

DESARROLLO DE UN SISTEMA INTERACTIVO
PARA E-CONSULTAS MÉDICAS
DEVELOPMENT OF AN INTERACTIVE SYSTEM
FOR E-MEDICAL CONSULTATIONS



TRABAJO FIN DE GRADO
CURSO 2023-2024

AUTOR
BEGOÑA PASCUAL GOENECHEA
PATRICIA PASTOR PUENTE

DIRECTOR
ANTONIO SARASA CABEZUELO
JAVIER CAMBRONERO SANTOS

GRADO EN INGENIERÍA DE SOFTWARE
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

DESARROLLO DE UN SISTEMA INTERACTIVO
PARA E-CONSULTAS MÉDICAS
DEVELOPMENT OF AN INTERACTIVE SYSTEM
FOR E-MEDICAL CONSULTATIONS

TRABAJO DE FIN DE GRADO EN INGENIERÍA DE SOFTWARE

AUTOR

BEGOÑA PASCUAL GOENECHEA
PATRICIA PASTOR PUENTE

DIRECTOR

ANTONIO SARASA CABEZUELO
JAVIER CAMBRONERO SANTOS

CONVOCATORIA: JUNIO 2024

GRADO EN INGENIERÍA DE SOFTWARE
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

27 DE MAYO DE 2024

DEDICATORIA

A nosotras mismas y a nuestra valía como
ingenieras de software a pesar de las
dificultades.

AGRADECIMIENTOS

Nos gustaría agradecer a todas las personas que han contribuido en el éxito de este trabajo de fin de grado, como posibles usuarios de la aplicación ya que nos han podido dar un enfoque más objetivo y a conseguir sacar su mejor versión.

En especial, a nuestro tutor Antonio Sarasa Cabezuelo por la paciencia brindada y los ánimos durante toda la fase del proyecto; así como a Javier Cambronero como cliente que nos ha dado total libertad a la hora de desarrollar la creatividad, así como un feedback positivo del desarrollo y evolución de nuestro trabajo.

Por último, a las familias, amigos y compañeros de clase que nos han alentado siempre a confiar en nosotras y mejorar.

RESUMEN

Desarrollo de un sistema interactivo para e-consultas médicas

El objetivo de este trabajo de fin de grado consiste en realizar una página web donde usuarios registrados como pacientes accederán a un portal de consultas médicas en el que podrán reservar citas, ver sus historiales clínicos además de subir sus propios informes procedentes de otras instituciones.

En cuanto a los usuarios registrados como médicos, tendrán acceso a una sección con diferentes listados en los que podrán llevar la gestión de todo lo mencionado anteriormente, así como de la información de cada paciente. Además, el portal cuenta con un sistema de mensajería instantánea en el que se facilita la relación médico-paciente para la resolución de dudas u obtener mayor información de cara al momento de la consulta.

Palabras clave

Consultas, medicina, informes, pacientes, página web, correspondencia, Stripe.

ABSTRACT

Development of an interactive system for e-medical consultations

The objective of this final degree project is to create a web page where users registered as patients will have access to a medical consultation portal where they will be able to book appointments, view their clinical histories and upload their own reports from other institutions.

As for users registered as doctors, they will have access to a section with different lists in which they will be able to manage everything mentioned above, as well as the information of each patient.

In addition, the portal has an instant messaging system that improves the doctor-patient relationship for the resolution of doubts or to obtain more information prior to or after the time of the consultation.

Keywords

Consultations, medicine, reports, patients, website, correspondence, Stripe.

ÍNDICE DE CONTENIDOS

Capítulo 1 - Introducción	1
1.1 Motivación	1
1.2 Objetivos	1
1.3 Plan de trabajo.....	2
1.3.1 Planificación.....	3
1.3.2 Desarrollo	3
1.3.3 Maquetación	4
1.3.4 Revisión	4
Chapter 1 - Introduction.....	5
1.1 Motivation	5
1.2 Goals.....	5
1.3 Work plan	6
1.3.1 Planning	7
1.3.2 Development	7
1.3.3 Layout	8
1.3.4 Review	8
Capítulo 2 - Tecnologías.....	9
2.1 Microsoft OneNote.....	9
2.2 Azure DevOps	9
2.3 Visual Studio Code	10
2.4 Github	11
2.5 XAMPP	11
2.6 HTML.....	11

2.7 CSS	11
2.8 Bootstrap	12
2.9 JavaScript.....	12
2.10 Ajax	12
2.11 Stripe	12
2.12 Ethereal Mail	12
Capítulo 3 - Estado del arte	13
3.1 Quironsalud.....	13
3.2 Top Doctors	14
3.3 HM Hospitales	14
3.4 Servicios salud Comunidad de Madrid	15
3.5 Hospital San Rafael	15
Capítulo 4 - Arquitectura y modelo de datos	17
4.1 Arquitectura de la aplicación	17
4.2 Estructura de la base de datos	19
4.2.1 Tabla usuarios.....	21
4.2.2 Tabla informes.....	22
4.2.3 Tabla citas	22
4.2.4 Tabla mensajes	23
4.2.5 Tabla sessions	24
4.2.6 Tabla tokens	25
Capítulo 5 - Diseño	27
Capítulo 6 - Funcionalidad	31
6.1 Módulo autenticación.....	31
6.2 Módulo gestión de pacientes	36

6.3 Módulo gestión de citas.....	40
6.4 Módulo gestión de informes	42
6.5 Módulo consultas médicas	44
Capítulo 7 - Pasarela de pago	47
7.1 Análisis de las herramientas	47
7.2 Investigación de la pasarela Stripe.....	47
7.3 Creación de la API.....	48
7.4 Implementación	48
Capítulo 8 - Conclusiones y trabajo futuro	55
8.1 Conclusiones.....	55
8.2 Trabajo futuro.....	55
Chapter 8 - Conclusions and future work	57
Contribuciones Personales.....	59
Apéndice A - Casos de uso	69
Apéndice B - Guía del usuario	103

ÍNDICE DE TABLAS

Figura 1. Diagrama de Gantt	2
Figura 2. Gantt chart.....	6
Figura 3. Bloc de notas en OneNote	9
Figura 4. Tablero en Azure DevOps.....	10
Figura 5. Proyecto en Visual Studio Code	10
Figura 6. Diagrama de arquitectura de la aplicación	17
Figura 7. Diagrama entidad-relación	19
Figura 8. Base de datos	20
Figura 9. Tabla usuarios.....	21
Figura 10. Tabla informes.....	22
Figura 11. Tabla citas	23
Figura 12. Tabla mensajes	24
Figura 13. Tabla sessions	24
Figura 14. Ejemplo sesión activa.....	24
Figura 15. Tabla tokens	25
Figura 16. Cabecera de una plantilla EJS	27
Figura 17. Hojas de estilos CSS	28
Figura 18. Ejemplo de navbar en Bootstrap.....	29
Figura 19. Ventana modal con características.....	29
Figura 20. Página inicio con modales y objetos incluidos.....	30
Figura 21. Formulario de registro I.....	32
Figura 22. Formulario de registro II.....	32
Figura 23. Vista registro	33

Figura 24. Formulario login.....	33
Figura 25. Vista login	34
Figura 26. Implementación recuperar contraseña.....	34
Figura 27. Vista recuperar contraseña	35
Figura 28. Función generar token de restablecimiento	35
Figura 29. Función enviar correo de restablecimiento.....	36
Figura 30. Correo electrónico con enlace para recuperar contraseña.....	36
Figura 31. Ventana modal editar perfil.....	37
Figura 32. Tabla lista pacientes	38
Figura 33. Vista tabla pacientes.....	38
Figura 34. Implementación ver paciente.....	39
Figura 35. Ficha técnica paciente	39
Figura 36. Barra de búsqueda	40
Figura 37. Función automática buscar.....	40
Figura 38. Vista de barra de búsqueda.....	40
Figura 39. Modal reservar cita	41
Figura 40. Modal reservar cita con opción "Pagar en consulta".....	42
Figura 41. Solicitud ver informe al servidor mediante end()	43
Figura 42. Vista mis informes desde paciente.....	43
Figura 43. Implementación bandeja de entrada	44
Figura 44. Vista mensajes recibidos.....	45
Figura 45. Script que genera el PDF a partir de la tabla	45
Figura 46. PDF generado para imprimir.....	46
Figura 47. Tienda online en Stripe.....	48
Figura 48. Claves secreta y pública de Stripe	49

Figura 49. Obtención de productos y precios desde Stripe	49
Figura 50. Asignación del atributo data-price a cada producto	50
Figura 51. Creación objeto con producto seleccionado.....	50
Figura 52. Pasarela de pago con consulta de tipo videoconferencia.....	51
Figura 53. Almacenar datos del formulario de manera local en objeto formData	52
Figura 54. Acceso al objeto formData para enviarlo al servidor	52
Figura 55. Ventana "Pagos" en Stripe.....	53
Figura 56. Diagrama de caso de uso UNR	69
Figura 57. Diagrama de caso de uso paciente	70
Figura 58. Diagrama de caso de uso médico.....	70
Figura 59. Inicio	103
Figura 60. Registro con errores.....	104
Figura 61. Inicio sesión.....	105
Figura 62. Modal inicio sesión desde "Pedir cita".....	105
Figura 63. Modal reservar cita	106
Figura 64. Ventana de pago	106
Figura 65. Vista mis citas desde paciente	107
Figura 66. Mis informes desde paciente	107
Figura 67. Ventana modal subir informe	108
Figura 68. Bandeja de entrada	108
Figura 69. Ver mensaje	109
Figura 70. Responder mensaje	109
Figura 71. Vista redactar mensaje	110
Figura 72. Eliminar perfil	110
Figura 73. Editar perfil.....	111

Figura 74. Barra de navegación con lista desplegable	111
Figura 75. Lista pacientes activos.....	112
Figura 76. Ver paciente	112
Figura 77. Pacientes archivados	113
Figura 78. Próximas citas.....	113
Figura 79. Citas con el filtro "telefónicas".....	114
Figura 80. PDF generado al descargar lista de citas	114

ÍNDICE DE TABLAS

Tabla 1. Registro	73
Tabla 2. Login.....	74
Tabla 3. Logout	75
Tabla 4. Recuperación de contraseña	76
Tabla 5. Dar de alta paciente	77
Tabla 6. Buscar paciente.....	78
Tabla 7. Dar de baja paciente	79
Tabla 8. Modificar perfil	80
Tabla 9. Eliminar perfil	81
Tabla 10. Reservar cita.....	82
Tabla 11. Cancelar cita.....	83
Tabla 12. Modificar cita.....	84
Tabla 13. Finalizar cita.....	85
Tabla 14. Ver citas	86
Tabla 15. Notificar cita.....	87
Tabla 16. Subir informe.....	88
Tabla 17. Modificar informe	90
Tabla 18. Eliminar informe.....	92
Tabla 19. Ver informe	93
Tabla 20. Guardar informe	94
Tabla 21. Buscar informe	95
Tabla 22. Listar informes.....	96
Tabla 23. Listar informes de un paciente.....	97

Tabla 24. Iniciar videoconferencia	98
Tabla 25. Finalizar videoconferencia	99
Tabla 26. Subir documento / imagen / vídeo	100
Tabla 27. Enviar mensaje	101
Tabla 28. Ver mensaje	102

Capítulo 1 - Introducción

1.1 Motivación

Este trabajo de Fin de Grado está basado en desarrollar una aplicación web para la gestión de consultas médicas, así como llevar un control de los distintos historiales médicos de los pacientes, para así poder gestionar sus tratamientos de manera online, rápida y sencilla.

La principal motivación de este trabajo viene dada por el aumento del uso de las tecnologías desde los últimos años y su rápida evolución, pretendiendo facilitar la relación médico-paciente a través de internet para casos en los que no se requiera la asistencia presencial. Para ello se hará uso de la aplicación implementada en este proyecto que permita al usuario realizar estas consultas intuitivamente.

1.2 Objetivos

El objetivo principal de este trabajo es ofrecer a los usuarios, tanto los médicos como los pacientes, una experiencia completa para llevar a cabo la gestión y organización de un sistema de consultas médicas virtual.

A su vez, este objetivo se divide en varios más concretos:

- Gestionar reservas de citas: el primer objetivo consiste en organizar mediante un calendario de reservas, todas las citas posibles a elegir por el usuario con los distintos médicos del hospital, dentro de los horarios pautados por estos.
- Realizar consultas médicas: este es el objetivo principal, poder realizar una consulta vía online, a través del teléfono o mediante videoconferencia con el médico para que se puedan tratar sus síntomas de manera eficiente.
- Interacción médico-paciente: además de las consultas, existe una bandeja de entrada con la posibilidad de mantener conversaciones entre el médico y el paciente de manera instantánea, sin tener que recurrir a otras herramientas como pueden ser el correo electrónico, y poder tenerlo todo en la misma aplicación.

- Gestionar informes: otro objetivo presente en el proyecto es la subida y visualización de informes tanto por parte del médico, como del paciente; que facilite el intercambio de información para llevar a cabo las consultas con mayor precisión. Además, el paciente también podrá subir sus informes de otros centros o consultas para brindar mayor detalle al médico.
- Gestionar pacientes: uno de los objetivos que facilitarán la tarea al médico es la gestión de pacientes mediante varias tablas en la aplicación que proporcionan a grandes rasgos toda la base de datos de los pacientes mediante distintas categorías (por ejemplo, pacientes pasados o archivados), además de información específica de cada uno mediante una ficha individual.
- Notificaciones: este objetivo pretende simplemente mantener al usuario informado en todo momento de las citas reservadas, así como de cualquier cambio o actividad realizado en la aplicación.

1.3 Plan de trabajo

En esta sección se presenta el plan de trabajo llevado a cabo a principio de curso para realizar este proyecto, que incluye todos los objetivos del apartado anterior, con sus fechas de estimación en la Figura 1 representado con un diagrama de Gantt realizado con la aplicación web Canva [1].

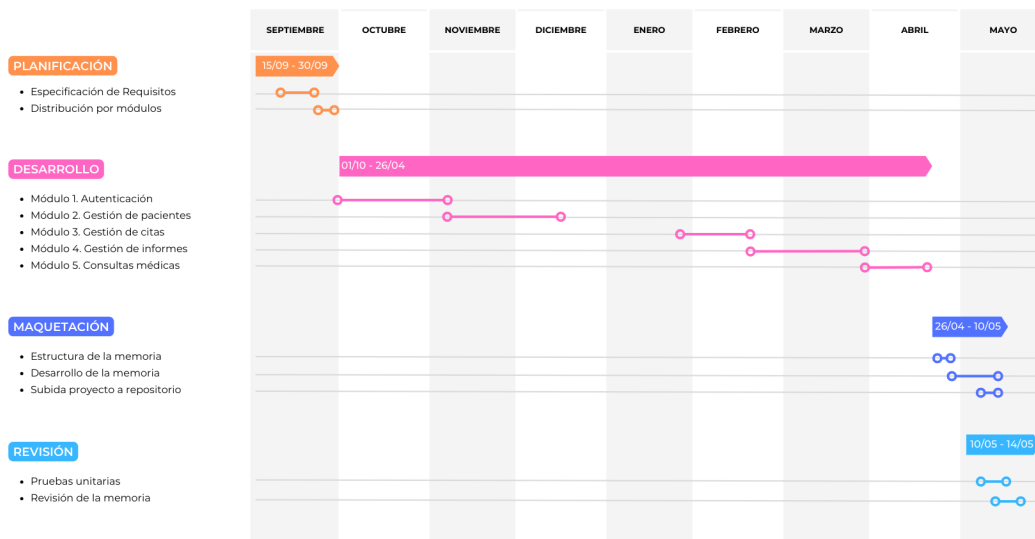


Figura 1. Diagrama de Gantt

A continuación, se explican las fases específicas que se han seguido para el completo desarrollo del trabajo.

1.3.1 Planificación

Esta primera fase sirvió para organizar el proyecto de cara al curso completo, definiendo de manera general el trabajo que se quería implementar y hasta dónde se pretendía llegar. Se empezó realizando una especificación de requisitos con todas las funcionalidades que irían dando forma a la aplicación web a través de hitos marcados junto al tutor.

1.3.2 Desarrollo

Una vez definidos los requisitos, se organizaron por módulos, de forma incremental, para que la aplicación tuviera sentido y fuera sencillo implementarla de manera lógica. Para ello, se establecieron los siguientes módulos:

1.3.2.1 Módulo de autenticación

En primer lugar, sería necesario un formulario tanto de registro, como de *login* para poder acceder al portal de consultas médicas.

1.3.2.2 Módulo de pacientes

Una vez iniciada sesión, sería importante gestionar todos los datos de los usuarios registrados mediante una base de datos local, pudiendo mostrar en una lista de manera visual todos los pacientes al médico correspondiente.

1.3.2.3 Módulo de citas

Ya implementado el módulo 2, los pacientes tendrían acceso a otro pequeño formulario en el que poder reservar sus citas, mediante un calendario que permitiera ver qué médicos o días disponibles pudieran solicitar una consulta. Cabe destacar, en este módulo se implementaron tres tipos de citas, habladas previamente con el cliente: telefónicas, por videoconferencia o de manera presencial.

1.3.2.4 Módulo de informes

El siguiente paso para darle forma al proyecto, sería implementar un sistema, también mediante un formulario, que permitiera subir archivos de tipo PDF donde los médicos, pudieran ver los informes que constituirían su historial médico. Se tuvo en cuenta la posibilidad de añadir esta funcionalidad también al paciente para que facilitara la relación con el médico a través de informes de otras consultas o instituciones.

1.3.2.5 Módulo de consultas médicas

Por último, se llevó a cabo el módulo que fusionaría todos los anteriores, dando lugar a la parte más técnica, las consultas médicas. Es entonces cuando se decidió que el médico y el paciente podrían comunicarse de manera rápida mediante una bandeja de entrada a modo de chat o correo electrónico interno para poder especificar los detalles de la cita, en el caso de ser vía online. Para la gestión de la cita como tal, se utilizará una tecnología ya existente como puede ser Google Meet, Zoom o Skype, de modo que el médico pueda enviarle un enlace privado al paciente para acceder a la sala.

1.3.3 Maquetación

Una vez finalizada la aplicación web, sería hora de llevar a cabo esta memoria, así como toda la documentación requerida para ello. Se tendría una reunión con el tutor para hablar acerca de la estructura de la memoria y el plazo de entrega.

Una vez finalizada, el proyecto estaría listo para ser subido de manera pública a un repositorio, al que se puede acceder mediante este [enlace](#).

1.3.4 Revisión

El último paso para dar por finalizado este trabajo de fin de grado, sería revisar tanto la memoria como toda la implementación del código, permitiendo así la gestión y análisis de posibles fallos que poder solucionar. Añadiendo a lo anterior, se realizarían pruebas con el cliente y con personas externas para obtener una valoración general y externa de la aplicación.

Chapter 1 - Introduction

1.1 Motivation

This Bachelor's Thesis is based on developing a web application for managing medical appointments, as well as keeping track of patients' medical records, in order to monitor their treatments online, quickly, and easily.

The main motivation for this work is given from the increased use of technologies in recent years and their rapid evolution, aiming to improve the doctor-patient relationship through the Internet for cases where in-person assistance is not required. To achieve this, the application implemented in this project will be used, allowing users to make these appointments intuitively.

1.2 Goals

The main objective of this work is to offer users, both doctors and patients, a complete experience to carry out the management and organization of a virtual medical consultation system.

In turn, this objective is divided into several more specific ones:

- Manage appointment booked: the first objective is to organize through a booking calendar, all possible appointments to be chosen by the user with the different doctors of the hospital, within the schedules set by them.
- Medical consultations: this is the main objective, to be able to make a consultation online, by telephone or by videoconference with the doctor so that your symptoms can be treated efficiently.
- Doctor-patient interaction: in addition to consultations, there is an inbox with the possibility of instant conversations between the doctor and the patient, without having to resort to other tools such as e-mail and having everything in the same application.
- Manage reports: another objective of the project is the uploading and visualization of reports by both the doctor and the patient, which makes more efficient the exchange of information to carry out consultations with greater

precision. In addition, the patient will also be able to upload reports from other centers or consultations to provide greater detail to the doctor in charge.

- Manage patients: one of the objectives that will facilitate the doctor's task is the management of patients through several tables in the application that provides, roughly speaking, the entire database of patients through different categories (for example, past or archived patients), as well as specific information on each one through an individual file.
- Notifications: this goal is simply to keep the user informed at all times of booked appointments, as well as any changes or activity performed in the application.

1.3 Work plan

This section presents the work plan carried out at the beginning of the course to work on this project, which includes all the objectives of the previous section, with their estimated dates, in Figura 2 represented with a Gantt chart made with the web application Canva [1].

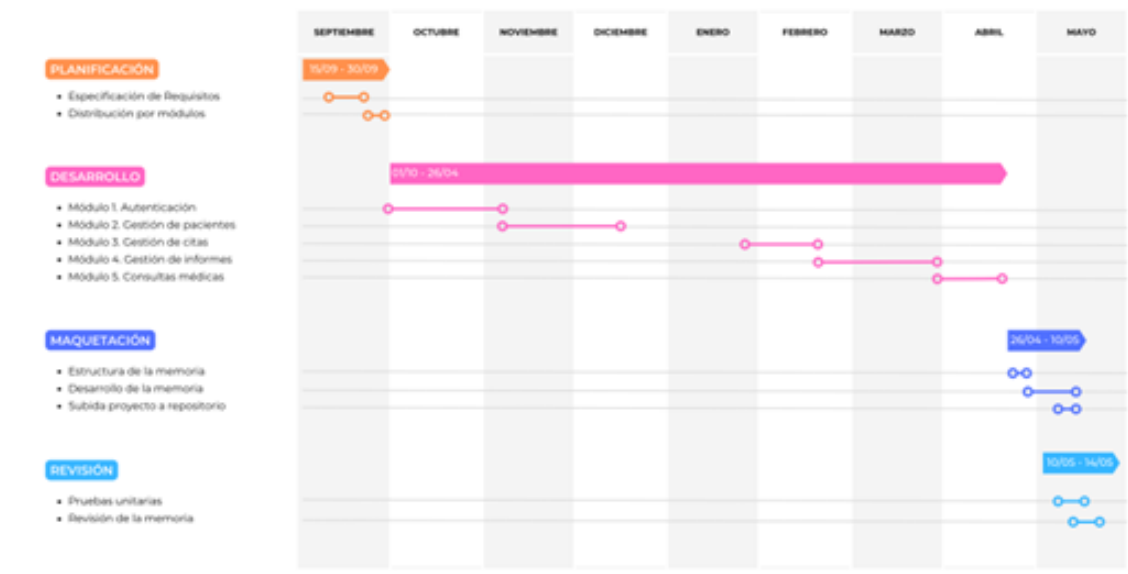


Figura 2. Gantt chart

The specific phases followed for the complete development of the work are explained below.

1.3.1 Planning

This first phase was used to organize the project for the whole course, defining in a general way, the work to be implemented and how far we wanted to go. We started by making a requirements specification with all the functionalities that would shape the web application through milestones set with the tutor.

1.3.2 Development

Once the requirements were defined, they were organized by modules, in an incremental way, so that the application would make sense and would be easy to implement in a logical way. For this purpose, the following modules were established:

1.3.2.1 Authentication module

First of all, a registration and login form would be necessary to access the medical consultation portal.

1.3.2.2 Patients' module

Once logged in, it would be important to manage all the data of the registered users through a local database, being able to visually display all the patients to the corresponding doctors in a list.

1.3.2.3 Appointment module

Once module 2 was implemented, patients would have access to another small form in which they could book their appointments, through a calendar that would allow them to see which doctors or available days they could request a consultation. It is worth highlighting that in this module three types of appointments were implemented, previously discussed with the client: telephone, videoconference or face-to-face.

1.3.2.4 Reports module

The next step to give shape to the project would be to implement a system, also through a form, that would allow uploading .pdf files where doctors could view the reports that would constitute the patient's medical history. We considered

the possibility of adding this functionality also to the patient to facilitate the interaction with the doctor through reports from other offices or institutions.

1.3.2.5 Medical consultation module

Finally, the module that would merge all the previous ones was implemented, giving rise to the most technical part, the medical consultations. It was then decided that the doctor and the patient could communicate quickly through an inbox as a chat or internal email for example to specify the details of the appointment, in the case of being online. For the management of the appointment, existing technology such as Google Meet, Zoom or Skype will be used, so that the doctor can send a private link to the patient to access the room.

1.3.3 Layout

Once the web application is finished, it would be time to carry out this report, as well as all the documentation required for it. A meeting would be held with the tutor to discuss the structure of the report and the deadline.

Once finalized, the project would be ready to be uploaded publicly to a repository, which can be accessed through this [link](#).

1.3.4 Review

The last step to finalize this final degree work would be to review both the memory and the entire implementation of the code, allowing the management and analysis of possible failures that can be solved.

In addition to the above, tests would be carried out with the client and with external people to obtain a general and external assessment of the application.

Capítulo 2 - Tecnologías

2.1 Microsoft OneNote

En primer lugar, se ha hecho uso de la herramienta Microsoft OneNote [2], que es un bloc de notas digital. En él se pueden crear diferentes blogs como si fueran cuadernos de papel en el que ir añadiendo páginas o secciones con apuntes. En este caso se ha utilizado para ir anotando todo lo necesario para la realización de esta memoria, como se puede ver en la Figura 3

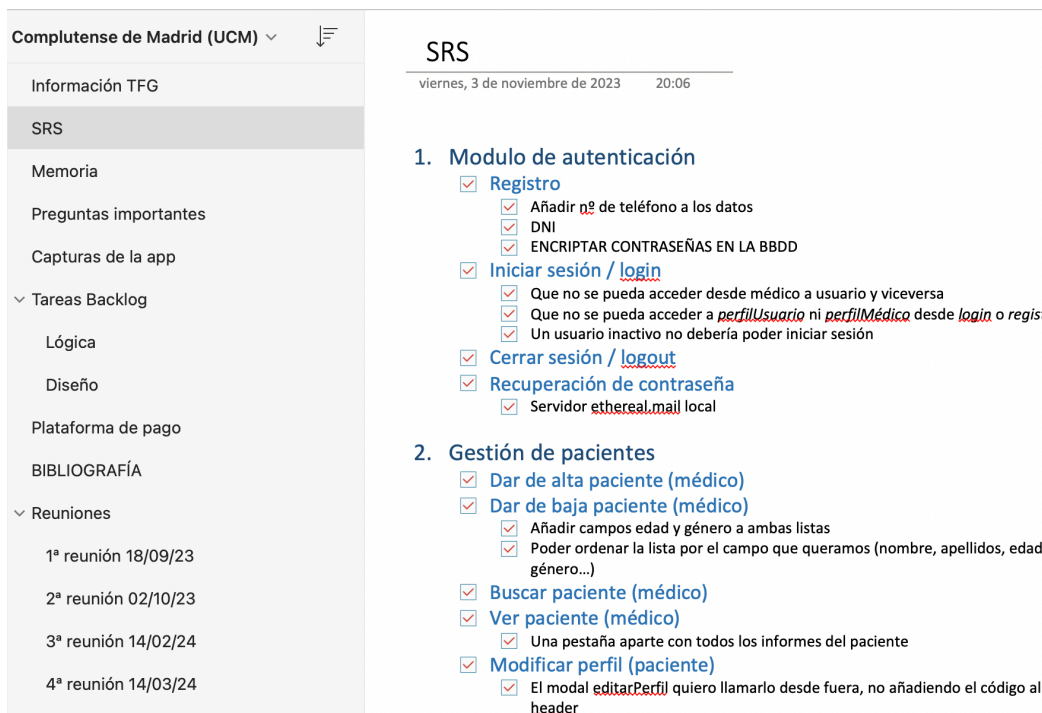


Figura 3. Bloc de notas en OneNote

2.2 Azure DevOps

Azure DevOps [3] es una herramienta en la nube proporcionada por Microsoft que permite organizar las tareas a través de tableros. En este caso se ha realizado la implementación de los módulos a través de Sprints, incluyendo dentro de cada uno las distintas tareas que lo formaban. Además, permite ir documentando mediante subtareas cómo se ha realizado la implementación de cada una de estas;

categorizando en diferentes estados: "To Do", "Doing" y "Done", como se muestra en la Figura 4, permitiendo ver al usuario en todo momento el estado del proyecto.

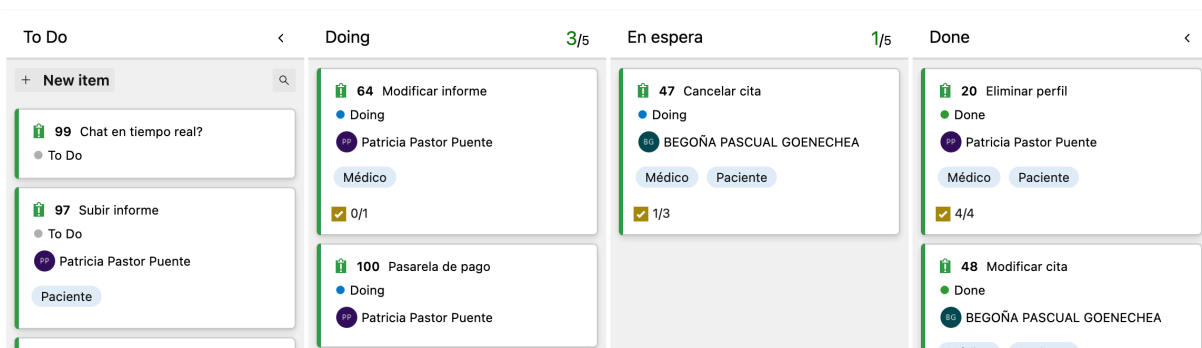


Figura 4. Tablero en Azure DevOps

2.3 Visual Studio Code

La herramienta más utilizada durante este proyecto ha sido Visual Studio Code [4], un editor de código fuente desarrollado por Microsoft que permite, gracias a sus numerosas extensiones y complementos, desarrollar código de manera muy intuitiva y sencilla en prácticamente todos los lenguajes de programación, organizado por carpetas, como se puede ver en la Figura 5. Proyecto en Visual Studio Code. Además, permite integrarse con la aplicación Github.

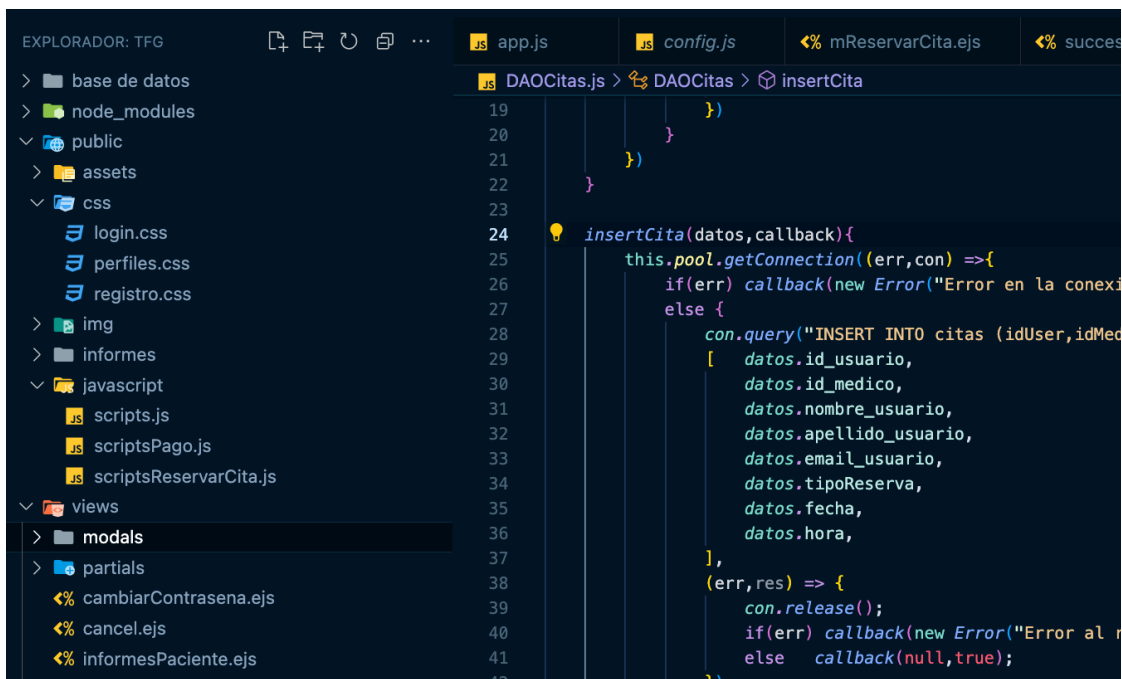


Figura 5. Proyecto en Visual Studio Code

2.4 Github

Github [5] es una herramienta que te permite alojar el código mediante repositorios a través de versiones, de manera online, para poder acceder a proyecto en múltiples dispositivos y estar sincronizado con otras personas. Cuenta también con una aplicación de escritorio integrada con Visual Studio Code, mencionada anteriormente, que permite la actualización de los cambios y su subida a la nube de manera rápida.

2.5 XAMPP

XAMPP [6] es una aplicación de escritorio de software libre que permite la gestión de bases de datos SQL con servidores de manera local sin necesidad de crear o tener un servidor complejo en la nube. Se puede descargar tanto para Mac como para Windows o Linux y simplemente iniciando los puertos que se necesiten ya se puede trabajar juntamente con la base de datos implementada en el proyecto. Sus siglas son un acrónimo que significan X (para múltiples dispositivos), A (servidor web Apache), M (base de datos MySQL), PP (PHP y Perl, dos lenguajes de programación).

2.6 HTML

Ya entrando en lo más técnico, cabe destacar el lenguaje más utilizado en el proyecto: HTML. [7] . Es un lenguaje utilizado para crear páginas web, con una sintaxis relativamente sencilla. Utiliza etiquetas para definir elementos como encabezados, títulos, párrafos, imágenes, formularios... facilitando la organización de lo que se desea mostrar. Junto con CSS y JavaScript, que se mencionan más adelante, permite una visualización completa y un diseño más personalizado para las páginas web.

2.7 CSS

Las hojas de estilo CSS. [8] , o *Cascading Style Sheets*, es un lenguaje utilizado en desarrollo web para definir el diseño y la presentación de páginas HTML. Permite separar el contenido de la presentación, ajustando los estilos como colores, fuentes o márgenes que se aplican en toda la página web. Junto a HTML, se pueden crear diferentes hojas de estilo para cada sección de la web, pudiendo diferenciar y organizar los estilos para cada usuario.

2.8 Bootstrap

Bootstrap [9] es una biblioteca de código abierto que permite a los desarrolladores la creación de páginas web con estilos *responsive* de manera fácil e intuitiva. Esto es, para la creación de aplicaciones móviles en conjunto con HTML y CSS también para organizar los elementos de la página a gusto de cada uno.

2.9 JavaScript

Es un lenguaje de programación de alto nivel utilizado por los programadores para darle dinamismo e interactividad a las páginas está presente del lado del cliente en el navegador web, no del servidor. JavaScript [10] es muy útil por ejemplo para la validación de formularios mencionada anteriormente.

2.10 Ajax

Ajax [11] es una técnica de desarrollo web que permite la actualización de partes específicas de las páginas web de manera asíncrona, permitiendo así una mayor fluidez y dinámica tanto en la visualización de la página como en la gestión de los datos de los formularios o la comprobación de estos sin necesidad de recargar la página completa.

2.11 Stripe

Stripe [12] es una plataforma de pago online preparada para ser utilizada en cualquier página web. Gracias a la documentación proporcionada, se permite la conexión, a través de una API, con una clave secreta única para poder llevar a cabo la venta de productos en un solo click.

2.12 Ethereum Mail

Ethereum Mail [13] [13] es un SMTP (Protocolo Simple de Transferencia de Correo) que actúa como si fuera un servidor real, para que los desarrolladores puedan realizar pruebas manera local. Gracias a su sencilla interfaz, permite la creación de una cuenta de correo de tipo Ethereum. Con el código necesario y el módulo nodemailer [14] será factible realizar y recibir correos electrónicos.

Capítulo 3 - Estado del arte

En este capítulo se ha llevado a cabo una investigación de aplicaciones web similares a este trabajo de fin de grado, con el objetivo de conseguir información e inspiración para la creación de la página web requerida para este trabajo de fin de grado. Además, han servido a modo de evaluación para ver qué funcionalidades se echan en falta como usuarios y poder implementarlas como desarrolladores.

3.1 Quironsalud

La aplicación de Quironsalud [15] permite a los usuarios utilizar una serie de funcionalidades y herramientas orientadas a la gestión de su salud y permitiendo mejorar y agilizar el proceso de atención médica.

A través de la aplicación una de las acciones que se pueden realizar es la reserva, modificación o cancelación de una cita médica de una forma más sencilla y cómoda, pudiendo evitar tener que hacer llamadas telefónicas o ponerse en contacto con el médico. Además, la aplicación permite un acceso rápido y seguro al historial médico del usuario, ofreciendo información sobre informes médicos, prescripciones médicas mediante recetas, resultados de pruebas y otros datos relevantes para la salud. Esto permite proporcionar al paciente un seguimiento detallado de su salud y a la hora de compartir la información con otros profesionales un fácil acceso.

La plataforma también tiene configurado un sistema de notificaciones y recordatorios que permite al paciente estar constantemente alerta de medicamentos que deben tomarse o próximas citas. En cuanto a las citas, aparte de poder pedir citas presenciales o telefónicas, los pacientes pueden realizarlas desde la comodidad de sus hogares mediante videollamadas, beneficiando así a aquellas consultas que no son urgentes o simplemente son para ver cómo va el seguimiento de un tratamiento ahorrando tiempo y desplazamiento al usuario.

3.2 Top Doctors

Top Doctors [16] es otra aplicación dentro del ámbito de la medicina que comparte los mismos fines comentados anteriormente en el apartado de la herramienta Quirónsalud.

A través de la aplicación, los usuarios pueden buscar fácilmente a especialistas y médicos altamente cualificados en sus áreas y pedir rápidamente cita con estos. Una de las principales características de la aplicación es el acceso rápido y seguro a información detallada sobre los profesionales, incluyendo su formación, su experiencia, sus áreas de especialidad y lo más relevante de cara al usuario, opiniones de otros pacientes. Esto permite al usuario tener toda la información en un mismo sitio y facilitar la toma de decisión para ver cuál será el profesional adecuado para lo que está buscando.

La aplicación también cuenta con la posibilidad de programar, modificar y cancelar citas, además de ofrecer notificación y recordatorio tanto de citas como de toma de medicamentos. Mantiene continuamente al usuario informado de todo lo relacionado con el cuidado de su salud mejorando el cumplimiento del tratamiento.

3.3 HM Hospitales

La aplicación HM Hospitales [17] es otra de las muchas dentro del ámbito de la medicina que también cuenta con las funciones básicas que estas proporcionan, pero centrándose que los hospitales de la cadena HM. A través de esta, los usuarios pueden buscar y encontrar fácilmente información sobre los hospitales y centros médicos HM, incluyendo la ubicación, los servicios que ofrecen y datos de contacto. Esto permite al usuario encontrar el centro médico más cercano a él y acceder a los servicios que este ofrece con facilidad. Como en las aplicaciones anteriores, esta también ofrece la posibilidad de programar, cancelar y modificar una cita médica dentro de su centro. También proporciona acceso rápido sobre los profesionales médicos que trabajan en HM hospitales incluyendo perfiles profesionales, áreas de especificación y opiniones de otros pacientes. Ayudando a los usuarios a tomar decisiones informadas sobre su

atención médica encontrando al profesional más adecuado. Los pacientes podrán tener acceso a todos los datos el historial médico y resultados de pruebas realizados en los centros HM.

3.4 Servicios salud Comunidad de Madrid

La página web de Servicios salud de la Comunidad de Madrid [18] ofrece una amplia gama de servicios dentro de la salud en la región.

En primer lugar, podemos encontrar información sobre hospitales y centros de salud, su ubicación, horarios y servicios que ofrecen. También ofrece información sobre recursos de salud pública y servicios de urgencia, ofreciendo consejos para situaciones de emergencia médica. En otros apartados podemos encontrar la solicitud de recetas electrónicas y gestión de medicamentos, así como información sobre su uso adecuado.

En la página también podemos encontrar datos acerca de diferentes tipos de atención médica disponibles, desde atención primaria hasta tratamientos especializados en hospitales, detalles sobre programas de salud mental, atención a personas mayores, cuidado infantil y noticias más actualizaciones de salud. Al igual que las anteriores páginas comentadas, esta también ofrece la posibilidad de reservar consultas médicas generales online.

3.5 Hospital San Rafael

La página web Hospital San Rafael [19] es similar al resto de las ya mencionadas. En este caso, para Pedir Cita online es necesario estar registrado, para ello tienen habilitado un formulario de registro en la misma página de inicio. Una vez registrado, basta con seguir las instrucciones indicadas, contestando primero una serie de preguntas respecto a los distintos tipos de servicios médicos y sus profesionales a cargo; para así poder elegir el que mejor se adapte a las necesidades del usuario.

Cabe destacar que están disponibles además de la página web, las aplicaciones móviles tanto para Android como para iOS, permitiendo la accesibilidad al portal del paciente desde varios dispositivos.

El resto de las secciones que se pueden encontrar en la página son informativas, ofreciendo multitud de datos con respecto a la historia del hospital., sus servicios y profesionales disponibles; así como la localización y contacto del centro para generar tranquilidad y fiabilidad al usuario en todo momento.

A modo de curiosidad, existe una ventana en la que cuentan con consejos de salud, así como de prevención de riesgos para poder mejorar la calidad de vida de los usuarios, con recomendaciones de profesionales.

Capítulo 4 - Arquitectura y modelo de datos

En este capítulo se explica en detalle todo lo relacionado con el modelo de datos utilizado, un diagrama más visual entidad-relación como se puede ver en la Figura 7 que permita entender la estructuración de los datos, así como la posterior base de datos, las tablas que la conforman y su conjunto de atributos.

4.1 Arquitectura de la aplicación

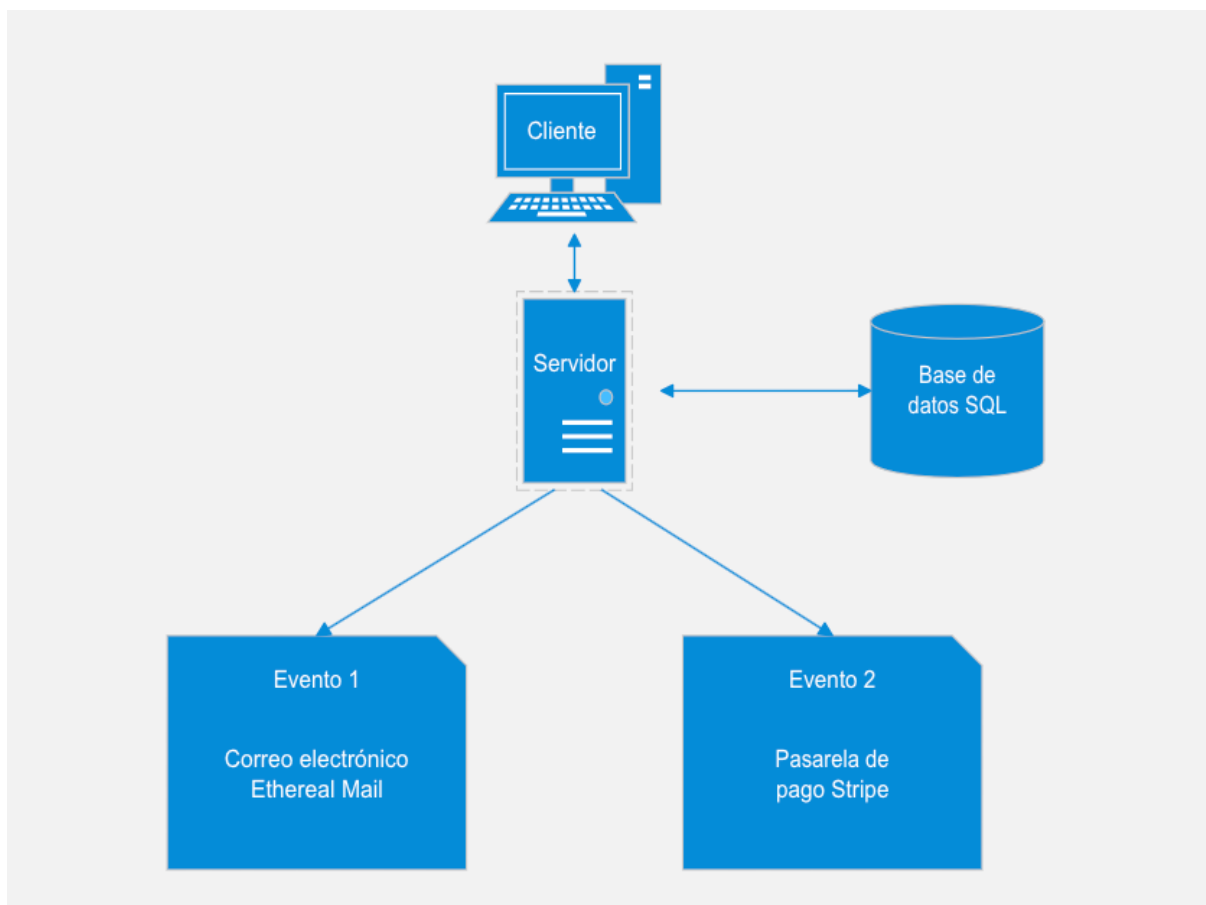


Figura 6. Diagrama de arquitectura de la aplicación

La arquitectura utilizada para esta aplicación web viene dada por una combinación de varias arquitecturas explicadas en la Figura 6, estando principalmente basada en una arquitectura cliente-servidor.

- **Cliente-Servidor:** Esta arquitectura se ha utilizado para la interacción entre el cliente (navegador web del usuario) y el servidor. El cliente envía numerosas solicitudes al servidor para realizar acciones como el registro en la página, reservar citas, procesar pagos, enviar mensajes entre usuarios... Y el servidor responde con un *login* correcto, la subida de documentos o mensajes de confirmación de que las acciones se han realizado correctamente.
- **Arquitectura basada en eventos:** esta arquitectura ha servido para manejar eventos como la reserva de una cita y el pago exitoso, o el envío automático de correos electrónicos una vez realizada, modificada o cancelada una cita. Cada acción desencadena un evento que será procesado por los microservicios correspondientes.
- **Arquitectura de microservicios:** por último, se ha hecho uso de una arquitectura de microservicios para gestionar diferentes aspectos de la aplicación de manera independiente. Por ejemplo, existe un servicio para gestionar la reserva de citas con la base de datos, otro servicio integrado con la plataforma de pago Stripe, y otro servicio para enviar correos electrónicos gracias a la herramienta Ethereal Mail.

4.2 Estructura de la base de datos

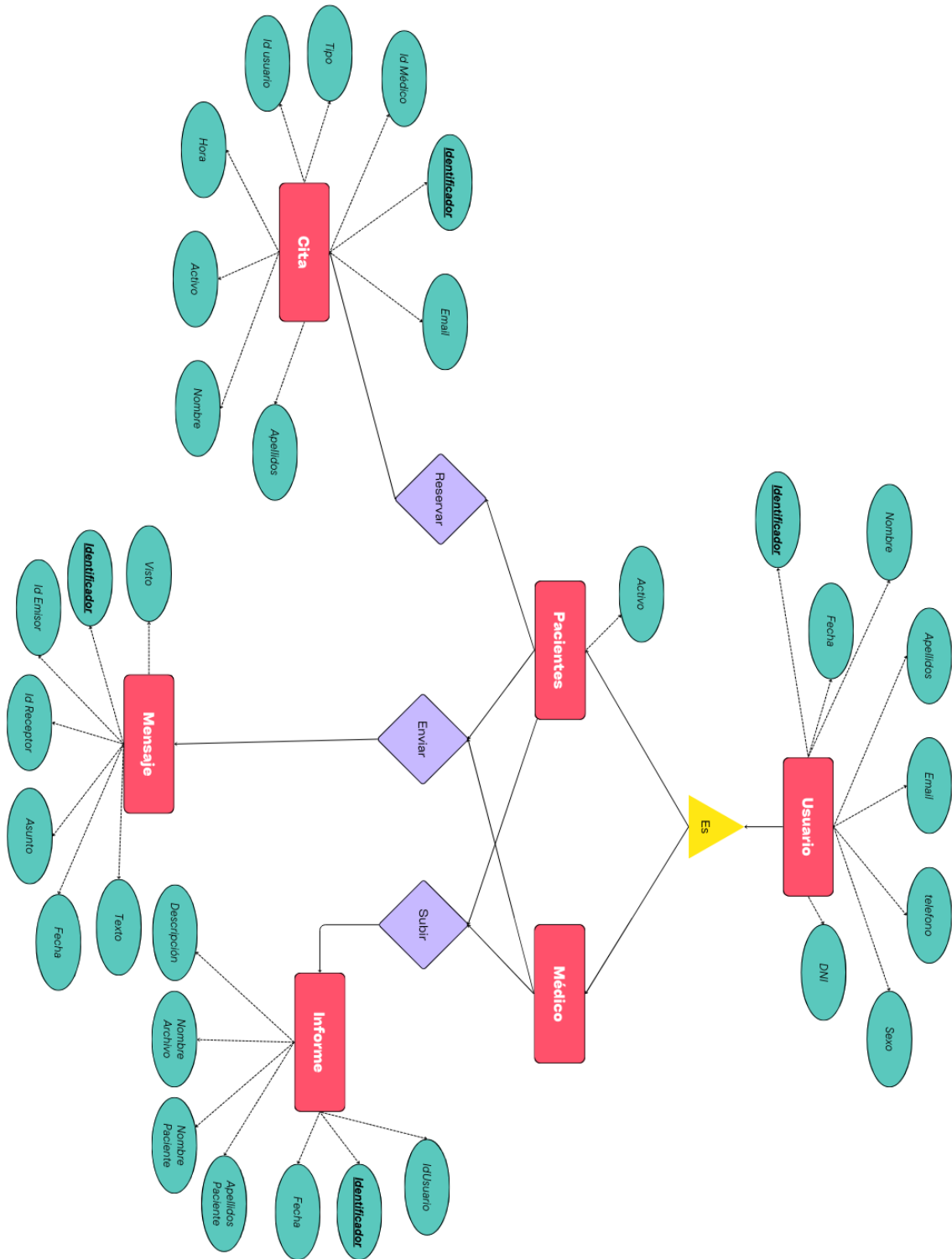


Figura 7. Diagrama entidad-relación

Para la estructuración de la base de datos se ha tenido en cuenta el diagrama entidad-relación de la Figura 7, donde se ven claramente las diferentes entidades junto con los atributos que componen cada una, así como las relaciones que formaban entre ellas para poder dar paso a la correcta funcionalidad de la aplicación.

El modelo de datos que compone las entidades principales de la aplicación se muestra en la Figura 8:

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño
<input type="checkbox"/> citas	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	15	InnoDB	utf8mb4_general_ci	32.0 KB
<input type="checkbox"/> informes	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	9	InnoDB	utf8mb4_general_ci	2.5 MB
<input type="checkbox"/> mensajes	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	20	InnoDB	utf8mb4_general_ci	16.0 KB
<input type="checkbox"/> sessions	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8mb4_general_ci	16.0 KB
<input type="checkbox"/> tokens	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8mb4_general_ci	32.0 KB
<input type="checkbox"/> usuario	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	10	InnoDB	utf8mb4_general_ci	272.0 KB

Figura 8. Base de datos

La herramienta utilizada para albergar la base de datos ha sido phpMyAdmin [20] en un modelo con una base de datos relacional estudiada durante el grado, que permite de manera intuitiva gestionar y mantener una base de datos local con las siguientes tablas:

- Tabla usuario
- Tabla citas
- Tabla informes
- Tabla mensajes
- Tabla sessions
- Tabla tokens

4.2.1 Tabla usuarios

En esta tabla se pueden encontrar los diferentes atributos que componen el perfil, tanto de pacientes como de médicos. En ella se almacena un id único siendo una *Primary Key*¹ auto incremental; el estado activo del paciente, con valor 1 si el usuario esta activo y valor 0 si ha sido dado de baja por el médico; el nombre, apellidos, email y contraseña; el tipo de perfil, “Médico” o “Paciente”; la fecha de nacimiento junto al sexo, que podrá elegir entre: “Mujer”, “Hombre” o “Prefiero no decirlo”; el D.N.I. y, por último, el teléfono e imagen de usuario, siendo estas dos últimas de carácter obligatorio, pero con posible valor nulo dentro de la base de datos. Tanto el correo como el D.N.I. son atributos únicos, es decir, en la base de datos no se podrán encontrar dos o más usuarios que compartan uno de estos campos. La estructura de la tabla descrita anteriormente se puede ver a continuación en la Figura 9:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/>	1	id 🔑			int(11)	No	Ninguna	AUTO_INCREMENT
<input type="checkbox"/>	2	activo			tinyint(1)	No	Ninguna	
<input type="checkbox"/>	3	nombre		utf8mb4_general_ci	varchar(20)	No	Ninguna	
<input type="checkbox"/>	4	apellidos		utf8mb4_general_ci	varchar(30)	No	Ninguna	
<input type="checkbox"/>	5	email		utf8mb4_general_ci	varchar(30)	No	Ninguna	
<input type="checkbox"/>	6	password		utf8mb4_general_ci	varchar(16)	No	Ninguna	
<input type="checkbox"/>	7	perfil		utf8mb4_general_ci	varchar(11)	No	Paciente	
<input type="checkbox"/>	8	fecha			date	No	Ninguna	
<input type="checkbox"/>	9	sexo		utf8mb4_general_ci	varchar(11)	No	Ninguna	
<input type="checkbox"/>	10	dni		utf8mb4_general_ci	varchar(9)	No	Ninguna	
<input type="checkbox"/>	11	telefono			int(9)	Sí	NULL	
<input type="checkbox"/>	12	img			blob	Sí	NULL	

Figura 9. Tabla usuarios

¹ Primary Key: es la clave primaria de una entidad, suele ser un identificador único como el DNI de una persona, o un número de identificación que no va a tener nadie más.

4.2.2 Tabla informes

En esta tabla se pueden encontrar todos los atributos que componen la subida al sistema de un informe. En ella se almacena un id que será la *Primary Key*, el id del paciente correspondiente al informe, haya sido subido por él o por su médico, además de su nombre y apellidos; la fecha de subida del informe, el título y una breve descripción de. Por último, el propio archivo en formato PDF. La estructura de la tabla descrita anteriormente se puede ver en la Figura 10:

	#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/>	1	id 	int(11)			No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/>	2	idUser	int(11)			No	Ninguna		
<input type="checkbox"/>	3	nombre	varchar(16)	utf8mb4_general_ci		No	Ninguna		
<input type="checkbox"/>	4	apellidos	varchar(30)	utf8mb4_general_ci		No	Ninguna		
<input type="checkbox"/>	5	fecha	date			No	Ninguna		
<input type="checkbox"/>	6	descripcion	varchar(200)	utf8mb4_general_ci		No	Ninguna		
<input type="checkbox"/>	7	archivo	longblob			No	Ninguna		
<input type="checkbox"/>	8	nombreArchivo	varchar(100)	utf8mb4_general_ci		No	Ninguna		

Figura 10. Tabla informes

4.2.3 Tabla citas

Esta tabla es utilizada para gestionar toda la información relacionada con las reservas de citas médicas. En ella aparece de nuevo un id único, los identificadores tanto del usuario que ha realizado la cita como el del médico seleccionado; el nombre y los apellidos del paciente junto con su email, el tipo de cita que ha solicitado recibir y la fecha y hora del evento. Además, esta tabla tiene un atributo "activo", siendo a 0 si la cita ya ha vencido, o 1 si aún está pendiente de realizar. En este caso, no hay atributo que tenga un valor único como en el caso del D.N.I., ya que no se puede dar el caso en el que dos personas hayan reservado una cita el mismo día con el mismo médico, pero sí con ambos. La estructura de la tabla descrita anteriormente se observa en la Figura 11

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/>	1 id 🔑	int(11)			No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/>	2 idUser	int(11)			No	Ninguna		
<input type="checkbox"/>	3 nombre	varchar(20)	utf8mb4_general_ci		No	Ninguna		
<input type="checkbox"/>	4 apellidos	varchar(30)	utf8mb4_general_ci		No	Ninguna		
<input type="checkbox"/>	5 email	varchar(30)	utf8mb4_general_ci		No	Ninguna		
<input type="checkbox"/>	6 tipo	varchar(20)	utf8mb4_general_ci		No	Ninguna		
<input type="checkbox"/>	7 idMedico	int(11)			No	Ninguna		
<input type="checkbox"/>	8 fecha	date			No	Ninguna		
<input type="checkbox"/>	9 hora	time			No	Ninguna		
<input type="checkbox"/>	10 activo	tinyint(4)			No	Ninguna		

Figura 11. Tabla citas

4.2.4 Tabla mensajes

Esta tabla contiene toda la información relacionada con los mensajes. Simula una bandeja de entrada de correo electrónico, pero alojada en la propia aplicación para no tener que utilizar otras herramientas. Tanto médico como paciente pueden enviar y recibir mensajes, así que en la base de datos se guardarán el id del emisor y el id del receptor, pudiendo ser de los dos tipos: médico y paciente. Además, existe un campo “Asunto” el cual contiene un título breve a modo descriptivo y el texto del mensaje, tal y como aparece en la base de datos almacenando el contenido completo de lo que se consideraría el cuerpo del mensaje. Esta tabla también contiene el atributo visto, que puede tomar dos valores: 0 si ha abierto el mensaje y 1 si no lo ha sido. La estructura de la tabla se puede ver en la Figura 12.

	#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/>	1	id	int(11)			No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/>	2	idEmisor	int(11)			No	Ninguna		
<input type="checkbox"/>	3	idReceptor	int(11)			No	Ninguna		
<input type="checkbox"/>	4	fecha	date			No	current_timestamp()		
<input type="checkbox"/>	5	asunto	varchar(50)	utf8mb4_general_ci		No	Ninguna		
<input type="checkbox"/>	6	texto	text	utf8mb4_general_ci		No	Ninguna		
<input type="checkbox"/>	7	visto	tinyint(1)			No	Ninguna		

Figura 12. Tabla mensajes

4.2.5 Tabla sessions

La tabla *sessions*, en la Figura 13, almacena las sesiones activas que tiene el usuario. A través de middlewares, se consigue que un paciente no pueda acceder a las ventanas del médico y viceversa, de igual manera que un usuario no registrado pueda interactuar con la página escribiendo la ruta a mano en el navegador. Contiene un *session_id* único, una fecha de expiración e información adicional necesaria para poder cargar la información del usuario actual, como se observa en la Figura 14.

	#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<input type="checkbox"/>	1	session_id	varchar(128)	utf8mb4_bin		No	Ninguna
<input type="checkbox"/>	2	expires	int(11)		UNSIGNED	No	Ninguna
<input type="checkbox"/>	3	data	mediumtext	utf8mb4_bin		Sí	NULL

Figura 13. Tabla sessions

	#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<input type="checkbox"/>	1	session_id	varchar(128)	utf8mb4_bin		No	Ninguna
<input type="checkbox"/>	2	expires	int(11)		UNSIGNED	No	Ninguna
<input type="checkbox"/>	3	data	mediumtext	utf8mb4_bin		Sí	NULL

Figura 14. Ejemplo sesión activa

4.2.6 Tabla tokens

Esta tabla ha sido creada exclusivamente para almacenar los tokens de restablecimiento de contraseña en la página de inicio. Así, los tokens se generan automáticamente una vez el usuario pide el restablecimiento de contraseña, siendo una cadena de caracteres aleatoria a través del módulo *crypto*, y caducando en 30 minutos desde la fecha de creación. La estructura de la tabla contiene el id del token para diferenciarlos en caso de haber varios, el id del usuario que ha solicitado la recuperación de contraseña, el propio token y la fecha de expiración como se aprecia en la Figura 15.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/>	1 idToken 	int(11)			No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/>	2 idUser	int(11)			No	Ninguna		
<input type="checkbox"/>	3 expirationDate	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()
<input type="checkbox"/>	4 token	varchar(255)	utf8mb4_general_ci		No	Ninguna		

Figura 15. Tabla tokens

Capítulo 5 - Diseño

En esta sección se va a tratar todo lo referente al diseño de la aplicación. Se parte de la idea de una interfaz sencilla y fácil de interpretar por el usuario para que su atención se centre en la funcionalidad, no tanto en el diseño; pero a su vez creando una página atractiva visualmente y con la idea de transmitir credibilidad y fiabilidad.

Se ha decidido escoger una paleta de colores en torno a verdes y azules puesto que son muy representativos del ámbito de la medicina. Además, el cotutor Javier dio el visto bueno ya que entraba dentro de sus preferencias.

Gracias al uso de plantillas dentro de las hojas de estilo CSS se permiten multitud de detalles que se dividen según la sección en la que se encuentra el usuario, cargando la plantilla deseada mediante un *import* como se puede ver a continuación en la Figura 16:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset='utf-8'>
5   <link rel="stylesheet" href="/css/login.css">
6   <link rel="icon" href="/img/iconos/salud.png" type="image/png">
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
8   <title>Login</title>
9 </head>
```

Figura 16. Cabecera de una plantilla EJS

En cada plantilla EJS de la aplicación, cada elemento de la página estará definido por un identificador único al que luego se podrá llamar desde la hoja de estilos importada y darle los detalles que se deseen como, por ejemplo: el tamaño de letra, los márgenes, una imagen de fondo, la anchura y altura del objeto... Esto se ve en la Figura 17.

```
.nav-link{
  color: white !important;
}

.bg-custom-card{
  background-color: rgb(57, 122, 122) !important;
}

.banner {
  color: #78d5cb !important;
  text-decoration: none !important;
}

.botonIniciarSesion {
  text-align: center;
  text-decoration: none !important;
  color: white !important;
}

.form-image {
  display: inline-block;
  border-radius: 20px;
  position: relative;
  cursor: pointer;
}

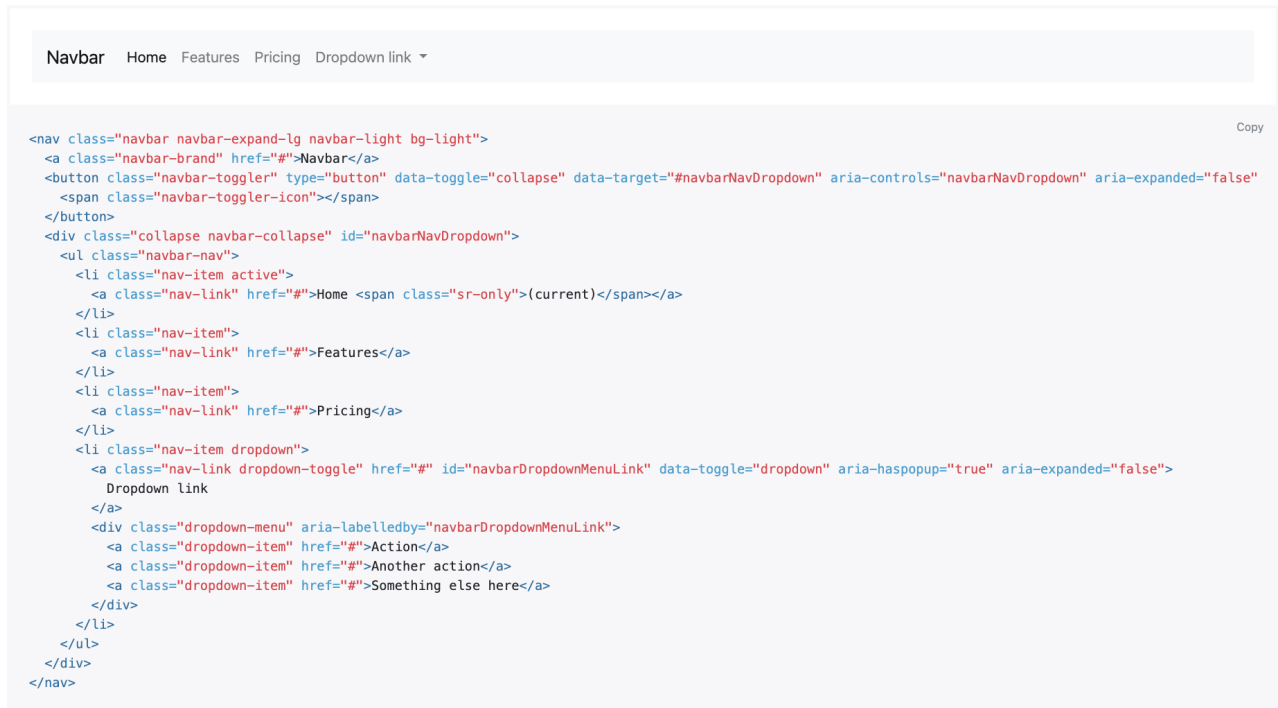
/* Estilo para ocultar el input de tipo file y posicionar la imagen sobre él */
.form-image input[type="file"] {
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  opacity: 0;
  cursor: pointer;
}
```

Figura 17. Hojas de estilos CSS

El uso de Bootstrap ha permitido poder organizar la página mediante bloques con etiquetas como títulos, encabezados, tarjetas y el uso de ventanas modales ha sido crucial en este proyecto.

En su página web se puede encontrar numerosa información acerca del objeto que se está implementando, con algunas palabras clave reservadas por la herramienta que hacen la aplicación más accesible al usuario, como se ve en la Figura 18. Además, ofrece diversas opciones de implementación según qué características quieras añadir a tu página, pudiendo acceder a todos los datos importando simplemente la biblioteca Bootstrap al inicio de la plantilla EJS.

You may also utilize dropdowns in your navbar. Dropdown menus require a wrapping element for positioning, so be sure to use separate and nested elements for `.nav-item` and `.nav-link` as shown below.

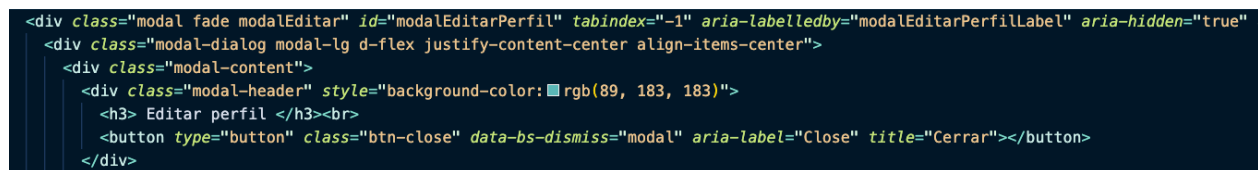


The image shows a screenshot of a Bootstrap navbar with a dropdown menu. The navbar has links for Home, Features, Pricing, and a dropdown link. Below the screenshot is the HTML code for the navbar, which includes a toggle button for the dropdown menu and a list of navigation items, including a dropdown menu with three items: Action, Another action, and Something else here.

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown" aria-controls="navbarNavDropdown" aria-expanded="false"
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNavDropdown">
    <ul class="navbar-nav">
      <li class="nav-item active">
        <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Features</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Pricing</a>
      </li>
      <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
          Dropdown link
        </a>
        <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
          <a class="dropdown-item" href="#">Action</a>
          <a class="dropdown-item" href="#">Another action</a>
          <a class="dropdown-item" href="#">Something else here</a>
        </div>
      </li>
    </ul>
  </div>
</nav>
```

Figura 18. Ejemplo de navbar en Bootstrap

Uno de los elementos de Bootstrap más utilizados en el proyecto ha sido las ventanas modales, que gracias a las características que se pueden añadir en el código, ver la Figura 19, permite dar al usuario una experiencia personalizada. Por ejemplo, la etiqueta “fade” le da un efecto a la apertura del modal de manera gradual.



The image shows a snippet of HTML code for a Bootstrap modal window. The modal has a header with a title and a close button. The code includes classes for modal, modal-dialog, modal-content, modal-header, and modal-close, along with attributes for tabindex, aria-labelledby, and data-bs-dismiss.

```
<div class="modal fade modalEditar" id="modalEditarPerfil" tabindex="-1" aria-labelledby="modalEditarPerfilLabel" aria-hidden="true"
  <div class="modal-dialog modal-lg d-flex justify-content-center align-items-center">
    <div class="modal-content">
      <div class="modal-header" style="background-color: #555555">
        <h3>Editar perfil </h3><br>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close" title="Cerrar"></button>
      </div>
```

Figura 19. Ventana modal con características

Son útiles a la hora de la optimización del código ya que no es necesario crear una página específica para su uso, simplemente con importarlos a la hoja EJS donde se quiere utilizar ya se puede acceder su funcionalidad. Aparte se pueden reutilizar en varias ventanas sin necesidad de reescribir el código completo, como se ve en la Figura 20.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <link rel="icon" href="/img/iconos/salud.png" type="image/png">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
  <!-- OJO: hoja de estilos debajo del Boostraop, si no no hace efecto -->
  <link rel="stylesheet" href="/css/perfiles.css">
  <title>Inicio</title>
</head>
<body>
  <%= include('./partials/header'); %>
  <% if (info.perfil === "Paciente") { %>
    <%= include('./partials/navbarPaciente'); %>
  <% } else if (info.perfil === "Médico") { %>
    <%= include('./partials/navbarMedico'); %>
  <% } else { %>
    <!-- Incluimos la clase resto porque la llevan implícita los navbar y en caso de UNR, se perdería es
  <div class="resto"></div>
  <% } %>
  <%= include('./partials/footer'); %>
  <h1 class="d-flex justify-content-center align-items-center">Cuida de ti y deja que te cuidemos</h1>
  <div class="container-fluid text-center text-black mt-5">...
</div>
<br><br>
  <%= include('./partials/cajasInfo'); %>
<br><br>
  <%= include('./partials/carousel'); %>
<br><br>
  <%= include('./partials/mapa'); %>
  <%= include('./modals/mIniciarSesion', {errMsg: []}); %>
  <%= include('./modals/mReservarCita', {info: info, medicos: medicos}); %>
  <div class="resto2"></div>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <script src="https://js.stripe.com/v3/"></script>
  <script src="/javascript/scriptsReservarCita.js" type="text/javascript"></script>
  <script src="javascript/scriptsPago.js" type="module"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

Figura 20. Página inicio con modales y objetos incluidos

Capítulo 6 - Funcionalidad

En este capítulo, se explica en detalle la funcionalidad de la aplicación desde el punto de vista del código y con sus equivalencias en la propia página una vez arrancado el proyecto.

Para ello, se ha organizado todo el desarrollo a través de los siguientes módulos:

- Módulo de autenticación
- Módulo gestión de pacientes
- Módulo gestión de citas
- Módulo gestión de informes
- Módulo de consultas médicas

Observar que, el archivo que contendrá toda la información relacionada con las rutas será el llamado "app.js", donde habrá funciones de tipo GET y POST para recoger y enviar datos al servidor, respectivamente. Es aquí donde aparecen los llamados *middlewares*, que son funciones intermedias que revisan las conexiones a modo de control, por ejemplo, para asegurarse que un paciente no puede acceder a las secciones del médico y viceversa; o que un usuario no registrado puede acceder al portal del paciente introduciendo la ruta manualmente (`localhost:4000/perfilPaciente`).

6.1 Módulo autenticación

Se han implementado las páginas de registro e inicio de sesión, mediante formularios creados gracias a Bootstrap para después configurarlo con el archivo app.js. Existen campos de varios tipos: *email* (comprueba si el campo tiene @), *password* (muestra el texto con caracteres ocultos), *text* (permite números y letras), *select* (permite elegir entre varias opciones) para permitir tener un formulario con datos deseados, como se ve en las Figura 21 y Figura 22, así como la gestión de estos en la posterior ruta POST que los recogerá y guardará en la base de datos.

```

<div class="card text-white mt-3" style="border-radius: 1rem;">
  <div class="card-body text-center">
    <div class="mt-3"></div>
    <h1>Registro</h1>
    <form action="/registro" method="POST" enctype="multipart/form-data">
      <div class="row mt-5 mb-5">
        <div class="col-6 form-outline">
          <label for="nombre" class="form-label">Nombre</label>
          <input type="text" id="nombre" name="nombre" class="form-control" placeholder="Juan" required/>
          <% if(validaciones.nombre){%>
            <span class="rojo" id="nombre"><%= validaciones.nombre.msg %></span>
          <% } %>
        </div>
        <div class="col-6 form-outline">
          <label for="apellidos" class="form-label">Apellidos</label>
          <input type="text" id="apellidos" name="apellidos" class="form-control" placeholder="García García" required/>
          <% if(validaciones.apellidos){%>
            <span class="rojo" id="apellidos"><%= validaciones.apellidos.msg %></span>
          <% } %>
        </div>
      </div>
      <div class="row mb-5">
        <div class="col-6 form-outline">
          <label class="form-label" for="email">Correo electrónico</label>
          <input type="email" id="email" name="email" class="form-control" placeholder="usuario@gmail.com" required/>
          <% if(validaciones.email){%>
            <span class="rojo" id="email1">
              <%= validaciones.email.msg %>
            </span>
          <% } %>
        </div>
      </div>
    </form>
  </div>
</div>

```

Figura 21. Formulario de registro I

```

<div class="col-6 form-outline">
  <label class="form-label" for="confPassword">Confirmar contraseña</label>
  <input type="password" name="confPassword" id="confPassword" class="form-control" placeholder="*****">
  <% if(validaciones.confPassword){%>
    <span class="rojo" id="">
      <%= validaciones.confPassword.msg %>
    </span>
  <% } %>
</div>
</div>
<div class="row mb-5">
  <div class="col-6 form-outline">
    <label class="form-label" for="image">Imagen de perfil</label>
    <input accept="image/png,image/jpeg,image/jpg" type="file" name="image" id="image" class="form-control" value="">
  </div>
  <div class="col-3 form-outline">
    <label for="dni" class="form-label">DNI</label>
    <input type="text" id="dni" name="dni" class="form-control" placeholder="00000000X" required/>
    <% if(validaciones.dni){%>
      <span class="rojo" id="dni"><%= validaciones.dni.msg %></span>
    <% } %>
  </div>
  <div class="col-3 form-outline">
    <label for="telefono" class="form-label">Teléfono</label>
    <input type="text" id="telefono" name="telefono" class="form-control" placeholder="600300100"/>
    <% if(validaciones.telefono){%>
      <span class="rojo" id="telefono"><%= validaciones.telefono.msg %></span>
    <% } %>
  </div>
</div>

```

Figura 22. Formulario de registro II

La vista final del registro se aprecia en la Figura 23.

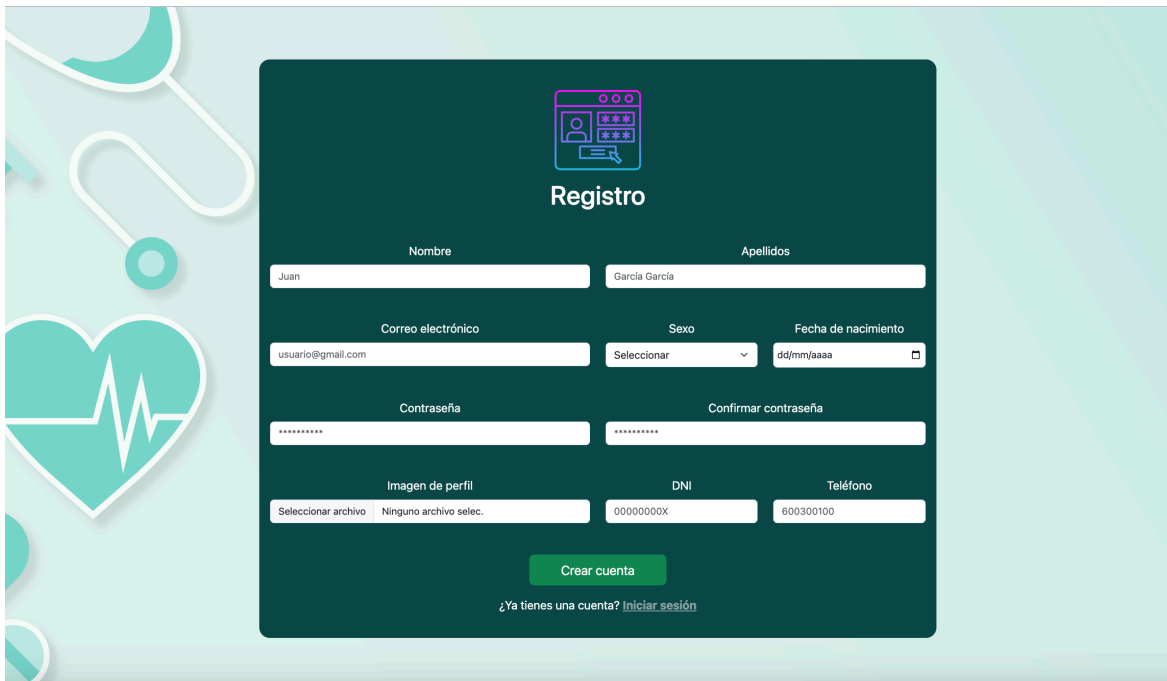


Figura 23. Vista registro

El formulario de *login*, ver Figura 24, comprobará que los campos email y contraseña introducidos coinciden en la base de datos con algún usuario registrado, que ha quedado visualmente como en la Figura 25

```
<div class="row justify-content-center">
  <div class="form-outline form-login mb-4 col-6">
    <label class="form-label" for="email">Correo electrónico</label>
    <input type="text" id="email" name="email" class="form-control form-control-lg" placeholder="paciente@gmail.com" />
  </div>
</div>
<div class="row justify-content-center">
  <div class="form-outline form-login mb-3 col-6">
    <label class="form-label" for="password">Contraseña</label>
    <input type="password" id="password" name="password" class="form-control form-control-lg" placeholder="*****" />
    <p class="mt-3" style="font-style: italic; text-align: center;"><a href="/recuperar-contrasena" class="text-white-50">Recuperar contraseña</a></p>
  </div>
</div>
<div class="d-grid mt-2">
  <button type="submit" class="btn btn-success btn-lg mx-auto mb-5">Iniciar sesión</button>
</div>
```

Figura 24. Formulario login

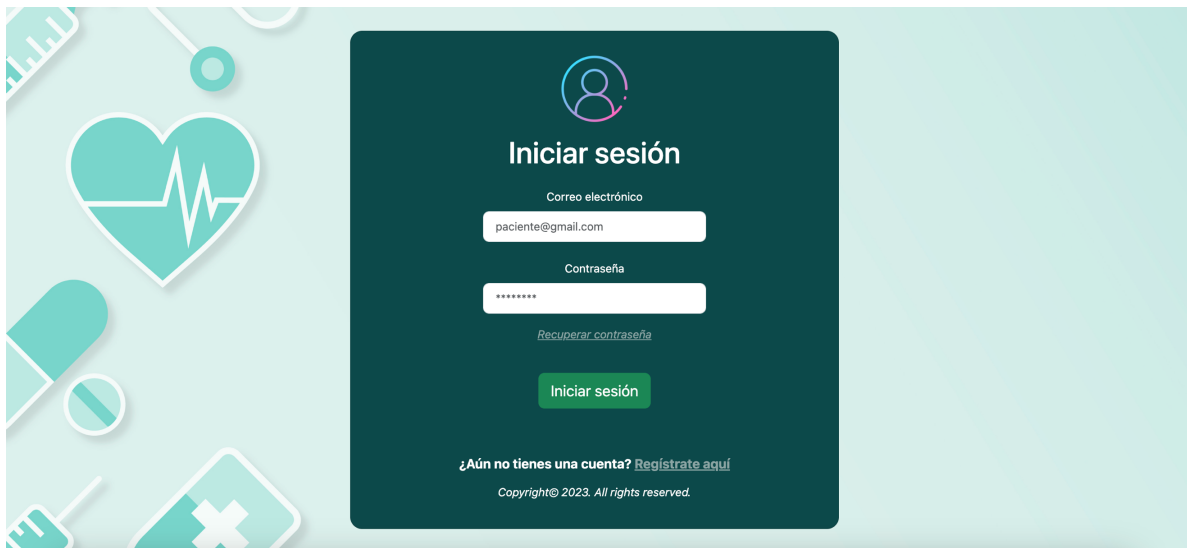


Figura 25. Vista login

Con un formulario muy sencillo, ver Figura 26, que permite al usuario introducir su correo electrónico y en caso de estar registrado enviarle un correo con un enlace de recuperación, se ha conseguido la página "Recuperación de contraseña", en la Figura 27.

```

<div class="container d-flex justify-content-center align-items-center vh-100">
  <div class="card bg-custom-card text-white mt-3 mb-5" style="border-radius:1rem;">
    <div class="card-body text-center">
      
      <form action="/recuperar-contrasena" method="POST">
        <div class="form-outline">
          <label for="email">Por favor, introduce tu correo electrónico para recuperar tu contraseña:</label>
          <input type="email" id="email" name="email" required>
        </div>
        <% if (error) { %>
          <p class="rojo"><%= error %></p>
          <% } %>
        <% if (success) { %>
          <p class="verde"><%= success %></p>
          <% } %>
        <button type="submit" class="btn btn-success">Recuperar contraseña</button>
      </form>
      <br><br>
    </div>
  </div>
</div>
</div>

```

Figura 26. Implementación recuperar contraseña

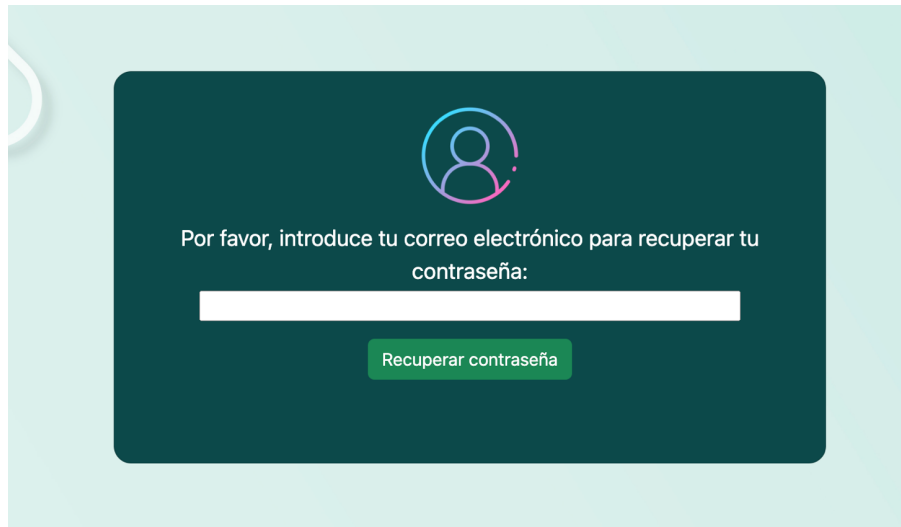


Figura 27. Vista recuperar contraseña

Se genera un token aleatorio como se observa en la Figura 28, que será utilizado para el enlace enviado al usuario indicado en la Figura 29, pasado como parámetro. Este token tiene una caducidad de 30 minutos.

```
// Generar token de restablecimiento de contraseña
function generateResetToken(email, callback) {

  daoUser.getUserID(email, (err, idUser) => {
    if(err) console.log(err);
    crypto.randomBytes(20, (err, buf) => {
      if (err) return callback(err);

      const token = buf.toString('hex');
      console.log("Token generado: " + token);

      // Obtener la fecha de expiración del token (por ejemplo, media hora desde ahora)
      const expirationDate = moment().add(30, 'minutes').toDate();
      console.log("Fecha de expiración " + expirationDate);

      // Guardar el token en la base de datos asociado al ID del usuario
      daoUser.saveResetToken(idUser, token, expirationDate, (err) => {
        if (err) return callback(err);
        callback(null, token);
      });
    });
  });
}
```

Figura 28. Función generar token de restablecimiento

```

// Enviar correo electrónico con el enlace de restablecimiento de contraseña
function sendResetEmail(email, token, callback) {
  const resetLink = `localhost:4000/cambiar-contrasena/${token}`;
  const mailOptions = {
    from: 'general.ward@ethereal.email',
    to: email,
    subject: 'Restablecimiento de contraseña',
    text: `Para restablecer tu contraseña, haz clic en el siguiente enlace: ${resetLink}`,
  };

  transporter.sendMail(mailOptions, callback);
}

```

Figura 29. Función enviar correo de restablecimiento

Para recuperar su contraseña, el usuario hace click en el enlace proporcionado, ver Figura 30 y se le redirige a la página para introducir la nueva; y en caso de éxito, volver a la página *login*.

Subject: Restablecimiento de contraseña
From: <general.ward@ethereal.email>
To: <jimenasantos@gmail.com>
Time: Today at 19:53
Message-ID: <87710ccc-1253-d02f-0583-ba080743abf6@ethereal.email>

HTML
 Plaintext

Para restablecer tu contraseña, haz clic en el siguiente enlace: <http://localhost:4000/cambiar-contrasena/2d634ee8b3639fcbd9d9f15630e08fd9988c811a>

Figura 30. Correo electrónico con enlace para recuperar contraseña

6.2 Módulo gestión de pacientes

En este módulo se tratará todo lo referente al manejo de datos del usuario. En el menú principal de la aplicación web, los usuarios, tanto médicos como pacientes podrán acceder a sus datos modificándolos si fuera conveniente mediante el uso de ventanas modales, implementado con un formulario que recoge solo los datos de los campos rellenos y los actualiza en la base de datos, implementado como se observa

en la Figura 31. De igual manera, podrán eliminar su perfil dándose de baja de la base de datos.

```
<div class="col-4">
  <div class="row form-outline mt-2">
    <div class="col-4"><label for="nombre" class="form-label">Nombre</label></div>
    <!-- No hace falta pasarle el objeto info por parámetro (al id del modal) porque lo coge el middleware "acceso" -->
    <div class="col">
      <span class="rojo" id="nombre-error"></span>
      <input type="text" id="nombre" name="nombre" class="form-control" placeholder="<%=info.nombre%>"/>
    </div>
  </div>
</div>
<div class="row form-outline mt-2">
  <div class="col-4"><label for="apellidos" class="form-label">Apellidos</label></div>
  <div class="col">
    <span class="rojo" id="apellidos-error"></span>
    <input type="text" id="apellidos" name="apellidos" class="form-control" placeholder="<%=info.apellidos%>"/>
  </div>
</div>
</div>
<div class="col-3 text-center">
  <label for="image" class="form-image" style="border-radius: 20px;">
    
    <input accept="image/png,image/jpeg,image/jpg" type="file" name="image" id="image">
  </label>
</div>
</div>
<div class="row form-outline mt-2">
  <div class="col-3"><label class="form-label" for="email">Correo electrónico</label></div>
  <div class="col">
    <span class="rojo" id="email-error"></span>
    <input type="email" id="email" name="email" class="form-control" placeholder="<%=info.email%>"/>
  </div>
</div>
<div class="row form-outline mt-2">
  <div class="col-3"><label class="form-label" for="password">Contraseña</label></div>
  <div class="col">
    <span class="rojo" id="password-error"></span>
    <input type="password" name="password" id="password" class="form-control" placeholder="*****">
  </div>
</div>
<div class="row form-outline mt-2">
  <div class="col-3"><label class="form-label" for="confPassword">Confirmar contraseña</label></div>
  <div class="col">
    <span class="rojo" id="confPassword-error"></span>
    <input type="password" name="confPassword" id="confPassword" class="form-control" placeholder="*****">
  </div>
</div>
<span class="rojo" id="image-error"></span>
```

Figura 31. Ventana modal editar perfil

Por parte del médico, se requiere una serie de funciones básicas para la gestión de los pacientes, pudiéndolos dar de alta o de baja, buscarlos y ver su ficha técnica. Un formato de tabla con varios botones permitirá al médico realizar estas acciones como se ha implementado en la Figura 32 dando lugar a una vista como la de la Figura 33.

```

<table class="table table-striped">
  <thead>
    <tr><!--Encabezado-->
      <!--Añado la clase cursor para en las css hacer que me aparezca clicable y el usuario lo pueda ver bien-->
      <th scope="col" class="cursor"></th>
      <th scope="col" class="cursor" id="nombre-ord">Nombre</th>
      <th scope="col" class="cursor" id="apellido-ord">Apellidos</th>
      <th scope="col" class="cursor" id="fecha-ord">Fecha de nacimiento</th>
      <th scope="col" class="cursor" id="sexo-ord">Género</th>
      <th scope="col"></th>
    </tr>
  </thead>
  <tbody id="cuerpoTablaPacientes" style="vertical-align: middle;">
    <!--Informacion de la tabla-->
    <% usuarios.forEach(u => {
      <tr class="fila">
        <td><a type="button" class="btn" href="/verPaciente/<%=u.id%>"><%=u.nombre%></td>
        <td class="busqueda"><%=u.apellidos%></td>
        <td class="busqueda"><%= new Date(u.fecha).toLocaleDateString() %></td>
        <td class="busqueda"><%=u.sexo%></td>
        <td>
          <a type="button" class="btn" title="Agregar" onclick="subirInforme('<%= u.id %>', '<%= u.nombre %>')">
            
          </a>
          <a type="button" class="btn" onclick="confirmarArchivado('<%= u.id %>')">
  <div class="card-header" style="background-color: #5bc0de;">
    <div class="row">
      <div class="col-8">
        <div class="row">
          <a type="button" class="btn" href="/listaPacientes"><%=paciente.nombre%> <%=paciente.apellidos%></h3>
          <h4 class="card-subtitle text-muted">Ficha del paciente</h4>
        </div>
        <div class="row">
          <ul class="datosPaciente">
            <li><strong>Cumpleaños: </strong><span id="fechaCumple"></span></li>
            <li><strong>Género: </strong><%=paciente.sex0%></li>
            <li><strong>DNI: </strong><%=paciente.dni%></li>
            <li><strong>Teléfono: </strong><%=paciente.telefono%><span id="telefono"></span></li>
          </ul>
        </div>
      </div>
      <div class="col-4">
  </div>
</div>

```

Figura 34. Implementación ver paciente

←
Carlos Pérez Pérez
 Ficha del paciente

- **Cumpleaños:**
- **Género:** Hombre
- **DNI:** 98172371V
- **Teléfono:** 788192815

Buscar

Fecha	Descripción	Archivo	Opciones
17/4/2024	Prueba Carlos	Informe Carloss	
17/4/2024	Informe 2	Urgencias Carloss	

Figura 35. Ficha técnica paciente

La barra de búsqueda, implementada con Ajax en la Figura 36, permite filtrar los resultados automáticamente sin necesidad de escribir la búsqueda entera y hacer click sobre el botón de lupa "Buscar", ver Figura 38. Esto se consigue gracias a la función `keyup()`, ver Figura 37.

```

<form class="d-flex align-items-center">
  <input class="form-control me-2" type="buscar" placeholder="Buscar" aria-label="Buscar" id="buscar">
  <button class="btn btn-dark" type="submit"></button>
</form>

```

Figura 36. Barra de búsqueda

```

$(document).ready(function(){
  $("#buscar").keyup(function(){
    var buscando = $(this).val().toLowerCase();

    $("table tbody tr").each(function(){
      var encontrado = false;
      $(this).find("td").each(function(){
        if ($(this).text().toLowerCase().indexOf(buscando) !== -1) {
          encontrado = true;
          return false;
        }
      });
      if (encontrado) {
        $(this).show();
      } else {
        $(this).hide();
      }
    });
  });
});

```

Figura 37. Función automática buscar

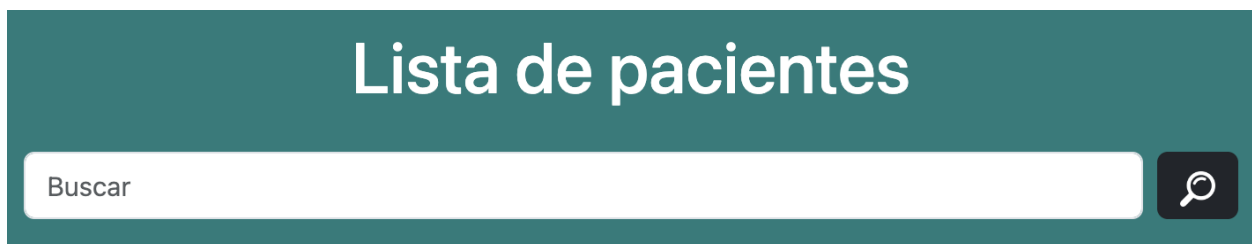


Figura 38. Vista de barra de búsqueda

6.3 Módulo gestión de citas

En este bloque se gestionará todo lo que engloba a las citas médicas. Es un apartado bastante complejo, ya que organiza los horarios de varios médicos a la vez.

Es decir, dos médicos podrán tener una cita a la misma hora con dos pacientes distintos. El modal "reservar cita" implementado en la Figura 39 está formado por una serie de campos a elegir por el usuario que una vez seleccionado el tipo de médico y el día filtran las horas disponibles.

```

<!-- cogemos el médico-->
<label for="nombre" class="form-label">
  <h6><strong>Selecciona un médico</strong></h6>
</label>
<select class="form-select" id="id_medico" name="id_medico" onchange="datosCita()" required>
  <option value="" disabled selected hidden>Selecione un médico...</option>
  <% medicos.forEach(m => { %>
    <option value="<%=m.id%>"><%=m.nombre%> <%=m.apellidos%></option>
  <% }>;%>
</select>
<!-- cogemos el tipo de reserva-->
<label for="nombre" class="form-label" id="tipo0cult" hidden>
  <h6><strong>Tipo de cita </strong></h6>
</label>
<select class="form-select" id="tipoReserva" name="tipoReserva" required hidden>
  <option value="" disabled selected hidden>Selecione el tipo de consulta...</option>
  <option value="Presencial">Presencial</option>
  <option value="Telefónica">Telefónica</option>
  <option value="Videoconferencia">Videoconferencia</option>
</select>
<!-- cogemos la fecha de la reserva-->
<label for="nombre" class="form-label" id="fecha0cult" hidden>
  <h6><strong>Fecha </strong></h6>
</label>
<input type="date" id="fecha" name="fecha" class="form-control"
  onchange="seleccionarHora(event, document.getElementById('id_medico').value);" required hidden>
<div class="valid-feedback">Fecha correcta</div>
<div class="mt-3" id="horasDispo"></div>

<!-- Habilitamos checkbox para poder pagar en consulta en caso de ser presencial -->
<div class="form-check mt-3" id="opcionesPago" style="display: none;">
  <input class="form-check-input" type="checkbox" value="pagarConsulta" id="pagarConsultaCheckbox">
  <label class="form-check-label" for="pagarConsultaCheckbox">Pagar en la consulta</label>
</div>
</div>

<div class="modal-footer" style="background-color: #556b2f;">
  <button type="submit" class="btn btn-success" id="reservarDirecto"
  style="display: none;" title="Reservar">
  Reservar
</button>
  <button type="submit" class="btn btn-primary" id="continuarAlPago"
  style="display: none;" title="Confirmar">
  Continuar al pago
</button>

```

Figura 39. Modal reservar cita

En caso de ser consulta presencial y con la opción "Pagar en consulta" activada se reserva directamente la cita. En caso contrario, el botón "Continuar al pago" redirigirá al paciente a la plataforma de pago, como se puede observar a continuación en la Figura 40.

The image displays two side-by-side screenshots of a 'Reservar nueva cita' (Book new appointment) modal form. Both forms share the same layout and fields: a teal header with a close button, a 'Selección de médico' dropdown (Belén Fernández Puertas), a 'Tipo de cita' dropdown (Presencial), a 'Fecha' date picker (22/05/2024), and a 'Hora' dropdown (10:00). The primary difference is the 'Pagar en la consulta' option at the bottom, which is unchecked in the left screenshot and checked in the right one. This state change is reflected in the bottom button, which is 'Continuar al pago' (blue) on the left and 'Reservar' (green) on the right.

Figura 40. Modal reservar cita con opción "Pagar en consulta"

Modificar cita permite al médico exclusivamente cambiar la fecha u hora en la que esta se iba a llevar a cabo en caso de imprevistos y se enviará un correo electrónico con los cambios al paciente.

Finalizar cita dejará que una vez haya concluido la consulta el médico pueda hacer click pasándola a una sección llamada "Citas archivadas".

Cancelar cita puede ser utilizada como paciente o como médico, dejando de nuevo el hueco libre para otro paciente y volviendo a aparecer la hora en el modal "Reservar cita".

6.4 Módulo gestión de informes

Este módulo permitirá tener un control de todo el historial médico con respecto a informes de tipo PDF que servirán de ayuda al médico de cara a las consultas.

La funcionalidad es igual para médico y para paciente mediante una ventana modal con una breve descripción del archivo a modo de título y la subida del archivo en sí.

Para el paciente será una opción de la barra de navegación llamada "Mis informes", con un listado de todos sus informes en formato de tabla. Para el médico será necesario acceder a la ficha del paciente, y una vez dentro aparecerá dicho listado con los informes subidos tanto por él como por el paciente. Se podrán visualizar, ver Figura 41 y descargar haciendo click sobre él gracias al visor de documentos del navegador existente, ver Figura 42.

```
app.get("/verInforme/:id", acceso, (req, res) => {
  const id = req.params.id;

  daoInformes.getUserInforme(id, (err, informe) => {
    if(err) console.log(err);
    else {
      if(informe) res.end(informe);
    }
  })
})
```

Figura 41. Solicitud ver informe al servidor mediante end()



Fecha	Descripción	Archivo
17/4/2024	Prueba Carlos	 Informe Carlos
17/4/2024	Informe 2	 Urgencias Carlos

Figura 42. Vista mis informes desde paciente

El médico tendrá tres opciones extra para la gestión de estos informes, pudiendo eliminarlos, modificarlos subiendo una nueva versión o descargarlos directamente mediante un click. De nuevo, está operativa la barra de búsqueda.

6.5 Módulo consultas médicas

El módulo final que constituirá la aplicación web y la hará funcional para ambos usuarios registrados, será el denominado "Consultas médicas".

Para esto, se hará uso de herramientas conocidas como Google Meet [21] o Zoom [22], que será labor del médico elegir y enviar un enlace al paciente. La implementación de un sistema de mensajería interno facilita tener todo el control médico en una única página web, sin el requerimiento de numerosas herramientas externas.

El sistema de mensajes está implementado, ver Figura 43, en una sección de la barra de navegación, con varias opciones como se muestra en la Figura 44:

- Bandeja de entrada: con un listado de todos los mensajes recibidos por parte del otro tipo de usuario, en el caso de pacientes únicamente por sus médicos y viceversa.
- Mensajes enviados: otro listado con los mensajes enviados por el usuario a modo de historial.
- Redactar mensaje: un último listado con los posibles destinatarios junto a un botón "Enviar" que abrirá una ventana modal con el asunto y cuerpo del mensaje a redactar.

```
div class="row">
  <div class="col">
    <form class="formCuerpo" id="verCorreos" method=" POST">
      <button type="submit" class="btn btn-primary mb-3 mt-3 ms-3" title="Ver" id="botonVerCorreo"
        onclick="cambioColorMensaje('botonVerCorreo')">Bandeja de entrada</button>
    </form>
  </div>
  <div class="col">
    <form class="formCuerpo" id="correosEnviados" method=" POST">
      <button type="submit" class="btn btn-primary mb-3 mt-3 ms-3" title="Enviados" id="botonEnviados"
        onclick="cambioColorMensaje('botonEnviados')">Enviados</button>
    </form>
  </div>
  <div class="col">
    <form class="formCuerpo" id="listaMensaje" method=" POST">
      <button type="submit" class="btn btn-primary mb-3 mt-3 ms-3" title="Enviar" id="botonEnviar"
        onclick="cambioColorMensaje('botonEnviar')">Redactar mensaje</button>
    </form>
  </div>
</div>
```

Figura 43. Implementación bandeja de entrada



Figura 44. Vista mensajes recibidos

Además, se ha desarrollado la posibilidad de descargar los listados completos en formato PDF para poder ser imprimibles, gracias a la librería JSPDF de Ajax, como se muestra a continuación en la Figura 45.

```

window.jspdf = window.jspdf.jspdf;
function imprimirContenido() {

    // Obtener la tabla de citas HTML
    var maintable = document.getElementById('tablaListaCitas');
    var pdfout = document.getElementById('pdfout');

    pdfout.onclick = function () {
        // generamos el PDF, p = portrait, en vertical
        // pt = points, conteo de puntos; y a4 = tamaño del papel
        var doc = new jsPDF('p', 'pt', 'a4');
        var margin = 20;
        var scale = (doc.internal.pageSize.width - margin * 2) / document.body.clientWidth;

        if(window.innerWidth > 768) {
            doc.html(maintable, {
                x:margin,
                y:margin,
                html2canvas:{
                    scale: scale,
                },
                callback: function(doc) {
                    doc.save('Próximas citas.pdf');
                }
            });
        }
    }
}

```

Figura 45. Script que genera el PDF a partir de la tabla

El archivo PDF se verá como se muestra a continuación en la Figura 46.

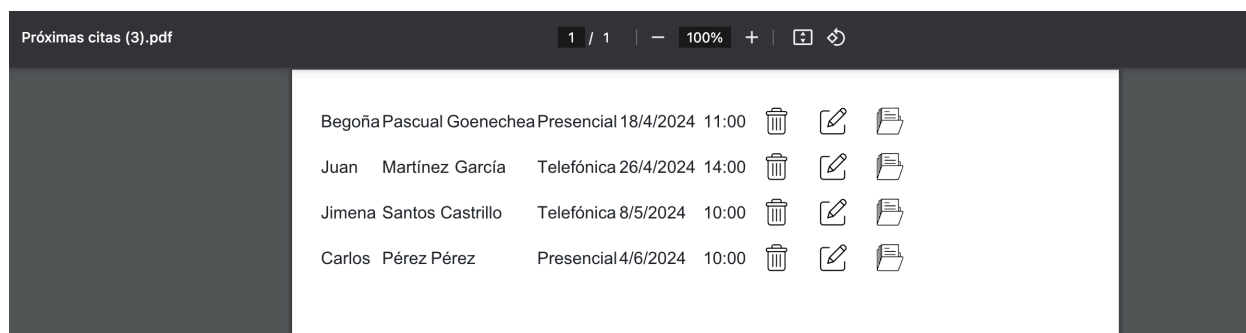


Figura 46. PDF generado para imprimir

Capítulo 7 - Pasarela de pago

En este capítulo, se va a tratar de manera más extensa la pasarela de pago implementada mediante una API de Stripe [12]. Se explicará en detalle cómo se ha conseguido la correcta unión del código ya existente en la aplicación, junto con los elementos proporcionados por la plataforma Stripe.

7.1 Análisis de las herramientas

En primer lugar, se realizó un análisis de tipos de plataformas de pago con el fin de ver cuál era la más adecuada para la aplicación. Se pensó en utilizar, entre otras, Paypal [23], RedSys [24], Amazon Pay [25], Square [26]... todas ofrecían lo mismo a nivel de servicios: poder efectuar un pago seguro de manera online.

Sin embargo, debido a su interfaz y la documentación en su página web, Stripe fue la que más se adecuaba a la forma de pago que se pretendía implementar; así como la facilidad con el usuario.

Otro de los motivos que hicieron a Stripe como la principal candidata, fue la libertad que ofrecía al no ser necesario tener una cuenta externa como ocurría con Amazon Pay o Paypal, sino simplemente utilizar los datos de una tarjeta bancaria cualquiera.

7.2 Investigación de la pasarela Stripe

Gracias a la inmensa cantidad de información existente hoy en día en internet, se realizó una tarea de investigación mediante vídeos explicativos y la lectura de la documentación que ofrece Stripe.

Al principio resultó ser una tarea sencilla, puesto que el objetivo de añadir la pasarela de pago era poder “vender productos” al cliente, en este caso, las consultas médicas. Es verdad que el enfoque de otras aplicaciones estaba centrado en una tienda online donde a través de un catálogo de productos el usuario podría hacer click en uno y se le añadiría al carrito.

La idea era la misma, pero habría que fusionarlo e integrarlo con la aplicación web de manera que se recogieran datos de otro tipo, como el médico, la fecha y hora de la reserva, y por último, el producto a vender (tipo de consulta médica).

7.3 Creación de la API

Una vez investigado y analizado las diferentes maneras de fusionar la pasarela con la aplicación se creó una nueva tienda en la plataforma Stripe, eligiendo el tipo de empresa y los productos a vender, como se observa en la Figura 47.

The screenshot displays the Stripe 'Catálogo de productos' interface. On the left, a sidebar contains navigation links: Inicio, Pagos, Saldos, Clientes, Billing, and Más. The main content area features a search bar and a title 'Catálogo de productos'. Below the title, there are tabs for 'Todos los productos', 'Cupones', 'Tasas de envío', 'Tipos de impuesto', 'Tablas de tarifas', and 'Medidores'. A filter section shows 'Todos' (3) and 'Activo' (3). Below this, there are buttons for 'Creación', 'Estado: Activo', 'Borrar los filtros', and 'Exportar precios'. The main table lists three products:

Nombre	Tarifas	Creado	Actualizado
Presencial	140,00 €	16 abr.	21 abr.
Telefónica	100,00 €	13 abr.	22 abr.
Videoconferencia	100,00 €	13 abr.	22 abr.

3 resultados

Figura 47. Tienda online en Stripe

7.4 Implementación

Se creó una pestaña adicional en la *navbar*² ya existente a modo de catálogo que mostrara simplemente los 3 productos a vender: consulta presencial, telefónica o videoconferencia.

² Menú de navegación que permite al usuario moverse entre las distintas secciones de la página. Suele ser una barra horizontal donde cada sección, está asociada a un link que te

Para acceder a los datos de los productos sería necesario el archivo "keys.js", ver en la Figura 48, donde se almacenarían la clave pública y clave secreta de la API creada en el apartado anterior.

```
// Importante guardar en documentos aparte las claves secretas para mejor seguridad
// copiamos las claves de nuestra API
export default {
  public: "pk_test_
  secret: "sk_test_
}
```

Figura 48. Claves secreta y pública de Stripe

Mediante la fusión del código implementado ya en la ventana modal "mReservarCita.ejs" y los IDs de los productos y precios alojados en la plataforma, como se observa en la Figura 49, ya se pueden utilizar estos datos para llevar al usuario al pago del producto seleccionado mediante el `select` del modal.

Productos: `Array(3)`

- `0`: {id: 'prod_', object: 'product', active: true, attributes: Array(0), created: 1713311400, ...}
- `1`: {id: 'prod_', object: 'product', active: true, attributes: Array(0), created: 1713006922, ...}
- `2`: {id: 'prod_', object: 'product', active: true, attributes: Array(0), created: 1713006841, ...}

length: 3
[[Prototype]]: Array(0)

Precios: `Array(3)`

- `0`: {id: 'price_', object: 'price', active: true, billing_scheme: 'per_unit', created: 1713311400, ...}
- `1`: {id: 'price_', object: 'price', active: true, billing_scheme: 'per_unit', created: 1713006922, ...}
- `2`: {id: 'price_', object: 'price', active: true, billing_scheme: 'per_unit', created: 1713006841, ...}

length: 3
[[Prototype]]: Array(0)

option 0	<code><option value="Presencial" data-price="price_</code>	<code>F52">Presencial</option></code>	<code>slot</code>
option 1	<code><option value="Telefónica" data-price="price_</code>	<code>hY2">Telefónica</option></code>	<code>slot</code>
option 2	<code><option value="Videoconferencia" data-price="price_</code>	<code>b2m">Videoconferencia</option></code>	<code>slot</code>

Figura 49. Obtención de productos y precios desde Stripe

Como se puede observar en la Figura 50, los precios de los productos están asociados a las opciones del `select` del formulario reservar cita, añadiéndoles un atributo llamado "data-price" que será el que luego se utilizará para redirigir al pago:

redirige a dicha pestaña, pudiendo moverte entre ellas sin necesidad de escribir la ruta completa en el navegador.

```

let products, prices;

// Realiza las peticiones a la API de Stripe para obtener productos y precios
Promise.all([
  fetch("https://api.stripe.com/v1/products", options),
  fetch("https://api.stripe.com/v1/prices", options)
])
.then(responses => Promise.all(responses.map(res => res.json())))
.then(json => {
  products = json[0].data;
  prices = json[1].data;

  console.log("Productos: ", products);
  console.log("Precios: ", prices);

  prices.forEach((price, index) => {
    // Busca el nombre del producto correspondiente al precio
    const productName = products.find(product => product.id === price.product).name;

    // Le ponemos index+1 porque la option = 0 es el título "Seleccione un tipo de consulta..."
    // y se estaba asignando el precio del primer producto (Presencial) a esa option
    const option = $formReserva.querySelector("#tipoReserva").options[index+1];

    // Establece el atributo data-price a cada consulta, con el valor del ID del precio
    option.setAttribute("data-price", price.id);

    console.log("option ", index, " ", option);
  });
});

```

Figura 50. Asignación del atributo data-price a cada producto

La función de Stripe `redirectToCheckout()` enviará los datos recogidos en el formulario a la pasarela de pago, ver en la Figura 51 y Figura 52.

```

// se crea el objeto de configuración para la redirección de checkout de Stripe
const redirectToCheckoutConfig = {
  lineItems: [{ price: idPrecio, quantity: 1 }],
  mode: "payment",
  successUrl: "http://localhost:4000/success",
  cancelUrl: "http://localhost:4000/cancel"
};

// Redirige al checkout de Stripe
Stripe(KEYS.public).redirectToCheckout(redirectToCheckoutConfig)
.then(res => {
  if (res.error) {
    console.error(res.error.message);
  }
});

```

Figura 51. Creación objeto con producto seleccionado

← TFG consultas médicas TEST MODE

Videoconferencia
100,00 €
Consulta por videoconferencia de unos 20 minutos de duración

Correo electrónico

Información de la tarjeta

1234 1234 1234 1234

MM / AA CVC

Nombre del titular de la tarjeta

Nombre completo

País o región

España

Guardar mis datos de forma segura para un proceso de compra en un clic
Paga más rápido en TFG consultas médicas y en todos los comercios que acepten Link.

Pagar

Figura 52. Pasarela de pago con consulta de tipo videoconferencia

Una vez en la página del pago, Stripe proporciona en su documentación [numerosas tarjetas de crédito de prueba](#) para realizarlos. En el caso de pago exitoso, esto es, que se hayan introducido todos los datos correctamente, se redirigirá a la página pasada en el objeto `redirectToCheckoutConfig successUrl`; en caso contrario será enviado a `cancelUrl`.

Un punto a tener en cuenta a la hora de controlar el orden de las acciones es mantener los datos de la reserva almacenados de manera local, como se observa en la Figura 53, hasta que se haya realizado el pago para evitar incongruencias con la base de datos en caso de que no se pudiera completar y así evitar que la reserva quedase guardada en el calendario del médico.

```

$d.addEventListener("submit", e => {
  if (e.target.matches("#formReserva")) {
    if(e.submitter.id === "continuarAlPago") {
      e.preventDefault();

      // Guarda los datos del formulario en un objeto
      const formData = {
        id_usuario: $d.getElementById("id_usuario").value,
        id_medico: $d.getElementById("id_medico").value,
        nombre_usuario: $d.getElementById("nombre_usuario").value,
        apellido_usuario: $d.getElementById("apellido_usuario").value,
        email_usuario: $d.getElementById("email_usuario").value,
        tipoReserva: $d.getElementById("tipoReserva").value,
        fecha: $d.getElementById("fecha").value,
        hora: $d.getElementById("hora").value,
      };

      // almacenamos los datos del formulario en el almacenamiento local
      localStorage.setItem("formData", JSON.stringify(formData));
    }
  }
});

```

Figura 53. Almacenar datos del formulario de manera local en objeto formData

En la página éxito "success.ejs" se recuperarán esos datos y se enviarán a la base de datos mediante el método POST, como se observa en la Figura 54.

```

<script>
  document.addEventListener("DOMContentLoaded", function() {
    // Recupera los datos del formulario que habíamos guardado localmente
    const formData = JSON.parse(localStorage.getItem("formData"));

    if (formData) {
      // Enviamos los datos del formulario al servidor mediante AJAX
      $.ajax({
        url: "/reservarCita",
        type: 'POST',
        data: formData,
        dataType: "JSON",
        success: (data) => {
          console.log(data);
        },
      });
    }
  });
</script>

```

Figura 54. Acceso al objeto formData para enviarlo al servidor

Para comprobar que se ha realizado el pago y la reserva se accede a la sección "Pagos" en Stripe y se ve si se han añadido, como se muestra a continuación en la Figura 55:

Pagos

Todos los pagos Disputas Todas las transacciones

Todos
61

Exitosos
38

Rebolsados
0

No capturados
0

Error
0

⊕ Fecha y hora
⊕ Importe
⊕ Divisa
⊕ Estatus
⊕ Método de pago
⊕ Más filtros

📄 Exportar

<input type="checkbox"/>	Importe		Método de pago	Descripción	Cliente	Fecha
<input type="checkbox"/>	100,00 € EUR	Exitoso ✓	•••• 3222	1x Videoconferencia	isabelmarquez@gmail.com	13 may. 10:42
<input type="checkbox"/>	140,00 € EUR	Cancelado ✕	—	1x Presencial		5 may. 11:40
<input type="checkbox"/>	100,00 € EUR	Exitoso ✓	•••• 5100	1x Telefónica	jimenasantos@gmail.com	2 may. 15:01
<input type="checkbox"/>	100,00 € EUR	Exitoso ✓	•••• 5556	1x Telefónica	jimenasantos@gmail.com	30 abr. 17:18
<input type="checkbox"/>	100,00 € EUR	Exitoso ✓	•••• 4444	1x Videoconferencia	begopascual@gmail.com	30 abr. 9:05
<input type="checkbox"/>	100,00 € EUR	Cancelado ✕	—	1x Telefónica		30 abr. 8:59
<input type="checkbox"/>	100,00 € EUR	Cancelado ✕	—	1x Telefónica		26 abr. 14:17
<input type="checkbox"/>	140,00 € EUR	Exitoso ✓	•••• 4444	1x Presencial	begopascual@gmail.com	25 abr. 10:23

Figura 55. Ventana "Pagos" en Stripe

Si en algún pago al final se ha decidido volver para atrás en la aplicación y no completarlo aparecerá como "Cancelado". Debido a las numerosas pruebas realizadas con respecto al tema de gestión de los datos, muchos pagos se quedaron en este estado.

Capítulo 8 - Conclusiones y trabajo futuro

8.1 Conclusiones

En este trabajo se ha desarrollado una página web con el objetivo de poder llevar a cabo diversas acciones en un mismo entorno para que, de forma intuitiva, poder facilitar la interacción de médicos y pacientes con su historial clínico.

De esta manera esta aplicación permite al usuario registrarse y crear un perfil y poder reservar citas con el médico deseado y el tipo de cita que se ajuste a sus necesidades, eligiendo entre: telefónica, presencial o videoconferencia. Añadiendo a lo anterior, los usuarios podrán subir y guardar informes de tipo PDF con información que ayudará en las sesiones reservadas.

El sistema de mensajería incorporado ofrece tanto a paciente como médico una manera rápida y eficaz de estar en contacto sin necesidad de herramientas externas.

Por último, el usuario permanecerá constantemente actualizado de los diversos cambios que se realicen en las acciones del sistema gracias a las notificaciones generadas automáticamente, en forma de correos electrónicos a la cuenta registrada.

8.2 Trabajo futuro

Una vez finalizado este proyecto y visto el alcance conseguido, se han propuesto algunas funcionalidades adicionales que podrían explotar el potencial de la aplicación:

- Un formulario con preguntas más específicas para reducir el tiempo de consulta. El usuario recibirá, previo a la cita, un formulario con una serie de preguntas guiadas para poder ofrecer mayor detalle al médico de cara a la consulta. Entre las preguntas se incluirán: los síntomas, alergias, patologías previas, medicamentos que este consumiendo actualmente... consiguiendo así agilizar la labor del médico a la hora de dar un diagnóstico final.
- Enviar una encuesta de satisfacción al cliente tras un cierto periodo de tiempo habiendo permitido al usuario interactuar con la aplicación. Con esto se desea

conseguir un *feedback*³ por parte del paciente que permita a los desarrolladores mejorar la aplicación en diferentes ámbitos para futuros clientes.

- Integración de IA en diferentes ámbitos. Por un lado, se pretende implementar una sección en la que el usuario, a través de conceptos más técnicos proporcionados por la aplicación, mediante un *ChatBot*⁴ que consiga un análisis o evaluación de lo podría padecer.

Por otro lado, el médico tendrá una sección en la que la IA pueda localizar o reconocer pacientes en el sistema con síntomas o tratamientos en común para el ámbito de la investigación. A su vez ayuda a identificar la enfermedad que puede estar padeciendo el paciente.

- Adaptación de la página a diferentes tamaños de pantalla. Se pretende implementar el código para poder ser utilizado en diversas plataformas como móvil o *tablet*, haciendo la aplicación más accesible en el día a día.

El acceso al código completo de la aplicación puede encontrarse en un repositorio en GitHub mediante este [enlace](#).

³ Retroalimentación por parte de un individuo a través de una opinión personal con un fin de mejora.

⁴ Conversaciones simuladas con una especie de robot que genera respuestas automáticas dependiendo de la interacción con él.

Chapter 8 - Conclusions and future work

Conclusions

In this assignment, a web page has been developed to be able to carry out various actions in the same environment so that, in an intuitive way, the interaction of doctors and patients with their clinical history is provided.

In this way, this application allows the user to register and create a profile and to book appointments with the desired doctor and the type of appointment that suits their needs, choosing between telephone, face-to-face or videoconference. In addition to the above, users will be able to upload and save PDF reports with information that will help in the booked sessions.

The built-in messaging system offers both patients and doctors a quick and efficient way to stay in touch without the need for external tools.

Finally, the user will remain constantly updated on the various changes in the system's actions thanks to automatically generated notifications in the form of e-mails to the registered account.

Future Works

Once this project was completed and had a clear view of the achievements, some additional functionalities were proposed that could exploit the potential of the application:

- A form with more specific questions to reduce appointments time. The user will receive, prior to the appointment, a form with a series of guided questions in order to provide more detail to the doctor for the consultation. The questions will include symptoms, allergies, previous pathologies, medications currently being taken, etc., speeding the doctor's work when it comes to making a final diagnosis.

- Send a customer satisfaction survey after a certain period, allowing the user to interact with the application. The aim is to obtain feedback⁵ from the patient that will allow the developers to improve the application in different areas for future customers.
- Integration of AI in different areas. On one hand, it is intended to implement a section in which the user, through more technical concepts provided by the application, by means of a ChatBot⁶ gets an analysis or evaluation of what he/she might suffer.

On the other hand, the doctor will have a section in which the AI can locate or recognize patients in the system with symptoms or treatments in common with the field of research. helping to identify the disease the patient may be suffering.

- Adaptation of the page to different screen sizes. It is intended to implement the code to be used on various platforms such as mobile or tablet, making the application more accessible on a daily basis.

The code can be found on a GitHub repository at the following [link](#).

⁵ Feedback is information about reactions to a product or a process, which is used as a basis for improvement

⁶ A Chatbot is a computer program or artificial intelligence software designed to simulate conversation with human users, especially over the internet.

CONTRIBUCIONES PERSONALES

Begoña Pascual Goenechea

Las contribuciones en este presente proyecto realizadas por Begoña Pascual Goenechea, junto con su compañera Patricia Pastor Puente, han abarcado múltiples módulos diseñados para optimizar la experiencia del usuario y mejorar la eficiencia en la gestión de datos. A continuación, se detallarán las funciones desarrolladas en cada módulo.

Inicio del proyecto

En este módulo ambas compañeras trabajamos mano junto con su tutor para planificar las fases y estructura que compondrían el proyecto a lo largo de estos meses, como a su vez el desarrollo de los módulos y de los requisitos de la aplicación. Una vez se recibió el visto bueno del tutor pasaron a la siguiente fase.

Creación del proyecto y base de datos

Ambas trabajaron en equipo para realizar la creación en Visual Studio Code que albergará el desarrollo de la página y pautar la organización en paquetes que se seguirá. En cuanto a la base de datos, se comenzó una creación de esta en phpMyAdmin muy básica con las principales tablas y un formato a seguir de los atributos que se iban a ir añadiendo, ya que a medida que el proyecto se fuese desarrollando se tendrían que ir realizando modificaciones.

Módulo de autenticación

En este módulo Begoña realizó la implementación de lo que sería iniciar sesión / *login* para que los usuarios pudiesen acceder de manera segura al sistema. Tras finalizar con esta tarea, desarrolló la función de cerrar sesión para que los usuarios pudiesen finalizar sesión de manera controlada. Estos dos métodos garantizarán al cliente una segura protección de los datos.

Módulo de gestión de pacientes

En este bloque comenzó programando las acciones que el médico tendría dentro de la aplicación con respecto a los pacientes. Entre ellas se encuentran, dar de alta a pacientes para que una vez dentro del sistema el médico pudiese verlos en formato de lista y tener un control de aquellos que actualmente están activos.

Dar de baja al paciente también es otra de las funcionalidades en las que participó por los mismos motivos comentados anteriormente, que el médico pudiese tener un control de los usuarios que ya han finalizado su tratamiento.

Por último, en este módulo, realizó una funcionalidad que se repetirá en muchos otros módulos: buscar paciente, que consistirá en una barra de búsqueda en la que el usuario o el médico introducirá una serie de caracteres y automáticamente los cogerá y actualizará el listado con aquellos usuarios que cumplen los requisitos escritos en la barra.

Módulo de gestión de citas

En este apartado comenzó con el desarrollo para poder reservar una cita que, sin lugar a duda, se convirtió en uno de los más difíciles de implementar, ya que el desarrollo terminó siendo más complicado de lo que se esperaba debido a las numerosas restricciones que se tuvieron que tener en cuenta: franja horaria, días restringidos... y, a su vez, el desarrollo para que la aplicación pudiese alojar varios médicos y tipos de citas.

Finalmente lo consiguió utilizando AJAX, haciendo que el usuario solo pudiera introducir los campos una vez seleccionase el médico con el que se desea tener la cita, facilitando así que las horas que aparecieran a continuación, la elección del tipo de cita y el día fueran lo más precisos posible para su correcto funcionamiento.

Una vez terminada esta parte, continuó con el desarrollo de funciones como cancelar y finalizar cita, implementadas simplemente con botones que el usuario pulsará y que se encargarán de devolver una acción a la base de datos, la cual se encargará de actualizar esta información y dejarla reflejada.

También se encargó de la parte de modificar cita, la cual realizó a través de un modal en el que el médico seleccionará la nueva hora en la que se podría impartir la cita previamente seleccionada. Una vez enviada la información, se actualizará automáticamente en la página en la cual se encuentra el listado de citas y en la base de datos.

Para finalizar este módulo desarrolló la función de finalizar cita para que los médico, una vez terminadas las sesiones con un paciente, pudiesen archivarlas y poder tener un historial con estas.

Módulo de gestión de informes

En este bloque comenzó implementando el poder subir un informe tanto por parte del médico al paciente, como también para que el paciente pudiese subir informes obtenidos en otras instituciones y que quisiese tener en su historial para uso del médico. Esto se consiguió gracias a una ventana modal en la que el usuario introdujese un breve asunto, descripción del contenido del informe, y por último realizase la subida desde el escritorio de su ordenador del informe en formato PDF para tenerlo en el historial. Tras finalizar este desarrollo, se encargó de que el usuario y el médico pudiesen visualizar este informe y a su vez descargarlo.

Para terminar, realizó una serie de funciones básicas como son eliminar informe, listar informes y buscar informes a través de la barra de búsqueda que siempre se encontrará en la parte superior de la vista.

Módulo de consultas médicas

Finalmente, en el último módulo desarrolló el sistema de mensajería interno que se iba a utilizar en la aplicación. El cual consiste en una bandeja de entrada en la que aparecen los mensajes que ha recibido el usuario. En azul aparecerán aquellos que no han sido leídos y en blanco los que sí han sido. Se encontrarán también dos iconos, uno para marcar el mensaje como leído sin necesidad de entrar en él y otro para visualizar el mensaje, ya que de primeras solo aparecerán los primeros 20 caracteres de este.

Dentro de ver mensaje implementó la posibilidad de responder automáticamente a este dando al botón de responder y redactando el nuevo mensaje.

En la parte superior trabajo en tres botones. El primero que redirige a la bandeja de entrada mencionada anteriormente. El segundo que mostrará los mensajes enviados con los mismos botones implementados en la bandeja de entrada y el tercer y último botón que es para poder enviar mensajes. En el caso del médico, aparecerá un listado de todos los usuarios registrado y simplemente, realizando clic en el botón que se encuentra a la derecha, podrá enviarle un correo a través de un modal en el que tendrán que redactar un asunto, el mensaje y proceder a enviarlo. Y en el caso del paciente, aparecerá un listado de todos los médicos y el proceso de enviar el mensaje es el mismo mencionado anteriormente en la parte del médico.

Para finalizar, el último desarrollo que realizó fue la implementación de cuatro botones en la vista de citas para poder organizar el contenido de esta dependiendo del tipo de cita que fuese, presencial, telefónica, videoconferencia o simplemente el listado de todas las citas pendientes. Junto con el cifrado de contraseñas dentro de la bases de datos importando un nuevo paquete al proyecto para llevarlo a cabo y garantizar siempre la seguridad de la cuenta del usuario.

Patricia Pastor Punte

La aportación a este trabajo de fin de grado de Patricia pastor Punte, como menciona su compañera Begoña en el apartado anterior, se ha visto dividida mediante una serie de módulos incrementales con el fin de obtener una aplicación funcional desde lo más básico hasta el objetivo final: el poder realizar consultas médicas vía online.

Inicio de proyecto

En primer lugar, fue necesaria una reunión a principio de curso con el tutor Antonio, para especificar los requisitos de la aplicación y cómo se iban a organizar a través de módulos. Fue entonces cuando se creó la denominada Especificación de requisitos. Para poder tener un orden de todos los módulos, se establecieron unos hitos de entrega, estimando mediante un diagrama de Gantt los tiempos necesarios para cada módulo dependiendo de su dificultad.

Creación del proyecto y base de datos

Para la creación del proyecto decidimos utilizar Visual Studio Code, GitHub y XAMPP, que les permitió trabajar de manera asíncrona. Una vez vieron que todo funcionaba correctamente y la conexión era estable, empezaron a elegir tareas para cada una. Todo lo referente a este apartado y el anterior fue de manera conjunta, para una correcta organización y división de tareas a posteriori.

Investigación y herramientas externas

En general, la aportación de patricia fue en su mayoría de investigación e implementación de herramientas externas que permitieron a la aplicación dar un punto de credibilidad y mejora más allá de una simple página web.

Su compañera Begoña se encargó mayormente del tema de funcionalidad en el *back-end*⁷ y ella más en el *front-end*⁸, en la parte más visual, además de las herramientas que se explicarán a continuación.

En primer lugar, estuvo investigando acerca de Nodemailer y vio que era de fácil instalación y manejable con el código. La idea principal era utilizar el servidor de correo más famoso Gmail, para poder hacer la aplicación más real. Sin embargo, debido a las restricciones de seguridad que tiene la aplicación, no dejaba acceder a su API de manera local debido a la posible aparición de virus. Tuvo que buscar una alternativa “más casera” y fue entonces cuando encontró Ethereum Mail, otro servidor que funciona de manera local, simplemente registrándote en él y utilizando esas credenciales en tu código para permitir enviar y recibir correos de manera simulada a cualquier dirección. Sin duda la tarea que más tiempo le llevó fue la de la pasarela de pago. Tuvo que informarse mucho acerca de cómo integrar la API con el código existente y tras muchas depuraciones, consiguió guardar los datos de los productos en venta (las consultas médicas).

Módulo de autenticación

Respecto a este módulo, se encargó del formulario de registro y de la sección “Recuperación de contraseña”. Aquí tuvo la primera toma de contacto con la herramienta Ethereum Mail, mediante el cual se enviaría un correo electrónico con un enlace generado aleatoriamente y que caducaba al cabo de un tiempo; haciendo los datos del usuario más seguros pero complejos. Realizó numerosas pruebas para comprobar que todo funcionaba bien, que los tokens caducaban e invalidaban el enlace o que la contraseña se cambiaba correctamente.

⁷ Es la parte que funciona del lado del servidor y gestionan las acciones de los botones, formularios...

⁸ Es la parte de la aplicación con la que interactúan los usuarios directamente, incluyendo toda la parte visual.

Módulo de gestión de pacientes

En este módulo, se encargó de realizar la vista “Ver paciente”, que era la que más estética tendría en cuanto a mostrar la información de los datos. Para ello, creó una pestaña nueva a la que se accediera desde la tabla principal implementada previamente por Begoña, mediante un icono en forma de ojo.

Es aquí donde a modo de información el médico podría ver con una tarjeta implementada con Bootstrap con una cabecera con los datos básicos del paciente alojados en la base de datos como el nombre completo, DNI, fecha de nacimiento... junto con su imagen de perfil, para permitir al médico identificar al paciente de manera rápida. En el cuerpo de la tarjeta irían entonces sus informes médicos, pudiendo ser subidos por él mismo o por el paciente, función que implementaría Begoña en este caso.

El uso de ventanas modales ha sido crucial en el proyecto, ya que les ha aportado una página web más interactiva y dinámica. Se encargó, de las opciones modificar y eliminar perfil, alojados en la cabecera de la aplicación mediante un menú desplegable. Parecía una tarea simple, pero tuvo demasiadas complicaciones, sobre todo por el tema de las cookies que guardaban los datos del usuario que actualmente ha iniciado sesión. Esto fue porque, si editabas su correo electrónico, la página se quedaba recargando ya que el campo elegido para controlar las sesiones en la base de datos era el email. Entonces tuvo que cambiar todas las funciones que requerían de este ‘currentUser’ para que se controlara mediante el ID único que no se iba a cambiar de ninguna manera.

De igual manera, realizó el modal eliminar perfil, este más sencillo dando de baja al usuario de la base de datos una vez comprobada su contraseña a modo de confirmación.

Módulo de gestión de citas

En este módulo se encargó de la pasarela de pago y la gestión de notificaciones al usuario. Gracias a la interfaz sencilla que ofrece Ethereum Mail y al uso de plantillas que pasamos como parámetro mantendrán informados al usuario tanto de sus nuevas citas, como de si hay alguna modificación o cancelación de estas.

Módulo de gestión de informes

En este módulo cogió la tarea de “Ver informes” de un paciente ya que ella había implementado la ventana “Ver Paciente” en el módulo 2 y estaría directamente relacionado con eso. Mediante una tabla que recorriera todos los informes de cada paciente, se mostrarían junto con el documento PDF adjunto para poder visualizarlo y descargarlo a su dispositivo.

Funciones extra

Una vez tenían la aplicación más o menos funcionando tuvieron una reunión con el cliente, Javier, y les pidió alguna funcionalidad extra. Entonces se encargó de añadir la función “Descargar lista completa” que permitiera al médico tener un documento PDF imprimible.

Por último, programó la pestaña principal de inicio, ya que le llamó mucho la atención la parte visual así que insertó un mapa de Google Maps [27] y un carrusel con varias noticias de medicina a modo informativo y decorativo. Esta página la tendrían los 3 tipos de usuario, así que creó una general para todos con comprobaciones que mostraran u ocultaran secciones reutilizando así código en la medida de lo posible.

BIBLIOGRAFÍA

- [1] «Canva,» [En línea]. Available: <https://www.canva.com/>.
- [2] «Microsoft OneNote,» [En línea]. Available: <https://www.microsoft.com/es-es/microsoft-365/onenote/digital-note-taking-app>.
- [3] «Azure DevOps,» [En línea]. Available: <https://azure.microsoft.com/es-es/products/devops>.
- [4] «Visual Studio Code,» [En línea]. Available: <https://code.visualstudio.com/>.
- [5] «Github,» [En línea]. Available: <https://github.com/>.
- [6] «XAMPP,» [En línea]. Available: <https://www.apachefriends.org/es/index.html>.
- [7] «HTML,» [En línea]. Available: <https://es.wikipedia.org/wiki/HTML>.
- [8] «CSS,» [En línea]. Available: <https://es.wikipedia.org/wiki/CSS>.
- [9] «Bootstrap,» [En línea]. Available: <https://getbootstrap.com/>.
- [10] «Javascript,» [En línea]. Available: <https://es.wikipedia.org/wiki/JavaScript>.
- [11] «AJAX,» [En línea]. Available: <https://es.wikipedia.org/wiki/AJAX>.
- [12] «Stripe,» [En línea]. Available: <https://stripe.com/es>.
- [13] «Ethereal Mail,» [En línea]. Available: <https://ethereal.email/login>.
- [14] «Nodemailer,» [En línea]. Available: <https://www.nodemailer.com/>.
- [15] «Quironsalud,» [En línea]. Available: <https://www.quironsalud.com/>.
- [16] «Top Doctors,» [En línea]. Available: <https://www.topdoctors.es>.

- [17] «HM Hospitales,» [En línea]. Available: <https://www.hmhospitales.com/>.
- [18] «Servicios de salud Comunidad de Madrid,» [En línea]. Available: <https://www.comunidad.madrid/servicios/salud>.
- [19] «Hospital San Rafael,» [En línea]. Available: <https://www.hospitalsanrafael.es/>.
- [20] «phpMyAdmin,» [En línea]. Available: <https://www.phpmyadmin.net/>.
- [21] «Google Meet,» [En línea]. Available: <https://meet.google.com/>.
- [22] «Zoom,» [En línea]. Available: <https://zoom.us/es>.
- [23] «Paypal,» [En línea]. Available: <https://www.paypal.com/es/home>.
- [24] «RedSys,» [En línea]. Available: <https://redsys.es/>.
- [25] «Amazon Pay,» [En línea]. Available: <https://pay.amazon.es/>.
- [26] «Square,» [En línea]. Available: <https://squareup.com/es/es>.
- [27] «Google Maps,» [En línea]. Available: <https://www.google.es/maps>.

APÉNDICES

Apéndice A - Casos de uso

Actores

- Usuario no registrado: este usuario podrá acceder a la página de inicio meramente informativa con las noticias y datos de la consulta. Para pedir cita será necesario que se registre, ver Figura 56.
- Usuario registrado (paciente): el paciente podrá realizar citas con su médico de forma presencial y mediante videoconferencias además de disponer de todos los informes y documentos que su médico le va subiendo a su perfil, ver Figura 57.
- Administrador (médico): el médico puede tener un listado de todos sus pacientes junto con acceso a sus perfiles y todos los documentos que se encuentran en él. También tiene una lista de las citas pendientes junto con toda su información, ver Figura 58.
- El sistema: el propio sistema podrá, de manera automática, notificar a los usuarios de sus eventos.

Diagramas de caso de uso

Diagrama de usuario no registrado

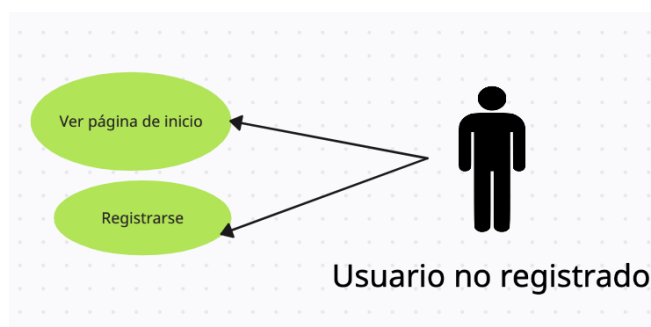


Figura 56. Diagrama de caso de uso UNR

Diagrama de paciente

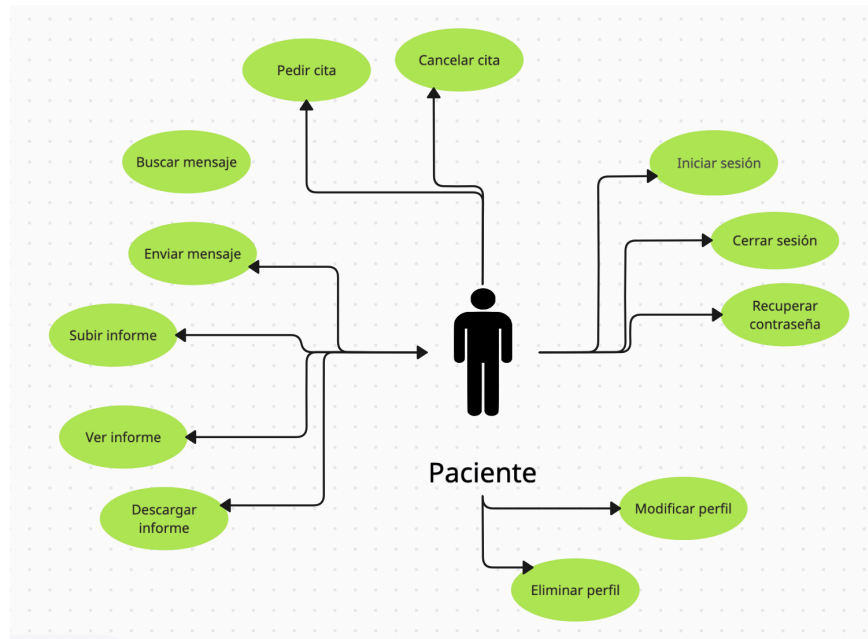


Figura 57. Diagrama de caso de uso paciente

Diagrama de médico

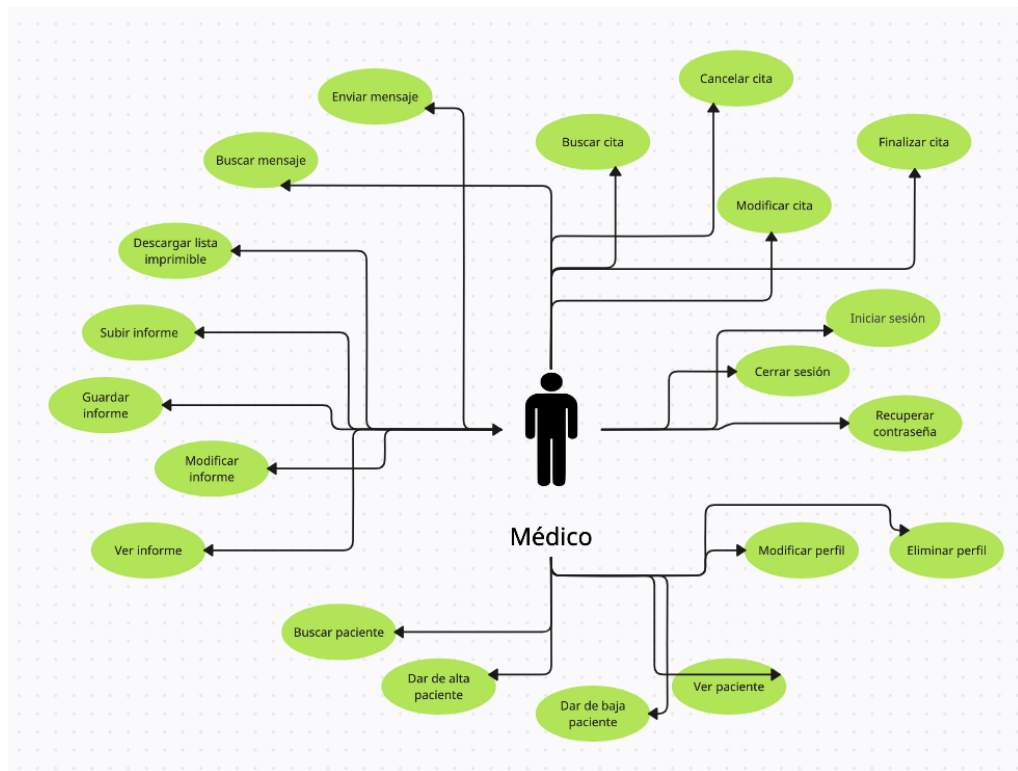


Figura 58. Diagrama de caso de uso médico

Especificación de requisitos

La agrupación de los casos de uso se ha realizado siguiendo los siguientes módulos con sus funcionalidades, de manera incremental:

- Módulo de autenticación
 - Registro
 - Iniciar de sesión / login
 - Cerrar sesión / logout
 - Recuperación de contraseña
- Gestión de pacientes
 - Crear paciente (médico)
 - Buscar paciente (médico)
 - Dar de baja paciente (médico)
 - Modificar perfil
 - Eliminar perfil
- Gestión de citas
 - Reservar cita nueva (paciente)
 - Cancelar cita (médico)
 - Modificar cita (médico)
 - Finalizar cita (médico)
 - Ver citas
 - Notificación de cualquier actualización de cita (administrador)
- Gestión de informes
 - Crear nuevo informe (médico)
 - Modificar informe existente (médico)
 - Eliminar informe (médico)
 - Ver informe (paciente)
 - Guardar informe (paciente)
 - Buscar informe
 - Listar informes (paciente)
 - Listar informes de un paciente (médico)

- Consultas médicas
 - Iniciar videoconferencia
 - Finalizar videoconferencia
 - Subir documento/imagen/vídeo (paciente)
 - Enviar mensajes
 - Ver mensajes

Módulo de autenticación

Requisito	Registro	
Identificador	1.1	
Prioridad	Alta	
Precondición	NA	
Descripción	El paciente puede crearse una cuenta en la aplicación	
Entrada	Nombre y apellidos, correo electrónico, contraseña, número de teléfono, DNI, género, fecha de nacimiento.	
Salida	NA	
Secuencia normal	Paso	Acción
	1	El usuario accede a la aplicación web y clic en "Regístrate"
	2	Aparecerá un formulario que deberá rellenar con sus datos personales y enviará el formulario.
	3	El sistema valida el formulario enviado.
	4	Se muestra de nuevo la ventana principal.
Postcondición	Se ha creado una nueva cuenta de usuario	
Excepciones	Paso	Acción
	2	Se muestra mensaje de error si el DNI o correo electrónico ya existen
	2	Se muestra mensaje de error si las contraseñas no coinciden
	2	Se muestra mensaje de error si la contraseña no tiene el formato requerido (una mayúscula, minúscula, número y es entre 8 y 16 caracteres)
	2	Se muestra mensaje de error si el correo electrónico no tiene el formato correcto
Comentarios	NA	
Actores	Usuario no registrado	

Tabla 1. Registro

Requisito	Inicio de sesión	
Identificador	1.2	
Prioridad	Alta	
Precondición	El usuario debe estar previamente registrado	
Descripción	El usuario iniciará sesión en su cuenta	
Entrada	Nombre y apellidos, correo electrónico, contraseña, número de teléfono, DNI, género, fecha de nacimiento.	
Salida	NA	
Secuencia normal	Paso	Acción
	1	En la ventana principal, el usuario rellenará el formulario de <i>login</i> introduciendo sus datos: DNI y contraseña
	2	El sistema validará los campos
	3	Se muestra la ventana de perfil
Postcondición	Se ha creado una nueva cuenta de usuario	
Excepciones	Paso	Acción
	2	Se muestra mensaje de error si el DNI o correo electrónico ya existen
	2	Se muestra mensaje de error si las contraseñas no coinciden
	2	Se muestra mensaje de error si la contraseña no tiene el formato requerido (una mayúscula, minúscula, número y es entre 8 y 16 caracteres)
	2	Se muestra mensaje de error si el correo electrónico no tiene el formato correcto
Comentarios	NA	
Actores	Usuario no registrado	

Tabla 2. Login

Requisito	Cierre de sesión	
Identificador	1.3	
Prioridad	Baja	
Precondición	El usuario debe estar previamente registrado y haber iniciado sesión	
Descripción	El usuario puede cerrar sesión en la aplicación	
Entrada	NA	
Salida	Se devuelve a la página principal de la aplicación	
Secuencia normal	Paso	Acción
	1	En la ventana principal, el usuario, hará click en su foto de perfil, y luego en "Cerrar sesión"
	2	El sistema cerrará la sesión y devolverá al usuario a la ventana principal.
Postcondición	El usuario ha cerrado su sesión	
Excepciones	Paso	Acción
	2	Se muestra un mensaje de error si no se puede cerrar sesión
Comentarios	NA	
Actores	Usuario registrado	

Tabla 3. Logout

Requisito	Recuperación de contraseña	
Identificador	1.4	
Prioridad	Alta	
Precondición	El usuario debe estar previamente registrado	
Descripción	El usuario puede recuperar su contraseña en caso de olvido	
Entrada	Correo electrónico	
Salida	Correo electrónico con un link para cambiar la contraseña	
Secuencia normal	Paso	Acción
	1	En la ventana principal, el usuario, hará click en "Recuperar contraseña"
	2	El sistema le pedirá que introduzca su correo electrónico para poder enviarle el enlace
	3	El sistema validará el correo
	4	Se le enviará un enlace para cambiar la contraseña
	5	El usuario deberá rellenar el formulario con la nueva contraseña, 2 veces y hacer <i>click</i> en "Cambiar contraseña"
	6	El sistema validará las contraseñas
	7	Se le enviará un correo electrónico al usuario informando el cambio de contraseña
Postcondición	El usuario ha iniciado sesión en la aplicación	
Excepciones	Paso	Acción
	3	Se muestra un mensaje de error si el correo no existe
	6	Se muestra un mensaje de error si las contraseñas no coinciden y tienen el formato requerido (una mayúscula, minúscula, número y es entre 8 y 16 caracteres)
Comentarios	NA	
Actores	Usuario registrado	

Tabla 4. Recuperación de contraseña

Módulo de gestión de pacientes

Requisito	Dar de alta paciente (médico)	
Identificador	2.1	
Prioridad	El paciente	
Precondición	El paciente debe estar registrado en la app	
Descripción	El médico dará de alta al paciente creando una ficha para ir gestionando sus consultas y su historial médico,	
Entrada	DNI, nombre y apellidos, edad, género, número de teléfono, correo electrónico, número de la seguridad social y dirección.	
Salida	Una ficha en formato de tabla con los datos introducidos.	
Secuencia normal	Paso	Acción
	1	El médico hace <i>click</i> en la sección "Crear ficha paciente"
	2	Le aparecerá un formulario con datos a rellenar
	3	Enviará el formulario con "Confirmar y crear nuevo paciente"
	4	Se devolverá a la página "Pacientes" con el nuevo paciente añadido a la lista.
Postcondición	Se ha creado una nueva ficha de paciente.	
Excepciones	Paso	Acción
	2	Se muestra un mensaje de error si alguno de los campos está en el formato incorrecto.
	2	Se muestra un mensaje de error si falta por rellenar algún campo de los obligatorios.
Comentarios	NA	
Actores	Administrador (médico)	

Tabla 5. Dar de alta paciente

Requisito	Buscar paciente	
Identificador	2.2	
Prioridad	Media	
Precondición	El paciente debe estar dado de alta en la base de datos.	
Descripción	El médico podrá buscar a un paciente específico, con la finalidad de facilitar y agilizar la consulta y modificación de su ficha.	
Entrada	Nombre y apellidos o DNI...	
Salida	Ficha del paciente buscado.	
Secuencia normal	Paso	Acción
	1	En la ventana "Mis pacientes" habrá una barra de búsqueda
	2	El médico hará click en ella y escribirá los datos del paciente que desea buscar
	3	Si existe dicho paciente, aparecerá en la pantalla y podrá acceder a su ficha y sus datos.
Postcondición	Se ha entregado una ficha de un paciente previamente buscado.	
Excepciones	Paso	Acción
	2	Si no existe el paciente buscado, simplemente aparecerá la ventana en blanco.
Comentarios	La búsqueda se realizará a tiempo real, es decir, carácter a carácter	
Actores	Administrador (médico)	

Tabla 6. Buscar paciente

Requisito	Dar de baja paciente	
Identificador	2.3	
Prioridad	Baja	
Precondición	El paciente debe estar dado de alta en la base de datos	
Descripción	El médico podrá dar de baja a un paciente que ya no vaya a atender	
Entrada	NA	
Salida	El paciente ha quedado "inactivo" en la base datos	
Secuencia normal	Paso	Acción
	1	El médico accederá al listado de pacientes (mediante búsqueda o manual)
	2	Hará click en la ficha de este y le aparecerán varias opciones
	3	Dará de baja al paciente en el botón "Dar de baja"
	4	Se abrirá una ventana flotante para confirmar y dará al botón "Confirmar"
Postcondición	Se devolverá a la ventana "Mis pacientes > Pacientes archivados"	
Excepciones	Paso	Acción
Comentarios	Si el paciente está "inactivo", no podrá realizar videoconferencias ni comunicarse con el médico, así como utilizar la mayoría de funcionalidad de la aplicación.	
Actores	Administrador (médico)	

Tabla 7. Dar de baja paciente

Requisito	Modificar perfil	
Identificador	2.4	
Prioridad	Media	
Precondición	El usuario debe estar previamente registrado y haber iniciado sesión	
Descripción	El usuario puede modificar algunos datos de su perfil en la aplicación	
Entrada	Datos personales a modificar	
Salida	Datos personales modificados	
Secuencia normal	Paso	Acción
	1	En la ventana principal, el usuario, hará click en su foto de perfil, y luego en "Modificar perfil"
	2	Aparecerá la ventana "Modificar perfil" y podrá modificar los datos permitidos mediante un formulario.
	3	El sistema validará el formulario con los nuevos datos.
	4	Se muestra la ventana principal y un mensaje de éxito "Los datos de usuario han sido modificados correctamente"
Postcondición	El usuario ha modificado sus datos personales	
Excepciones	Paso	Acción
	3	Se muestra un mensaje de error si el correo ya existe
	3	Se muestra un mensaje de error si el número de teléfono ya existe
Comentarios	3	Se muestra un mensaje de error si las contraseñas no coinciden y tienen el formato requerido (una mayúscula, minúscula, número y es entre 8 y 16 caracteres)
	<p>Los datos personales que el usuario podrá modificar son:</p> <ul style="list-style-type: none"> • Nombre y apellidos • Correo electrónico • Número de teléfono • Género • Contraseña <p>No podrá modificar el DNI porque es único.</p>	
Actores	Usuario registrado	

Tabla 8. Modificar perfil

Requisito	Eliminar perfil	
Identificador	2.5	
Prioridad	Baja	
Precondición	El usuario debe estar previamente registrado y haber iniciado sesión	
Descripción	El usuario puede eliminar su cuenta de usuario si ya no desea seguir utilizando la aplicación	
Entrada	NA	
Salida	Se devuelve a la página principal de la aplicación	
Secuencia normal	Paso	Acción
	1	En la ventana principal, el usuario, hará click en su foto de perfil, y luego en "Eliminar perfil"
	2	Aparecerá una ventana emergente con un mensaje que diga "¿Estás seguro de que deseas eliminar tu perfil?" en el que el usuario hará click "Sí, quiero eliminar mi perfil definitivamente"
	3	El sistema eliminará el perfil y se devolverá a la ventana principal
Postcondición	El usuario ha eliminado su perfil	
Excepciones	Paso	Acción
	3	Se muestra un mensaje de error si no se ha podido eliminar el perfil
Comentarios	NA	
Actores	Usuario registrado	

Tabla 9. Eliminar perfil

Módulo de gestión de citas

Requisito	Reservar nueva cita	
Identificador	3.1	
Prioridad	Alta	
Precondición	El usuario ha de estar registrado en la app y haber iniciado sesión	
Descripción	El usuario puede reservar una cita con su médico mediante la aplicación web	
Entrada	NA	
Salida	Se envía un correo electrónico de confirmación y la cita aparece en la ventana "Citas pendientes"	
Secuencia normal	Paso	Acción
	1	El paciente hace click en "Reservar cita"
	2	Elige el día y la hora a la que desea la cita de entre los disponibles
	3	Le da a "Confirmar"
	4	El sistema valida la cita y se redirige a ventana principal
	5	Se envía un correo al paciente con la reserva en modo "Pendiente de confirmación"
Postcondición	La cita aparece en la sección "Citas pendientes" y en el correo electrónico	
Excepciones	Paso	Acción
	4	Se muestra un mensaje de error si no se puede reservar una cita
Comentarios	Por ejemplo, otra persona realiza la reserva en el mismo turno y prácticamente en el mismo momento y, el paciente, al hacer click en "Confirmar" su página se actualiza y deja de estar disponible ese horario.	
Actores	Usuario registrado (paciente)	

Tabla 10. Reservar cita

Requisito	Cancelar cita	
Identificador	3.2	
Prioridad	Alta	
Precondición	El médico ha de estar registrado y haber iniciado sesión, y la cita debe existir	
Descripción	El médico puede rechazar una cita con un paciente	
Entrada	NA	
Salida	Mensaje de rechazo de la cita	
Secuencia normal	Paso	Acción
	1	El médico accede a la ventana "Citas pendientes"
	2	Abre la cita que desee anular, y hace click en "Cancelar cita"
	3	Se abre una ventana flotante y hace click en "Confirmar"
	4	El sistema elimina la cita de la base de datos y del calendario
	5	Se notifica al paciente de que su cita ha sido cancelada
Postcondición	La cita pasa a la ventana "Citas canceladas" y se notifica al paciente	
Excepciones	Paso	Acción
	3	Se muestra un mensaje de error si no se ha podido cancelar la cita
Comentarios	NA	
Actores	Usuario registrado (médico)	

Tabla 11. Cancelar cita

Requisito	Modificar cita	
Identificador	3.3	
Prioridad	Media	
Precondición	La cita tiene que existir	
Descripción	El médico puede modificar la cita con un paciente si lo necesitase	
Entrada	Fecha y hora de la cita, comentarios adicionales	
Salida	Fecha y hora de la cita nueva	
Secuencia normal	Paso	Acción
	1	El médico accede a su calendario de citas y selecciona la que quiera modificar
	2	Cambia los datos que oportune, como la fecha o la hora o incluso puede modificar comentarios adicionales acerca de esta
	3	Guarda la cita con los cambios realizados (click en "Aplicar cambios")
	4	El sistema valida los cambios
	5	Se notifica al paciente del cambio de cita mediante correo
Postcondición	La nueva cita ha quedado guardada en el calendario del paciente y el médico, y se notifica al paciente del cambio.	
Excepciones	Paso	Acción
	4	Se muestra un mensaje de error si no se ha podido modificar la cita
Comentarios	NA	
Actores	Administrador médico	

Tabla 12. Modificar cita

Requisito	Finalizar cita	
Identificador	3.4	
Prioridad	Media	
Precondición	La cita tiene que estar registrada.	
Descripción	El médico podrá marcar la cita como finalizada y registrarla como cita realizada.	
Entrada	Nombre del paciente y hora de la cita.	
Salida	Mensaje de finalizado o completado con éxito.	
Secuencia normal	Paso	Acción
	1	El médico tiene una lista de las citas por realizar y hace click en la cita que acaba de terminar y otro click sobre el botón de finalizada.
	2	El sistema muestra un mensaje preguntando por la confirmación.
	3	El médico hace click en "Continuar".
	4	El sistema muestra la lista de citas.
Postcondición	Se ha marcado como finalizada la cita.	
Excepciones	Paso	Acción
	3	El usuario hace click en "Cancelar".
Comentarios	NA	
Actores	Administrador médico	

Tabla 13. Finalizar cita

Requisito	Ver citas	
Identificador	3.5	
Prioridad	Alta	
Precondición	El usuario (médico) ha de estar registrado en la app y haber iniciado sesión, y la cita debe existir	
Descripción	El médico puede ver un listado de sus próximas citas, así como las canceladas o archivadas.	
Entrada	NA	
Salida	NA	
Secuencia normal	Paso	Acción
	1	El usuario accede a la ventana "Próximas citas"
	2	Se muestra un listado de todas las citas ordenadas cronológicamente
Postcondición	Ver la cita	
Excepciones	Paso	Acción
	2	Se muestra un mensaje si no hay ninguna cita programada.
Comentarios	Este caso de uso es idéntico para ver las "Citas canceladas" o "Citas archivadas". En el caso del paciente, sólo podrá tener una próxima cita . Sin embargo, el médico puede tener varias con distintos pacientes.	
Actores	Administrador (médico) y usuario registrado (paciente)	

Tabla 14. Ver citas

Requisito	Notificación de actualización de cita	
Identificador	3.6	
Prioridad	Media	
Precondición	Que el paciente haya realizado prioritariamente una cita.	
Descripción	El paciente 24 horas antes de la cita realizada recibe un correo de recordatorio con la fecha y la hora que ha solicitado.	
Entrada	Nombre del paciente, fecha y hora de la cita realizada.	
Salida	Mensaje de recordatorio con nombre del paciente, fecha y hora de la cita realizada.	
Secuencia normal	Paso	Acción
	1	24 horas antes de dicha cita el usuario recibe un correo con el recordatorio de esta.
Postcondición	Se recibe el correo con el recordatorio y los datos.	
Excepciones	Paso	Acción
Comentarios	NA	
Actores	El sistema	

Tabla 15. Notificar cita

Módulo de gestión de informes

Requisito	Crear nuevo informe	
Identificador	4.1	
Prioridad	Alta	
Precondición	Paciente registrado.	
Descripción	El médico podrá crear informes para luego poder compartirlos con el paciente o tenerlos en el historial de este.	
Entrada	Datos del paciente	
Salida	Nuevo documento del informe del paciente.	
Secuencia normal	Paso	Acción
	1	El médico accede a la página web, se le muestra la pantalla principal y hace click en "Pacientes".
	2	El sistema muestra una pantalla en la que se le pide al usuario que introduzca el nombre del paciente para el que va dirigido el documento.
	3	El médico introduce el nombre del paciente.
	4	El sistema muestra una lista con los nombres de pacientes que contengan la palabra introducida antes.
	5	El médico selecciona el paciente al que quiere subir el informe y hace click en "Documentos" y dentro de este hace click en "Subir Documento".
	6	Se muestra de nuevo una pantalla en la que debe seleccionar el documento que quiere subir, hace click sobre este y le da a "Subir".
	7	El sistema muestra un mensaje preguntando por la confirmación.
	8	El médico hace click en "Continuar".
	9	El sistema muestra el perfil del paciente.
Postcondición	Se ha subido el informe al perfil del cliente.	
Excepciones	Paso	Acción
	4	No se ha encontrado ningún paciente.
	7	El usuario hace click en "Cancelar".
Comentarios	NA	
Actores	Administrador médico.	

Tabla 16. Subir informe

Requisito	Modificar informe	
Identificador	4.2	
Prioridad	Alta	
Precondición	Informe creado	
Descripción	El médico podrá modificar los informes subidos en los historiales de los pacientes actualizando su información.	
Entrada	Informe del paciente	
Salida	Informe modificado	
Secuencia normal	Paso	Acción
	1	El médico accede a la página web, se le muestra la pantalla principal y hace <i>click</i> en "Pacientes"
	2	El sistema muestra una pantalla en la que se le pide al usuario que introduzca el nombre del paciente para el que va dirigido el documento.
	3	El médico introduce el nombre del paciente.
	4	El sistema muestra una lista con los nombres de pacientes que contengan la palabra introducida antes.
	5	El médico selecciona el paciente al que quiere subir el informe y hace <i>click</i> en "Documentos".
	6	Dentro del apartado de "Documentos" le da a "Buscar".
	7	El médico introduce el nombre del documento que desea modificar.
	8	El sistema muestra una lista con los documentos del paciente que contengan la palabra introducida antes. El médico hace <i>click</i> en el documento que desea modificar.
	9	Se muestra una nueva ventana y hacemos <i>click</i> en "Modificar Documento".
	10	De nuevo aparece una pantalla en la que debe seleccionar el documento que quiere subir, hace <i>click</i> sobre este y le da a "Subir"
11	El sistema muestra un mensaje preguntando por la confirmación.	

	12	El médico hace click en "Continuar".
	13	El sistema muestra el perfil del paciente.
Postcondición	Se ha modificado el documento del paciente.	
Excepciones	Paso	Acción
	4	No se ha encontrado ningún documento.
	12	El usuario hace click en "Cancelar".
Comentarios	NA	
Actores	Administrador médico	

Tabla 17. Modificar informe

Requisito	Eliminar informe	
Identificador	4.3	
Prioridad	Alta	
Precondición	Informe existente	
Descripción	El médico podrá eliminar documentos del paciente.	
Entrada	Documento	
Salida	NA	
Secuencia normal	Paso	Acción
	1	El médico accede a la página web, se le muestra la pantalla principal y hace click en "Pacientes"
	2	El sistema muestra una pantalla en la que se le pide al usuario que introduzca el nombre del paciente para el que va dirigido el documento.
	3	El médico introduce el nombre del paciente.
	4	El sistema muestra una lista con los nombres de pacientes que contengan la palabra introducida antes.
	5	El médico selecciona el paciente al que quiere subir el informe y hace click en "Documentos".
	6	Dentro del apartado de "Documentos" le da a "Buscar".
	7	El médico introduce el nombre del documento que desea eliminar.
	8	El sistema muestra una lista con los documentos del paciente que contengan la palabra introducida antes. El médico hace clic en el documento que desea eliminar.
	9	Se muestra una nueva ventana y hacemos click en "Eliminar Documento".
	10	El sistema muestra un mensaje preguntando por la confirmación.
11	El médico hace click en "Continuar".	
Postcondición	Se ha eliminado el documento.	

	Paso	Acción
Excepciones	4	No se ha encontrado ningún documento.
	10	El usuario hace click en "Cancelar".
Comentarios	NA	
Actores	Administrador médico	

Tabla 18. Eliminar informe

Requisito	Ver Informe	
Identificador	4.4	
Prioridad	Media	
Precondición	Informe existente	
Descripción	El paciente podrá ver todos los documentos subidos por el médico.	
Entrada	Documento	
Salida	NA	
Secuencia normal	Paso	Acción
	1	El paciente accede a la página web, se le muestra la pantalla principal y hace click en "Mis Documentos".
	2	Al entrar en ese apartado le da a "Buscar".
	3	El paciente introduce el nombre del documento que desea ver.
	4	El sistema muestra una lista con los documentos del paciente que contengan la palabra introducida antes. El médico hace click en el documento que desea ver.
Postcondición	El paciente puede visualizar el documento.	
Excepciones	Paso	Acción
	3	No se ha encontrado el documento.
Comentarios	Cuando el paciente quiere dejar de ver el documento simplemente cierra el documento y vuelve a la lista de documentos.	
Actores	Usuario registrado (paciente)	

Tabla 19. Ver informe

Requisito	Guardar Informe	
Identificador	4.5	
Prioridad	Media	
Precondición	Que el informe esté creado.	
Descripción	El paciente podrá descargarse los informes que le va subiendo el médico.	
Entrada	Documento	
Salida	Documento descargado en el dispositivo del paciente	
Secuencia normal	Paso	Acción
	1	El paciente accede a la página web, se le muestra la pantalla principal y hace click en "Mis Documentos".
	2	Al entrar en ese apartado le da a "Buscar".
	3	El paciente introduce el nombre del documento que desea descargar.
	4	El sistema muestra una lista con los documentos del paciente que contengan la palabra introducida antes. El médico hace click en el documento que desea descargar.
	5	El sistema abre el documento y el paciente hace click en el emoticono de descargar.
Postcondición	El documento se descarga en el su dispositivo.	
Excepciones	Paso	Acción
	3	No se ha encontrado el documento.
	5	No se ha podido completar la descarga
Comentarios	NA	
Actores	Usuario registrado (paciente)	

Tabla 20. Guardar informe

Requisito	Buscar Informe	
Identificador	4.6	
Prioridad	Media	
Precondición	Nombre del Informe.	
Descripción	Tanto el médico como el paciente pueden buscar cualquier informe subido.	
Entrada	Nombre del documento.	
Salida	Documento encontrado.	
Secuencia normal	Paso	Acción
	1	El paciente accede a la página web, se le muestra la pantalla principal y hace click en "Mis Documentos".
	2	Al entrar en ese apartado le da a "Buscar".
	3	El paciente introduce el nombre del documento que desea descargar.
	4	El sistema muestra una lista con los documentos del paciente que contengan la palabra introducida antes. El médico hace click en el documento que desea consultar.
Postcondición	Documento encontrado	
Excepciones	Paso	Acción
	4	No se ha encontrado el documento.
Comentarios	En el caso del médico primero tendrá que buscar entre sus pacientes al que quiere consultar dicho informe. Una vez encontrado sigue el mismo proceso de hacer click en mis documentos y buscar el que quería consultar.	
Actores	Usuario registrado (paciente) y administrador (médico)	

Tabla 21. Buscar informe

Requisito	Listar Informes	
Identificador	4.7	
Prioridad	Media	
Precondición	NA	
Descripción	El paciente puede ver una lista de todos los documentos que tienen subidos en la plataforma.	
Entrada	NA	
Salida	Listado de informes.	
Secuencia normal	Paso	Acción
	1	El paciente accede a la página web, se le muestra la pantalla principal.
	2	Hace click en "Mis Documentos"
Postcondición	Aparece una lista de documentos.	
Excepciones	Paso	Acción
Comentarios	NA	
Actores	Usuario registrado (paciente)	

Tabla 22. Listar informes

Requisito	Listar informes de un paciente	
Identificador	4.8	
Prioridad	Media	
Precondición	NA	
Descripción	El médico puede ver todos los documentos subidos en cada ficha del paciente.	
Entrada	NA	
Salida	Lista de documentos del paciente	
Secuencia normal	Paso	Acción
	1	El médico accede a la página web, se le muestra la pantalla principal y hace click en "Mis Pacientes"
	2	Al entrar en ese apartado le da a "Buscar"
	3	El médico introduce el nombre del paciente que desea buscar
	4	El sistema muestra una lista con los pacientes que contengan el nombre introducida antes. El médico hace click en el paciente que quiere consultar su historial
	5	Una vez dentro de su ficha hace click en "Mis documentos"
Postcondición	Lista de documentos del paciente	
Excepciones	Paso	Acción
	4	No se ha encontrado al paciente.
Comentarios	NA	
Actores	Administrador (médico)	

Tabla 23. Listar informes de un paciente

Módulo de consultas médicas

Requisito	Iniciar videoconferencia	
Identificador	5.1	
Prioridad	Media	
Precondición	Que el paciente esté registrado	
Descripción	Los pacientes entrarán al meet de citas y esperarán a que el médico les proporcione el acceso a la sala.	
Entrada	Enlace de meet	
Salida	NA	
Secuencia normal	Paso	Acción
	1	El paciente accede a la página web, se le muestra la pantalla principal y hace click en "Videoconferencia".
	2	Entrará en el entorno de "Videoconferencia" en el que el usuario encontrará el link a la sala meet y hará clic sobre él.
	3	El sistema le redirigirá a la sala de espera de dicha reunión donde el paciente deberá esperar a que el médico le proporcione acceso.
Postcondición	El paciente entra en la videoconferencia.	
Excepciones	Paso	Acción
		NA
Comentarios	NA	
Actores	Usuario registrado (paciente) y administrador (médico)	

Tabla 24. Iniciar videoconferencia

Requisito	Finalizar videoconferencia	
Identificador	5.2	
Prioridad	Media	
Precondición	Que paciente esté en la videoconferencia	
Descripción	El paciente sale de la videoconferencia finalizando su cita.	
Entrada	NA	
Salida	NA	
Secuencia normal	Paso	Acción
	1	El paciente hace click en "Salir de la sala"
	2	El sistema le redirige a la pantalla principal.
Postcondición	Ha finalizado la videoconferencia.	
Excepciones	Paso	Acción
		NA
Comentarios	NA	
Actores	Usuario registrado (paciente) y administrador (médico)	

Tabla 25. Finalizar videoconferencia

Requisito	Subir documento / imagen / vídeo	
Identificador	5.3	
Prioridad	Media	
Precondición	El usuario (paciente o médico) debe estar registrado y haber iniciado sesión	
Descripción	Tanto médico como paciente podrán subir imágenes, vídeos o documentos a través de la cita o informe para facilitar el reconocimiento de la dolencia o enfermedad.	
Entrada	NA	
Salida	NA	
Secuencia normal	Paso	Acción
	1	El usuario hará click en la pestaña "Subir contenido multimedia" dentro de las ventanas "Historial" o "Próximas citas"
	2	Se abrirá la carpeta del ordenador para seleccionar el/los archivo/s
	3	El usuario seleccionará el/los archivo/s y se subirán a la web.
	4	El sistema validará el/los archivo/s y se guardarán.
Postcondición	Los archivos se han guardado en la base de datos	
Excepciones	Paso	Acción
	4	Se muestra un mensaje de error si los archivos no están en el formato correcto (pdf, word, jpg, png, mp4, avi...)
Comentarios	NA	
Actores	Administrador (médico) y paciente)	

Tabla 26. Subir documento / imagen / vídeo

Requisito	Enviar mensaje	
Identificador	5.4	
Prioridad	Alta	
Precondición	El usuario debe estar registrado y haber iniciado sesión	
Descripción	El usuario podría enviar mensajes en el caso de los pacientes únicamente a los médicos y en el de los médicos a cualquiera de los pacientes. Con el fin de resolver cualquier duda o inquietud que surja de una forma instantánea.	
Entrada	Datos tanto del emisor como del receptor	
Salida	Asunto y cuerpo de un mensaje redactado por el usuario	
Secuencia normal	Paso	Acción
	1	El usuario hará click en la sección de "Mensajes"
	2	Una vez dentro le dará al botón de "Redactar Mensaje"
	3	Cuando ya haya encontrado el paciente o el médico al que desea enviarle el mensaje hará click en el icono de enviar de este.
	4	En el modal que se abre el usuario rellenara los campos con la información necesaria.
	5	Tras finalizar el paso anterior pulsara el botón de enviar.
Postcondición	Devuelve al usuario a la página principal de "Mensaje"	
Excepciones	Paso	Acción
		NA
Comentarios	NA	
Actores	Usuario registrado (paciente y médico)	

Tabla 27. Enviar mensaje

Requisito	Ver mensaje	
Identificador	5.5	
Prioridad	Media	
Precondición	El usuario debe estar registrado y haber iniciado sesión	
Descripción	Los usuarios solo podrán visualizar en la bandeja de entrada los primeros caracteres que componen el mensaje, pero no el mensaje completo para ello tendrá que visualizarlo.	
Entrada	Datos del mensaje recibido, almacenado en la base de datos.	
Salida	NA	
Secuencia normal	Paso	Acción
	1	El usuario hará click en la sección de "Mensajes"
	2	Una vez dentro predeterminado le aparecerá su bandeja de entrada y tendrá que hacer click en el icono de ver mensaje.
	3	Ya realizado el paso anterior se abrirá un modal con todo el contenido de dicho mensaje para visualizar que finalizará dándole al botón de cerrar.
Postcondición	Devuelve al usuario a la ventana "Mensajes" a su bandeja de entrada.	
Excepciones	Paso	Acción
		NA
Comentarios	NA	
Actores	Usuario registrado (paciente y médico)	

Tabla 28. Ver mensaje

Apéndice B - Guía del usuario

En este apéndice se explicará en detalle cómo hacer uso de la aplicación dependiendo del tipo de usuario.

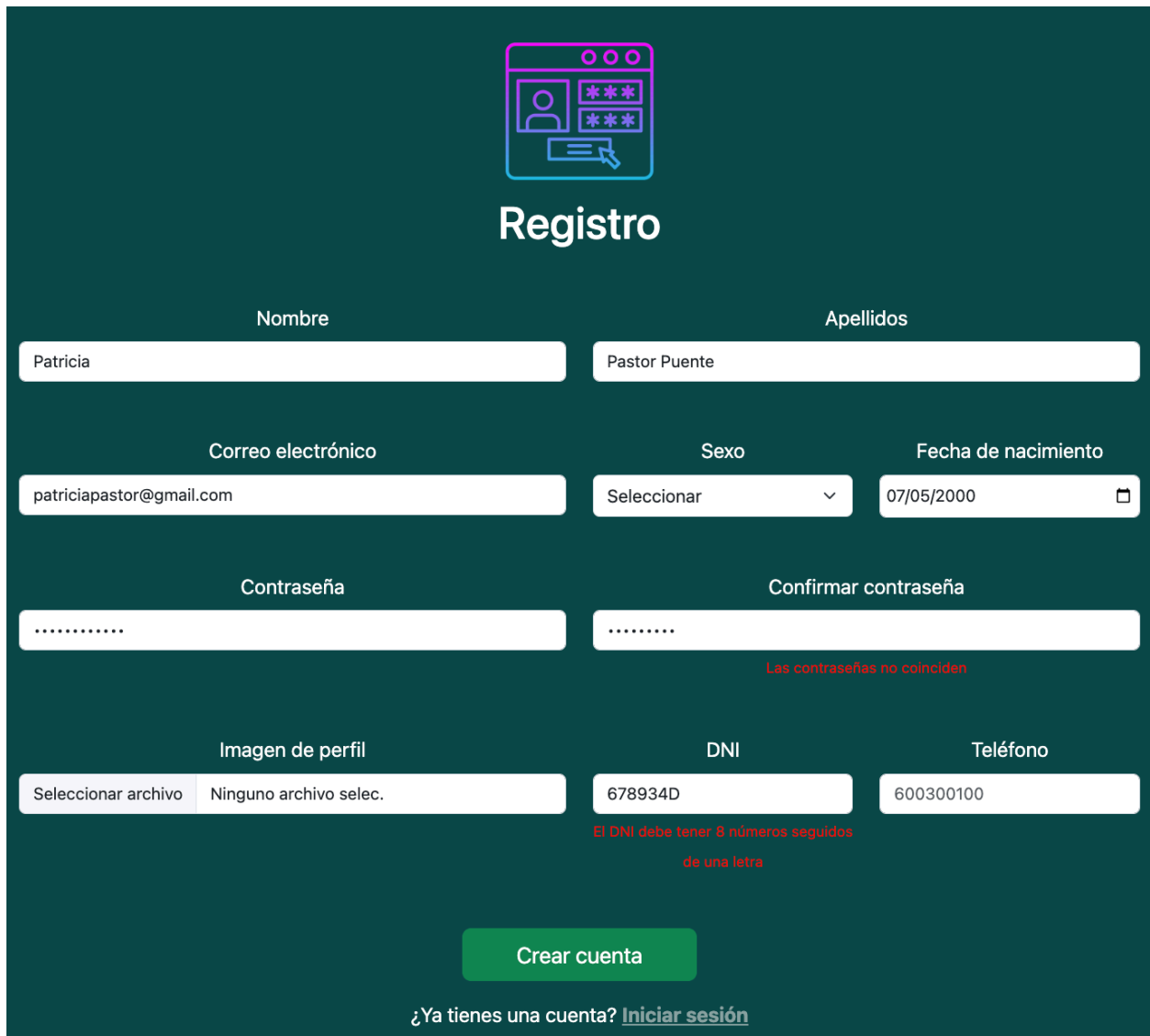
Usuario no registrado

En primer lugar, existe una vista general de la página como inicio, ver Figura 59, que muestra la información del hospital y las noticias de última hora; así como un mapa de localización que permite a cualquier persona no registrada informarse de los servicios que aporta la clínica.



Figura 59. Inicio

Además, se pueden apreciar los botones “Iniciar sesión” o “Pedir cita” que requerirán de un *login* para poder acceder al portal del paciente. En el caso de no estar registrado, la ventana de inicio de sesión tiene un enlace a la página de registro, en el que el usuario podrá completar un formulario introduciendo sus datos, como se muestra a continuación en la Figura 60.



The image shows a registration form titled "Registro" on a dark green background. At the top center is an icon of a computer monitor with a user profile, password fields, and a cursor. Below the title, the form is divided into several sections:

- Nombre:** Input field containing "Patricia".
- Apellidos:** Input field containing "Pastor Punte".
- Correo electrónico:** Input field containing "patriciapastor@gmail.com".
- Sexo:** Dropdown menu with "Seleccionar" and a downward arrow.
- Fecha de nacimiento:** Input field containing "07/05/2000" with a calendar icon.
- Contraseña:** Input field with masked characters ".....".
- Confirmar contraseña:** Input field with masked characters ".....". Below this field is a red error message: "Las contraseñas no coinciden".
- Imagen de perfil:** File selection area with "Seleccionar archivo" and "Ninguno archivo selec.".
- DNI:** Input field containing "678934D". Below this field is a red error message: "El DNI debe tener 8 números seguidos de una letra".
- Teléfono:** Input field containing "600300100".

At the bottom center is a green button labeled "Crear cuenta". Below the button is a link: "¿Ya tienes una cuenta? [Iniciar sesión](#)".

Figura 60. Registro con errores

Usuario paciente

Una vez ya registrado en la aplicación, el usuario podrá iniciar sesión en la aplicación mediante el botón "Iniciar sesión" como se ve en la Figura 61 en la cabecera, o haciendo click en el botón "Pedir cita" que le abrirá un modal para pedirle las credenciales como se ve en la Figura 62.

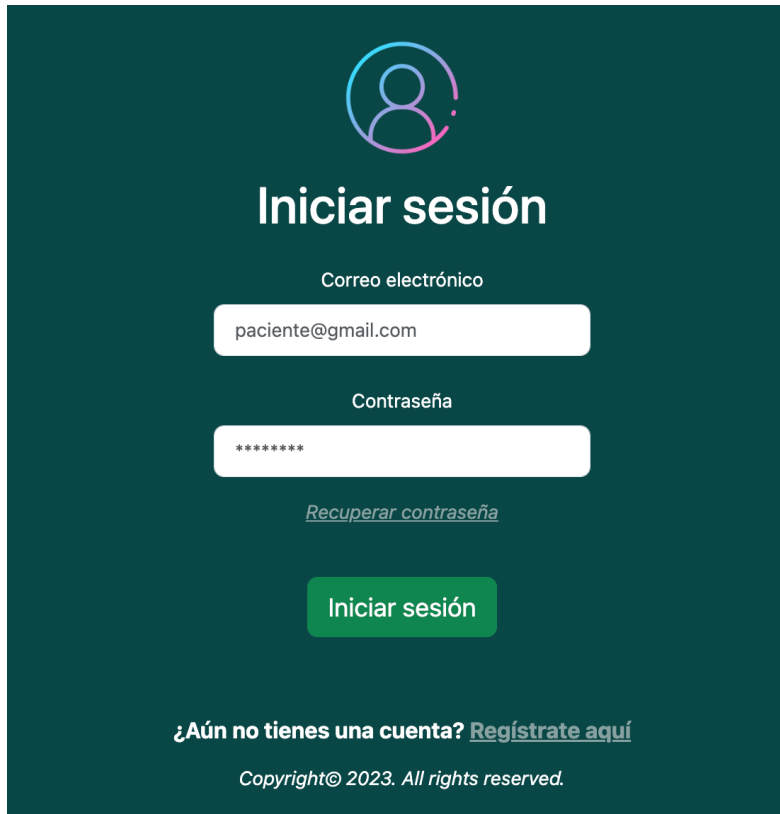


Figura 61. Inicio sesión

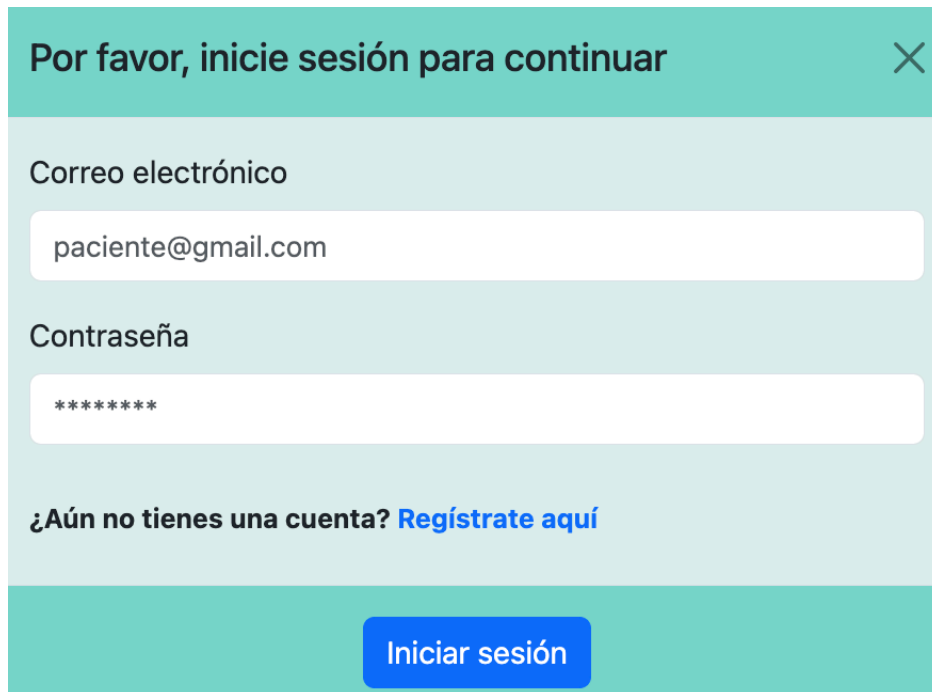


Figura 62. Modal inicio sesión desde "Pedir cita"

Tendrá visible una barra de navegación en la que acceder a las distintas vistas. El usuario ya podrá tener acceso a la página de inicio pudiendo hacer uso de la función "Pedir cita". Aquí, elegirá los datos con respecto a la cita que irán apareciendo a medida que elija el médico, el tipo de cita y la fecha y hora, ver Figura 63.



Reservar nueva cita

Selecciona un médico

Belén Fernández Puertas

Tipo de cita

Videoconferencia

Fecha

28/05/2024

12:00

Continuar al pago

Figura 63. Modal reservar cita

Dependiendo del tipo de cita y tipo de pago, se le redirigirá a la pasarela de pago donde introducirá sus datos bancarios para poder guardar su cita, como se ve en la Figura 64. En el caso de ser presencial y marcar la casilla "Pagar en consulta", no tendrá que hacer nada más y volverá automáticamente a la página de inicio.



← TFG consultas médicas TEST MODE

Videoconferencia

100,00 €

Consulta por videoconferencia de unos 20 minutos de duración



Pagar con [link](#)

O pagar con tarjeta

Correo electrónico

begopascual@gmail.com

Información de la tarjeta

4242 4242 4242 4242

07 / 28 829

Nombre del titular de la tarjeta

Begoña Pascual Goenechea

País o región

España

Guardar mis datos de forma segura para un proceso de compra en un clic

Paga más rápido en TFG consultas médicas y en todos los comercios que acepten Link.

Pagar

Figura 64. Ventana de pago

La sección "Mis citas" muestra una lista con la información de sus próximas reservas en modo de tabla, simplemente informativa, ver en la Figura 65.

Lista de mis citas



Fecha cita	Hora	Médico	Tipo	
18/4/2024	11:00	Javier Cambronero Santos	Presencial	
28/5/2024	12:00	Belén Fernández Puertas	Videoconferencia	

Figura 65. Vista mis citas desde paciente

La sección "Mis informes" también muestra una tabla con todos sus informes, subidos por él o por los médicos, como se observa en la Figura 66, además de permitirle mediante el botón "Subir informe" adjuntar un nuevo archivo de formato PDF mediante otro modal, ver Figura 67.

Lista de Informes

[Subir Informe](#)

Buscar 

Fecha	Descripción	Archivo
17/4/2024	Prueba Carlos	 Informe Carlos
17/4/2024	Informe 2	 Urgencias Carlos

Figura 66. Mis informes desde paciente

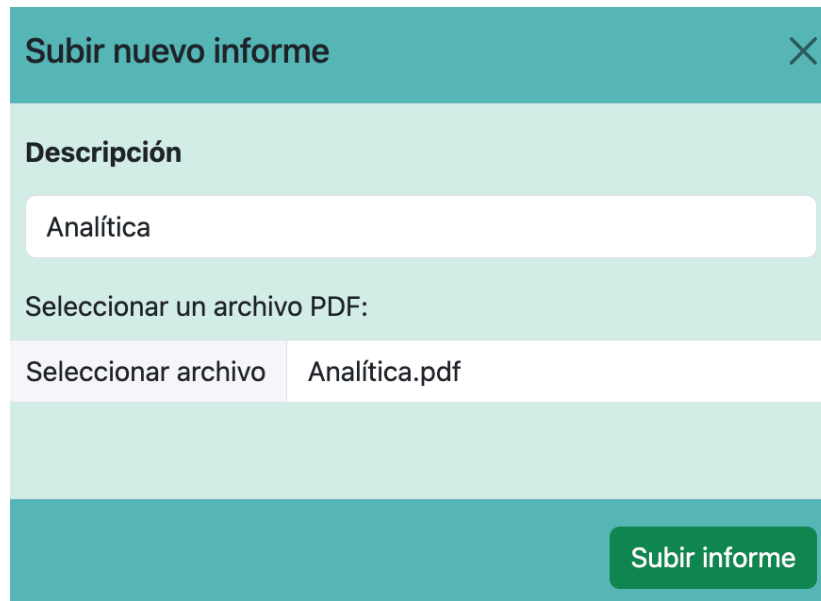


Figura 67. Ventana modal subir informe

La última sección disponible para el paciente es "Mensajes", donde tendrá una bandeja de entrada a modo de correo electrónico interno con sus mensajes recibidos, si están sin leer, aparecerán en azul como se puede observar en la Figura 68.



Figura 68. Bandeja de entrada

Aquí podrá marcar los mensajes como leídos con un solo click y ver el mensaje como en la Figura 69 para poder enviar una respuesta que abrirá otro modal como se muestra en la Figura 70.

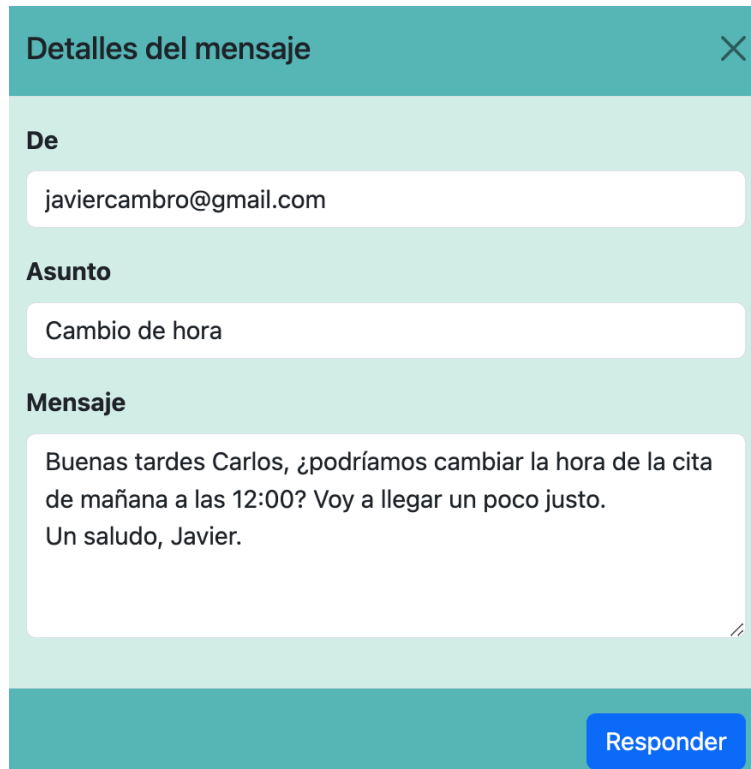


Figura 69. Ver mensaje

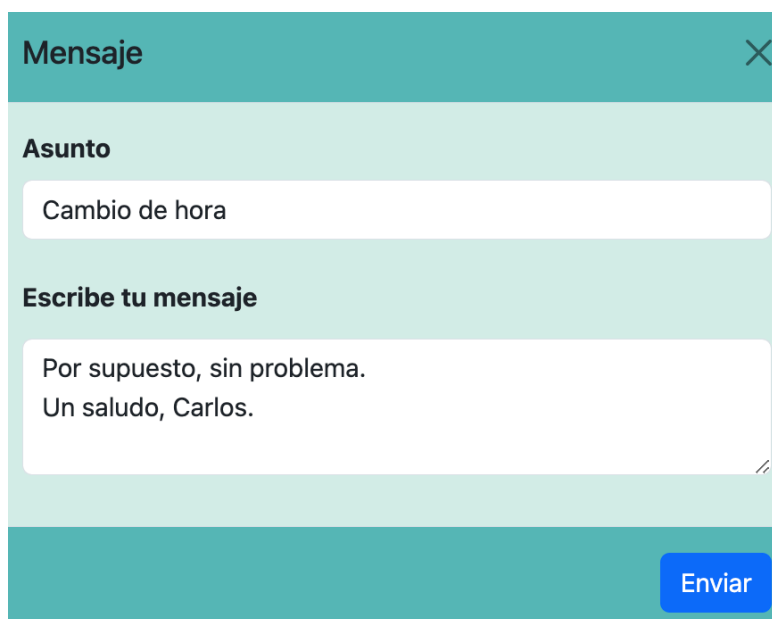


Figura 70. Responder mensaje

También podrá ver los “Mensajes enviados” y “Redactar un mensaje”, pudiendo elegir el médico destinatario al que enviarle dicho mensaje, con un campo Asunto y el cuerpo del mensaje, ver Figura 71.



Figura 71. Vista redactar mensaje

Otras opciones extra que puede realizar un paciente son la de "Eliminar perfil" (ver Figura 72) o "Editar perfil" (ver Figura 73), alojadas en la cabecera de la página a la derecha, haciendo click en su nombre a través de un dropdown⁹.

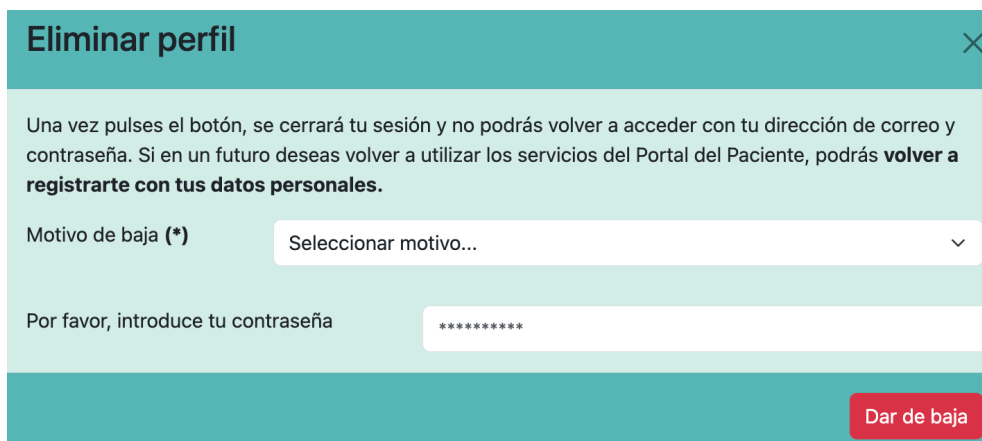


Figura 72. Eliminar perfil

⁹ Herramienta de Bootstrap que consiste en un menú desplegable que permite al usuario elegir entre varias opciones disponibles.

Editar perfil ×

Nombre

Apellidos

DNI

Teléfono

Correo electrónico

Contraseña

Confirmar contraseña

Figura 73. Editar perfil

Usuario médico

El último y más importante de los usuarios será el médico, quién tendrá la posibilidad de realizar multitud de funciones para llevar a cabo la correcta gestión de todos los pacientes, sus historiales médicos, así como de las consultas vía online.

Él también tendrá acceso a la página de inicio, pero, a diferencia de los anteriores, no tendrá el botón de "Pedir cita". Su barra de navegación albergará distintas opciones.

Tendrá dos *dropdown* para los pacientes y otro para las citas, que funcionarán de manera similar, categorizando ambas secciones, como se observa en la Figura 74.

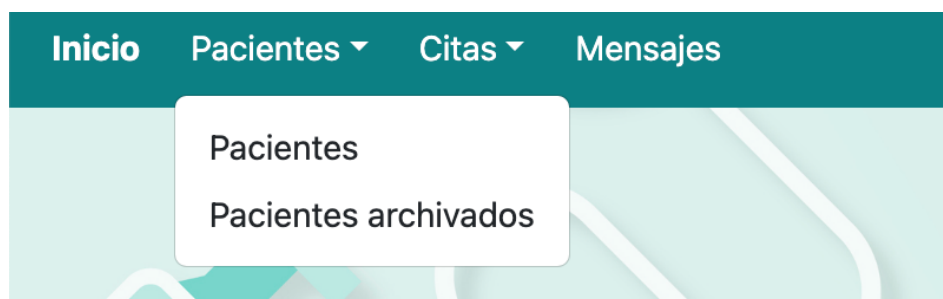


Figura 74. Barra de navegación con lista desplegable

En esta primera vista, "Pacientes", el médico puede visualizar de manera rápida todos los pacientes en activo (ver Figura 75), con varias opciones como subir un nuevo informe para dicho paciente o darlo de baja, pasándose automáticamente a "Pacientes archivados".



Figura 75. Lista pacientes activos

Dentro de cada paciente, hay un icono para visualizar su ficha técnica con mayor detalle de sus datos personales. Es aquí donde podrá ver el listado de informes de cada paciente, así como guardarlos, modificarlos subiendo una nueva versión o eliminarlos, como se observa a continuación en la Figura 76.

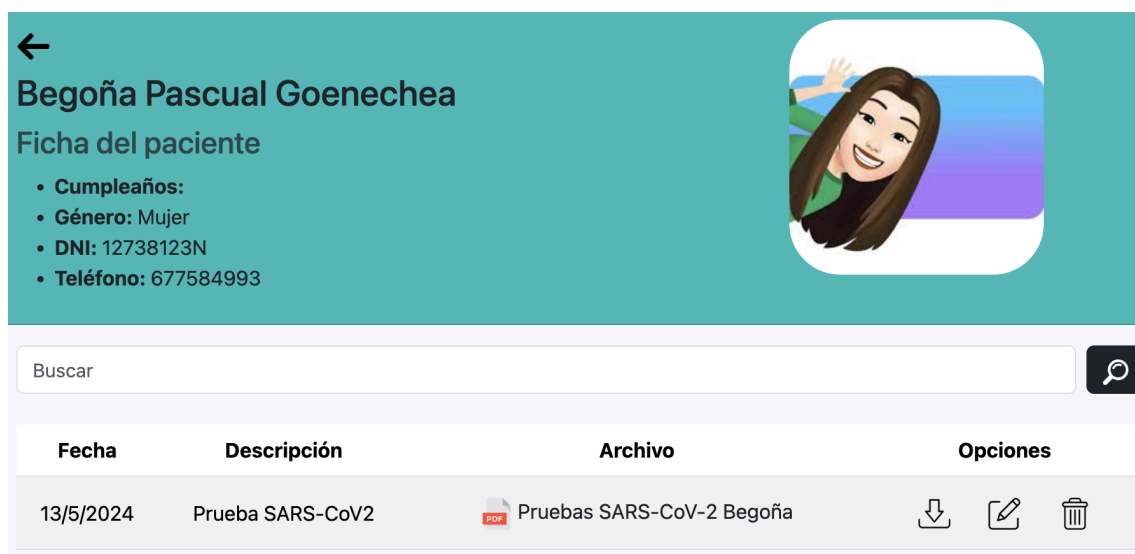


Figura 76. Ver paciente

Por último, tendrá visible otra subsección "Pacientes archivados" como la que se muestra en la Figura 77, con una tabla con los pacientes que hayan finalizado su tratamiento o ya no estén en la clínica, a modo de historial, pudiendo darlos de alta de nuevo en un futuro.

Pacientes archivados

Buscar

Descargar lista completa

	Nombre	Apellidos	Fecha de nacimiento	Género	
	José	Suárez Gómez	10/10/2000	Hombre	

Figura 77. Pacientes archivados

Respecto a la sección "Citas", también existen dos subsecciones, con las próximas o las pasadas, llamadas "Citas archivadas". Aquí el médico podrá ver en detalle todas sus próximas citas, con varios iconos también como "Cancelar cita", "Modificar cita" o "Finalizar cita", ver Figura 78.

Próximas citas

Todas Presenciales Telefónicas Videoconferencias

Buscar

Descargar lista completa

Nombre	Apellidos	Tipo	Fecha cita	Hora			
Isabel	Márquez Sevilla	Videoconferencia	23/5/2024	10:00			
Begoña	Pascual Goenechea	Videoconferencia	28/5/2024	12:00			

Figura 78. Próximas citas

Para facilitar la organización de estas, tiene disponible en la parte superior de la tabla distintos botones que filtran las reservas dependiendo del tipo de cita, siendo presencial, telefónica o videoconferencia como se puede ver en la Figura 79.



Figura 79. Citas con el filtro "telefónicas"

En la subsección "Citas archivadas" se muestra el historial de citas ya finalizadas. Como en la otra subsección, también tienen los botones que categorizan por el tipo de cita.

Otra opción integrada en estas dos secciones será la de "Descargar lista" con toda la información tanto de pacientes como de consultas en formato PDF, para así poder acceder a un listado imprimible como el que se muestra en la Figura 80 en un solo click.

Isabel	Márquez Sevilla	Videoconferencia	23/5/2024	10:00			
Begoña	Pascual Goenechea	Videoconferencia	28/5/2024	12:00			

Figura 80. PDF generado al descargar lista de citas

La última sección de la barra de navegación para el médico será la de "Mensajes" que funciona de igual manera que para los pacientes, siendo estos los únicos destinatarios disponibles.