muestra el hospital al que pertenece el paciente no puede ser editable.	El comportamiento es correcto.
3	Estado de las variables calculadas	En el caso de edición, las entradas que se calculan basándose en otras deben calcularse al cargar el formulario, sin necesidad de hacer ningún cambio en estas.	El comportamiento es correcto.

<sup>1</sup> Solo se ha comprobado su comportamiento en los navegadores principales como Google Chrome, Microsoft Edge, Mozilla Firefox y Safari.

4	Correcto funcionamiento de las entradas de tipo fecha.	En los navegadores que no cuentan con un soporte completo para este tipo de entradas, se debe mostrar cual es el formato correcto para insertar las fechas.	El comportamiento es correcto.
5	Editar dos veces el valor de la misma variable.	Se debe guardar el último valor asignado.	El comportamiento es correcto.
6	Edición de las variables múltiples y simples.	Todas las variables deberían poder ser editadas a la vez.	El comportamiento es correcto.
7	Decimales en las variables de tipo numérico.	Variables como la edad, talla o número de ganglios, entre otras, no deben permitir decimales.	El comportamiento es correcto.
8	Scripts que hacen aparecer y desaparecer variables.	Ciertas variables deben hacer aparecer otras cuando su valor es "sí" y hacerlas desaparecer cuando su valor es "no".	El comportamiento es correcto.
9	Todas las variables cumplen su función correctamente	Todas las variables actúan como deberían, no se ha encontrado ningún error y el formulario está listo para su uso.	Hay ciertas variables que necesitan recibir alguna modificación.

#### 6.4.3. Pruebas Entrega parcial 3

Recordar que en la tercera entrega parcial se ha implementado la importación de pacientes desde el registro Eurecca.

Tabla 4- Pruebas Entrega parcial 3

#	Descripción	Resultado esperado	Observaciones
1	Comprobar si se importan todos los pacientes.	Se deberían importar todos los pacientes del archivo Excel.	El comportamiento es correcto.
2	Comprobar que los seis tipos de variables se importan correctamente.	Todas las variables deberían importarse sin lanzar ningún error.	El comportamiento es correcto.
3	Comprobar el funcionamiento del caso en el que una variable de Eurecca se divide en dos en RENACE.	La variable de Eurecca debería dividirse y asignar a cada variable de RENACE su valor correspondiente.	El comportamiento es correcto.
4	Referencias de las variables múltiples en la base de datos.	Las variables múltiples deberían importarse a la base de datos haciendo referencia al usuario al paciente al que pertenecen.	El comportamiento es correcto.
5	Comprobar que se importan el mayor número de variables posibles.	Todas las variables con opción de ser importadas se añaden correctamente a nuestra base de datos.	Hay ciertas variables que se podrían importar que no han sido seleccionadas.

#### 6.4.4. Pruebas Entrega parcial 4

Recordar que en la cuarta entrega parcial se ha implementado la sección que muestra los informes.

Tabla 5- Pruebas Entrega parcial 4

#	Descripción	Resultado esperado	Observaciones
1	Comprobar que los cálculos están bien hechos y los datos mostrados son correctos.	Los datos mostrados son correctos.	El comportamiento es correcto.

#### 6.4.5. Pruebas Entrega parcial 5

Recordar que en la quinta entrega parcial se ha implementado la sección que muestra los gráficos.

Tabla 6- Pruebas Entrega parcial 5

#	Descripción	Resultado esperado	Observaciones
1	Los gráficos muestran los datos de todos los hospitales.	Todos los hospitales tienen su marca en todos los gráficos, incluso si su valor es 0.	El comportamiento es correcto.
2	Identificación de los datos.	Solo se marcarán los datos del usuario conectado, de manera que no pueda identificar a qué hospital pertenecen los demás datos.	El comportamiento es correcto.
3	Zoom de los gráficos.	Los gráficos pueden verse a pantalla completa al hacer <i>click</i> sobre su título. Y se saldrá de esta haciendo <i>click</i> sobre la "X" situada en la parte superior derecha.	El comportamiento es correcto.
4	Feedback de los gráficos.	Los títulos de los gráficos muestran el puntero con forma de mano. Al colocar el cursor encima de un punto del gráfico se mostrará una pestaña con los datos detallados.	El comportamiento es correcto.

#### 6.4.6. Pruebas con usuarios

Una vez hechas todas las correcciones que se veían oportunas por parte del desarrollador y del interesado, se ofreció el producto a los primeros usuarios y se les pidió que rellenasen una encuesta de satisfacción, donde puntuasen ciertos aspectos que el desarrollador y el interesado consideraban importantes, y, además, hiciesen propuestas con el fin de mejorar el registro.

En la Tabla 7 se muestran las opiniones de los usuarios agrupadas según su pertenencia o no al registro Eurecca, lo que condiciona el método utilizado para la inserción de datos.

Dentro de las respuestas hay que destacar el elevado grado de satisfacción de los usuarios. La única diferencia que llama la atención, aunque lógica, es el interés para la práctica clínica, mucho más acusado entre los hospitales que carecen de un registro previo.

Tabla 7- Resultados de la encuesta de satisfacción

	No EURECCA	EURECCA	TOTAL
Interés global	10	9,3 (8-10)	9,7 (8-10)
Intuitivo (manejo de la página Web)	9,6 (8-10)	9,5 (9-10)	9,7 (8-10)
Facilidad de rellenar las variables	9 (8-10)	9,3 (9-10)	9,1 (8-10)
Duplicidad del esfuerzo desde el EURECCA	No aplica	42,5% (20-70)	42,5% (20-70%)
Trabajo "extra" sobre la actividad clínica habitual	17,4% (2-30)	27,5% (10-70)	21,9% (2-70%)
Interés para la práctica clínica	9,6 (8-10)	7,8 (6-10)	8,8 (6-10)
Interés para publicaciones o registros multicéntricos	9 (7-10)	9,3 (9-10)	9,1 (7-10)
Interés de los Informes y gráficos generados automáticamente	9,2 (8-10)	9,8 (9-10)	9,4 (8-10)
Interés de extender el registro a otras patologías o secciones	10	9,7 (9-10)	9,9 (9-10)

(Datos expresados mediante media y rango)

Además de alguna sugerencia sobre la posible inclusión de nuevas variables concretas, se destaca la clara necesidad de ligar el registro con algún tipo de programa de cálculo de supervivencia global y libre de enfermedad (SPSS) basada en los datos de seguimiento.

La mayor parte de las sugerencias hacen referencia a la falta de ciertas variables en el formulario o a la posible mejora de alguna de ellas. Estas se discutirán con el interesado y se decidirá que variables merece la pena mejorar o añadir, analizando la mejora de calidad que puedan aportar al futuro del registro.

Otra parte de las sugerencias hacen referencia a la necesidad de poner este registro en manos de una empresa que pueda sacarle provecho y llevar un correcto manejo de los datos que interfieren. Éstas son propuestas con vistas al futuro, que se tendrán en cuenta más adelante, pero con las que ahora no se pueden tomar medidas.

En cuanto al diseño de la aplicación, la usabilidad de esta u otros aspectos que influyan directamente con el desarrollador solo se ha recibido una sugerencia: el aumento de tamaño de los símbolos que forman los gráficos.



# 7.

## Gestión y seguimiento del proyecto

En este capítulo se hará un análisis de la gestión llevada a cabo a lo largo del proyecto. Se detectarán los puntos en los que se haya desviado de la planificación, su repercusión en los plazos y la razón por la que no se había previsto. De esta manera, se tendrá más cuidado con estos aspectos en los futuros proyectos a realizar.

### 7.1. Gestión de la planificación

En este capítulo se hará un repaso a las desviaciones respecto a la planificación realizada en el comienzo del trabajo. A continuación, se muestra una lista con los factores que más han afectado a lo establecido en cronograma inicial.

**COVID-19:** la irrupción de la pandemia por coronavirus a principios de marzo cambió radicalmente la normalidad existente, y con ella cualquier tipo de planificación realizada. Los planes académicos cambiaron drásticamente y la carga de trabajo aumentó considerablemente, por el número de tareas enviadas a los alumnos. Esta repentina carga de trabajo, con plazos muy ajustados, supuso una pausa casi al completo en el desarrollo del TFG, lo que provocó cierto retraso en los plazos establecidos.

Además, parte del desarrollo de este registro pasaba por la parte de prueba a manos de los cirujanos. Para realizar estas pruebas, al tratarse de datos personales, se necesitaba un certificado por parte de los respectivos Comités de ética que permitiese la inserción de pacientes en el registro RENACE, (certificado que puede consultarse en la sección de anexos, Figura 49). Este proceso de certificación tanto del registro como de los usuarios de prueba se retrasó considerablemente, haciendo que las pruebas se pospusiesen y con ellas, las posteriores fases de corrección de errores previstas.

**Importación desde Eurecca:** este retraso fue causado por una falta de análisis durante el proceso de planificación y una subestimación al tiempo que llevaría la importación de datos de Eurecca a RENACE.

Debido a esa falta de previsión, se tuvo que dedicar mucho tiempo que no estaba en los planes al diseño de una solución viable para este objetivo imprescindible establecido por el interesado. Junto a eso, el hecho de que la implementación pasase por el trabajo con un archivo Excel, que implicaba la utilización de una librería nueva para el desarrollador, no ayudó a agilizar esta carga de trabajo extra.

**Gráficos:** la implementación de los gráficos trajo consigo dos variaciones respecto al planteamiento inicial, una positiva, ya que una parte se realizó más rápida de lo esperado, y una negativa, ya que otra de las partes requirió más tiempo del previsto.

En cuanto a la positiva, al encontrar la librería CanvasJS [17], que ayudaría a la generación de datos, se vio que traía una gran variedad de ejemplos y una documentación de calidad que aligeraría el proceso de familiarización con la herramienta. Sin embargo, no se esperaba que el uso de esta librería de la que no se tenía conocimiento previo fuese tan fácil, por lo que se requiso menos tiempo del esperado para diseñar esta generación de datos y amoldarlos a las exigencias del interesado. Esto ayudó a recuperar parte del tiempo perdido por las demás desviaciones negativas que habían sucedido hasta el momento.

Aun así, seguido de esta variación positiva en la planificación, vino la negativa. Este tiempo no previsto vino de la mano de las consultas a la base de datos para obtener los datos que se mostrarían en los gráficos. El principal problema que trajeron estas consultas fue que algunos hospitales no tenían datos en todas las estadísticas que se reflejaban en los gráficos, por lo que resultó un problema para llevar a cabo la solución diseñada en un principio. En busca de esta solución se acudió desde los foros más comunes [18] hasta otros que nunca antes se habían visitado, y al no encontrar una solución que se adecuase al caso concreto que se enfrentaba, se tuvo que diseñar e implementar una solución propia, lo que llevó más tiempo del esperado.

Y, por último, al ver la claridad con la que estos gráficos mostraban las diferentes estadísticas de la base de datos, el interesado propuso la creación de más gráficos de los que se habían acordado en un principio, creando finalmente siete gráficos.

**Estilos:** los estilos han ido causando problemas que se han ido solucionando a lo largo de todo el desarrollo, debido a la inexperiencia a la hora de crear el diseño de una página desde cero. Aun así, se han utilizado ayudas como las de Bootstrap para facilitar el diseño de ciertos elemento y estructuras, pero no ha evitado que se tenga que dedicar tiempo a la corrección de pequeños errores y, sobre todo, en intentar que el registro mantuviese una imagen correcta en el mayor número de buscadores posibles.

Cuantificando en horas la desviación causada por estos factores comentados, solo cabe comentar la referente a la exportación de los pacientes desde Eurecca. El trabajo extra que supuso por la parte de diseño, implementación y el aprendizaje de la nueva herramienta utilizada, conllevó a una desviación de alrededor de 50 horas. Las demás desviaciones no han supuesto un aumento significativo en el número de horas.

Por lo tanto, teniendo en cuenta que en un principio se previó que se dedicarían un total de 352 horas, el número de horas totales ha acabado siendo de 400.

## 7.2. Gestión del alcance

Dejando a un lado las desviaciones comentadas, en este apartado se analizará la gestión del alcance de este trabajo, echando la vista atrás a lo especificado en la sección 2.2 y analizando si se ha cumplido o no lo establecido.

En cuanto a la integración del registro en la AEC, el camino para lograrlo ya ha asentado sus bases durante el desarrollo de este TFG, logrando una excelente opinión por parte de los usuarios que lo han probado y cumpliendo los objetivos del interesado.

En lo que a la usabilidad de la página web se refiere, se considera que se ha conseguido hacer una interfaz limpia, sin distracciones y que permite centrarse en lo que es verdaderamente importante, haciendo así más llevadera y ágil la interacción entre el usuario y el registro.



La anonimización de los datos, tanto en el aspecto de su almacenamiento, como a la hora de limitar lo que cada usuario puede ver sobre los demás hospitales, se ha cumplido satisfactoriamente. En la base de datos no se guarda ningún dato identificativo, y será cada cirujano el que guarde de manera local la relación entre código y el nombre del paciente. En cuanto a la interfaz, no se permite a los usuarios acceder a los pacientes que no les pertenecen, y tanto en los informes como en los gráficos solo serán capaces de identificar los datos de su propio hospital.

Por último, y el punto más importante, se han conseguido cumplir todos los requisitos acordados con el interesado. De hecho, durante el desarrollo algunos de estos objetivos han sufrido cambios, como el número de variables a implementar, o qué estadísticas mostrarían los gráficos, y estos se han gestionado correctamente consiguiendo completar todos ellos. El objetivo opcional que supone la expansión del registro a otras secciones de la AEC es una cuestión que debe decidirse en la Asamblea General Ordinaria 2020, planificada para el mes de noviembre.

### 7.3. Gestión de los sistemas de información

La comunicación con la tutora y con el interesado durante el comienzo del desarrollo ha ido tal y como se había previsto, utilizando el mail para dudas puntuales y las tutorías o reuniones para la discusión de cuestiones más profundas.

Sin embargo, a causa del ya comentado COVID-19, estos sistemas de comunicación han tenido que ser actualizados para acomodarse a lo que ahora se llama la 'nueva normalidad'. El intercambio de información con la tutora se siguió haciendo a través del mail, pero las reuniones pasaron a hacerse vía BlackBoard Collaborate, plataforma que ha aportado la universidad para realizar tanto clases como reuniones durante este periodo de cuarentena.

Por otro lado, las reuniones con el interesado han sido más comunes y fáciles de lo previsto, ya que en un principio se iba a estar en provincias diferentes, lo que hacía complicadas las reuniones, pero en consecuencia de esta pandemia, se han podido hacer reuniones prácticamente dos veces por semana, para discutir cualquier aspecto del diseño y el progreso del registro.



# 8.

## Conclusiones

En este capítulo se sacan las conclusiones de los diferentes aspectos que han formado parte del desarrollo de este trabajo de fin de grado.

### 8.1. Conclusión general

Como conclusión que aborde todos los aspectos, se podría decir que el desarrollo de este trabajo de fin de grado ha acabado de la mejor manera posible. Se han cumplido todos los objetivos principales, y se ha dejado el camino preparado para completar en un futuro los demás objetivos solicitados por parte del interesado. Además, como se ha visto en las pruebas a los usuarios, y como se recoge en la encuesta de satisfacción, la satisfacción por parte de los cirujanos que han probado el registro es prácticamente absoluta, con pocas cosas a mejorar en lo que a las tareas del alumno desarrollador se refiere, pero a su vez, con la posibilidad de haber abierto un gran camino a recorrer.

### 8.2. Tecnologías y metodologías utilizadas

Las tecnologías utilizadas durante el desarrollo del registro han cumplido su labor tal y como se esperaba, desde las aplicaciones conocidas anteriormente hasta las herramientas que se han utilizado por primera vez como la librería CanvasJS o la aplicación web projectmanager.com para la creación del diagrama Gantt.

Sin embargo, ha habido una aplicación con la que no se había contado desde el principio, y que se comenzó a utilizar tras el mal funcionamiento de la herramienta utilizada hasta el momento. Este cambio se produjo en el sistema de transferencia de datos del ordenador al servidor provisto por la AEC. En un principio se utilizó el sistema propio de Windows, pero tras unos días de prueba se vio que era muy lento y con muy pocas facilidades, por lo que se optó por pasar a FileZilla, herramienta que facilita este intercambio de archivos de la que ya se ha hablado en el capítulo 5.2.4.

### 8.3. Respecto al grado

Como bien se ha explicado a lo largo de esta memoria, la implementación de este registro gira en torno al desarrollo de una página web con su base de datos correspondiente. Por ello, las asignaturas que más presentes se han tenido han sido las de Sistemas Web, Bases de Datos y Diseño de Bases de Datos. Además, debido a la gestión que se ha tenido que realizar a lo largo de todo el desarrollo también se han tenido que aplicar los conocimientos aprendidos en la asignatura Gestión de Proyectos.

Como conclusión en cuanto al grado se refiere, se ha de remarcar la importancia que se les dan a todas y cada una de las asignaturas cursadas ahora que echa la vista atrás. Ya que muchas veces, como estudiante, no se encuentra el sentido a todas las asignaturas o ciertos aspectos que se tratan en estas, y no es hasta que pasa el tiempo y se mira atrás, desde un punto de vista

diferente, cuando uno se da cuenta de que todas aportan conocimiento, herramientas y nuevas experiencias que luego serán útiles, en mayor o menor medida, a la hora de tener que afrontar las dificultades que se nos presentan en situaciones más complejas como puede ser el completo desarrollo de un trabajo de fin de grado.

#### 8.4. Futuro de la aplicación

Presentado el registro RENACE a una parte representativa de la Junta Directiva que será ratificada en la próxima Asamblea General de la AEC (presidente electo y próxima presidenta del Comité Científico incluidos), se muestra como un arma fácil de aplicar, necesaria y que ofrece un *feedback* inmediato, ayudando a saber cuáles son los principales indicadores, cómo se sitúa cada hospital en relación a los demás y respecto a los estándares internacionales reconocidos (los denominados “benchmarks”), y a observar de forma gráfica con qué factores se relacionan las complicaciones o los resultados de efectividad a largo plazo.

Si se consigue garantizar la confidencialidad de los datos según la ley y los derechos de los pacientes, si se limitan las variables y se generaliza su uso, el registro RENACE representará el germen del tan anhelado Registro Nacional de Cirugía, como marca de identidad de la AEC y recurso incomparable para poner a disposición de todos y todas la experiencia y resultados de todos y todas, plasmada de forma objetiva y visual, de tal modo que puedan evolucionar los estándares de tratamiento dentro de cada patología.

Manteniendo apartados comunes (datos demográficos, estudio de extensión, anatomía patológica, seguimiento) y otros específicos de cada patología (tipo de tratamiento o técnica quirúrgica), el RENACE podría evolucionar hacia un REN-AEC (Registro Nacional AEC). La solicitud de un presupuesto y el desarrollo del proyecto ya han sido encargados al coordinador de Cirugía esófago-gástrica, que ha participado activamente en este TFG.

# Bibliografía

1. National Oesophago-Gastric Cancer Audit 2018. An audit of the care received by people with Oesophago-Gastric Cancer in England and Wales: 2018 Annual Report. National Health System, United Kingdom, 2019.
2. Dutch Upper GI Cancer Audit (DUCA). 2017 Yearly Report.
3. Van der Werf LR, Busweiler LAD, van Sandick JW, van Berge Henegouwen MI, Wijnhoven BPL and the Dutch Upper GI Cancer Audit (DUCA) group. Reporting national outcomes after esophagectomy and gastrectomy according to the Esophageal Complications Consensus Group (ECCG). Ann Surg. 2019 doi: 10.1097/SLA.0000000000003210
4. Messager M, de Steur W, Boelens PG, Jensen LS, Mariette C, Reynolds JV et al. Description and Analysis of Clinical Pathways for Oesophago-Gastric Adenocarcinoma, in 10 European Countries (The EURECCA Upper Gastro Intestinal Group - European Registration of Cancer Care). EJSO 2016;42:1432-47.
5. De Steur WO, Hennemann D, Allum WH, Dikken JL, Van Sandick JW, Reynolds J et al. Common Data Items in Seven European Oesophagogastric Cancer Surgery Registries: Towards a European Upper GI Cancer Audit (EURECCA Upper GI). EJSO 2014;40:325-9.
6. Escuela de Negocios de la Innovación y los Emprendedores (IEBS). Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa. En: <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>
7. Teamleader. ¿Qué es y para qué sirve un diagrama de Gantt? En: <https://blog.teamleader.es/diagrama-de-gantt>
8. Charlson ME, Pompei P, Ales KL, MacKenzie CR. A new method of classifying prognostic comorbidity in longitudinal studies: development and validation. J Chronic Dis 1987;40:373-83.
9. PHP Documentation Group. Manual de PHP. En: Cowburn P Eds, 1997-2020. <https://www.php.net/manual/es/mysqli.overview.php>
10. OpenJS Foundation. En: <https://jquery.com/>
11. FileZilla. En: <https://filezilla-project.org/>
12. Bootstrap. En: <https://getbootstrap.com/docs/4.5/getting-started/introduction/>
13. Colceriu I. CSS Toggle Switch. En: <https://ghinda.net/css-toggle-switch/index.html>
14. PhpSpreadsheet's documentation. En: <https://phpspreadsheet.readthedocs.io/>
15. Schmidt HM, Gisbertz SS, Moons J, Rouvelas I, Kauppi J, Brown A et al. Defining benchmarks for transthoracic esophagectomy. A multicenter analysis of total minimally invasive esophagectomy in low risk patients. Ann Surg 2017;266:814-21.
16. Low DE, Kuppusamy MK, Alderson D, Cecconello I, Chang AC, Darling G et al. Benchmarking complications associated with esophagectomy. Ann Surg 2019;269:291-8.
17. Fenopix. Charts for PHP Applications. En: <https://canvasjs.com/php-charts/>
18. Stackoverflow. En: <https://stackoverflow.com/>



# Anexos

Este último apartado recoge una serie escogida de anexos con información de interés tanto para las partes interesadas actualmente en este proyecto, como futuros desarrolladores que deseen extenderlo. La recopilación en concreto es la siguiente:

1. Informe favorable del CEImE a la realización de este proyecto
2. Diagrama de Gantt detallado
3. Formulario completo "Añadir nuevo paciente"
4. Modelo de datos íntegro de la base de datos de RENACE
5. Gráficos

## 1. Informe favorable del CEImE a la realización de este proyecto



### **INFORME DEL COMITÉ DE ÉTICA DE LA INVESTIGACIÓN CON MEDICAMENTOS DE EUSKADI (CEIm-E)**

Nagore García Goitiandia  
Secretaria del CEIm de Euskadi (CEIm-E)

#### **CERTIFICA**

Que este Comité, de acuerdo a la ley 14/2007 de Investigación Biomédica, Principios éticos de la declaración de Helsinki y resto de principios éticos y legislación aplicables, ha evaluado el proyecto de investigación titulado **Registro Nacional de Cirugía Esofágica de la Asociación Española de Cirujanos (RENACE)**.

Código interno: PI2020046 (DOCENTE)

Versión del Protocolo: Versión 1 de 17 de febrero de 2020

Versión de la Hoja de Información al Paciente: GENERAL / Versión 2.2020

Y que este Comité reunido el día 08/04/2020 (recogido en Acta 06/2020) ha decidido emitir informe favorable a que dicho proyecto sea realizado por el siguiente personal investigador:

- Jon Díez Barrios (*Sistemas Informáticos*) Universidad del País Vasco UPV/EHU
- José Ignacio Asensio Gallego (*Cirugía General y Aparato Digestivo*) Hospital Universitario Donostia
- Emma Eizaguirre Letamendia (*Cirugía General y Aparato Digestivo*) Hospital Universitario Donostia
- Ismael Díez del Val (*Cirugía General y Aparato Digestivo*) Hospital Universitario Basurto
- Saioa Leturio Fernández (*Cirugía General y Aparato Digestivo*) Hospital Universitario Basurto
- Gaizka Errazti Olartekoetxea (*Cirugía General y Aparato Digestivo*) Hospital Universitario Cruces
- Aingeru Sarriugarte Lasarte (*Cirugía General y Aparato Digestivo*) Hospital Universitario Cruces
- Lorena Reka Mediavilla (*Cirugía General y Aparato Digestivo*) Hospital Universitario de Araba
- José M<sup>a</sup> Vitores López (*Cirugía General y Aparato Digestivo*) Hospital Universitario de Araba

Lo que firmo en Vitoria, a 22 de abril de 2020



Nagore García Goitiandia  
Secretaria del CEIm de Euskadi (CEIm-E)

**Nota:** Se recuerda la obligación de:

- Incluir en Osabide la alerta correspondiente a cada paciente, de que se encuentra bajo estudio o ensayo clínico
- Enviar un **informe de seguimiento anual** y el **informe final** que incluya los resultados del estudio (si el estudio dura menos de un año, con el informe final será suficiente). Más información en la página web del CEIm-E: <http://www.euskadi.eus/comite-etico-investigacion-clinica/>

Figura 49- Informe favorable del CEImE a la realización de este proyecto



## 2. Diagrama de Gantt detallado

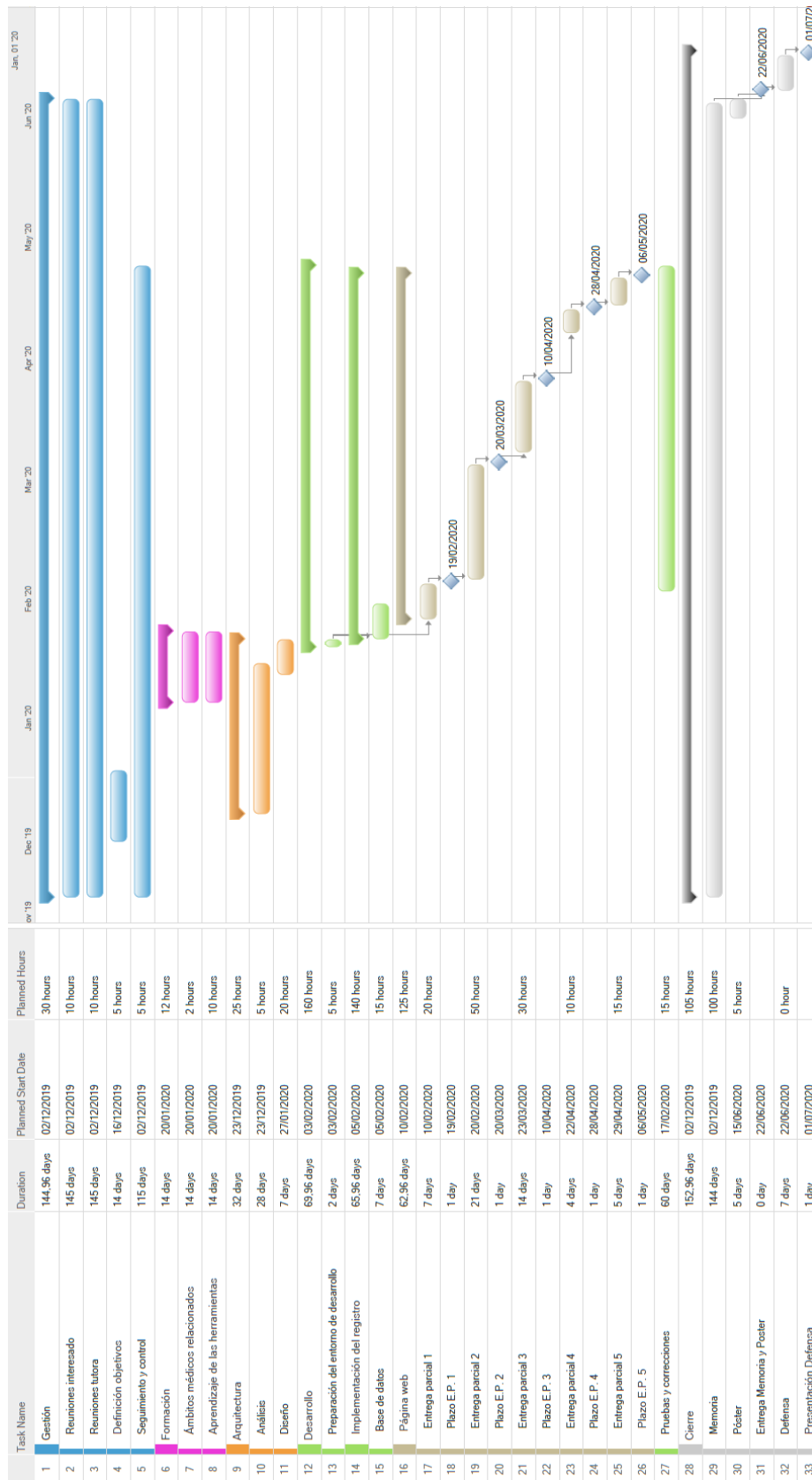




Figura 51- Diagrama de Gantt detallado

### 3. Formulario completo “Añadir nuevo paciente”



ASOCIACIÓN ESPAÑOLA  
DE CIRUJANOS

Registro Nacional de Cirugía Esofágica



Jordi

Inicio

Pacientes

Gráficas

Informes

## Añadir nuevo paciente

---

**Datos del paciente**

Centro/Hospital

Edad

Sexo

Peso

Talla  cm

IMC

Situación Funcional

Valoración nutricional subjetiva

Índice ASA

**Comorbilidad**

- Infarto de miocardio
- Insuficiencia cardíaca congestiva
- Enfermedad vascular periférica
- Enfermedad cerebrovascular
- Demencia
- Enfermedad pulmonar crónica
- Patología del tejido conectivo
- Enfermedad ulcerosa
- Patología hepática ligera
- Patología hepática moderada o grave
- Diabetes
- Diabetes con lesión orgánica
- Hemiplegia
- Patología renal (moderada o grave)
- Neoplasias
- Leucemias
- Linfomas malignos
- Metastasis sólida
- SIDA

**Índice de Comorbilidad de Charlson**

Revisar la edad para poder aplicar el cálculo.

---

**Diagnóstico y estudio de extensión (datos de la enfermedad)**

Fecha del diagnóstico

Localización del tumor

Hemoglobina

Al diagnóstico  g/dl

Próximo a la cirugía  g/dl

Albúmina preoperatoria  g/dl

Ca 19.9  U/ml

Estadificación clínica (cTNM)

cT

cN  Negativo Positivo

cM  M0 Indet. M1

**Pruebas diagnósticas**

- Endoscopia
- TC
- Fosfoscopia
- Resección mucosa/submucosa
- Broncoscopia
- Laparoscopia exploradora
- PET TC
- Gammagrafía ósea
- MRI
- Citología biopsia

---

**Optimización preoperatoria**

Transfusión preoperatoria

(Número de unidades)

Ferrotterapia

Soporte nutricional

---

**Tratamiento neoadyuvante**

Fecha inicio

Fecha fin

Tipo de tratamiento

---

**Cirugía**

Fecha

Reconstrucción

Yeyunostomía de alimentación

Intervención quirúrgica asociada

**Abordaje**

Abdomen

Tórax

Posición

**Linfadenectomía**

Abdominal:

Torácica:

Cervical:  No Sí

**Anastomosis**

Localización

Técnica

---

**Anatomía Patológica**

Figura 52- Parte superior del formulario completo “Añadir nuevo paciente”

### Anatomía Patológica

---

<p><b>Grado de diferenciación</b>  <input type="text" value="Seleccione una opción"/></p> <p><b>Número de ganglios resecaados</b>  <input type="text"/></p> <p><b>Radicalidad de la resección</b>  <input type="text" value="Seleccione una opción"/></p> <p><b>Invasión perineural</b>  <input type="checkbox"/> 0 <input checked="" type="checkbox"/> 1</p> <p><b>Determinación de HER-2</b>  <input checked="" type="checkbox"/> Negativo <input type="checkbox"/> Positivo</p>	<p><b>p/ypT</b>  <input type="text" value="Seleccione una opción"/></p> <p><b>Nº de ganglios afectados por el tumor</b>  <input type="text"/></p> <p><b>Invasión linfática</b>  <input type="checkbox"/> 0 <input checked="" type="checkbox"/> 1</p> <p><b>Presencia de Barrett adyacente</b>  <input type="checkbox"/> No <input checked="" type="checkbox"/> Sí</p>	<p><b>p/ypN</b>  <input type="text" value="Seleccione una opción"/></p> <p><b>pM</b>  <input type="checkbox"/> 0 <input checked="" type="checkbox"/> 1</p> <p><b>Invasión vascular</b>  <input type="checkbox"/> 0 <input checked="" type="checkbox"/> 1</p> <p><b>Grado de regresión tumoral de Mandard (GRT)</b>  <input type="text" value="Seleccione una opción"/></p>
--	---	--

### Evolución postoperatoria

---

<p><b>Fecha de alta hospitalaria</b>  <input type="text" value="dd/mm/aaaa"/></p> <p><b>Complicaciones anastomóticas (fistula)</b>  <input type="text" value="Seleccione una opción"/></p> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> <p style="text-align: center; margin: 0;"><b>Complicaciones respiratorias</b></p> <p><input type="checkbox"/> Derrame</p> <p><input type="checkbox"/> Neumotórax</p> <p><input type="checkbox"/> Fistula TL</p> <p><input type="checkbox"/> Fuga aéreas mantenida</p> <p><input type="checkbox"/> Neumonía</p> <p><input type="checkbox"/> Nelelectasia</p> <p><input type="checkbox"/> SEDRA</p> <p><input type="checkbox"/> TTP</p> </div> <p><b>Complicaciones urológicas</b>  <input type="text" value="Seleccione una opción"/></p> <p><b>Complicaciones infecciosas</b>  <input type="text" value="Seleccione una opción"/></p> <p><b>Otras</b>  <input type="checkbox"/> No <input checked="" type="checkbox"/> Sí</p> <p><b>Reintervención</b>  <input type="checkbox"/> No <input checked="" type="checkbox"/> Sí</p>	<p><b>Complicaciones (Clavien-Dindo)</b>  <input type="text" value="Seleccione una opción"/></p> <p><b>Tratamiento de la fistula</b>  <input type="text" value="Seleccione una opción"/></p> <p><b>Tratamiento de complicación pulmonar</b>  <input type="text" value="Seleccione una opción"/></p>	<p><b>Complicaciones hemorrágicas</b>  <input type="text" value="Seleccione una opción"/></p>
--	---	---

### Seguimiento

---

<p><b>Tratamiento adyuvante</b>  <input type="text" value="Seleccione una opción"/></p>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> <p style="text-align: center; margin: 0;"><b>Tipo de recidiva</b></p> <p><input type="checkbox"/> No</p> <p><input type="checkbox"/> Local</p> <p><input type="checkbox"/> Regional</p> <p><input type="checkbox"/> Carcinomatosis</p> <p><input type="checkbox"/> Sistémica</p> </div> <p><b>Fecha recidiva</b>  <input type="text" value="dd/mm/aaaa"/></p> <p><b>Tratamiento de la recidiva</b>  <input type="text" value="No"/></p>	<p><b>Fecha último control</b>  <input type="text" value="dd/mm/aaaa"/></p> <p><b>Estado</b>  <input type="text" value="Seleccione una opción"/></p> <p><b>Fecha exitus</b>  <input type="text" value="dd/mm/aaaa"/></p>
---	--	--

Figura 53- Parte inferior del formulario completo "Añadir nuevo paciente"

#### 4. Modelo de datos íntegro de la base de datos de RENACE

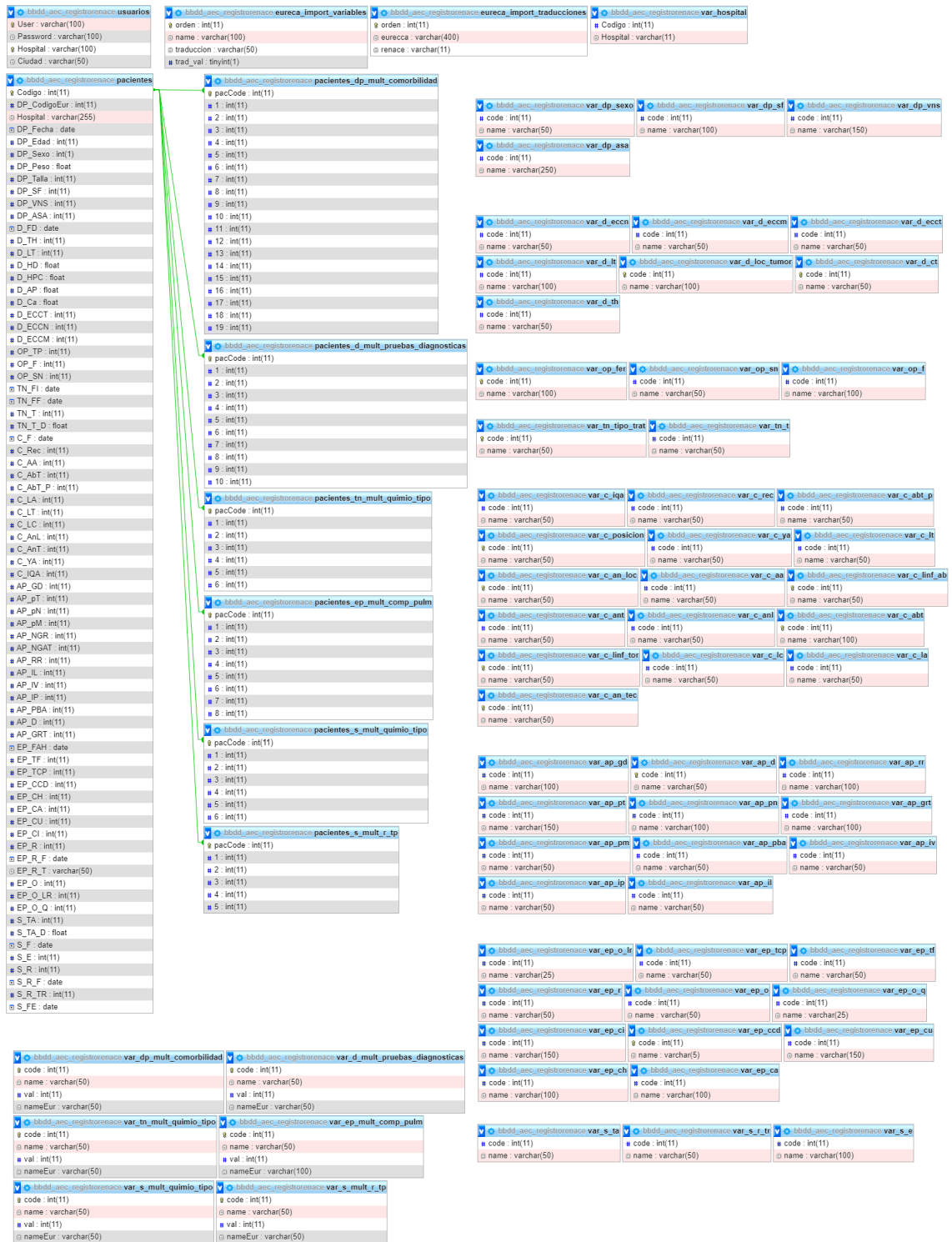


Figura 54- Estructura de la base de datos

## 5. Gráficos

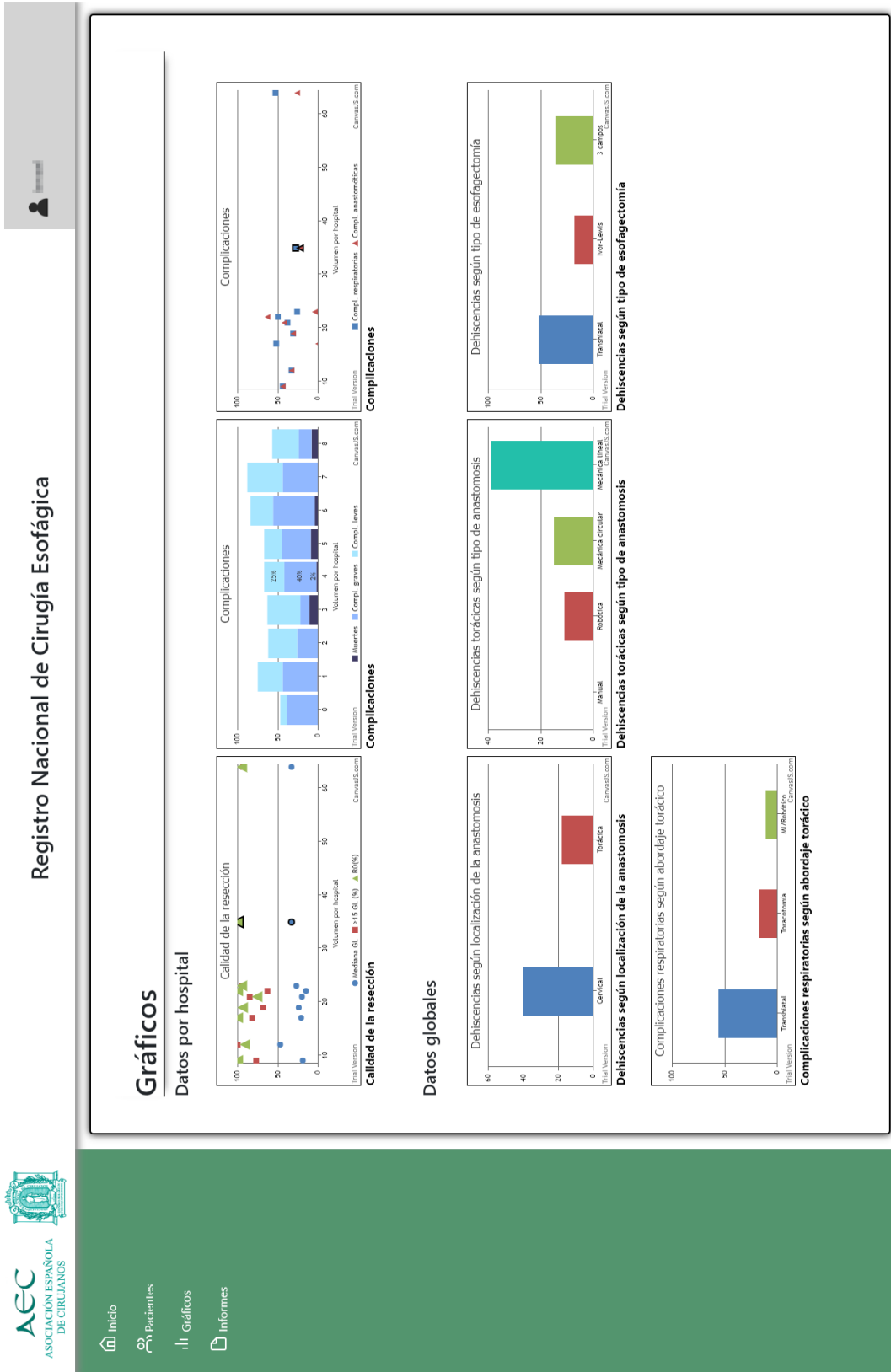


Figura 55- Página de visualización general de Gráficos