



GRADO EN INGENIERÍA INFORMÁTICA

TRABAJO FINAL DE GRADO

**Desarrollo de herramienta informática para
la adquisición de competencias en la
interpretación del electrocardiograma en el
estudiante de Medicina**

Autor:
Albert ESCRICH CLIMENT

Supervisor:
David SAEZ BAIXAULI
Tutor académico:
Begoña MARTÍNEZ SALVADOR

Fecha de lectura: 17 de Septiembre de 2019
Curso académico 2018/2019

Resumen

En esta memoria se describe la estancia en prácticas realizada en el Hospital General Universitario de Castelló (HGUCS). Durante la estancia, se ha realizado el diseño e implementación de un sistema para la adquisición de competencias en la interpretación del electrocardiograma en el estudiante de Medicina. Para la implementación de este sistema, se ha utilizado un conjunto de tecnologías, como son los lenguajes Java, JavaScript, HTML, CSS o el sistema de gestión de bases de datos PostgreSQL.

Palabras clave

Aprendizaje interactivo, electrocardiograma, MVC, JavaScript.

Keywords

Interactive learning, electrocardiogram, MVC, JavaScript.

Índice general

1. Introducción	11
1.1. Contexto y motivación del proyecto	11
1.1.1. Empresa	11
1.1.2. Proyecto	11
1.2. Objetivos del proyecto	12
1.3. Estructura de la memoria	13
2. Descripción del proyecto	15
2.1. Situación inicial del proyecto	15
2.2. Tecnologías y herramientas utilizadas en el proyecto	15
2.2.1. HTML5	15
2.2.2. CSS	16
2.2.3. JavaScript	16
2.2.4. PostgreSQL	16
2.2.5. Java	16
2.2.6. Apache Tomcat	17
2.2.7. UMLet	17
2.2.8. MockFlow	17

3. Planificación del proyecto	19
3.1. Metodología	19
3.2. Planificación	19
3.3. Estimación de recursos y costes del proyecto	22
3.3.1. Recursos	22
3.3.2. Costes	22
3.4. Seguimiento del proyecto	23
4. Análisis y diseño del sistema	25
4.1. Análisis del sistema	25
4.1.1. Casos de uso del sistema	25
4.1.2. Requisitos de datos del sistema	36
4.2. Diseño de la arquitectura del sistema	38
4.2.1. Diseño lógico de la base de datos	39
4.2.2. Diseño físico de la base de datos	40
4.3. Diseño de la interfaz	42
4.3.1. Interfaces comunes	42
4.3.2. Estudiante	43
4.3.3. Profesor	46
4.3.4. Administrador	48
5. Implementación y pruebas	51
5.1. Detalles de implementación	51
5.1.1. Periodo 1	51
5.1.2. Periodo 2	51
5.1.3. Periodo 3	53

5.1.4. Periodo 4	55
5.1.5. Periodo 5	56
5.1.6. Periodo 6	58
5.2. Verificación y validación	59
5.2.1. Trabajo futuro	59
6. Conclusiones	61
A. Guía de usuario de la herramienta	65
A.1. Estudiante	65
A.2. Profesor	65
A.2.1. Introducción de un nuevo caso	65
A.2.2. Añadir una nueva pregunta a un caso	66

Índice de figuras

3.1. Diagrama de Gantt.	21
4.1. Diagrama de casos de uso del sistema	26
4.2. Diagrama de clases del sistema	38
4.3. Interfaz de la pagina de inicio	42
4.4. Interfaz de la pagina de acceso	42
4.5. Interfaz del índice de contenidos	43
4.6. Interfaz de analizar y responder caso	43
4.7. Interfaz visualizar la corrección al responder incorrectamente	44
4.8. Interfaz visualizar la explicación al responder correctamente	44
4.9. Interfaz visualizar perfil y cambiar contraseña	45
4.10. Interfaz visualizar la lista de casos	46
4.11. Interfaz añadir nuevo caso	46
4.12. Interfaz añadir nueva pregunta	47
4.13. Interfaz gestionar usuarios	48
4.14. Interfaz nuevo usuario	48
4.15. Interfaz editar usuario	49
4.16. Interfaz reiniciar contraseña	49
5.1. Herramienta pgAdmin.	52

5.2. Diseño interfaz administrar casos.	52
5.3. Diseño interfaz nuevo caso.	53

Índice de tablas

3.1. Lista de desglose de tareas, su duración y dependencias entre ellas.	20
3.2. Costes de los recursos del proyecto.	23
4.1. Resumen casos de uso.	27
4.2. CU1. Introducir un caso práctico.	28
4.3. CU2. Modificar los datos de un caso práctico.	29
4.4. CU3. Visualizar y analizar el ECG mostrado por la aplicación.	30
4.5. CU4. Responder las preguntas presentadas.	31
4.6. CU5. Visualizar la corrección o explicación de la pregunta.	32
4.7. CU6. Administrar perfil.	33
4.8. CU7. Administrar estudiantes.	34
4.9. CU8. Administrar profesores.	35
4.10. RD01. Profesor	36
4.11. RD02. Estudiante	36
4.12. RD03. Caso práctico	36
4.13. RD04. Pregunta	36
4.14. RD05. Respuesta	37
4.15. RD06. Explicación	37
4.16. Relaciones de la base de datos.	39

Capítulo 1

Introducción

1.1. Contexto y motivación del proyecto

En este apartado se describe la empresa en la que se ha desarrollado la estancia en prácticas y el proyecto desarrollado, presentando las necesidades que lo han motivado.

1.1.1. Empresa

En 1967 se inauguró la residencia sanitaria de Nuestra Señora del Sagrado Coro, el primer hospital público de referencia para ingresos de la ciudad de Castellón. En 1986 éste es reinaugurado como Hospital General. El Hospital General Universitari de Castelló (UGUCS) es un hospital ubicado en la ciudad de Castellón de la Plana, es el centro de referencia del Departamento de Salud de Castellón.

La estancia ha sido realizada en la unidad de informática del hospital, la cual se encarga, con la colaboración en la atención y soporte de la empresa externa Dedalus GS, de garantizar el óptimo funcionamiento de las redes y aplicaciones de los Sistemas de Información Sanitarios en el Hospital General así como en los centros de salud que de éste dependen.

1.1.2. Proyecto

El proyecto realizado y descrito en este documento consiste en el desarrollo de una herramienta para ayudar a los estudiantes de Medicina a adquirir las competencias correspondientes a la interpretación de electrocardiogramas. Según los expertos, la interpretación correcta del ECG es una de las habilidades más complejas de aprender.

El método de aprendizaje actual consiste en una colección de casos prácticos, recopilados en un cuaderno físico, divididos en secciones con los distintos aspectos a tener en cuenta. Estos casos están acompañados de su informe completo.

Tomando esto como punto de partida se quiere desarrollar una herramienta que permita al estudiante visualizar el electrocardiograma y analizarlo tomando medidas sobre la propia imagen del electrocardiograma. Ésta es una de las características más importantes, incluso la más importante, de la herramienta. Con las conclusiones obtenidas de este análisis, el estudiante responderá a una pregunta de tipo test relacionada con el caso práctico, y, dependiendo de si la respuesta es correcta o incorrecta, se le proporcionará la explicación completa de la pregunta o solamente la corrección de la respuesta errónea que haya elegido.

Esta herramienta beneficiará a los alumnos ya que podrán recibir feedback inmediato mientras aprenden a analizar electrocardiogramas mediante ejemplos y pruebas sobre distintos aspectos presentes en el electrocardiograma que se ilustre. La herramienta deberá permitir el aprendizaje según los resultados previos y el nivel del alumno.

El alcance de este proyecto incluye todos los elementos a desarrollar para realizar el sistema, como son el diseño de la base de datos, el diseño de las interfaces de usuario o el desarrollo de la parte técnica de la aplicación web. El alcance no incluye la creación de los datos a introducir en la base de datos, los cuales serán suministrados por profesionales del área de cardiología.

1.2. Objetivos del proyecto

El objetivo principal del proyecto es desarrollar una herramienta informática que permita al estudiante y/o profesional sanitario adquirir progresivamente y de forma amena e interactiva los conocimientos prácticos y teóricos para analizar de manera sistemática el electrocardiograma. Esta competencia es necesaria para una correcta interpretación y una adecuada toma de decisiones clínicas.

El objetivo principal se puede dividir en varios objetivos parciales:

1. Diseñar e implementar una base de datos para los casos prácticos, almacenando el electrocardiograma, las preguntas y las respuestas, así como los datos de los estudiantes y su progreso.
2. Diseñar e implementar un sistema de medición directo sobre los electrocardiogramas.
3. Diseñar e implementar un sistema que muestre a los alumnos los electrocardiogramas de manera adecuada, teniendo en cuenta el progreso de cada alumno para así ir adquiriendo las competencias progresivamente a la par que aumenta este nivel.
4. Diseñar e implementar una herramienta para que los médicos o administradores puedan mantener la base de datos de electrocardiogramas, pudiendo añadir, borrar o modificar casos y las preguntas y respuestas pertenecientes a estos.

1.3. Estructura de la memoria

En el capítulo 2 de la memoria se describirá el proyecto, presentando el entorno en el que se realiza, su motivación, objetivos y alcance. También se describirá la metodología y las tecnologías que se utilizan en el desarrollo del proyecto.

En el capítulo 3 se presentará la planificación del proyecto, tanto la planificación inicial como las conclusiones sacadas sobre ésta al finalizar. También se realizará un cálculo de los recursos utilizados y los costes de éstos.

En el capítulo 4 se muestra el análisis del sistema y se describe la arquitectura utilizada para la base de datos. Además se presentan las interfaces de usuario creadas para la herramienta.

En el capítulo 5 se presenta la implementación del sistema, detallando paso a paso como se ha ido desarrollando la aplicación utilizando las tecnologías escogidas.

En el capítulo 6 se encuentran las conclusiones sacadas al finalizar este proyecto, tanto en el ámbito formativo y profesional como en el personal.

Capítulo 2

Descripción del proyecto

Se describe con detalle cuales van a ser las tecnologías a usar en el proyecto y la situación inicial a partir de la cual se va a desarrollar el proyecto.

2.1. Situación inicial del proyecto

La realización de este proyecto ha sido solicitada por Patricia Palau Sampio y Eloy Domínguez Mafé, cardiólogos del Hospital General de Castelló y profesores asociados de la Universitat Jaume I.

El método de aprendizaje de las competencias, actualmente, consiste en un libro [1] cuyos autores son los anteriormente presentados. Este libro reúne una gran cantidad de casos prácticos, divididos por aspectos de los electrocardiogramas, todos ellos analizados y con un informe completo.

Los objetivos de la herramienta, descritos en el capítulo anterior, son presentados al inicio del proyecto por los cardiólogos mediante una presentación.

2.2. Tecnologías y herramientas utilizadas en el proyecto

2.2.1. HTML5

HTML, que significa Lenguaje de Marcado para Hipertextos (HyperText Markup Language) es el elemento de construcción más básico de una página web y se usa para crear y representar visualmente una página web. Determina el contenido de la página web, pero no su funcionalidad. Otras tecnologías distintas de HTML son usadas generalmente para describir la apariencia/presentación de una página web (CSS) o su funcionalidad (JavaScript) [2].

2.2.2. CSS

Como ya hemos mencionado, CSS es un lenguaje que se usa para diseñar la forma de presentar los documentos a los usuarios, su estilo, diseño, etc.

Un documento suele ser un archivo de texto estructurado usando un Lenguaje de marcado, HTML es el más común, pero hay otros como SVG o XML.

Presentar un documento a un usuario implica convertirlo en algo que este pueda utilizar. Navegadores (Browsers), como Firefox, Chrome o Internet Explorer, están diseñados para presentar documentos de forma visual, por ejemplo, en una pantalla de ordenador, un proyector o una impresora [3].

2.2.3. JavaScript

Javascript es un lenguaje de programación que surgió con el objetivo inicial de programar ciertos comportamientos sobre las páginas web, respondiendo a la interacción del usuario y la realización de automatismos sencillos. En ese contexto podríamos decir que nació como un "lenguaje de scripting" del lado del cliente, sin embargo, hoy Javascript es mucho más. Las necesidades de las aplicaciones web modernas y el HTML5 ha provocado que el uso de Javascript que encontramos hoy haya llegado a unos niveles de complejidad y prestaciones tan grandes como otros lenguajes de primer nivel [4].

2.2.4. PostgreSQL

PostgreSQL es un sistema de gestión de base de datos de código abierto que extiende el lenguaje SQL combinado con muchas otras características que almacenan de forma segura y escalable las cargas de datos mas complicadas. PostgreSQL es compatible con los principales sistemas operativos y se ha ganado una gran reputación debido a el uso probado de su arquitectura, integridad de datos y la robustez de su conjunto de características [5].

2.2.5. Java

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos diez millones de usuarios reportados [6].

2.2.6. Apache Tomcat

Apache Tomcat es una implementación de código abierto de las especificaciones de los servlets y de JavaServer Pages(JSP) de Oracle [7].

2.2.7. UMLet

UMLet es una herramienta UML gratuita con una interfaz de usuario simple, sirve para realizar diagramas UML o de uso fácilmente con la posibilidad de exportarlos en multitud de formatos [8].

2.2.8. MockFlow

MockFlow proporciona una herramienta para diseñar interfaces de usuario con elementos de distintos frameworks, también sirve para la creación de sitemaps, documentación y validaciones de diseño [9].

Capítulo 3

Planificación del proyecto

3.1. Metodología

Al tratarse de un proyecto a desarrollar desde cero, partiendo de las ideas de los médicos que solicitan la herramienta, la elección de la metodología a utilizar es una decisión propia. Se ha decidido utilizar una metodología predictiva ya que se trata de un proyecto unipersonal. En esta metodología se creará una lista inicial de tareas a realizar para desarrollar el proyecto. Se realiza una estimación de la duración de cada tarea, como veremos a continuación en la planificación.

3.2. Planificación

Al inicio del proyecto, se realiza una planificación utilizando una lista de tareas, que se puede observar en la Tabla 3.1 y su correspondiente diagrama de Gantt, este diagrama se puede ver en la Figura 3.1. En este diagrama se puede observar el tiempo estimado para la realización de cada una de las tareas planificadas para desarrollar el proyecto y las dependencias entre ellas.

Durante la estancia y al finalizar el proyecto se realiza una comparación entre la planificación de tareas y el tiempo de realización real de cada una de ellas y se llega a la conclusión de que no se observan variaciones significantes, por lo que podemos decir que la planificación ha sido buena y realista.

Nº	Herramienta aprendizaje electrocardiogramas	Tiempo (h.)	
1	Desarrollar la propuesta técnica		
1.1	Inicio	13h	
1.1.1	Definir el sistema con la tutora y los médicos cardiólogos	1	
1.1.2	Definir el proyecto con la tutora y el supervisor	2	1.1.1
1.1.3	Definir método de trabajo y documentación	5	1.1.2
1.1.4	Definir formato y estándares de trabajo	5	1.1.2
1.2	Documentar y planificar el proyecto	24h	
1.2.1	Revisar contexto y buscar información	16	1.1.2
1.2.2	Identificar alcance y objetivos	8	1.1.2
1.3	Planificar el proyecto	32h	
1.3.1	Definir tareas y estimar fechas	8	1.2
1.3.2	Crear diagrama de Gantt	8	1.3.1
1.3.3	Documentar la propuesta del proyecto	16	1.3.1
1.3.4	Entregar la propuesta técnica	0	1.3.3
2	Desarrollo técnico del proyecto		
2.1	Definir requisitos del proyecto	41h	
2.1.1	Crear diagrama de casos de uso	16	
2.1.2	Definir y documentar los requisitos de datos	10	
2.1.3	Definir requisitos tecnológicos y de plataforma	15	
2.2	Análisis	60h	
2.2.1	Crear diagrama de clases	15	
2.2.2	Documentar clases	15	2.2.1
2.2.3	Diagrama de actividades	15	
2.2.4	Validar análisis	15	2.2.3
2.3	Diseño	56h	
2.3.1	Identificar y clasificar usuarios	8	
2.3.2	Diseñar interfaces gráficas	16	
2.3.3	Refinar diagrama de clases de diseño	8	
2.3.4	Proponer plataforma tecnológica	8	
2.3.5	Validar diseño	16	
2.4	Desarrollar el producto	85h	
2.4.1	Programación	50	
2.4.2	Pruebas	35	
2.5	Puesta en marcha	6h	
2.5.1	Implantación y formación	6	
2.5.2	Entrega final	0	2.5.1
4	Documentación y presentación del TFG	120h	
4.1	Redacción de informes quincenales	4	
4.2	Redacción de la memoria técnica	85	
4.3	Entrega de la memoria técnica	0	4.2
4.4	Preparación de la presentación oral	30	4.3
4.5	Presentación oral	1	4.4

Tabla 3.1: Lista de desglose de tareas, su duración y dependencias entre ellas.

Diagrama de Gantt

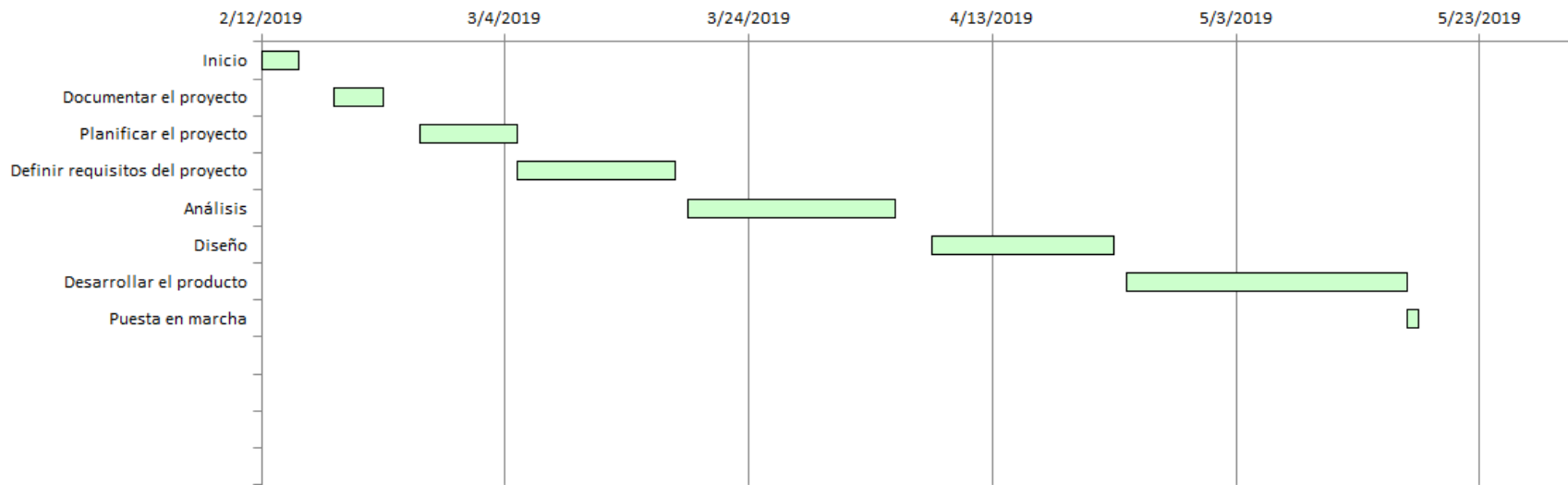


Figura 3.1: Diagrama de Gantt.

3.3. Estimación de recursos y costes del proyecto

3.3.1. Recursos

- Hardware proporcionado por la empresa:
 - Ordenador: HP EliteDesk 800 G4.
 - Monitores: 2x HP EliteDisplay e233.
 - Teclado: HP SmartCard CCID.
 - Ratón.
- Software:
 - Sistema Operativo: Maqueta de Windows 7 Professional y Windows Server 2016 en el dominio del hospital.
 - Herramientas utilizadas, descritas en el capítulo 2.
- Humanos:
 - Programador junior.

3.3.2. Costes

El coste del hardware utilizado se amortiza en 4 años y se desglosa en:

- Ordenador: $800\text{€} \rightarrow 16.6\text{€/mes} \rightarrow \text{Total: } 50\text{€}$
- Monitores: $2 \cdot 250\text{€} = 500\text{€} \rightarrow 10.4\text{€/mes} \rightarrow \text{Total } 31\text{€}$
- Teclado: $50\text{€} \rightarrow 1\text{€/mes} \rightarrow \text{Total: } 3\text{€}$
- Ratón: $20\text{€} \rightarrow 0.4\text{€/mes} \rightarrow \text{Total: } 1.2\text{€}$

El coste del software se calcula el precio de las licencias, el precio de Windows 7 Professional no se necesita tener en cuenta ya que se trata de una licencia de dominio y Windows Server 2016 cuesta 475€ en su versión Standard, esta licencia es vitalicia y se calcula un uso durante un mínimo de 4 años. Por tanto, el precio durante los 3 meses de trabajo es de 30€.

El sueldo neto de un programador junior está al rededor de 1200€ al mes, teniendo en cuenta que una jornada laboral es de 8 horas, tenemos un total de 160 horas de trabajo al mes. Dividiendo los 1200€ entre las 160 horas obtenemos el sueldo por hora: 7,5€/hora.

Como la duración de la estancia es de 300 horas presenciales, se estima que el programador cobra 2250€. Al ser neto hay que tener en cuenta la cuota de seguridad social y otras retenciones, con lo que el coste asciende a 2800€.

Los costes de los recursos utilizados se pueden ver en la Figura 3.2

Tipo	Recurso	Precio
Hardware	PC	50€
	Monitores	31€
	Teclado	3€
	Ratón	1.2€
Software	Windows Server 2016	30€
Humano	Programador	2800€
Total		2915.2€

Tabla 3.2: Costes de los recursos del proyecto.

3.4. Seguimiento del proyecto

Para realizar el seguimiento del proyecto, y puesto que se realizan informes quincenales para el seguimiento de la estancia, se han realizado periodos de quince días en los que se trabaja en las tareas planeadas según la lista de tareas y el diagrama de Gantt y, al finalizar el periodo, se evalúa el progreso comparándolo con la planificación de tareas para ese periodo.

Capítulo 4

Análisis y diseño del sistema

4.1. Análisis del sistema

En este capítulo se muestra en análisis del sistema, presentando los requisitos funcionales y de datos a cumplir por la herramienta. Posteriormente se muestran las interfaces de usuario de la aplicación.

4.1.1. Casos de uso del sistema

A continuación se presentan los casos de uso del sistema, diferenciados en 3 actores:

- Estudiante: representa a cualquier estudiante de Medicina con acceso a la herramienta.
- Profesor: representa a los profesores encargados de la herramienta.
- Administrador: representa al administrador del sistema.

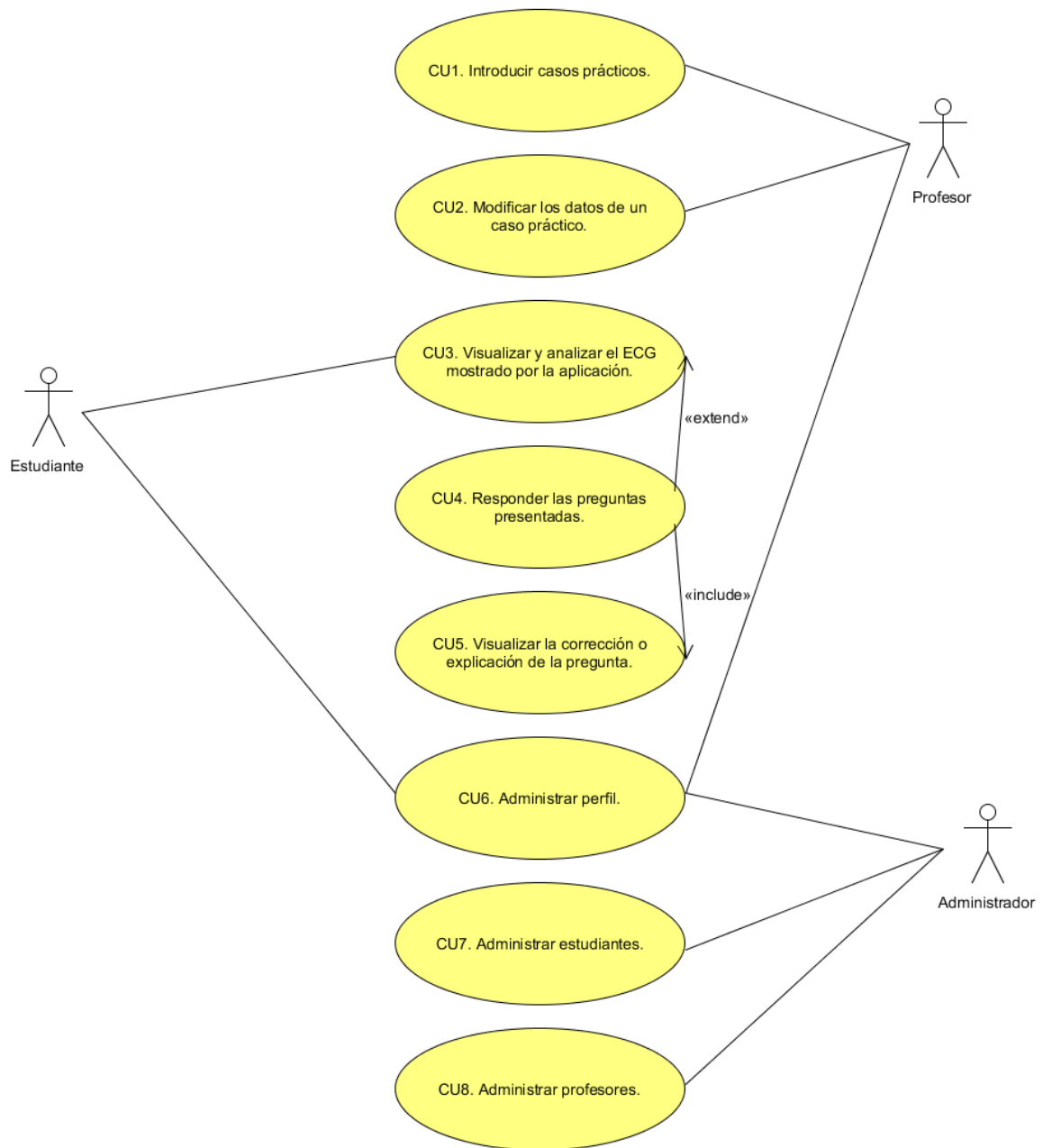


Figura 4.1: Diagrama de casos de uso del sistema

Actor primario	Casos de uso
Profesor	<ul style="list-style-type: none"> ▪ CU1. Introducir casos prácticos. ▪ CU2. Modificar los datos de un caso práctico.
Estudiante	<ul style="list-style-type: none"> ▪ CU3. Visualizar y analizar el ECG mostrado por la aplicación. ▪ CU4. Responder las preguntas presentadas. ▪ CU5. Visualizar la corrección o explicación de la pregunta.
Todos	<ul style="list-style-type: none"> ▪ CU6. Administrar perfil.
Administrador	<ul style="list-style-type: none"> ▪ CU7. Administrar estudiantes. ▪ CU8. Administrar profesores.

Tabla 4.1: Resumen casos de uso.

Nombre caso de uso	CU1. Introducir un caso práctico.
Complejidad	Complejidad media
Descripción	Permite al profesor encargado de la aplicación introducir nuevos casos en la base de datos de la aplicación y preguntas para evaluar a los estudiantes.
Actores	Profesor
Objetivo	Permitir la introducción de casos prácticos.
Pre-requisitos	Para introducir casos prácticos, el profesor debe estar registrado en el sistema y contar con los permisos necesarios.
Relaciones	
Asociación	Actor Profesor
Escenarios	
Flujo básico de eventos	<p>Secuencia normal de pasos:</p> <ol style="list-style-type: none"> 1. El profesor accede al sistema. 2. El profesor introduce los datos del caso, incluyendo la imagen del ECG. <ol style="list-style-type: none"> 2.1. Si el título introducido coincide con uno ya existente, se muestra un error. 3. El sistema guarda los datos del caso práctico introducido en la base de datos. <ol style="list-style-type: none"> 3.1. En caso de error al guardar en la base de datos, se avisa del fallo.

Tabla 4.2: CU1. Introducir un caso práctico.

Nombre caso de uso	CU2. Modificar los datos de un caso práctico.
Complejidad	Complejidad media
Descripción	Permite al profesor encargado de la aplicación modificar los datos de un caso práctico que ya está almacenado en la base de datos de la aplicación.
Actores	Profesor
Objetivo	Permitir la modificación de casos prácticos.
Pre-requisitos	Para modificar casos prácticos, debemos contar con algún caso almacenado en la base de datos. Además, el profesor debe estar registrado en el sistema y contar con los permisos necesarios.
Relaciones	
Asociación	Actor Profesor
Escenarios	
Flujo básico de eventos	<p>Secuencia normal de pasos:</p> <ol style="list-style-type: none"> 1. El profesor inicia sesión en el sistema y accede a los casos almacenados en este. 2. El profesor escoge el caso a modificar, realiza los cambios oportunos y los guarda. 3. El sistema guarda los cambios realizados al caso práctico en la base de datos. <ol style="list-style-type: none"> 3.1. En caso de error al modificar los datos de la base de datos, se avisa del fallo.

Tabla 4.3: CU2. Modificar los datos de un caso práctico.

Nombre caso de uso	CU3. Visualizar y analizar el ECG mostrado por la aplicación.
Complejidad	Complejidad alta
Descripción	Permite al estudiante visualizar un caso práctico y analizarlo, midiendo los valores que considere necesarios para el correcto diagnóstico del mismo.
Actores	Estudiante
Objetivo	Permitir visualizar y analizar un ECG.
Pre-requisitos	Para visualizar un caso práctico, este debe haber sido introducido en la base de datos de la aplicación. Además, el estudiante debe haber accedido al sistema.
Relaciones	
Asociación	Actor Estudiante
Escenarios	
Flujo básico de eventos	<p>Secuencia normal de pasos:</p> <ol style="list-style-type: none"> 1. El estudiante seleccionará un apartado del índice de contenidos mostrado en pantalla. <ol style="list-style-type: none"> 1.1. En caso de no existir casos prácticos del apartado seleccionado, se mostrará un aviso. 2. El sistema le muestra un caso práctico entre los almacenados correspondientes a ese apartado. 3. El estudiante utiliza las herramientas proporcionadas por la aplicación para analizar el ECG del caso práctico.

Tabla 4.4: CU3. Visualizar y analizar el ECG mostrado por la aplicación.

Nombre caso de uso	CU4. Responder las preguntas presentadas.
Complejidad	Complejidad media
Descripción	Permite al estudiante responder a las preguntas formuladas por los profesores encargados y visualizar las explicaciones correcciones oportunas.
Actores	Estudiante
Objetivo	Permitir responder a las preguntas de evaluación.
Pre-requisitos	Para responder a las preguntas sobre un caso práctico, previamente se debe haber accedido a este. CU03 finalizado con éxito.
Relaciones	
Asociación	Actor Estudiante
Escenarios	
Flujo básico de eventos	Secuencia normal de pasos: 1. El estudiante responde a la pregunta mostrada, esta pregunta será de tipo test con varias opciones.

Tabla 4.5: CU4. Responder las preguntas presentadas.

Nombre caso de uso	CU5. Visualizar la corrección o explicación de la pregunta.
Complejidad	Complejidad media
Descripción	Permite al estudiante saber si ha respondido correctamente o no a la pregunta y, dependiendo de esto, visualiza la explicación del caso o la corrección de su respuesta. En caso de ser incorrecta, el estudiante puede volver a contestar la pregunta.
Actores	Estudiante
Objetivo	Permitir al estudiante recibir feedback sobre su respuesta.
Pre-requisitos	Cualquier estudiante que conteste una pregunta recibirá este feedback. Es un include de CU04.
Relaciones	
Asociación	Actor Estudiante
Escenarios	
Flujo básico de eventos	<p>Secuencia normal de pasos:</p> <ol style="list-style-type: none"> 1. El sistema mostrará, en función de la respuesta introducida por el estudiante, las explicaciones de cada respuesta, tanto las correctas, como las incorrectas.

Tabla 4.6: CU5. Visualizar la corrección o explicación de la pregunta.

Nombre caso de uso	CU6. Administrar perfil.
Complejidad	Complejidad media
Descripción	Permite a cualquier usuario visualizar sus datos y cambiar su contraseña.
Actores	Administrador, profesor y estudiante
Objetivo	Permitir a cualquier usuario gestionar su perfil.
Pre-requisitos	Estar registrado y acceder al sistema.
Relaciones	
Asociación	Actor Administrador, Profesor y Estudiante
Escenarios	
Flujo básico de eventos	<p>Secuencia normal de pasos:</p> <ol style="list-style-type: none"> 1. Acceder al apartado de perfil. 2. El sistema mostrará los datos del usuario. 3. El usuario pulsa el botón de cambiar contraseña. 4. El usuario introduce la nueva contraseña y la envía. 5. El sistema guarda los nuevos datos del usuario en la base de datos. <ol style="list-style-type: none"> 5.1. En caso de error al modificar los datos en la base de datos, se avisa del fallo.

Tabla 4.7: CU6. Administrar perfil.

Nombre caso de uso	CU7. Administrar estudiantes.
Complejidad	Complejidad media
Descripción	Permite al administrador visualizar, modificar y añadir estudiantes.
Actores	Administrador
Objetivo	Permitir al administrador gestionar los estudiantes.
Pre-requisitos	Acceder al sistema como administrador.
Relaciones	
Asociación	Actor Administrador
Escenarios	
Flujo básico de eventos	<p>Secuencia normal de pasos:</p> <ol style="list-style-type: none"> 1. El sistema mostrará los estudiantes actuales y una opción para añadir uno nuevo. 2. El administrador selecciona la operación a realizar (editar, borrar o añadir nuevo). <ol style="list-style-type: none"> 2.1. Editar: el administrador modifica los campos que desea editar y lo envía. 2.2. Borrar: el administrador confirma la eliminación mediante un modal mostrado en pantalla. 2.3. Añadir: el administrador rellena los campos del formulario de creación de estudiante y lo envía. 3. El sistema modifica, elimina o añade los datos del usuario en la base de datos. <ol style="list-style-type: none"> 3.1. En caso de error en la base de datos, se informa del fallo..

Tabla 4.8: CU7. Administrar estudiantes.

Nombre caso de uso	CU8. Administrar profesores.
Complejidad	Complejidad media
Descripción	Permite al administrador visualizar, modificar y añadir profesores.
Actores	Administrador
Objetivo	Permitir al administrador gestionar los profesores.
Pre-requisitos	Acceder al sistema como administrador.
Relaciones	
Asociación	Actor Administrador
Escenarios	
Flujo básico de eventos	<p>Secuencia normal de pasos:</p> <ol style="list-style-type: none"> 1. El sistema mostrará los profesores actuales y una opción para añadir uno nuevo. 2. El administrador selecciona la operación a realizar (editar, borrar o añadir nuevo). <ol style="list-style-type: none"> 2.1. Editar: el administrador modifica los campos que desea editar y lo envía. 2.2. Borrar: el administrador confirma la eliminación mediante un modal mostrado en pantalla. 2.3. Añadir: el administrador rellena los campos del formulario de creación de profesor y lo envía. 3. El sistema modifica, elimina o añade los datos del usuario en la base de datos. <ol style="list-style-type: none"> 3.1. En caso de error en la base de datos, se informa del fallo..

Tabla 4.9: CU8. Administrar profesores.

4.1.2. Requisitos de datos del sistema

RD01. Profesor

Código	RD01
Nombre	Profesor
Fuentes	Administrador
Datos	Usuario, contraseña.
Comentarios	El sistema almacena los datos de cada usuario profesor.

Tabla 4.10: RD01. Profesor

RD02. Estudiante

Código	RD02
Nombre	Estudiante
Fuentes	Administrador
Datos	Usuario, contraseña, progreso.
Comentarios	El sistema almacena los datos de cada usuario estudiante.

Tabla 4.11: RD02. Estudiante

RD03. Caso práctico

Código	RD03
Nombre	Caso práctico
Fuentes	Administrador, profesor
Datos	idCaso, apartado, imagen, preguntas, respuestas.
Comentarios	El profesor introduce los datos de cada caso práctico.

Tabla 4.12: RD03. Caso práctico

RD04. Pregunta

Código	RD04
Nombre	Pregunta
Fuentes	Administrador, profesor
Datos	idCaso, idPregunta, enunciado.
Comentarios	El profesor introduce el enunciado de la pregunta.

Tabla 4.13: RD04. Pregunta

RD05. Respuesta

Código	RD05
Nombre	Respuesta
Fuentes	Administrador, profesor
Datos	idCaso, idPregunta, idRespuesta, texto, correcto.
Comentarios	El profesor introduce los datos de cada respuesta.

Tabla 4.14: RD05. Respuesta

RD06. Explicación

Código	RD06
Nombre	Explicación
Fuentes	Administrador, profesor
Datos	idCaso, idPregunta, idRespuesta, explicación.
Comentarios	El sistema muestra la explicación o solución de la respuesta.

Tabla 4.15: RD06. Explicación

4.2. Diseño de la arquitectura del sistema

Para satisfacer estos requisitos de datos se crea un diagrama de clases, el cual se puede ver en la Figura 4.2.

A continuación se presenta el diseño lógico y físico de la base de datos creada.

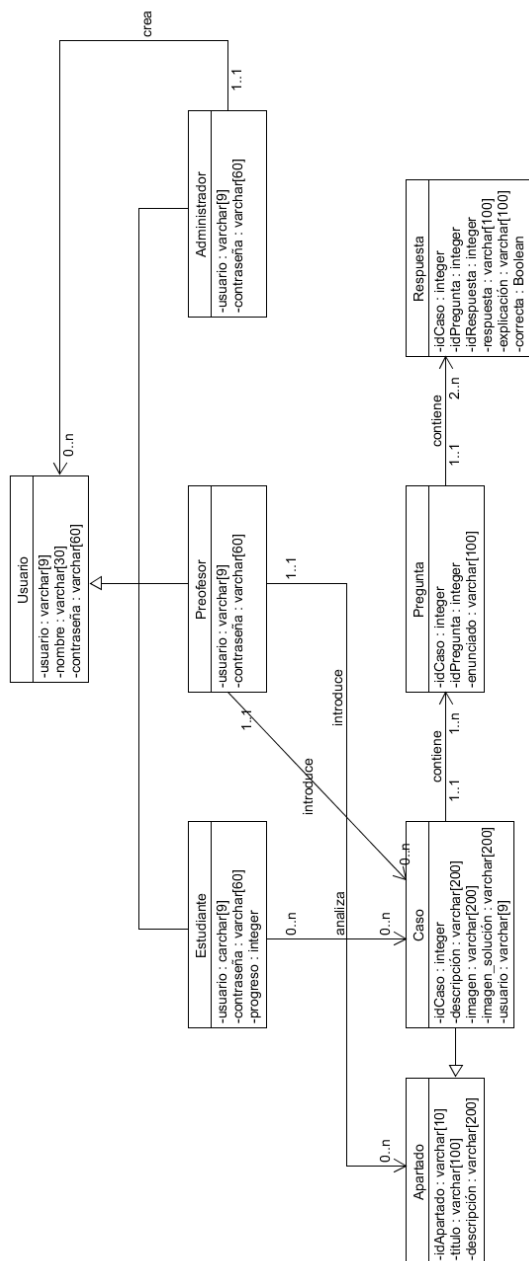


Figura 4.2: Diagrama de clases del sistema

4.2.1. Diseño lógico de la base de datos

En este apartado se muestra el diseño lógico de las tablas, relaciones y restricciones de la base de datos.

Tablas

Usuario(usuario, password, progreso, tipo, nombre)

Caso(idCaso, descripcion, nivel, imagen, imagenSolucion, idApartado, usuario)

Pregunta(idPregunta, idCaso, enunciado)

Respuesta(idRespuesta, idCaso, idPregunta, respuesta, explicacion, correcta)

Apartado(idApartado, titulo, descripcion)

Analiza(usuario, idCaso, idRespuesta)

Relaciones

Tabla 1	Clave ajena	Tabla 2	Borrado	Modificación	Nulos
Usuario	usuario	Caso	Restringir	Propagar	No
Caso	idCaso	Pregunta	Propagar	Propagar	No
Pregunta	idPregunta	Respuesta	Propagar	Propagar	No
Respuesta	idRespuesta	Analiza	Restringir	Restringir	No
Usuario	usuario	Analiza	Restringir	Propagar	No
Caso	idCaso	Analiza	Restringir	Propagar	No

Tabla 4.16: Relaciones de la base de datos.

Restricciones

Usuario.tipo in {'estudiante', 'profesor', 'administrador'}

Esta restricción sirve para definir los roles de usuario, éstos serán utilizados por la aplicación para identificar el tipo del usuario que accede y darle acceso a las funcionalidades que le corresponden.

4.2.2. Diseño físico de la base de datos

En este apartado se muestra el diseño físico realizado a partir del diseño lógico anterior.

```
1 CREATE TABLE public.analiza (
2     usuario character varying(9) NOT NULL,
3     idcaso integer NOT NULL,
4     idpregunta integer
5 );
6
7 ALTER TABLE ONLY public.analiza ADD CONSTRAINT analiza_pk PRIMARY KEY (usuario,
8     idcaso);
```

Listing 4.1: Tabla analiza.

```
1 CREATE TABLE public.apartado (
2     idapartado character varying(10) NOT NULL,
3     titulo character varying(100),
4     descripcion character varying(200)
5 );
6
7 ALTER TABLE ONLY public.apartado ADD CONSTRAINT apartado_pkey PRIMARY KEY (
8     idapartado);
```

Listing 4.2: Tabla apartado.

```
1 CREATE TABLE public.caso (
2     idcaso integer NOT NULL,
3     descripcion character varying(200) NOT NULL,
4     nivel integer NOT NULL,
5     imagen character varying(200) NOT NULL,
6     usuario_usuario character varying(9) NOT NULL,
7     idapartado character varying(10),
8     imagen_solucion character varying(200)
9 );
10
11 ALTER TABLE ONLY public.caso ALTER COLUMN idcaso SET DEFAULT nextval('public.
12     caso_idcaso_seq ':: regclass);
13
14 ALTER TABLE ONLY public.caso ADD CONSTRAINT caso_pk PRIMARY KEY (idcaso);
15
16 ALTER TABLE ONLY public.caso ADD CONSTRAINT usuario_caso FOREIGN KEY (
17     usuario_usuario) REFERENCES public.usuario(usuario) ON UPDATE CASCADE ON
18     DELETE RESTRICT;
```

Listing 4.3: Tabla caso.


```

1 CREATE TABLE public.pregunta (
2     idcaso integer NOT NULL,
3     idpregunta integer DEFAULT nextval('public.pregunta_idpregunta_seq ':: regclass
4     ) NOT NULL,
5     enunciado character varying(500) NOT NULL
6 );
7 ALTER TABLE ONLY public.pregunta ADD CONSTRAINT pregunta_pk PRIMARY KEY (idcaso
8     , idpregunta);
9 ALTER TABLE ONLY public.pregunta ADD CONSTRAINT pregunta_caso FOREIGN KEY (
10    idcaso) REFERENCES public.caso(idcaso) ON UPDATE CASCADE ON DELETE CASCADE;

```

Listing 4.4: Tabla pregunta.

```

1 CREATE TABLE public.respuesta (
2     idcaso integer NOT NULL,
3     idpregunta integer NOT NULL,
4     idrespuesta integer DEFAULT nextval('public.respuesta_idrespuesta_seq '::
5     regclass) NOT NULL,
6     respuesta character varying(100) NOT NULL,
7     explicacion character varying(1000) NOT NULL,
8     correcta boolean NOT NULL
9 );
10 ALTER TABLE ONLY public.respuesta ADD CONSTRAINT respuesta_pk PRIMARY KEY (
11    idcaso , idpregunta , idrespuesta);
12 ALTER TABLE ONLY public.respuesta ADD CONSTRAINT respuesta_pregunta FOREIGN KEY
13    (idcaso , idpregunta) REFERENCES public.pregunta(idcaso , idpregunta) ON UPDATE
    CASCADE ON DELETE CASCADE;

```

Listing 4.5: Tabla respuesta.

```

1 CREATE TABLE public.usuario (
2     usuario character varying(9) NOT NULL,
3     password character varying(60) NOT NULL,
4     progreso integer NOT NULL,
5     tipo character varying(25) NOT NULL,
6     nombre character varying(30) ,
7     CONSTRAINT check_tipo CHECK (((tipo)::text = ANY ((ARRAY['estudiante '::
8     character varying , 'administrador ':: character varying , 'profesor ':: character
9     varying]) :: text [])))
10 );
11 ALTER TABLE ONLY public.usuario ADD CONSTRAINT estudiante_pk PRIMARY KEY (
    usuario);

```

Listing 4.6: Tabla usuario.

4.3. Diseño de la interfaz

En esta sección se presentan las interfaces de la aplicación, separadas en roles de usuario. Así tenemos interfaces comunes, del estudiante, del profesor y del administrador.

4.3.1. Interfaces comunes

La página de bienvenida contiene una breve explicación y el botón de acceso, ésta se observa en la Figura 4.3.

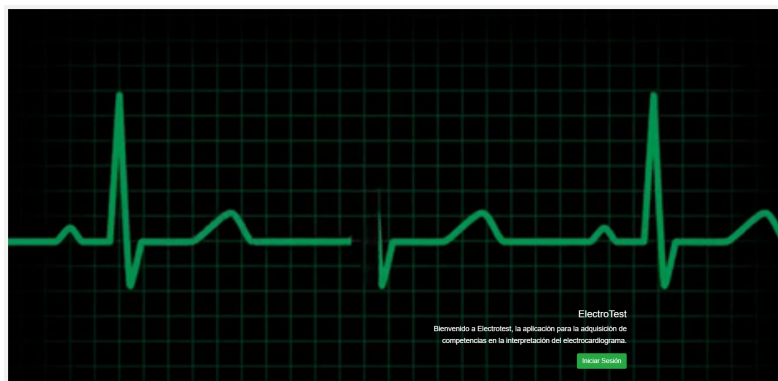


Figura 4.3: Interfaz de la pagina de inicio

En la página de acceso se introduce el usuario y la contraseña, ésta se observa en la Figura 4.4.

Figura 4.4: Interfaz de la pagina de acceso

4.3.2. Estudiante

En el índice de contenidos se muestra cada apartado junto al botón para seleccionarlo y un cuadro de búsqueda, éste se observa en la Figura 4.5.

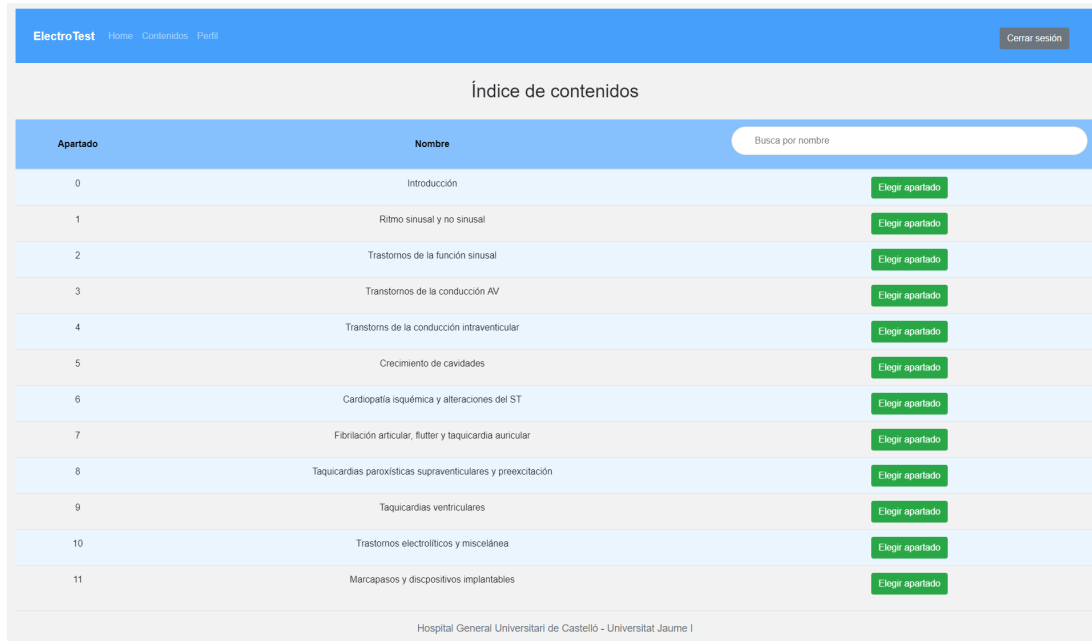


Figura 4.5: Interfaz del índice de contenidos

La página para analizar el caso y responder es la más compleja, por una parte contiene la imagen del electrocardiograma y los botones para manejar los datos de la medición y por otra parte contiene la pregunta y sus respuestas, ésta se observa en la Figura 4.6.



Figura 4.6: Interfaz de analizar y responder caso

La página para visualizar la corrección en caso de respuesta errónea es muy parecida a la anterior, pero cambiando la parte de test por la explicación, ésta se observa en la Figura 4.7.

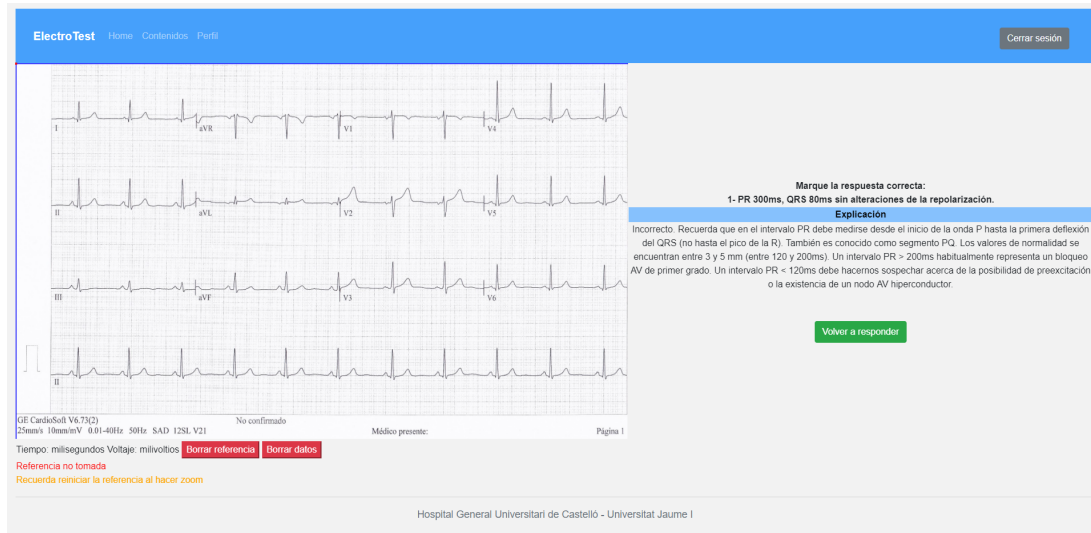


Figura 4.7: Interfaz visualizar la corrección al responder incorrectamente

La página para visualizar la explicación del caso cuando se acierta la respuesta contiene la imagen de solución junto con las explicaciones de cada una de las respuestas, esta se observa en la Figura 4.8.



Figura 4.8: Interfaz visualizar la explicación al responder correctamente

La página de administración del perfil permite modificar la contraseña, como se observa en la Figura 4.9.

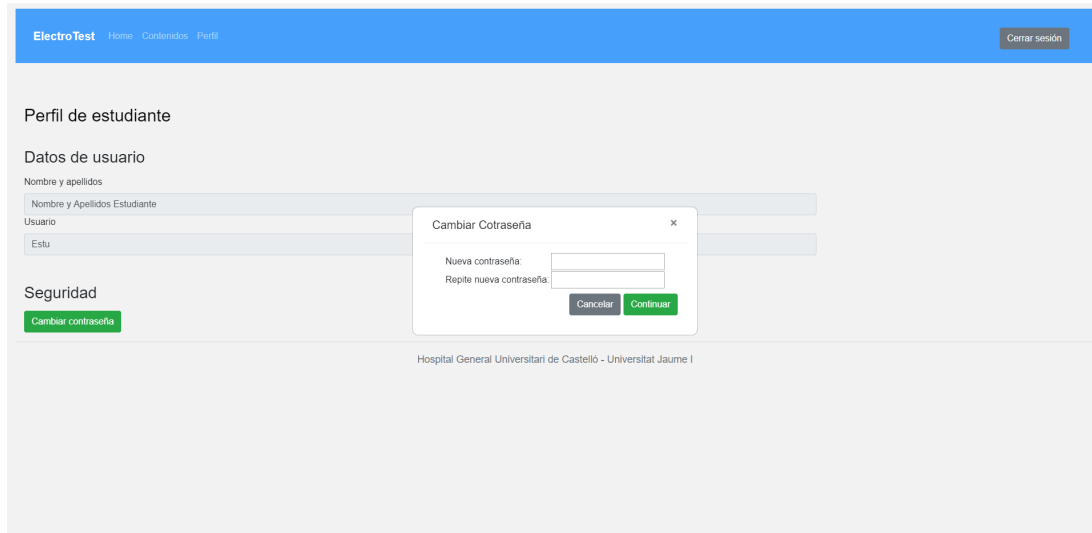


Figura 4.9: Interfaz visualizar perfil y cambiar contraseña

4.3.3. Profesor

La página para administrar los casos permite añadir una pregunta a un caso, editarlo, borrarlo o añadir un nuevo caso, como se muestra en la Figura 4.10.

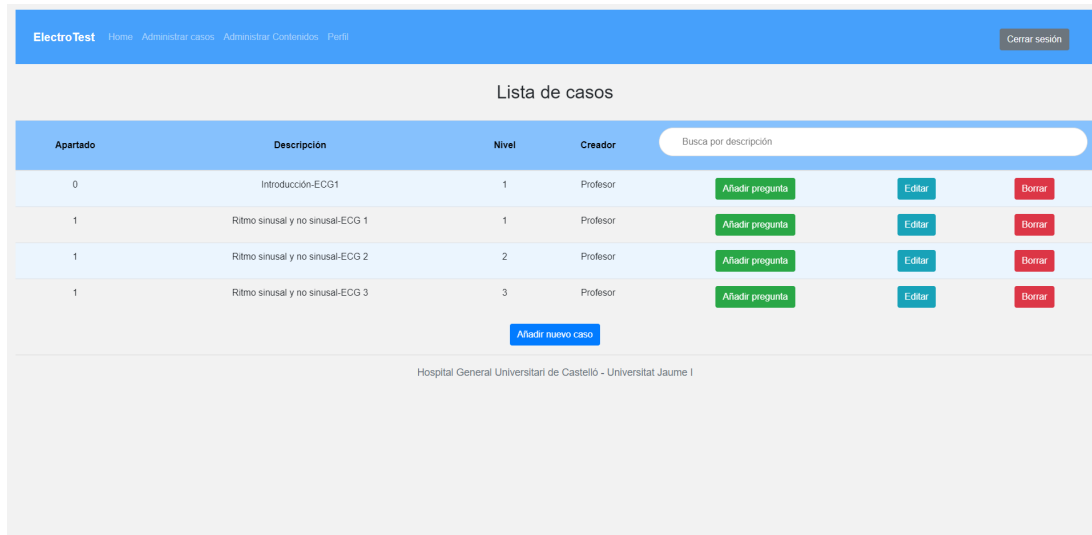


Figura 4.10: Interfaz visualizar la lista de casos

La interfaz para introducir los datos de un nuevo caso se puede ver en la Figura 4.11.

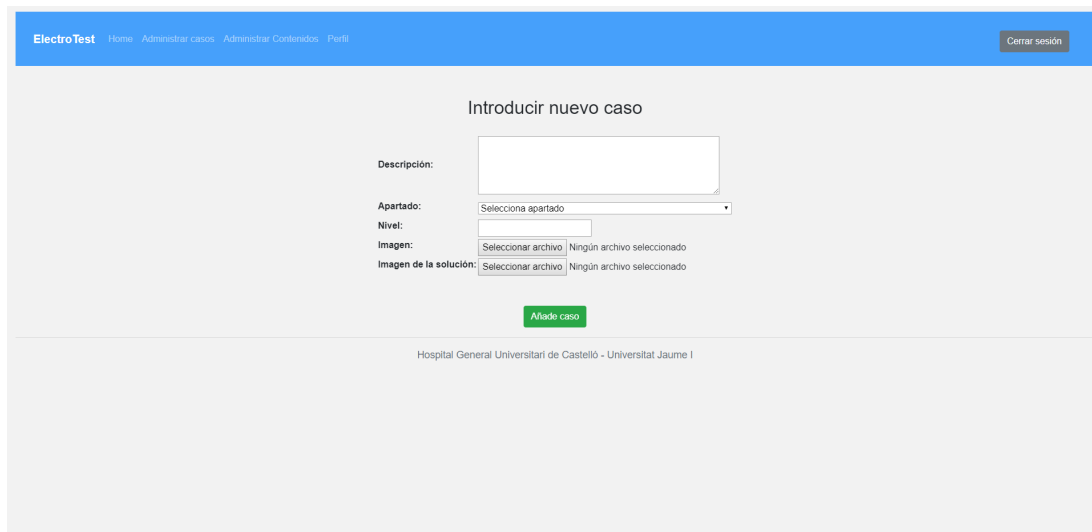


Figura 4.11: Interfaz añadir nuevo caso

La página para introducir una nueva pregunta consta con un número de campos dinámico, acorde con el número de respuestas indicadas, como se muestra en la Figura 4.12.

The screenshot shows a web interface for adding a new question. At the top, there is a blue navigation bar with the text 'ElectroTest' and links for 'Home', 'Administrar casos', 'Administrar Contenidos', and 'Perfil'. A 'Cerrar sesión' button is located in the top right corner. The main content area is titled 'Introducir nueva pregunta'. It contains a large text input field for the 'Enunciado'. Below this, there is a 'Número de respuestas' field with the value '1'. To the right of this field is a blue 'Añadir' button. Underneath, there are three input fields: 'Respuesta 1', 'Respuesta correcta', and 'Corrección 1'. A green 'Añadir pregunta' button is positioned below these fields. At the bottom of the page, the text 'Hospital General Universitari de Castelló - Universitat Jaume I' is displayed.

Figura 4.12: Interfaz añadir nueva pregunta

4.3.4. Administrador

Para gestionar los usuarios se muestran en una lista, junto con las opciones de editar, borrar y añadir, como se observa en la Figura 4.13.

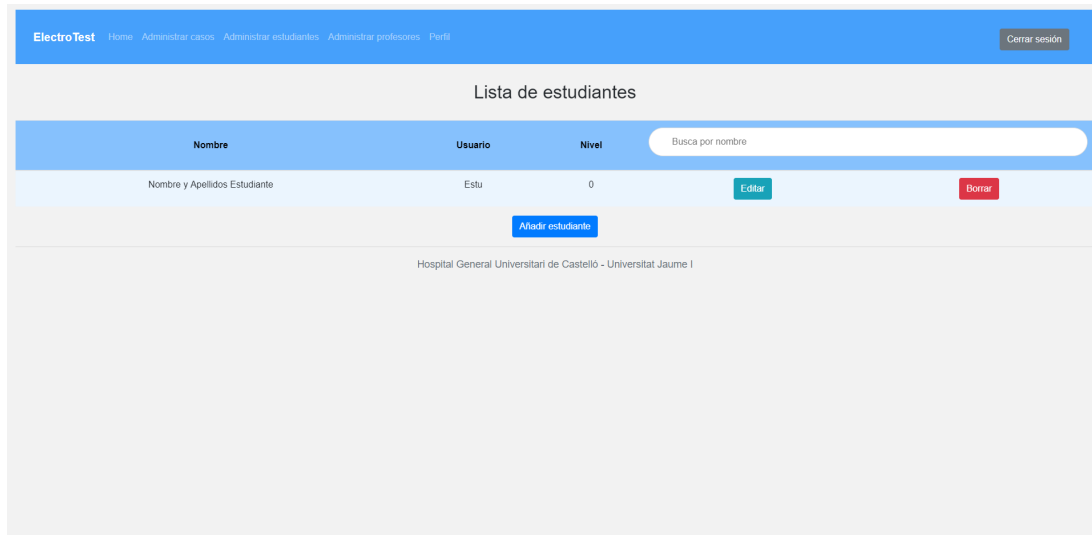


Figura 4.13: Interfaz gestionar usuarios

La interfaz para introducir los datos de un nuevo usuario se muestra en la Figura 4.14.

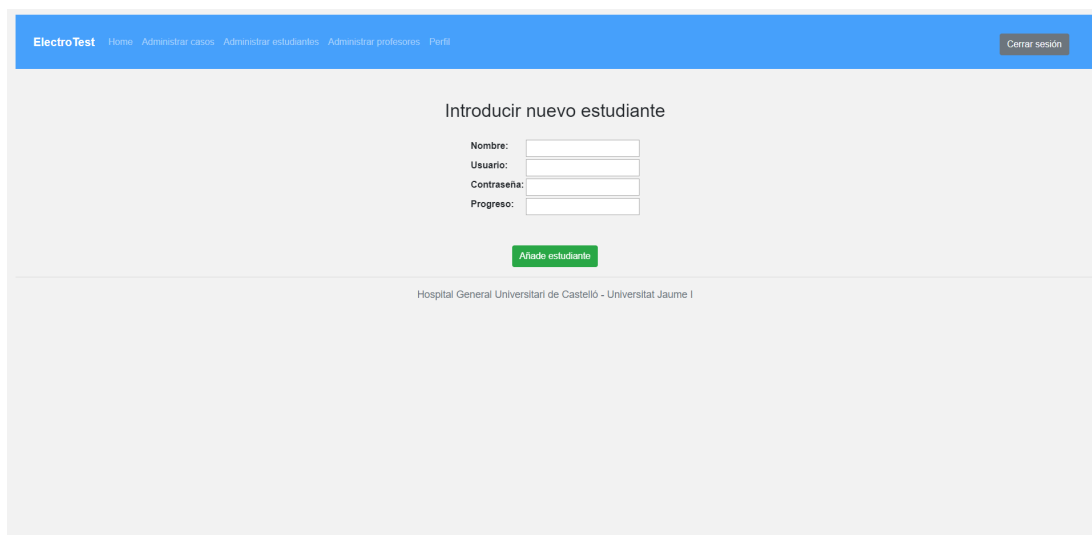


Figura 4.14: Interfaz nuevo usuario

En la interfaz de editar usuario que se muestra en la Figura 4.15 se incluye la opción de reiniciar contraseña, como se muestra en la Figura 4.16.

ElectroTest Home Administrar casos Administrar estudiantes Administrar profesores Perfil Cerrar sesión

Editar estudiante

Nombre: Nombre y Apellidos Estudia
Usuario: Estu
Progreso: 0

Reiniciar Contraseña Editar estudiante

Hospital General Universitari de Castelló - Universitat Jaume I

Figura 4.15: Interfaz editar usuario

ElectroTest Home Administrar casos Administrar estudiantes Administrar profesores Perfil Cerrar sesión

Editar estudiante

Nombre: Nombre y Apellidos Estudia
Usuario: Estu
Progreso: 0

Reiniciar Contraseña Editar estudiante

Hospital General Universitari de Castelló - Universitat Jaume I

Confirmar Reinicio

Al hacer click en Reiniciar Contraseña se reestablecerá para coincidir con el nombre de usuario.

Cancelar Reiniciar Contraseña

Figura 4.16: Interfaz reiniciar contraseña

Capítulo 5

Implementación y pruebas

5.1. Detalles de implementación

En esta sección se presentan los detalles sobre la implementación del sistema, para observar el trabajo realizado de forma más detallada se presenta el periodo completo de implementación dividido en periodos de quince días.

5.1.1. Periodo 1

Durante las primeras semanas se dedica la mayor parte del tiempo a definir el proyecto, para ello se realizan varias reuniones con la tutora, el supervisor y los médicos.

Con la información adquirida en las distintas reuniones se definen los objetivos del proyecto. Se establece como objetivo principal de la herramienta que los estudiantes sean capaces de tomar medidas directas sobre la imagen del electrocardiograma que se presenta en cada caso práctico.

Una vez definidos los objetivos del proyecto, se realiza la búsqueda de información sobre las posibles tecnologías que se pueden utilizar para la realización del proyecto. Como resultado de la búsqueda de información se definen las tecnologías a utilizar, descritas en el capítulo 2.

5.1.2. Periodo 2

Las primeras tareas de implementación que se realizan son las del diseño lógico y físico de la base de datos, que ya se ha visto en la sección de diseño del sistema.

El diseño de la base de datos se realiza mediante la herramienta pgAdmin. Esta es una herramienta de código abierto para administrar bases de datos en PostgreSQL. Cuenta con una interfaz gráfica administrativa, herramienta para introducir consultas SQL entre otras funciones. [10] Podemos ver como se presenta la base de datos en esta herramienta en la Figura 5.1

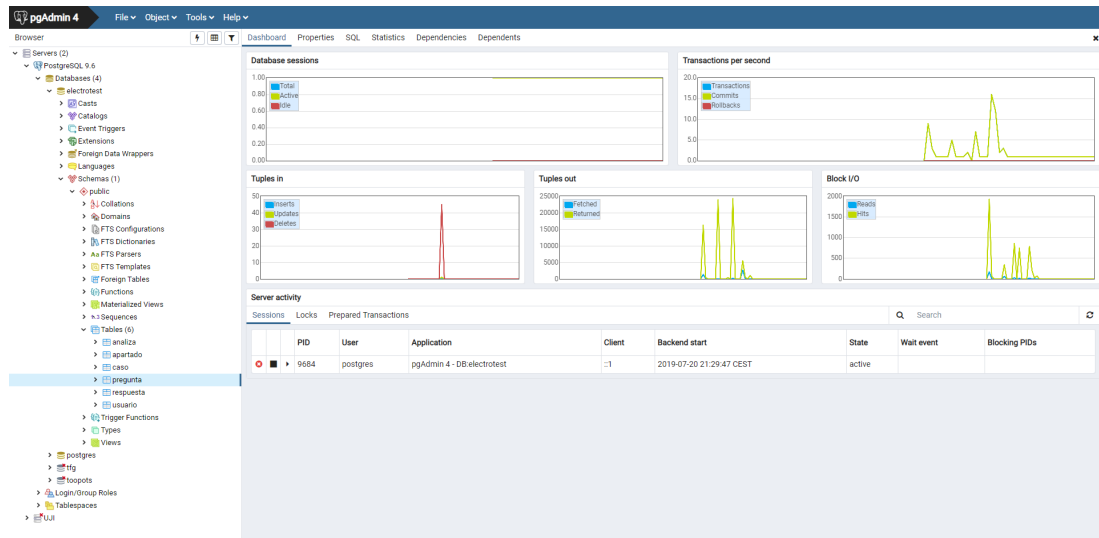


Figura 5.1: Herramienta pgAdmin.

Una vez realizado el diseño de la base de datos, se realiza el diseño de las interfaces de usuario de la aplicación web. El resultado de alguno de estos diseños se puede ver en la Figura 5.2 y la Figura 5.3

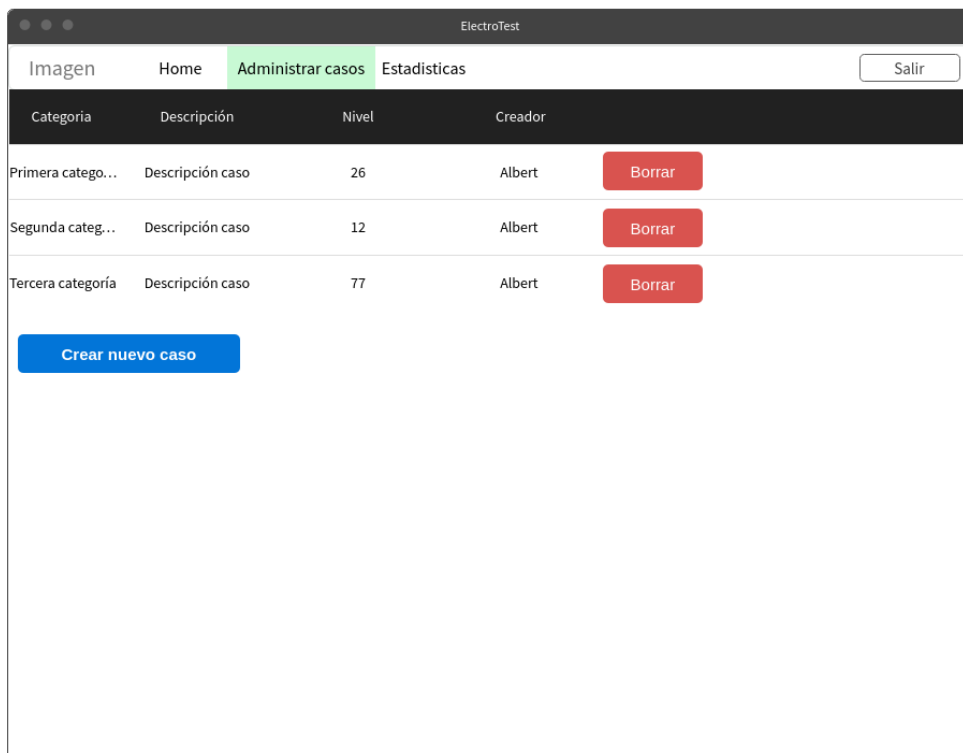


Figura 5.2: Diseño interfaz administrar casos.

Figura 5.3: Diseño interfaz nuevo caso.

5.1.3. Periodo 3

En este periodo se inicia la implementación del código de la aplicación web, empezando por la gestión de usuarios y del acceso de estos al sistema.

Para la conexión con la base de datos en PostgreSQL se utiliza JDBC, que permite ejecutar operaciones sobre la base de datos desde Java. La forma de conectarse a la base de datos se define en el fichero `application.properties`, el contenido de este fichero se puede ver en el código del Listado 5.1. En él se define la base de datos a la que conectarse y el usuario y contraseña administrador de PostgreSQL.

```

1  spring.datasource.type=org.springframework.jdbc.datasource.
   DriverManagerDataSource
2  spring.datasource.driver-class-name=org.postgresql.Driver
3  spring.datasource.url=jdbc:postgresql://localhost/electrotest
4  spring.datasource.username=postgres
5  spring.datasource.password=*****
6

```

Listing 5.1: Conexión a la base de datos con JDBC.

En cuanto a la gestión de usuarios hay que destacar que la contraseña no se guarda directamente en la base de datos, si no que se genera un hash salteado derivado de esta y este hash es el que se almacena. El salteado evita que se pueda descubrir la contraseña a partir de su hash utilizando tablas rainbow”. Para generar el hash salteado se utiliza la librería Jasypt (Java Simplified Encryption) que permite añadir capacidades de cifrado básicas al proyecto. [11].

Para crear un usuario nuevo del tipo estudiante a partir del formulario de creación, se implementa un método en el controlador como se muestra en el código del Listado 5.2. Este método llama a otro método del DAO de usuarios, el cual añade el nuevo usuario a la base de datos, este método puede verse en el Listado 5.3.

```
1  @RequestMapping(value = "/anadirEstudiante", method = RequestMethod.POST)
2  public String procesAddEstudiante(@RequestParam("nombre") String nombre,
3     @RequestParam("usuario") String usuario, @RequestParam("pass") String pass,
4     @RequestParam("progreso") int progreso) {
5     BasicPasswordEncryptor passwordEncryptor = new BasicPasswordEncryptor();
6     UserDetails nuevoEstudiante = new UserDetails();
7     nuevoEstudiante.setUsername(usuario);
8     nuevoEstudiante.setPassword(passwordEncryptor.encryptPassword(pass));
9     nuevoEstudiante.setNombre(nombre);
10    nuevoEstudiante.setTipo("estudiante");
11    nuevoEstudiante.setProgreso(progreso);
12    userDao.addUsuario(nuevoEstudiante);
13    return "redirect:administrarEstudiantes";
14 }
```

Listing 5.2: Método del controlador que recibe los datos de usuario.

```
1  public void addUsuario(UserDetails usuario) {
2     this.jdbcTemplate.update("insert into usuario(usuario, password, nombre, tipo
3     , progreso) values (?, ?, ?, ?, ?)", usuario.getUsername(), usuario.getPassword(),
4     usuario.getNombre(), usuario.getTipo(), usuario.getProgreso());
5 }
```

Listing 5.3: Método del DAO que inserta el usuario en la base de datos.

5.1.4. Periodo 4

En este periodo se implementan las funcionalidades de la aplicación pertenecientes al usuario profesor, estas incluyen la introducción de nuevos casos, nuevas preguntas y nuevas respuestas.

Para la introducción de nuevos casos se crea una vista en la que se presentará un formulario con los campos a rellenar para cada caso y las imágenes del electrocardiograma y solución asociadas.

Una vez se envía este formulario, los datos son recogidos por el controlador, el cual crea un nuevo objeto caso y se lo pasa al DAO de casos para que lo inserte en la base de datos, como sucedía con los nuevos usuarios.

Para introducir una nueva pregunta y sus respuestas se crea una vista en la que se podrá escoger el número de respuestas de la pregunta y, mediante el código JavaScript del Listado 5.4, se mostrará un número de campos de introducción de texto acorde al número de respuestas, acompañados de un checkbox para indicar la respuesta correcta.

Como con los objetos usuario y caso, este es recogido por el controlador e introducido en la base de datos por el DAO.

```
1 function addFields() {
2     // Número de inputs a crear
3     var number = document.getElementById("n_resp").value;
4     // Contenedor <div> donde se colocara el contenido dinámico
5     var container = document.getElementById("container");
6     var container2 = document.getElementById("container2");
7     // Limpiar contenedor
8     while (container.hasChildNodes()) {
9         container.removeChild(container.lastChild);
10    }
11    while (container2.hasChildNodes()) {
12        container2.removeChild(container2.lastChild);
13    }
14    for (i = 0; i < number; i++) {
15        var j = i + 1;
16        var input = document.createElement("input");
17        var correccion = document.createElement("input");
18        var checkbox = document.createElement("input");
19        var label = document.createElement("label");
20        input.type = "text";
21        input.name = "respuesta" + i;
22        input.placeholder = "Respuesta_" + j;
23        correccion.type = "text";
24        correccion.name = "correccion" + i;
25        correccion.placeholder = "Corrección_" + j;
26        checkbox.type = "checkbox";
27        checkbox.name = "correcta" + i;
28        checkbox.id = "correcta" + i;
29        checkbox.value = "correcta" + i;
30        label.htmlFor = "checkbox" + i;
31        label.appendChild(document.createTextNode("Respuesta_ correcta"));
32        container2.appendChild(input);
33        container2.appendChild(checkbox);
34        container2.appendChild(label);

```

```

35     container.appendChild(document.createElement("br"));
36     container2.appendChild(document.createElement("br"));
37     container2.appendChild(correccion);
38     container.appendChild(document.createElement("br"));
39     container2.appendChild(document.createElement("br"));
40 }
41 }
42

```

Listing 5.4: Código JavaScript para el número dinámico de campos.

5.1.5. Periodo 5

En este periodo se implementan las funcionalidades de la aplicación pertenecientes al usuario estudiante, es decir, visualizar y analizar un caso, responder una pregunta asociada a este y visualizar las correcciones.

Para visualizar y analizar un caso se crea una vista en la que se muestra la imagen del electrocardiograma a analizar y la pregunta a responder. Como queremos garantizar la correcta visualización del electrocardiograma se implementa una función JavaScript que permite hacer zoom sobre la imagen con la rueda del ratón.

Para la herramienta de medición se han realizado diversas funciones JavaScript, una función para tomar las referencias que se utilizarán para calcular la escala y tomar los puntos sobre los que realizar las mediciones y otras dos funciones para borrar los datos de cálculo y referencias, estas tres funciones se pueden ver en los Listados 5.5 y 5.6. El estado de la referencia y los datos de la medición siempre es visible en la pantalla.


```

1  var ref_x = null, ref_y = null, ref2_x = null, ref2_y = null, pos_x = null,
    pos_y = null, tiempo, voltaje;
2
3  //Se tomará una referencia de 10mm sobre el estado de la imagen en el que se
    quiera medir
4  function point_it(event) {
5      if (ref_x == null) { //Si no se ha tomado el inicio de la referencia, este se
        guarda
6          ref_x = event.offsetX ? (event.offsetX) : event.pageX
7          - document.getElementById("imagen").offsetLeft;
8          ref_y = event.offsetY ? (event.offsetY) : event.pageY
9          - document.getElementById("imagen").offsetTop;
10         document.getElementById("ref1").style.left = ref_x + "px";
11         document.getElementById("ref1").style.top = ref_y + "px";
12         document.getElementById("referencia").textContent = "Creando□referencia";
13         document.getElementById("referencia").style.color = "orange";
14     } else if (ref2_x == null) { //Si no se ha tomado el fin de la referencia,
        este se guarda
15         ref2_x = event.offsetX ? (event.offsetX) : event.pageX
16         - document.getElementById("imagen").offsetLeft;
17         ref2_y = event.offsetY ? (event.offsetY) : event.pageY
18         - document.getElementById("imagen").offsetTop;
19         distref_x = Math.abs(ref2_x - ref_x);
20         distref_y = Math.abs(ref2_y - ref_y);
21         document.getElementById("ref2").style.left = ref2_x + "px";
22         document.getElementById("ref2").style.top = ref2_y + "px";
23         document.getElementById("referencia").textContent = "Referencia□guardada";
24         document.getElementById("referencia").style.color = "green";
25     } else if (pos_x == null) { //Si es el preimer click de la medición
26         pos_x = event.offsetX ? (event.offsetX) : event.pageX
27         - document.getElementById("imagen").offsetLeft;
28         pos_y = event.offsetY ? (event.offsetY) : event.pageY
29         - document.getElementById("imagen").offsetTop;
30         document.pointform.form_x.value = pos_x;
31         document.pointform.form_y.value = pos_y;
32         document.getElementById("vert1").style.left = pos_x + "px";
33         document.getElementById("hor1").style.top = pos_y + "px";
34     } else { //Toma de punto final de la medición y cálculo de las distancias y
        valores en función de la referencia
35         distancia_x = event.offsetX ? (event.offsetX) : event.pageX
36         - document.getElementById("imagen").offsetLeft;
37         distancia_y = event.offsetY ? (event.offsetY) : event.pageY
38         - document.getElementById("imagen").offsetTop;
39
40         document.getElementById("vert2").style.left = distancia_x + "px";
41         document.getElementById("hor2").style.top = distancia_y + "px";
42         distancia_x = (distancia_x - pos_x) * 10 / distref_x;
43         distancia_y = (distancia_y - pos_y) * 10 / distref_y;
44         document.pointform.dist_x.value = distancia_x;
45         document.pointform.dist_y.value = distancia_y;
46         tiempo = distancia_x / 25 * 1000;
47         voltaje = distancia_y / 10;
48         document.getElementById("tiempo").innerHTML = Math.abs(tiempo)
49         .toFixed(1);
50         document.getElementById("voltaje").innerHTML = Math.abs(voltaje)
51         .toFixed(1);
52     }
53 }
54

```

Listing 5.5: Función JavaScript para tomar referencia y puntos para la medición.

```
1 //Se eliminan los valores referencia para la medición, por lo que el siguiente
  click será tratado como el primero.
2 function borrar_medicion() {
3     pos_x = null;
4     pos_y = null;
5     document.getElementById("vert1").style.left = "0px";
6     document.getElementById("vert2").style.left = "0px";
7     document.getElementById("hor1").style.top = "0px";
8     document.getElementById("hor2").style.top = "0px";
9 }
10
11 //Se eliminan los valores de referencia y se ejecuta el borrado de datos
12 function borrar_referencia() {
13     ref_x = null;
14     ref_y = null;
15     ref2_x = null;
16     ref2_y = null;
17     document.getElementById("ref1").style.left = "0px";
18     document.getElementById("ref1").style.top = "0px";
19     document.getElementById("ref2").style.left = "0px";
20     document.getElementById("ref2").style.top = "0px";
21     document.getElementById("referencia").textContent = "Referencia no tomada";
22     document.getElementById("referencia").style.color = "red";
23     borrar_medicion();
24 }
25
```

Listing 5.6: Funciones JavaScript para borrar datos y referencia.

Una vez el usuario envía una respuesta, el sistema decide si es correcta o incorrecta y almacena la respuesta en la base de datos. Tras esto redirige al usuario a la página correspondiente, le muestra la corrección de su respuesta si esta era incorrecta o la solución completa del caso si era correcta.

5.1.6. Periodo 6

Durante este último periodo se dedica la mayor parte del tiempo a perfilar detalles de la implementación y a desarrollar medidas de seguridad, por ejemplo, la restricción de acceso a cada parte de la herramienta para los usuarios no autorizados.

Una vez finalizada la herramienta, esta se construye en un archivo WAR que se pueda desplegar en un servidor Tomcat y se realizan pruebas tanto en local como en un servidor del hospital facilitado para esta tarea.

5.2. Verificación y validación

A lo largo del desarrollo del proyecto se han ido realizando todo tipo de pruebas sobre cada función que se iba implementando, cada una de estas funciones ha sido probada como útil y funcional. Además, tras finalizar la implementación de cada parte importante, se realizaba una reunión con los médicos para mostrar el avance y probar que la herramienta satisficiera los requisitos y objetivos marcados.

Al finalizar la herramienta esta ha sido desplegada en un servidor del dominio del hospital, no accesible desde el exterior y presentada a los médicos y a la tutora en una demostración de las distintas funcionalidades.

5.2.1. Trabajo futuro

En cuanto al trabajo futuro a realizar para la total implantación de la herramienta surgen 2 necesidades principales:

1. ¿Dónde se alojará la herramienta? Debido a que, inicialmente, ésta herramienta va a ser utilizada por estudiantes de la Universitat Jaume I, se plantea la opción de que el almacenamiento sea realizado en los servidores de la UJI.
2. ¿Cómo se realizará el registro de estudiantes? Igual que en el punto anterior, como los estudiantes serán alumnos de la UJI, se plantea que la creación de usuarios estudiante pueda realizarse tomando como fuente la lista de estudiantes de una asignatura específica.

Capítulo 6

Conclusiones

En el ámbito formativo he conseguido aplicar los conocimientos adquiridos durante el grado en Ingeniería Informática, así como completar estos conocimientos con otros que he ido adquiriendo a lo largo del desarrollo del proyecto. Sobre todo he podido aprender bastante más JavaScript ya que es un lenguaje que no se utiliza mucho durante la carrera.

En el ámbito profesional ha sido la primera vez dentro de un organismo del sector público, esto me ha hecho conocer como funciona una unidad de informática dentro de un hospital, las funciones que se desempeñan y la organización de cada apartado.

En el ámbito personal me he sentido capaz de desarrollar un proyecto con unos objetivos y un plazo marcados. Además me he sentido integrado entre los demás miembros de la unidad de informática. Gran parte de esto la tiene que, aunque no lo sabía antes de llegar, a la mayoría los había conocido con anterioridad.

Bibliografía

- [1] Página del artículo en la tienda de la UJI. http://www.tenda.uji.es/pls/iglu/!GCPA00.GCPPR0002?id_art=1607&lg=UK. [Consulta: 10 de Junio de 2019].
- [2] Documentación sobre html en el soporte a desarrolladores de Mozilla. <https://developer.mozilla.org/es/docs/Web/HTML>. [Consulta: 1 de Junio de 2019].
- [3] Documentación sobre CSS en el soporte a desarrolladores de Mozilla. https://developer.mozilla.org/es/docs/Learn/CSS/Introduction_to_CSS/Como_funciona_CSS. [Consulta: 1 de Junio de 2019].
- [4] JavaScript a fondo. <https://desarrolloweb.com/javascript>. [Consulta: 1 de Junio de 2019].
- [5] Documentación sobre PostgreSQL en su página web. <https://www.postgresql.org/about>. [Consulta: 5 de Junio de 2019].
- [6] Definición de Java en Wikipedia. [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language)). [Consulta: 1 de Junio de 2019].
- [7] Página web de Apache Tomcat. <https://tomcat.apache.org/>. [Consulta: 2 de Junio de 2019].
- [8] Página web de UMLet. <https://www.umlet.com/>. [Consulta: 7 de Junio de 2019].
- [9] Página web de MockFlow. <https://mockflow.com>. [Consulta: 3 de Junio de 2019].
- [10] Página web de pgAdmin. <https://www.pgadmin.org/>. [Consulta: 7 de Junio de 2019].
- [11] Página web de Jasypt. <http://www.jasypt.org/>. [Consulta: 10 de Junio de 2019].

Anexo A

Guía de usuario de la herramienta

A.1. Estudiante

1. Al acceder a la página web, nos encontraremos con la página de inicio, en la que pulsamos en iniciar sesión.
2. Introduciremos nuestro nombre de usuario y contraseña para acceder al sistema.
3. Una vez identificados nos encontraremos con el índice de contenidos, pulsaremos en el apartado sobre el que queramos trabajar y accedemos a un caso práctico.
4. Ahora nos encontramos con la imagen del electrocardiograma y una serie de respuestas a una pregunta planteada. Para tomar medidas sobre la imagen mostrada, debemos:

Tomar una referencia de 10mmX10mm en la imagen, recuerda que debes tomar la referencia en el estado de zoom sobre el que vayas a realizar la medición (esto nos lo recuerda la aplicación mediante un mensaje en pantalla).

Tomar dos puntos sobre los que se desee realizar la medición.

Analizar los resultados obtenidos.

5. Una vez analizada la pregunta, enviamos la respuesta que creemos oportuna y nos mostrará el resultado.

A.2. Profesor

A.2.1. Introducción de un nuevo caso

1. Accedemos al apartado de casos del sistema.
2. En la parte inferior de la pantalla encontramos el botón de añadir caso.
3. Nos encontraremos con el formulario para introducir los datos del nuevo caso, lo rellenamos y enviamos.

4. Una vez creado el caso, lo encontraremos en el apartado de casos.

A.2.2. Añadir una nueva pregunta a un caso

1. Accedemos al apartado de casos del sistema.
2. Pulsamos sobre añadir pregunta en el caso deseado.
3. Nos encontraremos con el formulario para introducir el enunciado y las respuestas de la pregunta.
4. Una vez creadas todas las respuestas pulsamos en añadir pregunta y ya tendremos la nueva pregunta relacionada con el caso deseado.