



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA  
Mención en Ingeniería de Software

---

Gestión de citas del Gabinete de Logopedia de la  
Universidad de Valladolid

---

Alumno: Enrique Rojo Álvarez

Tutores: Yania Crespo González-Carvajal  
Alfonso Jesús Población Sáez





...





# Agradecimientos

En primer lugar, a los tutores del proyecto. A Yania, por su flexibilidad, paciencia y dedicación durante todo el proyecto. A Alfonso, por ser el precursor de la idea inicial.

Al STIC de la Universidad de Valladolid y al responsable de privacidad, David Sanz, por la ayuda que proporcionaron para la implementación del *Single Sign On*.

A las logopedas del Gabinete de Logopedia de la Universidad de Valladolid, por el tiempo, atención e ideas que dedicaron al proyecto.

A mis amigos, por apoyarme y esperarme durante todo el proyecto.

A mi pareja, que ha sido uno de mis pilares fundamentales, siempre confiando en mí y soportando mis momentos de mayor estrés.

A mis familiares, entre ellos a mis hermanos, por toda la motivación que me han dado y ser mi ejemplo a seguir. Y especialmente a mi madre, por ser la persona que más me ha apoyado, creyendo siempre en mí y aguantándome durante toda la carrera.



# Resumen

*Consultas médicas* es una aplicación web de gestión de citas del Gabinete de Logopedia de la Universidad de Valladolid. Esta solución surgió como respuesta a la necesidad de las logopedas del gabinete para renovar la aplicación web que estaba activa en ese momento.

La aplicación original carecía de algunas funcionalidades y necesitaba de una renovación de su stack tecnológico. La versión renovada de la aplicación se ha construido con la integración del sistema de inicio de sesión unificado de la Universidad de Valladolid. Se ha diseñado en torno a dos roles principales: el de solicitante, que pide y consulta sus citas, y el de gestor, que consulta las citas de los solicitantes y administra los horarios.

La aplicación ha sido desarrollada con una interfaz *responsive*, lo que permite ser utilizada tanto en ordenadores como en móviles y tablets. Para el desarrollo del frontend, se han empleado las tecnologías de HTML, CSS y JavaScript. En cuanto al backend, se ha utilizado Spring Boot. También se ha utilizado la herramienta Thymeleaf, para la integración de plantillas HTML5, tanto en el frontend como en el backend. Como base de datos relacional se ha utilizado MySQL.



# Abstract

*Consultas médicas* is a web application for managing appointments at the *Gabinete de Logopedia de la Universidad de Valladolid*. This solution emerged in response to the need of the speech therapists in the office to renew the web application that was active at that time.

The original application lacked some functionalities and needed a renewal of its technological stack. The renewed version of the application has been built with the integration of the single sign on of the University of Valladolid. It has been designed around two main roles: the applicant, who requests and consults their appointments, and the manager, who consults the appointments of the applicants and manages the schedules.

The application has been developed with a responsive interface, which allows it to be used on computers, mobiles and tablets. For the development of the frontend, HTML, CSS, and JavaScript technologies have been used. As for the backend, Spring Boot has been used. The Thymeleaf tool has also been used for the integration of HTML5 templates, both in the frontend and in the backend. MySQL has been used as the relational database.



# Índice general

<b>Agradecimientos</b>	<b>III</b>
<b>Resumen</b>	<b>V</b>
<b>Abstract</b>	<b>VII</b>
<b>Lista de figuras</b>	<b>XV</b>
<b>Lista de tablas</b>	<b>XXI</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Contexto . . . . .	1
1.2. Normas de seguridad y privacidad . . . . .	1
1.3. Motivación de este proyecto . . . . .	2
1.4. Aplicación actual en funcionamiento . . . . .	3
1.4.1. Interfaz para gestoras . . . . .	3
1.4.2. Interfaz para solicitantes . . . . .	7
1.4.3. Limitaciones . . . . .	12
1.5. Objetivos . . . . .	14
1.6. Estructura de la memoria . . . . .	14
<b>2. Requisitos y Planificación</b>	<b>17</b>
2.1. Stakeholders y usuarios . . . . .	17

2.1.1. Stakeholders . . . . .	17
2.1.2. Usuarios finales . . . . .	18
2.2. Elicitación de requisitos . . . . .	18
2.2.1. Requisitos funcionales . . . . .	18
2.2.2. Requisitos de información . . . . .	19
2.2.3. Reglas de negocio . . . . .	20
2.2.4. Requisitos no funcionales . . . . .	20
2.2.5. Casos de uso . . . . .	21
2.3. Plan de proyecto . . . . .	29
2.3.1. Fases del proyecto . . . . .	29
2.3.2. Artefactos del proyecto . . . . .	30
2.3.3. Planificación inicial . . . . .	31
2.4. Análisis de riesgos del proyecto . . . . .	38
2.5. Replanificación . . . . .	45
2.6. Presupuesto . . . . .	50
2.6.1. Presupuesto simulado . . . . .	51
2.6.2. Presupuesto real . . . . .	51
2.7. Metodologías para el desarrollo . . . . .	52
2.7.1. Privacidad desde el diseño . . . . .	52
<b>3. Análisis</b>	<b>55</b>
3.1. Diagrama casos de uso . . . . .	55
3.2. Modelo de dominio . . . . .	57
3.3. Análisis de procesos . . . . .	60
3.4. Análisis de estados . . . . .	63
3.5. Análisis de riesgos de seguridad de la información . . . . .	65



<b>4. Tecnologías utilizadas</b>	<b>67</b>
4.1. Herramientas de documentación, análisis y diseño . . . . .	67
4.1.1. Overleaf . . . . .	67
4.1.2. Astah Professional . . . . .	67
4.1.3. Balsamiq Wireframes . . . . .	68
4.1.4. PILAR Basic . . . . .	68
4.2. Herramientas de comunicación . . . . .	68
4.2.1. Microsoft Teams . . . . .	68
4.2.2. Outlook . . . . .	68
4.3. Herramientas de desarrollo . . . . .	68
4.3.1. Visual Studio Code . . . . .	69
4.3.2. Git . . . . .	69
4.3.3. GitLab . . . . .	69
4.3.4. git-crypt . . . . .	69
4.3.5. Java . . . . .	69
4.3.6. Spring Boot . . . . .	69
4.3.7. Nginx . . . . .	69
4.3.8. MySQL . . . . .	70
4.3.9. HTML . . . . .	70
4.3.10. CSS . . . . .	70
4.3.11. JavaScript . . . . .	70
4.3.12. Thymeleaf . . . . .	70
4.3.13. Docker . . . . .	70
4.3.14. Maven . . . . .	70
<b>5. Diseño</b>	<b>71</b>
5.1. Diseño de interfaz de usuario . . . . .	71

5.1.1. Interfaz gestora . . . . .	72
5.1.2. Interfaz solicitante . . . . .	87
5.2. Arquitectura del sistema . . . . .	95
5.3. Diagrama de despliegue . . . . .	99
5.4. Modelo de bases de datos . . . . .	101
<b>6. Implementación y pruebas</b>	<b>103</b>
6.1. Bases de datos . . . . .	103
6.2. Seguridad . . . . .	104
6.2.1. Single Sign-On . . . . .	104
6.2.2. Diseño para evitar inyecciones de código . . . . .	106
6.2.3. git-crypt . . . . .	106
6.3. Estructura del código . . . . .	106
6.4. Dificultades y decisiones . . . . .	109
6.5. Organización a la hora de desarrollar . . . . .	112
6.6. Pruebas . . . . .	112
6.6.1. Inicio de sesión . . . . .	113
6.6.2. Solicitante - Pedir cita . . . . .	114
6.6.3. Solicitante - Añadir información de paciente . . . . .	117
6.6.4. Solicitante - Consultar citas . . . . .	118
6.6.5. Solicitante - Cancelar cita . . . . .	121
6.6.6. Solicitante - Consultar información de contacto del gabinete . . . . .	122
6.6.7. Gestora - Consultar citas de todas las gestoras . . . . .	123
6.6.8. Gestora - Cancelar cita de solicitante . . . . .	125
6.6.9. Gestora - Modificar fecha y/u hora de una cita . . . . .	126
6.6.10. Gestora - Cambiar estado de cita . . . . .	127
6.6.11. Gestora - Consultar su horario de disponibilidad . . . . .	129

6.6.12. Gestora - Modificar el horario de disponibilidad . . . . .	130
6.6.13. Gestora - Bloquear días . . . . .	134
6.6.14. Gestora - Bloquear horas . . . . .	135
6.6.15. Gestora - Consultar calendario . . . . .	137
6.6.16. Valoración tras finalizar las pruebas . . . . .	138
<b>7. Seguimiento del proyecto</b>	<b>141</b>
7.1. Fase de inicio . . . . .	141
7.2. Fase de elaboración . . . . .	144
7.3. Fase de construcción . . . . .	148
7.3.1. Primera iteración . . . . .	148
7.3.2. Segunda iteración . . . . .	153
7.4. Fase de transición . . . . .	153
7.5. Resumen de la ejecución del proyecto . . . . .	154
<b>8. Conclusiones</b>	<b>155</b>
8.1. Conclusiones . . . . .	155
8.2. Líneas de trabajo futuras . . . . .	156
<b>Bibliografía</b>	<b>157</b>
<b>A. Manuales</b>	<b>161</b>
A.1. Manual de despliegue e instalación . . . . .	161
A.1.1. Requisitos de despliegue . . . . .	161
A.1.2. Construcción y despliegue . . . . .	162
A.2. Manual de mantenimiento . . . . .	162
A.3. Manual de usuario . . . . .	162
<b>B. Resumen de enlaces adicionales</b>	<b>171</b>



# Lista de Figuras

1.1. Interfaz actual de Gestoras - inicio de sesión. . . . .	3
1.2. Interfaz actual de Gestoras - inicio. . . . .	4
1.3. Interfaz actual de Gestoras - gabinetes. . . . .	4
1.4. Interfaz actual de Gestoras - edición de gabinete 1/2. . . . .	5
1.5. Interfaz actual de Gestoras - edición de gabinete 2/2. . . . .	5
1.6. Interfaz actual de Gestoras - patologías. . . . .	6
1.7. Interfaz actual de Gestoras - edición de patologías. . . . .	6
1.8. Interfaz actual de Gestoras - lista de próximas citas. . . . .	7
1.9. Interfaz actual de Gestoras - editar cita. . . . .	7
1.10. Interfaz actual de Solicitante - inicio de sesión. . . . .	9
1.11. Interfaz actual de Solicitante - inicio. . . . .	9
1.12. Interfaz actual de Solicitante - Solicitar cita 1/4. . . . .	10
1.13. Interfaz actual de Solicitante - Solicitar cita 2/4. . . . .	10
1.14. Interfaz actual de Solicitante - Solicitar cita 3/4. . . . .	11
1.15. Interfaz actual de Solicitante - Solicitar cita 4/4. . . . .	11
1.16. Interfaz actual de Solicitante - mis citas. . . . .	12
1.17. Interfaz actual de Solicitante - contacto. . . . .	12
2.1. Matriz casos de uso - requisitos funcionales. . . . .	29

3.1. Diagrama casos de uso. . . . .	56
3.2. Modelo de dominio inicial. . . . .	58
3.3. Modelo de dominio modificado. . . . .	59
3.4. Diagrama de actividad del solicitante. . . . .	61
3.5. Diagrama de actividad de las gestoras. . . . .	62
3.6. Máquina de estados de cita. . . . .	64
3.7. Máquina de estados de día. . . . .	65
5.1. Boceto de inicio de sesión. . . . .	72
5.2. Boceto móvil de inicio de sesión. . . . .	72
5.3. Boceto de gestora - inicio. . . . .	73
5.4. Boceto móvil de gestora - inicio. . . . .	73
5.5. Boceto de gestora - consulta de citas. . . . .	74
5.6. Boceto móvil de gestora - consulta de citas. . . . .	74
5.7. Boceto de gestora - detalle de cita 1. . . . .	74
5.8. Boceto móvil de gestora - detalle de cita 1. . . . .	74
5.9. Boceto de gestora - detalle de cita 2. . . . .	75
5.10. Boceto móvil de gestora - detalle de cita 2. . . . .	75
5.11. Boceto de gestora - confirmación cancelar cita. . . . .	75
5.12. Boceto móvil de gestora - confirmación cancelar cita. . . . .	75
5.13. Boceto de gestora - confirmación, modificar estado de cita. . . . .	76
5.14. Boceto móvil de gestora - confirmación, modificar estado de cita. . . . .	76
5.15. Boceto de gestora - modificar fecha y hora de cita. . . . .	77
5.16. Boceto móvil de gestora - modificar fecha y hora de cita. . . . .	77
5.17. Boceto de gestora - confirmación, modificar fecha y hora de cita. . . . .	78
5.18. Boceto móvil de gestora - confirmación, modificar fecha y hora de cita. . . . .	78
5.19. Boceto de gestora - horario de disponibilidad. . . . .	79

5.20. Boceto móvil de gestora - horario de disponibilidad. . . . .	79
5.21. Boceto de gestora - modificar horario de disponibilidad. . . . .	79
5.22. Boceto móvil de gestora - modificar horario de disponibilidad. . . . .	79
5.23. Boceto de gestora - confirmación de modificar horario de disponibilidad. . . .	80
5.24. Boceto móvil de gestora - confirmación de modificar horario de disponibilidad.	80
5.25. Boceto de gestora - bloqueo de días. . . . .	81
5.26. Boceto móvil de gestora - bloqueo de días. . . . .	81
5.27. Boceto de gestora - confirmación de bloqueo de días. . . . .	82
5.28. Boceto móvil de gestora - confirmación de bloqueo de días. . . . .	82
5.29. Boceto de gestora - resultado del bloqueo de días. . . . .	83
5.30. Boceto móvil de gestora - resultado del bloqueo de días. . . . .	83
5.31. Boceto de gestora - bloqueo de horas. . . . .	84
5.32. Boceto móvil de gestora - bloqueo de horas. . . . .	84
5.33. Boceto de gestora - confirmación de bloqueo de horas. . . . .	85
5.34. Boceto móvil de gestora - confirmación de bloqueo de horas. . . . .	85
5.35. Boceto de gestora - calendario mensual. . . . .	86
5.36. Boceto móvil de gestora - calendario mensual. . . . .	86
5.37. Boceto de gestora - calendario anual. . . . .	87
5.38. Boceto móvil de gestora - calendario anual. . . . .	87
5.39. Boceto de solicitante - inicio. . . . .	88
5.40. Boceto móvil de solicitante - inicio. . . . .	88
5.41. Boceto de solicitante - mis citas. . . . .	89
5.42. Boceto móvil de solicitante - mis citas. . . . .	89
5.43. Boceto de solicitante - detalle de cita 1. . . . .	89
5.44. Boceto móvil de solicitante - detalle de cita 1. . . . .	89
5.45. Boceto de solicitante - detalle de cita 2. . . . .	90

5.46. Boceto móvil de solicitante - detalle de cita 2. . . . .	90
5.47. Boceto de solicitante - confirmación de cancelar cita. . . . .	90
5.48. Boceto móvil de solicitante - confirmación de cancelar cita. . . . .	90
5.49. Boceto de solicitante - pedir cita, calendario. . . . .	91
5.50. Boceto móvil de solicitante - pedir cita, calendario. . . . .	91
5.51. Boceto de solicitante - pedir cita para estudiante. . . . .	92
5.52. Boceto móvil de solicitante - pedir cita para estudiante. . . . .	92
5.53. Boceto de solicitante - pedir cita para PDI/PAS. . . . .	92
5.54. Boceto móvil de solicitante - pedir cita para PDI/PAS. . . . .	92
5.55. Boceto de solicitante - pedir cita 1a consulta. . . . .	93
5.56. Boceto móvil de solicitante - pedir cita 1a consulta. . . . .	93
5.57. Boceto de solicitante - pedir cita revisión. . . . .	94
5.58. Boceto móvil de solicitante - pedir cita revisión. . . . .	94
5.59. Boceto de solicitante - información de contacto del Gabinete de Logopedia. . . . .	95
5.60. Boceto móvil de solicitante - información de contacto del Gabinete de Logopedia. . . . .	95
5.61. Capas de la “Clean Architecture”. Tomada de [37] . . . . .	97
5.62. Diagrama de dependencias de la arquitectura. . . . .	98
5.63. Clases tipo participantes en un flujo de trabajo habitual en la arquitectura. . . . .	98
5.64. Diagrama de despliegue inicial, para la fase de diseño. . . . .	100
5.65. Modelo de bases de datos inicial, para la fase de diseño. . . . .	101
6.1. Diagrama de despliegue final, decidido en la implementación. . . . .	111
6.2. Modelo de bases de datos, decidido en la implementación. . . . .	111
A.1. Vista de inicio sin usuario autenticado. . . . .	163
A.2. Vista de inicio de sesión. . . . .	164
A.3. Vista de inicio con usuario autenticado. . . . .	164



A.4. Vista de calendario al solicitar cita. . . . .	165
A.5. Vista de información del usuario al solicitar cita. . . . .	166
A.6. Vista de pacientes al solicitar cita. . . . .	167
A.7. Vista de nuevo paciente al solicitar cita. . . . .	167
A.8. Vista de información al solicitar cita. . . . .	168
A.9. Vista de éxito al solicitar cita. . . . .	168



# Lista de Tablas

2.1. Planificación fases. . . . .	31
2.2. Planificación inicio. . . . .	32
2.3. Planificación elaboración. . . . .	34
2.4. Planificación construcción 1. . . . .	35
2.5. Planificación construcción 2. . . . .	36
2.6. Planificación transición. . . . .	37
2.7. Replanificación fases. . . . .	46
2.8. Replanificación elaboración. . . . .	47
2.9. Replanificación construcción 1. . . . .	48
2.10. Replanificación construcción 2. . . . .	49
2.11. Replanificación transición. . . . .	50
7.1. Seguimiento planificación inicio. . . . .	143
7.2. Seguimiento planificación elaboración 1/3. . . . .	145
7.3. Seguimiento planificación elaboración 2/3. . . . .	146
7.4. Seguimiento planificación elaboración 3/3. . . . .	147
7.5. Seguimiento planificación construcción 1. . . . .	153
7.6. Seguimiento planificación transición. . . . .	154



# Capítulo 1

## Introducción

### 1.1. Contexto

La Universidad de Valladolid (UVa) dispone de la unidad clínica de Logopedia, denominada Gabinete de Logopedia [42], la cual presta servicios a muchos pacientes pertenecientes a la comunidad universitaria, pudiendo llevar a cabo labores de evaluación y tratamiento. Con el fin de alcanzar su objetivo, los miembros de la Universidad de Valladolid interesados en recibir atención deben realizar una solicitud de cita previa. Dicho proceso se lleva a cabo a través de una aplicación web dentro de la Universidad de Valladolid, donde los interesados pueden seleccionar la fecha y hora que más les convenga e introducir varios datos personales, así como el motivo de su consulta. Una vez realizada la reserva de cita, la solicitud es recibida por el Gabinete de Logopedia, lo cual facilita la organización de sus profesionales.

Es de suma importancia que esta aplicación web funcione de manera correcta, tanto para los solicitantes, a fin de que no encuentren dificultades durante el proceso de solicitud de cita, como para el personal del Gabinete de Logopedia, que requiere contar con una visión y cierto grado de control sobre sus próximas citas, así como una visión completa de la información de los pacientes que serán atendidos. Además, resulta fundamental garantizar la seguridad y privacidad de los datos que se manejan en la aplicación web, dado que algunos de estos datos se consideran de tipo médico, son más sensibles de lo habitual.

### 1.2. Normas de seguridad y privacidad

Antes de continuar, se presentarán las normas relativas a la legislación actual sobre seguridad y privacidad que se tendrán en cuenta durante el proyecto. Se proporcionará la abreviatura correspondiente a cada una de ellas para facilitar la lectura del documento:

- REGLAMENTO (UE) 2016/679 DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 27 de abril de 2016 relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos y por el que se deroga la Directiva 95/46/CE [13] (Reglamento general de protección de datos, en adelante, RGPD).
- Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales [11], (en adelante, LOPDGDD).
- Real Decreto 311/2022, de 3 de mayo, por el que se regula el Esquema Nacional de Seguridad [12], (en adelante, ENS).

### 1.3. Motivación de este proyecto

Este proyecto nace a petición del propio Gabinete de Logopedia que plantea algunas quejas sobre la aplicación web que daba servicio a la fecha de inicio de este proyecto. Entre estas quejas se encontraban la falta de algunas funcionalidades, así como distintos apartados que les gustaría cambiar o mejorar. Las profesionales que gestionan el Gabinete de Logopedia comentan que esto entorpece su trabajo diario, en ocasiones, faltando información de los pacientes, ya sea porque no existen los campos necesarios a rellenar por parte del solicitante al pedir la cita o porque estos son opcionales y algunos deciden dejarlos vacíos. Aparte de lo anterior, también reportaba la falta de funcionalidad a la hora de gestionar dichas citas, impidiendo, por ejemplo, mover la fecha de una cita.

Teniendo en cuenta la gran labor que hace el Gabinete de Logopedia con sus pacientes, la calidad de sus servicios no debería verse afectada por una aplicación web. En consecuencia, se propone un nuevo desarrollo de la aplicación web de citas del Gabinete de Logopedia de la UVa, partiendo desde cero. Son varias las razones por las que no se parte de la aplicación actual realizando una labor de mantenimiento, casi reingeniería de la misma.

En primer lugar, la aplicación actual no tiene en cuenta todo lo necesario para cumplir el RGPD y el ENS por lo que haría falta una profunda revisión de su diseño y desarrollo. En segundo lugar, la aplicación fue desarrollada con un stack tecnológico actualmente en retirada por los servicios de informática de la UVa, requeriría de una gran actualización y revisión. Por otra parte, se prefiere rediseñar también las interfaces, así como modificar las funcionalidades actuales y añadir nuevas. Para llevar a cabo los procesos anteriores con éxito, es preferible abordar el desarrollo mediante metodologías y herramientas más modernas y adecuadas, teniendo en cuenta la privacidad desde el diseño y la seguridad de la aplicación y sus datos. Estas metodologías se explicarán en el Capítulo 2.

Finalmente, partiendo del contexto en el que se trata (trabajo académico de fin de grado de un estudiante) el completo desarrollo desde el principio de una aplicación web aplicando métodos y tecnologías modernas será de gran ayuda para el aprendizaje del estudiante.

La descripción de la aplicación web actual y sus limitaciones se detallan en el siguiente apartado.

## 1.4. Aplicación actual en funcionamiento

En esta sección se explica el funcionamiento y limitaciones de la aplicación web en vigor en el momento del inicio de este proyecto [40].

Esta aplicación web tiene dos tipos de accesos: el de los solicitantes que quieren reservar una cita y el de las profesionales del Gabinete de Logopedia (en adelante, “las gestoras”).

A fecha de la escritura de esta sección (14 de junio de 2023) la aplicación tiene las siguientes vistas para cada uno de los actores (gestoras y solicitantes):

### 1.4.1. Interfaz para gestoras

Primeramente, las gestoras se identifican en la aplicación a partir de la pantalla que se muestra en la Figura 1.1. Una vez iniciada sesión, se muestra la pantalla de inicio (Figura 1.2); en ella aparecen las siguientes secciones:

- Gabinetes (Figura 1.3): Se muestra el nombre del gabinete y un botón para editar la información relativa al mismo (Figuras 1.4 y 1.5).
- Patologías (Figura 1.6): Se ve la lista de patologías disponibles, cada una junto a dos botones, para editarla (Figura 1.7) y eliminarla de la lista.
- Citas (Figura 1.8, se ha pixelado la información sensible): Se muestra el listado de próximas citas, cada una junto a dos botones, para editarla (Figura 1.9) y eliminarla de la lista.
- Estadísticas: Este apartado se ha omitido debido a que se encuentra en desuso.

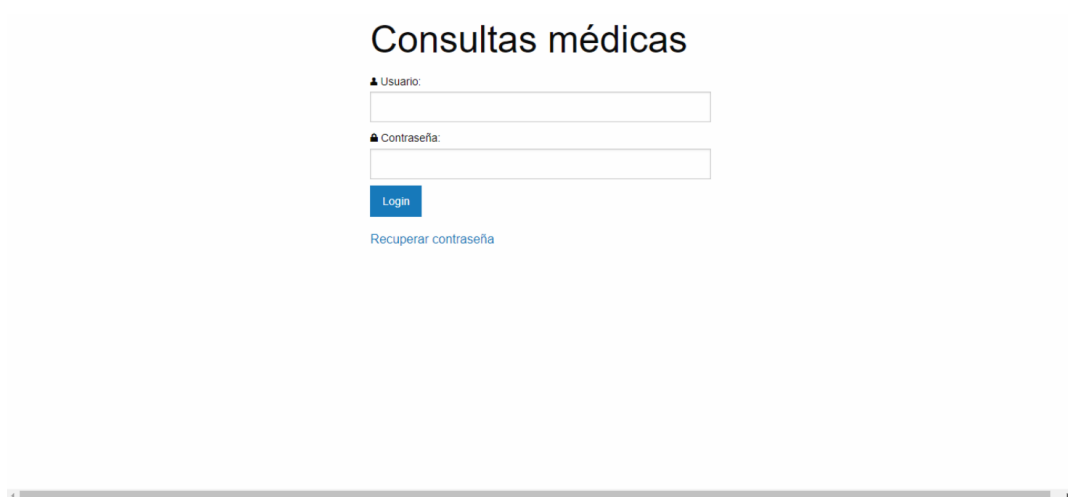


Figura 1.1: Interfaz actual de Gestoras - inicio de sesión.

## 1.4. APLICACIÓN ACTUAL EN FUNCIONAMIENTO

---

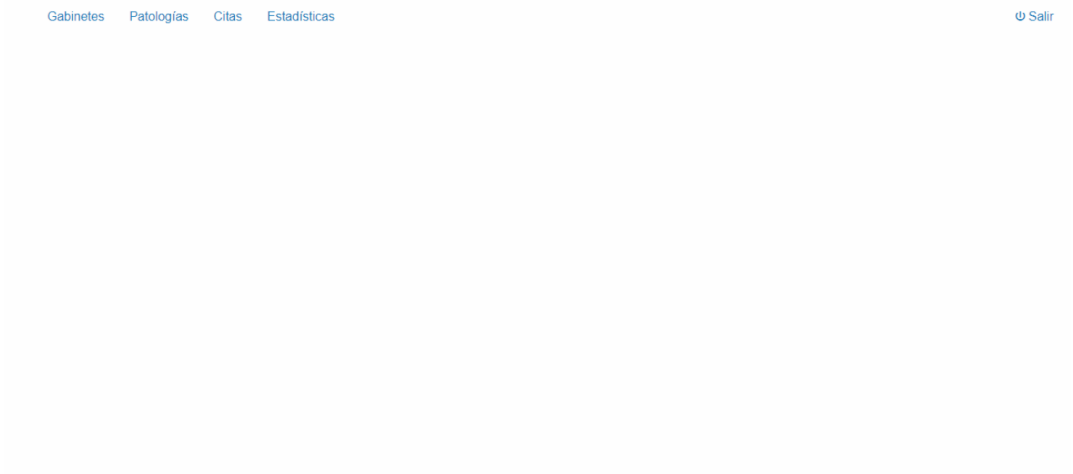


Figura 1.2: Interfaz actual de Gestoras - inicio.

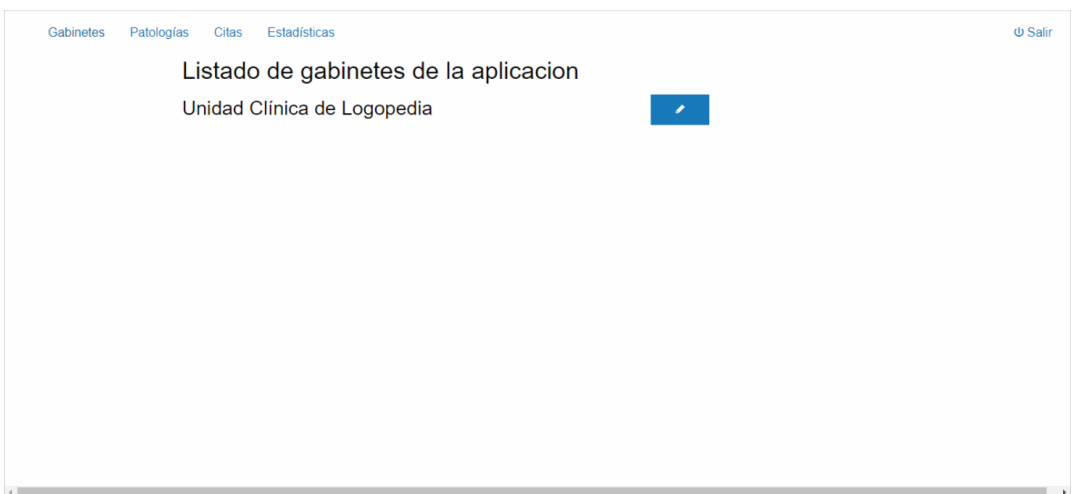


Figura 1.3: Interfaz actual de Gestoras - gabinetes.



Gabinetes Patologías Citas Estadísticas Salir

Nombre  
Unidad Clínica de Logopedia

Campus  
Valladolid

Localización  
Facultad de Medicina. 3ª Planta (Ala izquierda). Avda. Ramón y Cajal s/n, 47005 Valladolid. Por favor, deben citarse 24h ar

Teléfono de contacto  
983423021 / 608838683

Correo electrónico  
unidad.logopedia@uva.es

Tiempo por cita (en minutos)  
60

Número máximo de citas anuales por persona  
4

Figura 1.4: Interfaz actual de Gestoras - edición de gabinete 1/2.

Semana del mes con cita

Todas  Primera  Segunda  Tercera  Cuarta

	Primera	Segunda	Tercera	Cuarta
Lunes <input type="checkbox"/>	Hora de inicio mañana (HH:MM) <input type="text" value="Escribe el apellido"/>		Hora de fin mañana (HH:MM) <input type="text" value="Escribe el apellido"/>	Hora de inicio tarde (HH:MM) <input type="text" value="Escribe el apellido"/>
Martes <input checked="" type="checkbox"/>	Hora de inicio (HH:MM) <input type="text" value="9:00"/>		Hora de fin (HH:MM) <input type="text" value="12:00"/>	Hora de inicio tarde (HH:MM) <input type="text" value="17:00"/>
Miércoles <input checked="" type="checkbox"/>	Hora de inicio (HH:MM) <input type="text" value="9:00"/>		Hora de fin (HH:MM) <input type="text" value="12:00"/>	Hora de inicio tarde (HH:MM) <input type="text" value="Escribe el apellido"/>
Jueves <input type="checkbox"/>	Hora de inicio (HH:MM) <input type="text" value="Escribe el apellido"/>		Hora de fin (HH:MM) <input type="text" value="Escribe el apellido"/>	Hora de inicio tarde (HH:MM) <input type="text" value="Escribe el apellido"/>
Viernes <input type="checkbox"/>	Hora de inicio (HH:MM) <input type="text" value="Escribe el apellido"/>		Hora de fin (HH:MM) <input type="text" value="Escribe el apellido"/>	Hora de inicio tarde (HH:MM) <input type="text" value="Escribe el apellido"/>
Sábado <input type="checkbox"/>	Hora de inicio (HH:MM) <input type="text" value="Escribe el apellido"/>		Hora de fin (HH:MM) <input type="text" value="Escribe el apellido"/>	Hora de inicio tarde (HH:MM) <input type="text" value="Escribe el apellido"/>
Domingo <input type="checkbox"/>	Hora de inicio (HH:MM) <input type="text" value="Escribe el apellido"/>		Hora de fin (HH:MM) <input type="text" value="Escribe el apellido"/>	Hora de inicio tarde (HH:MM) <input type="text" value="Escribe el apellido"/>

Figura 1.5: Interfaz actual de Gestoras - edición de gabinete 2/2.

## 1.4. APLICACIÓN ACTUAL EN FUNCIONAMIENTO

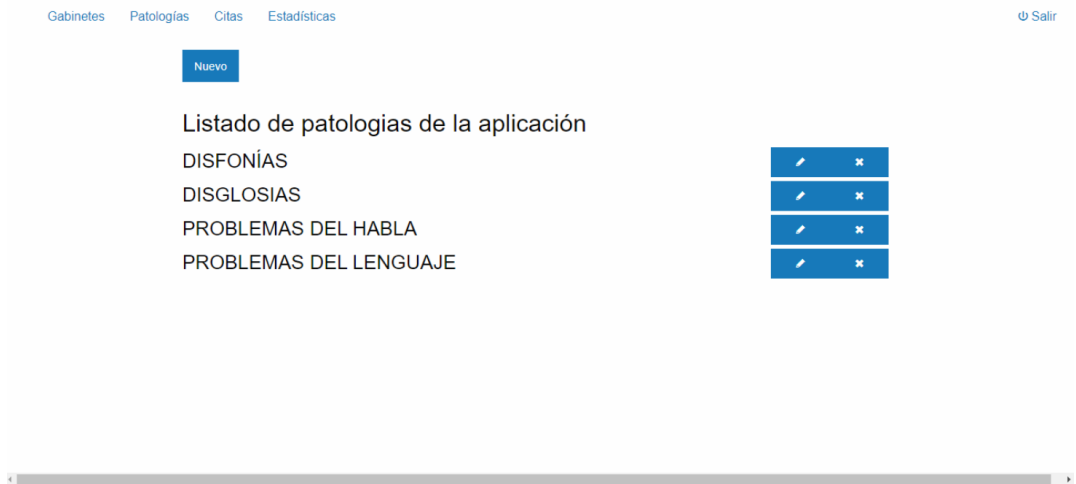


Figura 1.6: Interfaz actual de Gestoras - patologías.

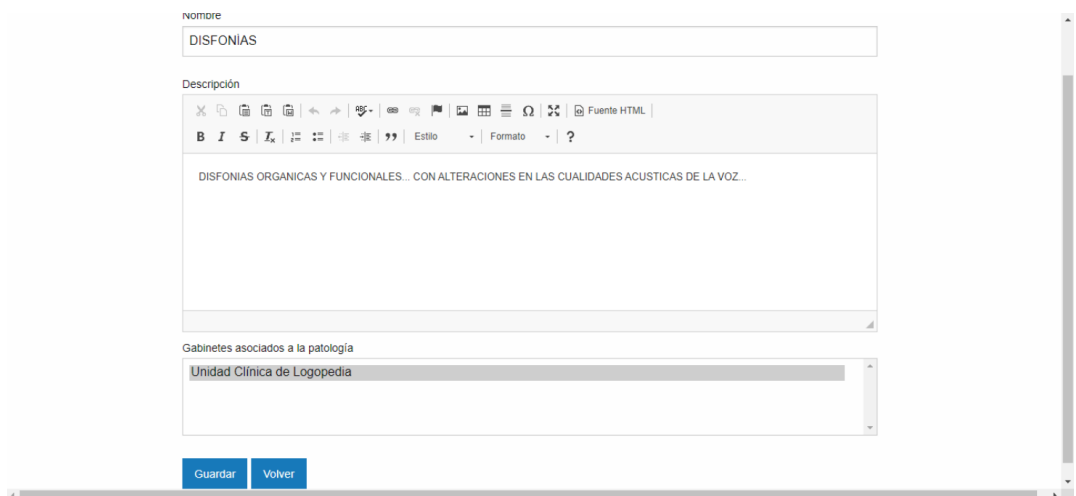


Figura 1.7: Interfaz actual de Gestoras - edición de patologías.

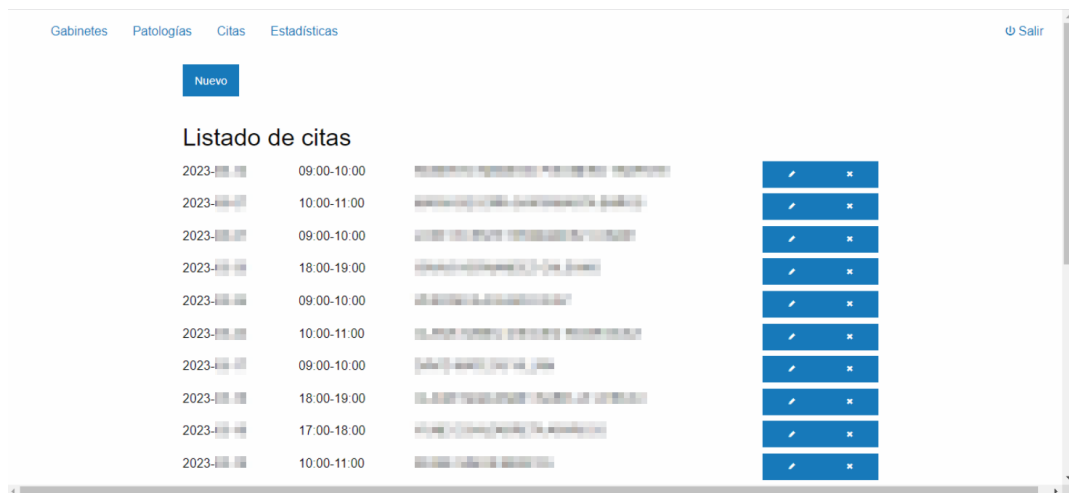


Figura 1.8: Interfaz actual de Gestoras - lista de próximas citas.

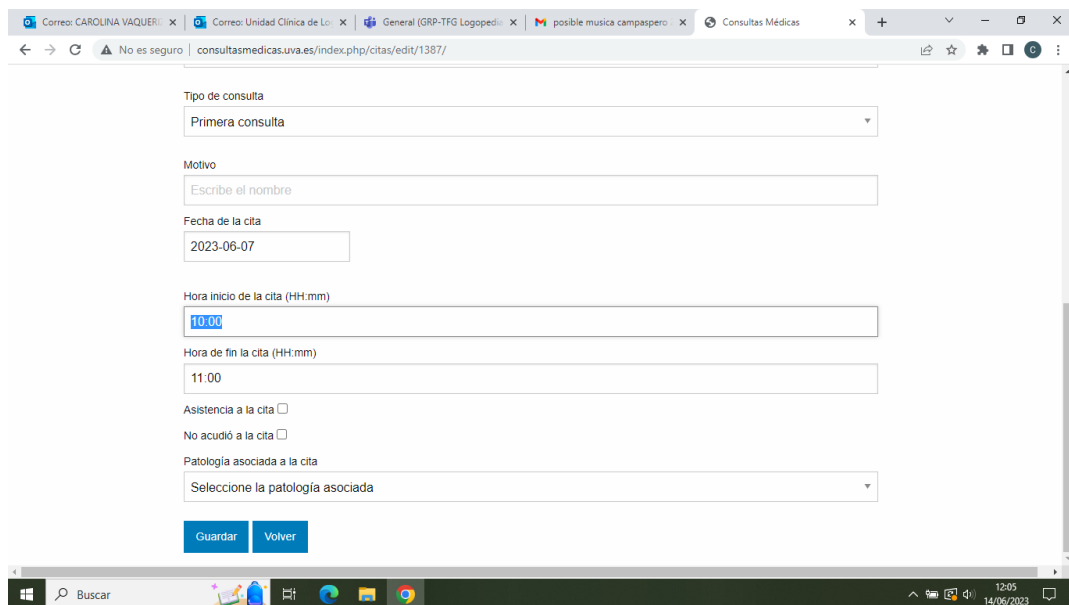


Figura 1.9: Interfaz actual de Gestoras - editar cita.

### 1.4.2. Interfaz para solicitantes

En este caso, el inicio de sesión se hace a través de un sistema de inicio de sesión centralizado de la UVa (Figura 1.10). Se trata de un Single Sign-On (en adelante, SSO); es un

proceso de inicio de sesión que habilita a un usuario para que pueda acceder a varios sistemas con un mismo método de verificación.

Una vez se ha iniciado sesión, se muestra una vista de inicio (Figura 1.11) con la información sobre cuando se puede pedir una cita, el aviso legal y un *banner* con las distintas opciones que podemos realizar:

- Inicio (Figura 1.11): Se muestra la página principal con los avisos legales correspondientes. Se trata de una página considerablemente larga y puede llegar a ser difícil de leer para el usuario.
- Solicitar cita (Figura 1.12): Se muestra una primera pantalla con un desplegable para seleccionar el gabinete deseado; al ser seleccionado, se ve la información del mismo. Este aspecto se podría mejorar, ya que son necesarios varios pasos para llegar a la información de contacto del gabinete.
  1. Al presionar sobre el botón de siguiente, se muestra la vista (Figura 1.13) con los días y horas disponibles. Cabe destacar que esta visualización de las horas y fechas podría resultar confusa.
  2. Al seleccionar la hora deseada y presionar en siguiente, se muestra la vista (Figura 1.14, se han pixelado algunos datos personales) con los campos de información necesarios a rellenar.
  3. Se rellenan los datos, se presiona en siguiente y se muestra un resumen de la cita (Figura 1.15).
  4. Para finalizar el proceso, se presiona en solicitar cita para que se formalice la solicitud y termine el proceso.
- Mis citas (Figura 1.16): Se muestra una vista con la información de la próxima cita y un botón de anular cita.
- Contacto (Figura 1.17): Se trata de una vista muy similar a la de la vista de inicio. Al igual que la página de inicio, se trata de una página considerablemente extensa.

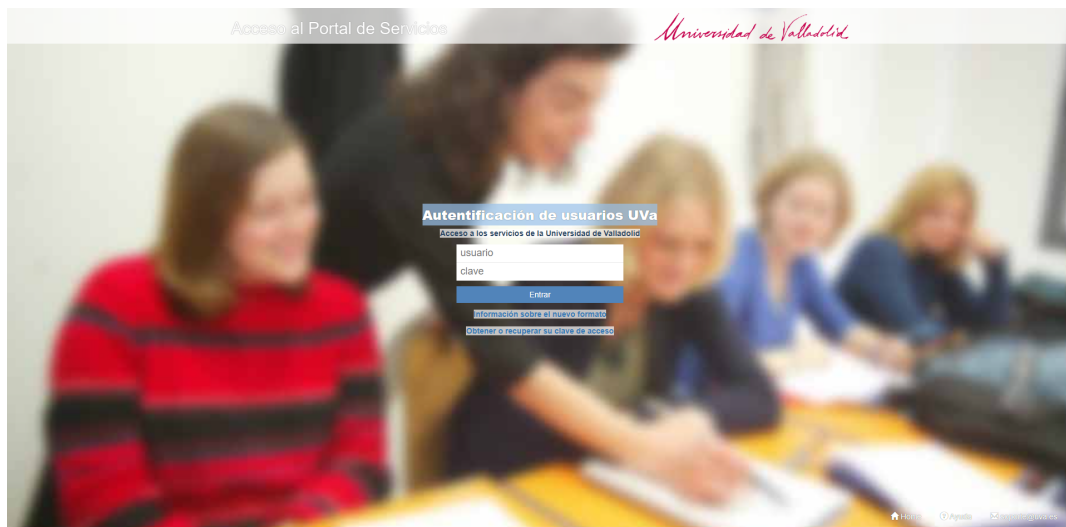


Figura 1.10: Interfaz actual de Solicitante - inicio de sesión.



Figura 1.11: Interfaz actual de Solicitante - inicio.

## 1.4. APLICACIÓN ACTUAL EN FUNCIONAMIENTO



Figura 1.12: Interfaz actual de Solicitante - Solicitar cita 1/4.

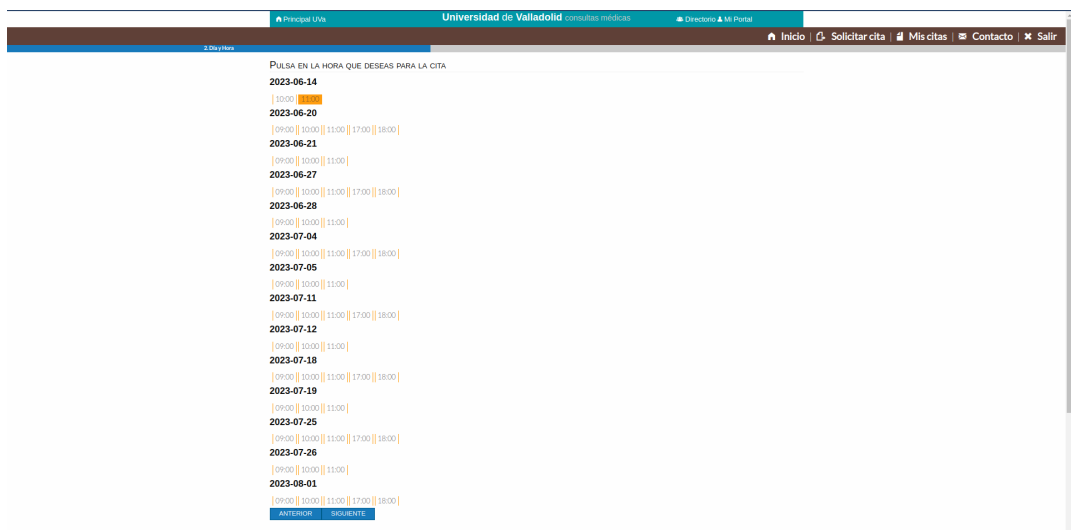


Figura 1.13: Interfaz actual de Solicitante - Solicitar cita 2/4.

The screenshot shows a web browser window with the URL 'consultas médicas' and navigation links: Inicio, Solicitar cita, Mis citas, Contacto, and Salir. The page title is '3. Datos de contacto'. The main heading is 'Rellene los siguientes datos'. The form contains the following fields:

- NIF/NIE/Pasaporte: [Redacted]
- Colectivo: Alumno (dropdown)
- Nombre y apellidos: Enrique Rojo Alvarez
- Teléfono de contacto: [Redacted]
- Correo electrónico de contacto: enrique.rojo@estudiantes.uva.es
- Género: Hombre (dropdown)
- Tipo de consulta: Primera consulta (dropdown)
- Motivo de la cita (opcional): Prueba TEG

Navigation buttons: ANTERIOR, SIGUIENTE

Footer navigation: Estudios, Te interesa, Descubre, Participa

Figura 1.14: Interfaz actual de Solicitante - Solicitar cita 3/4.

The screenshot shows a web browser window with the URL 'consultas médicas' and navigation links: Inicio, Solicitar cita, Mis citas, Contacto, and Salir. The page title is '4. Resumen de la cita'. The main heading is 'Resumen de la cita'. Below the heading, it says 'Compruebe la información y si está correcto pulse Solicitar cita'. The main content area is titled 'Unidad Clínica de Logopedia' and contains the following information:

- Localización: Facultad de Medicina, 3ª Planta (Ata izquierda), Avda. Ramón y Cajal s/n, 47305 Valladolid. Por favor, deben citarse 24h antes de la consulta, como mínimo y no realicen citaciones los días no lectivos de la Universidad.
- Correo: unidad.logopedia@uva.es
- Teléfono: 983423021 / 608838683
- Día: 2023-06-14
- Hora: 11:00

A map shows the location of the 'Unidad Clínica de Logopedia' at the 'Facultad de Medicina, Universidad de Valladolid'. Other landmarks include 'Residencia Universitaria Alfonso VIII', 'Estación de buses - BIK', 'Museo Anatómico (UVA)', 'Residencia Universitaria Monfermat', and 'Monasterio de Santa María la Real de...'. Map data is from 8/2023, Intel, Geogr. Nacional.

Navigation buttons: ANTERIOR, SOLICITAR CITA

Footer navigation: Estudios, Te interesa, Descubre, Participa

Footer content: 800 años de innovación, Sapientia Aedificavit, Sibi Domvum

Figura 1.15: Interfaz actual de Solicitante - Solicitar cita 4/4.

## 1.4. APLICACIÓN ACTUAL EN FUNCIONAMIENTO

The screenshot shows the 'mis citas' (my appointments) page for the 'Unidad Clínica de Logopedia'. The page header includes navigation links for 'Inicio', 'Solicitar cita', 'Mis citas', 'Contacto', and 'Salir'. The main content area displays the appointment date as '2023-06-14 de 11:00 a 12:00'. To the right, contact information is provided: 'Facultad de Medicina, 3ª Planta (Ala Izquierda), Avda. Ramón y Cajal s/n, 47005 Valladolid. Por favor, deben citarse 24h antes de la consulta, como mínimo y no realicen citaciones los días no lectivos de la Universidad.' The contact details include the phone number '983423021 / 608838683' and the email 'unidad.logopedia@uva.es'. A button labeled 'ANULAR CITA' is located at the bottom of the appointment card. Below the appointment card, there is a section with four columns of links: 'Estudios' (Estudios de Grados de la UVA, Estudios de Máster, Títulos Propios, Estudios de Doctorado), 'Te interesa' (Español en Valladolid, Cursos de Formación), 'Descubre' (Bibliotecas Universitarias, IEE, Parque Científico, Centro de Idiomas, Museo), and 'Participa' (Deportes, Centro Buena Vista, Gabinete Médico, Asociaciones). To the right of these links is a logo for '800 años de innovación Sapientia Aedificavit Sibi Domvum'. At the bottom, a footer contains the text: 'UNIVERSIDAD DE VALLADOLID // Palacio de Santa Cruz, 47002 Valladolid (España)' and a disclaimer: 'Los contenidos suministrados por la web están sujetos a los derechos de propiedad intelectual e industrial y son titularidad exclusiva de Universidad de Valladolid. La explotación de algún prestador de servicios no confiere al abastecedor ningún derecho de alteración, explotación, reproducción o distribución del mismo fuera de lo estrictamente controlado reservándose Universidad de Valladolid todos los derechos.'

Figura 1.16: Interfaz actual de Solicitante - mis citas.

The screenshot shows the 'contacto' (contact) page for the University of Valladolid. The page header includes navigation links for 'Inicio', 'Solicitar cita', 'Mis citas', 'Contacto', and 'Salir'. The main content area features the title 'Universidad de Valladolid' and a paragraph: 'La Universidad de Valladolid pone a disposición de estudiantes, profesorado y personal la Unidad Clínica de Logopedia, donde varios Logopedas de la UVA, con una amplia experiencia asistencial, realizarán labores de evaluación y tratamiento de pacientes adultos, ancianos y/o niños, con trastornos del lenguaje, habla, voz, audición, deglución y otros problemas logopédicos. Podrán acceder a la Unidad Clínica de Logopedia y solicitar sus servicios de forma gratuita, los siguientes miembros de la Comunidad Universitaria: PDI, PAS, cónyuges, hijos o padres del PDI y/o PAS, así como cualquier estudiante universitario.' Below this, it states: 'Se trata de un servicio gratuito, anónimo y confidencial cuyo objetivo es mejorar la salud foniátrica de las personas de la Comunidad Universitaria de la UVA.' A note follows: 'Para recibir consulta, la persona tiene que haber tramitado la solicitud de cita con al menos 24 horas de antelación al día solicitado. No se permite formalizar cita en día festivo, puente, o en periodo vacacional en la UVA.' The page is divided into sections: 'Aviso Legal', 'INFORMACIÓN SOBRE PROTECCIÓN DE DATOS' (El sistema permite elegir el día y la hora de consulta. Las consultas son totalmente gratuitas para los miembros de la comunidad universitaria (estudiantes, profesorado y personal).), '¿Con qué finalidad recabamos tu información personal?' (Trataremos tus datos de identificación con la finalidad de prestarle labores de evaluación y tratamiento de pacientes adultos, ancianos y/o niños, con trastornos del lenguaje, habla, voz, audición, deglución y otros problemas logopédicos.), '¿Quién es el responsable del tratamiento?' (Universidad de Valladolid. CIF: Q4718001C. Domicilio: Plaza del Colegio de Santa Cruz, 8 - 47002 Valladolid), '¿Cómo puedes contactar con nuestro delegado de protección de datos?' (El delegado de protección de datos es la persona encargada de supervisar que cumplimos las normas sobre protección de datos y ayudarte. Si tienes alguna duda o consulta sobre cómo tratamos los datos puedes contactar con el delegado de protección de datos en delegado.proteccion.datos@uva.es), and '¿Con qué derecho o base legal trata la Universidad de Valladolid tu información personal?' (La Universidad de Valladolid trata los datos en virtud del consentimiento de la

Figura 1.17: Interfaz actual de Solicitante - contacto.

### 1.4.3. Limitaciones

En su mayoría, las limitaciones son relativas a las funciones de las gestoras del Gabinete de Logopedia. Aquí se enumeran:

- No se almacena toda la información sobre pacientes/solicitantes que requieren las ges-



toras para su trabajo.

- Las gestoras describen algunas restricciones necesarias como “No se puede reservar cita 24 horas antes”, que no están realmente implementadas, se dejan a confianza del solicitante para que las cumpla, ya que el sistema le permite reservar con menos de 24 horas de antelación.
- Cuando ocurre el caso de que un miembro de la comunidad UVa ha reservado una cita que no es para sí mismo/a (por ejemplo, un familiar), las gestoras no tienen forma directa de saberlo. Esto dificulta su trabajo a la hora de organizarse y atender dicha cita.
- La información de las citas y solicitantes no queda almacenada dentro de la aplicación actual. Sin embargo, se guarda en un documento de Microsoft Excel que gestionan las profesionales del Gabinete de Logopedia, lo cual no cumple con el RGPD, ya que se trata de un documento sin ningún tipo de protección al que podría acceder cualquier persona.
- Solo hay una cuenta de usuario común para entrar con el rol gestora.

Las limitaciones que tienen las personas que solicitan cita no son tantas como las de las gestoras, pero también existen algunas:

- Cuando alguien quiere reservar una cita para otra persona distinta, la aplicación actual no lo contempla, se deja a confianza de la persona que solicita que detalle la información de la otra persona en el campo “Motivo de la cita (opcional)”. Aun así, en la aplicación actual no hay nada que indique que para estos casos se debe de actuar de esa forma.

Finalmente, estas son algunas limitaciones y carencias de tipo general que posee la aplicación actual:

- No posee un diseño *responsive*, actualmente no se visualiza correctamente en teléfonos móviles ni tablets.
- Se podría dar el caso de que dos personas distintas pidan una cita a la vez para un mismo día y hora.
- La tecnología utilizada (parte de ella está escrita en PHP) necesita actualizarse, ya que para que la aplicación pueda mantenerse deberá de hacer uso de las tecnologías utilizadas actualmente por la UVa.

Cabe destacar que a simple vista la mayoría de limitaciones son de funcionalidades, sobre lo que la aplicación permite hacer y no hacer para los usuarios, pero el grueso del proyecto y la razón más importante por la que se requiere rehacer la aplicación desde cero, es la actualización de las tecnologías en las que está desarrollada la aplicación y el necesario cumplimiento del RGPD y el ENS. Para llevarlo a cabo se requerirá de desarrollar un back-end robusto, que cumpla con el RGPD y tenga medidas de seguridad adecuadas, las cuales necesitan estar integradas desde el inicio del desarrollo de la aplicación web.

### 1.5. Objetivos

El objetivo de este proyecto se centra en el desarrollo de una aplicación web que permita la gestión de citas para el Gabinete de Logopedia de la Universidad de Valladolid.

Dentro de la aplicación web se declaran los siguientes puntos como alcance del proyecto, siendo las funcionalidades principales:

- Solicitud de citas por parte de los miembros de la comunidad universitaria.
- Facilitar a los solicitantes la consulta de sus citas.
- Gestión de citas.
- Facilitar la diferenciación entre paciente y solicitante.
- Prevenir malentendidos y desinformación de la cita entre solicitante y administrador
- Mantener almacenados los datos de los pacientes, solicitantes y citas, teniendo en cuenta un diseño e implementación que facilite la privacidad y la seguridad.
- Adaptable a las pantallas de ordenadores y teléfonos móviles.

Aparte de estas funcionalidades, el objetivo prioritario es que la aplicación sea implantada dentro del servicio de la Universidad de Valladolid, permitiendo el uso por parte de la comunidad y personal perteneciente a la UVa. Para lograr el cometido se declaran los siguientes objetivos:

- Diseño intuitivo que facilite la navegación por la aplicación.
- Aplicación robusta y estable, sin errores que entorpezcan el uso diario.
- Garantizar la seguridad de los datos personales.
- Un código de programación y documentación claro, y entendible para facilitar su mantenimiento y/o mejoras futuras.

### 1.6. Estructura de la memoria

Esta memoria está dividida en varios capítulos. Su estructura es la siguiente:

**Capítulo 1 Introducción:** Se hace una introducción al proyecto, empezando con un contexto, su motivación, una explicación de la aplicación web actual junto a sus limitaciones, los objetivos y el presente punto de estructura de memoria.

**Capítulo 2 Requisitos y planificación:** Se definen las partes interesadas del proyecto, se elicitan todos los requisitos y casos de uso, se describe un plan, se hace un análisis de riesgos, presupuestos y finalmente cuáles son las metodologías utilizadas para el desarrollo.

**Capítulo 3 Análisis:** Se detalla el modelo de casos de uso, modelo de dominio, diagrama de actividad y máquinas de estados.

**Capítulo 4 Tecnologías utilizadas:** En esta parte se detallan cuáles son las herramientas y tecnologías que han sido utilizadas para desarrollar el proyecto.

**Capítulo 5 Diseño:** Se detalla el diseño de la aplicación: bocetaje de la interfaz, la arquitectura lógica y el diseño de despliegue del sistema.

**Capítulo 6 Implementación y pruebas:** Se muestra como se ha implementado el proyecto, las dificultades, decisiones tomadas que difieren del diseño, y las pruebas realizadas para comprobar su correcto funcionamiento.

**Capítulo 7 Seguimiento del proyecto:** Se explica como ha sido el transcurso del proyecto, según como se ha llevado la planificación inicial y como evolucionó.

**Capítulo 8 Conclusiones:** Se da una valoración final en cuanto al trabajo realizado, el aprendizaje adquirido, y se exponen las líneas futuras del proyecto.



## Capítulo 2

# Requisitos y Planificación

### 2.1. Stakeholders y usuarios

Se detallan los principales interesados en el proyecto y los usuarios finales que interactuarán con la aplicación web. Una vez definidos, se enumeran los requisitos recogidos.

#### 2.1.1. Stakeholders

Los stakeholders son los principales interesados en el proyecto, ellos pueden imponer necesidades y requisitos para la aplicación. Por lo que es muy importante identificarlos y escuchar sus necesidades.

Las logopedas del Gabinete de Logopedia de la UVa son las primeras interesadas en el presente proyecto, ya que fueron las personas que llegaron con la necesidad de una renovación de su aplicación web.

Los tutores de este TFG: Yania Crespo y Alfonso Población, quienes monitorizarán el desarrollo del mismo.

Los profesionales del Servicio de Tecnologías de la Información y las Comunicaciones (STIC) de la UVa, dado que la aplicación web podría ser alojada en los servidores de la UVa, se deberán de cumplir ciertos requisitos que ellos impondrán para que finalmente pueda ser implantada.

El responsable de privacidad de la UVa, al tratarse datos personales y relacionados con la salud, será de vital importancia poner énfasis en la seguridad y privacidad de la aplicación. Se deberán seguir sus indicaciones en todo el desarrollo y despliegue.

Los pacientes y solicitantes, ya que interactuarán con el sistema una vez este completado. El estudiante y tutores son una representación de este colectivo.

### 2.1.2. Usuarios finales

Los usuarios finales son aquellos que utilizarán la aplicación una vez esté finalizada y desplegada. Entre ellos se encuentran las logopedas del Gabinete de Logopedia de la UVa. Debido a que utilizarán el sistema en la parte de gestionar las citas, de aquí en adelante y de cara al sistema se las llamará gestoras. Los solicitantes se refiere al tipo de usuarios del sistema que utilizarán la parte de solicitar las citas. Dentro de ellos se distinguen: estudiantes, PDI y PAS. Durante todo el desarrollo del proyecto se habla del tipo de solicitante PDI/PAS [43]. Se trata del personal trabajador en la UVa: Personal Docente e Investigador y Personal de Administración y Servicios, respectivamente.

## 2.2. Elicitación de requisitos

La elicitación de requisitos se realizó fundamentalmente a partir de dos fuentes: entrevistas con los stakeholders y revisión de la aplicación existente. A continuación se especifica la lista de requisitos obtenidos.

### 2.2.1. Requisitos funcionales

- RF1 El sistema deberá permitir a los usuarios iniciar sesión.
- RF2 El sistema deberá permitir a los solicitantes solicitar una cita para ellos mismos como pacientes (ver RN4).
- RF3 El sistema deberá permitir a algunos solicitantes solicitar una cita para otra persona, la cual será el paciente (ver RN1) (ver RN2) (ver RN4).
- RF4 El sistema deberá permitir a los solicitantes consultar sus citas.
- RF5 El sistema deberá permitir a los solicitantes consultar información de contacto del gabinete.
- RF6 El sistema deberá permitir a los solicitantes cancelar una cita.
- RF7 El sistema deberá permitir a las gestoras visualizar sus citas y las de todas las gestoras.
- RF8 El sistema deberá permitir a las gestoras cancelar citas.
- RF9 El sistema deberá permitir a las gestoras modificar la fecha y/u hora de una cita.
- RF10 El sistema deberá permitir a las gestoras consultar la información de los pacientes y solicitantes.
- RF11 El sistema deberá permitir a las gestoras consultar el horario de disponibilidad (se trata del horario en el que las gestoras estarán disponibles para atender citas).
- RF12 El sistema deberá permitir a las gestoras modificar la información de una o varias fechas concretas del horario de disponibilidad (ver RN6) (ver RN8).

- RF13 El sistema deberá permitir a las gestoras bloquear franjas de días completos, un día completo, franjas de horas o una sola hora, en los que el sistema no podrá aceptar citas para esos días (ver RN5) (ver RN6).
- RF14 El sistema deberá de enviar un e-mail a las gestoras cada vez que se recibe una nueva cita.
- RF15 El sistema deberá de enviar un e-mail a las gestoras cuando un solicitante cancela una cita.
- RF16 El sistema deberá de enviar un e-mail a los pacientes cuando una gestora cancele o modifique una cita.
- RF17 El sistema deberá permitir a las gestoras modificar el estado de una cita.

### 2.2.2. Requisitos de información

- RI1 El sistema deberá permitir, para cada solicitante, guardar la siguiente información: NIF/DNI, nombre, apellidos, colectivo UVa (ver RI5), edad, dirección postal, teléfono móvil y e-mail.
- RI2 El sistema deberá permitir, para cada cita guardar la siguiente información: estado, tipo de consulta, motivo de consulta, especificación del motivo de consulta, información del paciente en el caso de que no sea el solicitante y en caso de consultas de tipo revisión, un texto libre con información relacionada sobre la persona que les atendió anteriormente.
- RI3 El sistema deberá de contemplar los distintos estados de las citas: prerreservada, bloqueada, cancelada, pendiente, realizada y no presentado.
- RI4 El sistema deberá permitir, para cada paciente, guardar la siguiente información: nombre, apellidos, edad, dirección postal, teléfono móvil y e-mail.
- RI5 El sistema contemplará 3 tipos de colectivo UVa: PAS, PDI y estudiante.
- RI6 El sistema contemplará 2 tipos de consulta: 1<sup>a</sup> consulta y revisión (ver RN7).
- RI7 El sistema contemplará 6 tipos de motivos de consulta: lenguaje, habla, voz, audición, deglución y otros.
- RI8 El sistema almacenará la siguiente información para el horario de disponibilidad: un conjunto de fechas y por cada fecha: hora de inicio, hora de finalización y duración de los 2 tipos de consulta.
- RI9 El sistema almacenará la siguiente información del horario de disponibilidad semanal predeterminado: día de la semana, duraciones de primera consulta y revisión, horas de inicio, partida, segunda de inicio y final.
- RI10 El sistema deberá de contemplar los distintos estados para cada día: libre o bloqueado.
- RI11 El sistema deberá guardar la siguiente información de contacto del gabinete: e-mail, 2 números de teléfono y una extensión para teléfonos móviles.

### 2.2.3. Reglas de negocio

Se identificaron las siguientes reglas de negocio que regulan algunas operaciones:

- RN1 Los solicitantes que pertenecen al colectivo de estudiantes no pueden solicitar una cita para otra persona.
- RN2 Los solicitantes que pertenecen al colectivo PAS o PDI pueden solicitar una cita para otra persona que deberá ser un familiar de primer grado.
- RN3 Los solicitantes solo podrán tener una cita vigente a la vez por paciente.
- RN4 Las citas deben de solicitarse con una antelación mínima de 48 horas.
- RN5 El sistema debe bloquear automáticamente los días de vacaciones impuestos por el calendario PAS de la UVa.
- RN6 En el caso de que ya hubiese citas dentro de los días/horas a bloquear y la modificación del horario de disponibilidad, el sistema deberá de alertar al usuario para que cambie la fecha de esas citas, hasta su modificación, esos días y las citas quedarán bloqueados/modificados.
- RN7 El sistema debe de comprobar que, para solicitar una cita de tipo revisión, antes debe de haberse solicitado una de tipo primera consulta.
- RN8 La duración de las citas de tipo primera consulta y revisión, debe de estar comprendida de 15 a 120 minutos y 10 a 60 minutos, respectivamente.

### 2.2.4. Requisitos no funcionales

Para obtener estos requisitos no funcionales se utiliza un framework conocido por el acrónimo de FURPS+ [31]. Está compuesto por las palabras en inglés: funcionalidad, usabilidad, fiabilidad, rendimiento, soporte y el símbolo +, el cual amplía las categorías que lo siguen para tener en cuenta: requisitos de diseño, implementación, interfaz y físicos. Para contar con unos requisitos no funcionales completos se analiza cada una de las categorías (a excepción de la funcionalidad, la cual ya ha sido cubierta en los requisitos funcionales en el apartado 2.2.1). Además de ello, se especifican algunas exigencias que el responsable de privacidad y los técnicos de seguridad del STIC comentaron para que la aplicación pueda ser implantada dentro de la red de la UVa.

- RNF1 El sistema deberá de permitir realizar las acciones habituales de solicitar una cita en menos de 5 minutos.
- RNF2 El sistema deberá de tener filtros y ordenaciones a la hora de hacer consultas.
- RNF3 El sistema deberá de tener una interfaz, al menos, en el idioma español.



- RNF4 El sistema deberá, en caso de ocurrir un error para el usuario, informarlo en todo momento por medio de la interfaz.
- RNF5 El sistema deberá, en caso de ocurrir un error para el desarrollador, informarlo en todo momento por medio de *logs*.
- RNF6 El sistema deberá de ser capaz de recibir peticiones simultáneas por parte de, al menos, cinco usuarios.
- RNF7 El sistema deberá de dar respuestas en menos de cinco segundos y evitar posibles esperas al hacer peticiones.
- RNF8 El sistema deberá de reducir sus tiempos de mantenimiento a horarios con la mínima demanda, por lo menos *00:00-09:00*.
- RNF9 El sistema deberá de ser desplegado, como máximo, con tres comandos y en menos de dos minutos.
- RNF10 El sistema deberá de producir *logs* para poder comprobar su funcionamiento.
- RNF11 El sistema deberá de recibir revisiones de seguridad y de código por parte del personal técnico y de seguridad.
- RNF12 El sistema deberá de estar escrito en un código lo más entendible posible, tomando como referencia las revisiones por parte del personal técnico y de seguridad.
- RNF13 El sistema deberá de garantizar la privacidad y seguridad de los datos personales de todos los usuarios de acuerdo con [11] [12] [13].
- RNF14 El sistema deberá de tener un front-end desarrollado en HTML5, sin sobrecargar la interfaz.
- RNF15 El sistema deberá de tener alguna librería de estilo para lograr una interfaz más intuitiva.
- RNF16 El sistema deberá de tener un back-end desarrollado en Spring Boot.
- RNF17 El sistema deberá de tener el back-end contenerizado con Docker.
- RNF18 El sistema deberá de tener una herramienta de gestión automática de dependencias.
- RNF19 El sistema deberá de tener un front-end adaptable a las pantallas de dispositivos móviles, tablets y ordenadores.

### 2.2.5. Casos de uso

En esta parte se detallan todos los casos de uso, cada uno de ellos describe el flujo normal y las posibles bifurcaciones de una acción dentro del sistema.

Se considera como escenario con éxito los que siguen la secuencia principal, los pasos seguidos de un paréntesis y una letra “2(a)” son los escenarios alternativos y excepciones.

### CU1 Caso de uso: Identificarse

Actor: Usuario

Precondición: El actor no debe de estar identificado.

Secuencia principal:

- 1- El actor selecciona iniciar sesión.
- 2- El sistema solicita identificador y contraseña.
- 3- El actor introduce identificador y contraseña.
- 4- El sistema comprueba la existencia de un usuario con el identificador y contraseña, entonces muestra las opciones disponibles según el tipo de usuario.

Bifurcaciones:

- 1(a),3(a)- El actor cancela y el caso de uso queda sin efecto.
- 4(a)- El sistema comprueba que no existe ningún usuario con el identificador y contraseña, entonces muestra un mensaje de error de acceso incorrecto y vuelve al paso 1.

Postcondición: El usuario está identificado en el sistema.

### CU2 Caso de uso: Solicitar cita

Actor: Solicitante y servicio e-mail.

Precondición: El actor debe de estar identificado.

Secuencia principal:

- 1- El actor selecciona solicitar cita.
- 2- El sistema muestra las fechas, los tipos de consulta y las horas disponibles.
- 3- El actor selecciona la fecha, tipo de consulta y la hora que desea.
- 4- El sistema valida los datos, crea una cita con los datos introducidos y el estado: prerreservada.
- 5- El sistema muestra la información del solicitante y comprueba el colectivo UVa, si es "Estudiante" no muestra ninguna información más.
- 6- El actor selecciona continuar.
- 7- El sistema valida los datos y almacena la información del solicitante como paciente.
- 8- El sistema solicita la información acerca de la cita, en el caso de primera consulta: Motivo de consulta y más información acerca del motivo de consulta.
- 9- El actor introduce todos los datos.
- 10- El sistema comprueba la validez de los datos, actualiza la información de la cita y cambia su estado a Pendiente.
- 11- El servicio de e-mail envía un mensaje a las gestoras con el aviso de una nueva cita.

Bifurcaciones:

- 1(a), 3(a), 6(a), 9(a)- El actor cancela y el caso de uso queda sin efecto.
- 4(a), 7(a), 10(a)- El sistema comprueba que los datos son erróneos, muestra un mensaje de error y vuelve al paso anterior.

5(a)- El sistema valida los datos, muestra la información del solicitante y comprueba el colectivo UVa, si es “PDI/PAS” se inicia el CU3 y después se continúa en el paso 8.

7(b)- En caso de revisión, el sistema comprueba si existe una primera consulta previa, en caso de no existir, se muestra un aviso.

8(a)- El sistema solicita la información acerca de la cita, en el caso de revisión: Quién le atendió en la anterior cita, motivo de consulta y más información acerca del motivo de consulta.

Postcondición: Se ha registrado una nueva cita, un paciente y se ha enviado un e-mail.

### CU3 Caso de uso: Solicitante - Añadir información de paciente.

Actor: Solicitante

Precondición: El actor debe de estar identificado y tener colectivo UVa de PDI/PAS.

Secuencia principal:

1- El sistema muestra la información del solicitante y una opción para solicitar la cita para otra persona.

2- El actor selecciona la opción de pedir cita para otra persona.

3- El sistema solicita el nombre, apellidos, dirección postal, localidad, número de teléfono, email y fecha de nacimiento

4- El actor introduce los datos solicitados.

5- El sistema valida los datos y almacena la información del paciente.

Bifurcaciones:

2(a), 5(a)- El actor cancela y el caso de uso queda sin efecto.

2(b)- El actor responde que no, el caso de uso termina y continua en el caso de uso anterior, en el paso 7.

3(a)- En caso de que el solicitante tenga pacientes creados, el sistema los muestra y presenta la opción de seleccionarlos y modificarlos.

5(b)- En caso de revisión, el sistema comprueba si existe una primera consulta previa, en caso de no existir, se muestra un aviso.

5(c)- El sistema comprueba que los datos son erróneos, muestra un mensaje de error y vuelve al paso anterior.

5(d)- En caso de haber seleccionado un paciente, el sistema comprueba que los datos son válidos y actualiza el paciente existente.

Postcondición: Se ha registrado un paciente.

### CU4 Caso de uso: Solicitante - Consultar citas.

Actor: Solicitante.

Precondición: El actor debe de estar identificado.

Secuencia principal:

1- El actor selecciona consultar citas.

2- El sistema muestra la cita o las citas que el actor ha solicitado (en estado pendiente).

3- El actor selecciona una cita.

4- El sistema muestra, en caso de tipo de consulta “revisión”, la información acerca de la cita: estado, tipo de consulta, motivo de consulta, especificación del motivo de consulta e información del paciente. También se muestra la información relacionada sobre la persona que les atendió anteriormente, y la información del solicitante y paciente.

5- El actor selecciona volver a atrás.

6- El sistema vuelve a mostrar la cita o las citas que el actor ha solicitado (en estado pendiente)

7- El actor selecciona filtrar citas por estado, seleccionando uno o varios entre cancelada, pendiente, realizada y no presentado. También pudiendo seleccionar el orden según la fecha.

8- El sistema muestra las citas según el estado seleccionado.

Bifurcaciones:

1(a), 3(a), 5(a), 7(a)- El actor cancela y el caso de uso queda sin efecto.

4(a)- El sistema, en caso de tipo de consulta “1<sup>a</sup> consulta”, muestra la información acerca de la cita: estado, tipo de consulta, motivo de consulta, especificación del motivo de consulta, información del solicitante y paciente.

5(a)- El actor selecciona cancelar cita, se inicia el caso de uso 5.

Postcondición: -

### CU5 Caso de uso: Solicitante - Cancelar cita

Actor: Solicitante y servicio de e-mail.

Precondición: El actor debe de estar identificado.

Secuencia principal:

1- El actor selecciona cancelar cita

2- El sistema valida que la cita puede ser cancelada.

4- El sistema actualiza el estado de la cita a “cancelado”.

5- El servicio de e-mail envía un mensaje a las gestoras avisando de la cancelación de una cita.

Bifurcaciones:

1(a)- El actor cancela y el caso de uso queda sin efecto.

Postcondición: Se ha actualizado el estado de una cita y enviado un correo electrónico.

### CU6 Caso de uso: Solicitante - Consultar información de contacto del gabinete.

Actor: Solicitante.

Precondición: El actor debe de estar identificado.

Secuencia principal:

1- El actor selecciona consultar la información de contacto del gabinete.

2- El sistema muestra el e-mail, los números de teléfono y la extensión para teléfonos móviles.

Bifurcaciones:

1(a)- El actor cancela y el caso de uso queda sin efecto.

**CU7 Caso de uso: Gestora - Consultar citas de todas las gestoras.**

Actor: Gestora.

Precondición: El actor debe de estar identificado.

Secuencia principal:

1- El actor selecciona consultar citas.

2- El sistema muestra todas las citas (en estado pendiente).

3- El actor selecciona filtrar las citas por estado, seleccionando uno o varios entre: bloqueada, cancelada, pendiente, realizada y no presentado. También pudiendo seleccionar el orden según la fecha.

4- El sistema muestra las citas según el estado seleccionado.

5- El actor selecciona una cita.

6- El sistema muestra, en caso de tipo de consulta “revisión”, la información acerca de la cita: estado, tipo de consulta, motivo de consulta, especificación del motivo de consulta e información del paciente. También se muestra la información relacionada sobre la persona que les atendió anteriormente, y la información del solicitante y paciente.

7- El actor selecciona volver a atrás, se vuelve al paso 3.

Bifurcaciones:

1(a), 3(a), 5(a), 7(a)- El actor cancela y el caso de uso queda sin efecto.

6(a)- El sistema, en caso de tipo de consulta “1ª consulta”, muestra la información acerca de la cita: estado, tipo de consulta, motivo de consulta, especificación del motivo de consulta, e información del paciente y solicitante.

7(a)- El actor selecciona cancelar cita, se inicia el caso de uso 8.

7(b)- El actor selecciona modificar fecha y hora cita, se inicia el caso de uso 9.

7(c)- El actor selecciona un estado y pulsa confirmar estado, se inicia el caso de uso 10.

**CU8 Caso de uso: Gestora - Cancelar cita de solicitante.**

Actor: Servicio de e-mail.

Precondición: El actor debe de estar identificado.

Secuencia principal:

1- El actor selecciona cancelar cita

2- El sistema valida que la cita puede ser cancelada.

4- El sistema actualiza el estado de la cita a “cancelado”.

5- El servicio de e-mail envía un mensaje al solicitante y paciente avisando de la cancelación de una cita.

Bifurcaciones:

1(a)- El actor cancela y el caso de uso queda sin efecto.

Postcondición: Se ha actualizado el estado de una cita y enviado un correo electrónico.

**CU9 Caso de uso: Gestora - Modificar fecha u/y hora de una cita.**

Actor: Gestora y servicio de e-mail.

Precondición: El actor debe de estar identificado.

Secuencia principal:

- 1- El sistema pide la nueva fecha y hora.
- 2- El actor modifica los campos disponibles: fecha y hora.
- 3- El sistema comprueba que los datos sean válidos y actualiza la fecha, hora y estado a "Pendiente" de la cita.
- 4- El servicio de e-mail envía un mensaje al solicitante y paciente avisando de la modificación de una cita.

Bifurcaciones:

- 2(a)- El actor cancela y el caso de uso queda sin efecto.
- 3(a)- El sistema comprueba que los datos son inválidos, avisa al actor y vuelve al paso 1.

Postcondición: Se ha actualizado el estado de una cita y se ha enviado un e-mail al paciente.

**CU10 Caso de uso: Gestora - Cambiar estado de cita.**

Actor: Gestora.

Precondición: El actor debe de estar identificado.

Secuencia principal:

- 1- El sistema valida que el estado es uno de los siguientes: pendiente, realizada y no presentado.
- 2- El sistema actualiza el estado de la cita

Bifurcaciones:

- 1(a)- El sistema comprueba que el estado no es válido. Termina el caso de uso.

Postcondición: Se ha actualizado el estado de una cita.

**CU11 Caso de uso: Gestora - Consultar su horario de disponibilidad.**

Actor: Gestora.

Precondición: El actor debe de estar identificado.

Secuencia principal:

- 1- El actor selecciona consultar su horario de disponibilidad.
- 2- El sistema muestra la información del horario de disponibilidad de la semana actual.
- 3- El actor selecciona otra semana.
- 4- El sistema actualiza la semana a mostrar y vuelve al paso 1.

Bifurcaciones:

- 1(a), 3(a)- El actor cancela y el caso de uso queda sin efecto.
- 3(b)- Si el actor selecciona la opción, modificar horario, se inicia el caso de uso CU12.

**CU12 Caso de uso: Gestora - Modificar el horario de disponibilidad.**

Actor: Gestora.

Precondición: El actor debe de estar identificado.

Secuencia principal:

- 1- El actor selecciona modificar el horario de disponibilidad.
- 2- El sistema solicita, para cada día de la semana: hora de inicio, finalización y la duración de los 2 tipos de consulta.
- 3- El actor introduce los nuevos valores.
- 4- El sistema comprueba que los datos sean válidos, los almacena y desbloquea o bloquea las citas convenientes según el nuevo horario.

Bifurcaciones:

- 1(a), 3(a)- El actor cancela y el caso de uso queda sin efecto.
- 4(a)- El sistema comprueba que los datos son inválidos, muestra un mensaje de error y vuelve al paso 1.

Postcondición: Se han actualizado los datos del horario de disponibilidad.

**CU13 Caso de uso: Gestora - Bloquear días.**

Actor: Gestora.

Precondición: El actor debe de estar identificado.

Secuencia principal:

- 1- El actor selecciona bloquear días.
- 2- El sistema pide una fecha de inicio y otra de fin.
- 3- El actor introduce las fechas.
- 4- El sistema comprueba que los datos sean válidos, cambia el estado de los días ha bloqueado y bloquea las citas que había en los días bloqueados.

Bifurcaciones:

- 1(a), 3(a)- El actor cancela y el caso de uso queda sin efecto.
- 4(a)- El sistema comprueba que los datos son inválidos, muestra un mensaje de error y vuelve al paso 1.

Postcondición: Se han actualizado los datos del horario de disponibilidad y bloqueado las citas convenientes.

**CU14 Caso de uso: Gestora - Bloquear horas.**

Actor: Gestora.

Precondición: El actor debe de estar identificado.

Secuencia principal:

- 1- El actor selecciona bloquear horas.
- 2- El sistema solicita una fecha, hora de inicio y hora de fin.
- 3- El actor introduce los datos.

## 2.2. ELICITACIÓN DE REQUISITOS

---

4- El sistema comprueba que los datos sean válidos, cambia el estado de esas horas a bloqueado y bloquea las citas que están dentro de las horas.

Bifurcaciones:

1(a), 3(a)- El actor cancela y el caso de uso queda sin efecto.

4(a)- El sistema comprueba que los datos son inválidos, muestra un mensaje de error y vuelve al paso 1.

Postcondición: Se han actualizado los datos del horario de disponibilidad y bloqueado las citas convenientes.

### CU15 Caso de uso: Gestora - Consultar calendario.

Actor: Gestora.

Precondición: El actor debe de estar identificado.

Secuencia principal:

1- El actor selecciona nuestro calendario.

2- El sistema muestra un calendario con el mes actual, el calendario muestra los días bloqueados y los libres.

3- El actor selecciona vista anual.

4- El sistema muestra los calendarios de todos los meses del año actual, cada calendario muestra los días bloqueados y los libres.

Bifurcaciones:

1(a), 3(a)- El actor cancela y el caso de uso queda sin efecto.

A partir del listado de requisitos funcionales y la especificación de casos de uso, se realiza la matriz de correspondencia entre los mismos. La siguiente tabla representa dicha matriz (Figura 2.1), lo que nos permite comprobar que todos los requisitos funcionales están cubiertos en algún caso de uso.



	CU1	CU2	CU3	CU4	CU5	CU6	CU7	CU8	CU9	CU10	CU11	CU12	CU13	CU14	CU15
RF1	X														
RF2		X													
RF3			X												
RF4				X											
RF5						X									
RF6					X										
RF7							X								
RF8								X							
RF9									X						
RF10							X								
RF11											X				X
RF12												X			
RF13													X	X	
RF14		X													
RF15					X										
RF16								X	X			X	X	X	
RF17										X					

Figura 2.1: Matriz casos de uso - requisitos funcionales.

## 2.3. Plan de proyecto

Para la elaboración del proyecto se ha optado por seguir UPEdu, una adaptación de la metodología RUP (Rational Unified Process) [59]. Se trata de un proceso dividido en 4 fases y caracterizado por ser iterativo e incremental, centrado en la arquitectura y dirigido por los casos de uso. Incluye artefactos, que se tratan de los resultados tangibles que el proyecto va produciendo. UPEdu se definió para aligerar RUP para un contexto educativo en el que el proyecto no tiene grandes dimensiones y normalmente hay un solo miembro del equipo, el estudiante.

Es de difícil acceso un recurso fiable que muestre unas pautas a seguir para planificar un proyecto con UPEdu así que se ha optado por utilizar como referencias para la planificación algunos proyectos realizados con anterioridad por otros compañeros, siendo de temática similar como [14, 32].

### 2.3.1. Fases del proyecto

El proyecto se compone de las siguientes fases [3]:

- **Inicio:** En esta primera fase se establecen los objetivos y alcance del proyecto de acuerdo a las partes interesadas, se analizan los posibles riesgos que pueden repercutir en el proyecto, se establecen las fases del proyecto y su presupuesto.
- **Elaboración:** Se comprueba y completa la fase anterior y se prepara la arquitectura que se utilizará en el proyecto.
- **Construcción:** Es la fase en la que se desarrolla la aplicación web en cuanto a programación se refiere, teniendo en cuenta las fases anteriores, cumpliendo los requisitos ya establecidos y adaptando los nuevos requisitos en el caso de que los haya.
- **Transición:** Se prueba a fondo la aplicación web para depurarla de todos los errores posibles, se prepara para el usuario final y finalmente se despliega.

### 2.3.2. Artefactos del proyecto

En cada fase se obtienen nuevos artefactos y se refinan los anteriores, de cada fase se obtendrán los siguientes:

- Fase de inicio:
  - Contexto, alcance y objetivos del proyecto.
  - Planificación del proyecto.
  - Especificación de requisitos.
  - Especificación de casos de uso y diagrama.
- Fase de elaboración:
  - Modelo de dominio.
  - Diagrama máquina de estados.
  - Diagrama de actividad.
  - Modelos de diseño.
  - Prototipo de interfaz.
- Fase de construcción:
  - Aplicación utilizable.
  - Pruebas de los casos de uso.
- Fase de transición:
  - Pruebas finales.
  - Manual de despliegue.
  - Manual de usuario.

Al ser iterativo, se pueden definir iteraciones que abarquen todas las actividades de los flujos principales de trabajo (requisitos, análisis, diseño, implementación y pruebas).

### 2.3.3. Planificación inicial

Para la creación del plan de trabajo se ha tenido en cuenta el número de horas que indica la guía docente para realizar el trabajo de fin de grado, mención en ingeniería del software [41]: 300 horas. También se tiene en cuenta el límite de solicitud de defensa, según la escuela de ingeniería informática [44] para alumnos que realizan prácticas curriculares obligatorias en los meses de julio y agosto. El límite de solicitud de defensa es el 18 de septiembre de 2023 y finalmente la fecha de inicio de proyecto, siendo esta la primera reunión con los tutores e interesados del gabinete, 13 de abril de 2023. Se pone como fecha de finalización del proyecto el 6 de septiembre de 2023, una semana y media antes de la solicitud de defensa, teniendo así un poco de margen antes de la solicitud. Estas fechas dan un total de 21 semanas de trabajo, para cumplir las 300 horas mínimas que requiere un TFG, las horas mínimas de dedicación semanal tendrían que ser 15, con ello se hacen 315 horas, sin embargo, la duración total del proyecto son 380 horas, lo que da un trabajo semanal de 18 horas de media, siendo más bajo en las fases con menos trabajo y superior en otras.

Dentro de la fase de construcción se realizarán dos iteraciones. En la primera se desarrollarán todos los casos de uso y en la segunda se revisarán, pulirán y se añade lo necesario. Para ello se plantea la siguiente división de las fases (Tabla 2.1):

Etapa	Duración	Fecha inicio	Fecha fin
Fase de inicio	57 horas	13/04/2023	23/05/2023
Fase de elaboración	75 horas	24/05/2023	18/06/2023
Fase de Construcción (Iteración 1)	178 horas	18/06/2023	02/08/2023
Fase de Construcción (Iteración 2)	44 horas	09/08/2023	20/08/2023
Fase de transición	26 horas	23/08/2023	06/09/2023
Límite solicitud de defensa	-	18/09/2023	
Límite de realización de tribunales	-	28/09/2023	

Tabla 2.1: Planificación fases.

La calendarización y tareas de cada fase se muestran en las siguientes tablas: Fase de inicio, Tabla 2.2; fase de elaboración, Tabla 2.3; fase de construcción 1, Tabla 2.4; fase de construcción 2, Tabla 2.5 y fase de elaboración, Tabla 2.3.

### 2.3. PLAN DE PROYECTO

Tarea	Duración	Fecha inicio	Fecha fin	Anotaciones
Primera reunión	1 hora	13/04/2023	13/04/2023	Reunión con tutores y las logopedas del Gabinete de Logopedia.
Contexto, alcance y objetivos	5 horas	13/04/2023	14/04/2023	
Identificar interesados	2 horas	15/04/2023	15/04/2023	
Elicitación de requisitos	8 horas	15/04/2023	17/04/2023	
Reunión semanal	1 hora	10/05/2023	10/05/2023	Reuniones semanales con tutores.
Revisión de requisitos	3 horas	11/05/2023	12/05/2023	
Especificación de casos de uso	8 horas	14/05/2023	15/05/2023	
Diagrama de casos de uso	2 horas	16/05/2023	16/05/2023	
Reunión semanal	1 hora	17/05/2023	17/05/2023	Reunión con tutores y gabinete.
Revisar requisitos y casos de uso	3 horas	17/05/2023	17/05/2023	
Investigar RUP	3 hora	17/05/2023	17/05/2023	
Planificación	4 horas	18/05/2023	18/05/2023	
Identificación de riesgos	2 horas	19/05/2023	19/05/2023	
Seguimiento	4 horas	20/05/2023	20/05/2023	
Presupuestos	2 horas	21/05/2023	21/05/2023	
Investigar Privacidad desde el diseño	4 horas	21/05/2023	22/05/2023	
Investigar Dev-SecOps	4 hora	22/05/2023	23/05/2023	

Tabla 2.2: Planificación inicio.

CAPÍTULO 2. REQUISITOS Y PLANIFICACIÓN

---

Tarea	Duración	Fecha inicio	Fecha fin	Anotaciones
Reunión semanal	1 hora	24/05/2023	24/05/2023	Reuniones semanales con tutores.
Modelo de dominio	4 horas	25/05/2023	25/05/2023	
Prototipos de interfaz	8 horas	26/05/2023	27/05/2023	
Diagrama de actividad	4 horas	28/05/2023	28/05/2023	
Diagramas de máquinas de estados	5 horas	29/05/2023	30/05/2023	
Reunión semanal	1 hora	31/05/2023	31/05/2023	Reuniones semanales con tutores.
Revisión de diagramas de análisis	5 horas	01/06/2023	02/06/2023	
Revisión de prototipos de interfaz	3 horas	03/06/2023	03/06/2023	
Análisis arquitectónico	2 horas	04/06/2023	04/06/2023	
Investigar análisis de riesgos de seguridad	8 horas	04/06/2023	06/06/2023	
Reunión semanal	1 hora	07/06/2023	07/06/2023	Reuniones semanales con tutores.
Análisis de riesgos de seguridad	8 horas	07/06/2023	09/06/2023	
Diseño arquitectónico	2 horas	10/06/2023	10/06/2023	
Diseño de almacenamiento	3 horas	12/06/2023	12/06/2023	
Reunión semanal	1 hora	14/06/2023	14/05/2023	Reuniones semanales con tutores.
Revisión de análisis de riesgos de seguridad	5 horas	14/06/2023	14/06/2023	

### 2.3. PLAN DE PROYECTO

---

Revisión de diagramas de análisis	3 horas	15/06/2023	16/06/2023	
Revisión de prototipos de interfaz	3 horas	16/06/2023	16/06/2023	
Revisión de parte de diseño	4 horas	17/06/2023	17/06/2023	
Rellenar memoria	4 horas	18/06/2023	18/06/2023	

Tabla 2.3: Planificación elaboración.

CAPÍTULO 2. REQUISITOS Y PLANIFICACIÓN

Tarea	Duración	Fecha inicio	Fecha fin	Anotaciones
Preparación de entorno de desarrollo	4 horas	18/06/2023	18/05/2023	
Revisión de prototipos de interfaz, requisitos, casos de uso y modelos de análisis y diseño	3 horas	19/06/2023	19/05/2023	
Desarrollo Casos de uso 1/3	40 horas	19/06/2023	26/06/2023	CU1-CU5
Pruebas Casos de uso 1/3	5 horas	26/06/2023	27/06/2023	
Reunión semanal	1 hora	28/06/2023	28/06/2023	Reuniones semanales con tutores.
Revisión Casos de uso 1/3	5 horas	29/06/2023	30/06/2023	
Desarrollo Casos de uso 2/3	40 horas	30/06/2023	10/07/2023	CU6-CU10
Pruebas Casos de uso 2/3	5 horas	10/07/2023	11/06/2023	
Reunión semanal	1 hora	12/07/2023	12/07/2023	Reuniones semanales con tutores.
Revisión Casos de uso 2/3	5 horas	13/07/2023	14/07/2023	
Desarrollo Casos de uso 3/3	40 horas	14/07/2023	24/07/2023	CU11-CU15
Pruebas Casos de uso 3/3	5 horas	24/07/2023	25/06/2023	
Reunión semanal	1 hora	26/07/2023	26/07/2023	Reuniones semanales con tutores.
Revisión Casos de uso 3/3	7 horas	27/07/2023	28/07/2023	
Contenerización	4 horas	29/07/2023	30/07/2023	
Pruebas sistema contenerizado	5 horas	31/07/2023	01/08/2023	
Completar capítulos de la memoria	7 horas	01/08/2023	02/08/2023	

Tabla 2.4: Planificación construcción 1.

### 2.3. PLAN DE PROYECTO

---

<b>Tarea</b>	<b>Duración</b>	<b>Fecha inicio</b>	<b>Fecha fin</b>	<b>Anotaciones</b>
Reunión semanal	1 hora	09/08/2023	09/08/2023	Reuniones semanales con tutores.
Revisión de prototipos de interfaz, requisitos, casos de uso y modelos de análisis y diseño	3 horas	10/08/2023	10/08/2023	
Revisión de los casos de uso desarrollados	20 horas	11/08/2023	17/08/2023	
Pruebas generales	6 horas	12/08/2023	13/08/2023	
Completar capítulos de la memoria restantes	8 horas	15/08/2023	18/08/2023	
Preparar despliegue servidores UVa	6 horas	18/08/2023	20/08/2023	

Tabla 2.5: Planificación construcción 2.



Tarea	Duración	Fecha inicio	Fecha fin	Anotaciones
Reunión semanal	1 hora	23/08/2023	23/08/2023	Reuniones semanales con tutores.
Despliegue servidores UVa	4 horas	24/08/2023	24/08/2023	
Pruebas sobre servidores UVa	5 horas	25/08/2023	26/08/2023	
Manuales de usuario y despliegue	4 horas	27/08/2023	28/08/2023	
Reunión semanal	1 hora	30/08/2023	30/08/2023	Reuniones semanales con tutores.
Revisión de sistema	6 horas	31/08/2023	01/09/2023	
Revisión memoria	4 horas	02/08/2023	03/09/2023	
Reunión semanal	1 hora	06/09/2023	06/09/2023	Reuniones semanales con tutores.

Tabla 2.6: Planificación transición.

## 2.4. Análisis de riesgos del proyecto

Con una primera planificación inicial se pueden analizar los posibles riesgos que pueden afectar al transcurso del proyecto. En este apartado se detallarán para más adelante poder planificar una nueva calendarización que sirva de contingencia en caso de que ocurra alguno de los riesgos.

Cuando se habla de riesgo en la planificación y gestión de proyectos [10], se trata de un evento o condición que puede o no ocurrir. En caso de que ocurra, tendrá un efecto en el proyecto, ya sea positivo o negativo. Identificarlos y prepararse para ellos es muy importante para que el proyecto salga adelante lo mejor posible.

En primer lugar, se identifican todos los riesgos, después, a cada uno se le adjudica la probabilidad de ocurrir, el daño potencial, un plan de mitigación y un plan de contingencia. La probabilidad de ocurrir se medirá en tres niveles: baja, media y alta. El daño potencial se medirá en tres niveles: bajo, medio, alto. La mitigación consta de distintas actuaciones que tienen como objetivo reducir la probabilidad de que el riesgo se haga realidad. La contingencia consta de actuaciones que deberán de llevarse a cabo en el caso de que el riesgo haya ocurrido.

Riesgo R1	
Nombre	Requisitos incompletos
Descripción	El caso de que las partes interesadas propongan algún requisito nuevo o la modificación de alguno ya existente
Probabilidad	Baja
Daño potencial	Medio
Mitigación	<ul style="list-style-type: none"> <li>■ Buena comunicación con las partes interesadas.</li> <li>■ Prestar mucha atención y darle vueltas a los requisitos ofrecidos antes de pasar a la fase de análisis.</li> <li>■ Insistir en sí todos los requisitos están nombrados antes de pasar a la fase de análisis.</li> </ul>
Contingencia	<ul style="list-style-type: none"> <li>■ Aumentar las horas de trabajo para poder añadir los nuevos requisitos.</li> <li>■ Alargar la fase actual para poder añadir los nuevos requisitos.</li> </ul>

Riesgo R2	
Nombre	Avería o funcionamiento incorrecto del equipo de desarrollo
Descripción	El caso de que el equipo de desarrollo deje de funcionar correctamente o llegue a averiarse
Probabilidad	Baja
Daño potencial	Medio
Mitigación	<ul style="list-style-type: none"> <li>■ Un buen trato del equipo, manteniéndolo cuidado y libre de espacio.</li> <li>■ Realizar copias de seguridad.</li> </ul>
Contingencia	<ul style="list-style-type: none"> <li>■ Migrar a otro dispositivo disponible, gracias a las copias de seguridad.</li> </ul>

Riesgo R3	
Nombre	Imposibilidad de trabajar por asuntos familiares
Descripción	El caso en que, por cualquier motivo familiar, este implique desplazarse o no poder trabajar por la necesidad de dar atención a alguien.
Probabilidad	Baja
Daño potencial	Medio
Mitigación	<ul style="list-style-type: none"> <li>■ Un seguimiento de aquellas personas posibles para estar preparado.</li> </ul>
Contingencia	<ul style="list-style-type: none"> <li>■ En caso de desplazamiento intentar llevarse el equipo de trabajo.</li> <li>■ En caso de atención, intentar trabajar en momentos libres.</li> </ul>

## 2.4. ANÁLISIS DE RIESGOS DEL PROYECTO

---

Riesgo R4	
Nombre	Imposibilidad de trabajar por enfermedad propia
Descripción	El caso de caer enfermo puede dificultar el concentrarse para trabajar.
Probabilidad	Media
Daño potencial	Bajo
Mitigación	<ul style="list-style-type: none"><li>■ En verano, en caso de poder padecer de alergias, no salir mucho fuera.</li></ul>
Contingencia	<ul style="list-style-type: none"><li>■ En caso de ser grave e imposibilite trabajar, aplazar la planificación.</li><li>■ En caso de poder seguir trabajando, mantener la planificación si es posible y realizar tareas más sencillas, dejando otras más complicadas para cuando se mejore.</li></ul>

Riesgo R5	
Nombre	Uso de una nueva tecnología
Descripción	El uso de una tecnología desconocida puede llegar a complicar el transcurso del proyecto.
Probabilidad	Media
Daño potencial	Medio
Mitigación	<ul style="list-style-type: none"><li>■ Durante las etapas iniciales del proyecto, identificar las tecnologías desconocidas.</li></ul>
Contingencia	<ul style="list-style-type: none"><li>■ Buscar fuentes fiables donde aprender a usar la tecnología.</li><li>■ Preguntar a tutores de TFG.</li></ul>

Riesgo R6	
Nombre	Atasque en una tarea
Descripción	Durante la realización de las tareas puede dar lugar a quedarse atasgado con un problema y no conseguir resolverlo.
Probabilidad	Media
Daño potencial	Bajo
Mitigación	<ul style="list-style-type: none"> <li>■ Realizar las actividades en el tiempo y fechas planificadas.</li> <li>■ Investigar antes de hacer una tarea.</li> </ul>
Contingencia	<ul style="list-style-type: none"> <li>■ Buscar fuentes fiables donde corregir el error.</li> <li>■ Preguntar a tutores de TFG.</li> </ul>

Riesgo R7	
Nombre	Compatibilizar con prácticas de empresa
Descripción	Durante los meses de julio y agosto se tiene previsto realizar prácticas de empresa, compatibilizarlas con el TFG puede ser complicado.
Probabilidad	Alta
Daño potencial	Bajo
Mitigación	<ul style="list-style-type: none"> <li>■ Antes de empezar prácticas tener un horario de trabajo diario establecido.</li> </ul>
Contingencia	<ul style="list-style-type: none"> <li>■ Dedicar lo máximo posible al TFG los fines de semana.</li> <li>■ Descansar entre prácticas y TFG para estar despejado.</li> </ul>

## 2.4. ANÁLISIS DE RIESGOS DEL PROYECTO

---

Riesgo R8	
Nombre	Sobrecarga con fechas de exámenes ordinarios
Descripción	Durante el mes de junio se realizan los exámenes de convocatoria ordinaria y pueden afectar al desarrollo del proyecto.
Probabilidad	Alta
Daño potencial	Medio
Mitigación	<ul style="list-style-type: none"><li>▪ Llevar un buen horario de estudio y dedicación al TFG.</li><li>▪ Estudiar con antelación antes de las fechas.</li></ul>
Contingencia	<ul style="list-style-type: none"><li>▪ Retrasar las fechas previstas del proyecto.</li></ul>

Riesgo R9	
Nombre	Exámenes extraordinarios
Descripción	Durante el mes de junio se realizan los exámenes de convocatoria extraordinaria y pueden afectar al desarrollo del proyecto.
Probabilidad	Media
Daño potencial	Medio
Mitigación	<ul style="list-style-type: none"><li>▪ Llevar un buen horario de estudio y dedicación al TFG.</li><li>▪ Intentar aprobar en convocatoria ordinaria.</li></ul>
Contingencia	<ul style="list-style-type: none"><li>▪ Retrasar las fechas previstas del proyecto.</li></ul>

Riesgo R10	
Nombre	Error en la planificación
Descripción	Errores en la planificación del proyecto, ya sea por duración, orden de tareas o tareas no contempladas.
Probabilidad	Media
Daño potencial	Medio
Mitigación	<ul style="list-style-type: none"> <li>■ Utilizar proyectos semejantes para tomar referencia de las planificaciones.</li> <li>■ Pensar detenidamente en cada tarea y su planificación.</li> </ul>
Contingencia	<ul style="list-style-type: none"> <li>■ Reorganizar las tareas necesarias.</li> <li>■ Atrasar fechas si es necesario.</li> </ul>

Riesgo R11	
Nombre	Imposibilidad de entregar el TFG en fecha estimada
Descripción	Debido al suspenso de alguna asignatura, esto imposibilitaría entregar el trabajo en la fecha de septiembre.
Probabilidad	Media
Daño potencial	Bajo
Mitigación	<ul style="list-style-type: none"> <li>■ Aprovechar la convocatoria extraordinaria.</li> <li>■ Preparar bien los exámenes.</li> </ul>
Contingencia	<ul style="list-style-type: none"> <li>■ Retrasar fecha de entrega.</li> <li>■ Utilizar convocatoria de fin de carrera.</li> </ul>

## 2.4. ANÁLISIS DE RIESGOS DEL PROYECTO

---

Riesgo R12	
Nombre	Despliegue en servidores de la UVa
Descripción	No se garantiza que nos permitan desplegar la aplicación web en los servidores de la UVa.
Probabilidad	Media
Daño potencial	Bajo
Mitigación	<ul style="list-style-type: none"><li>▪ Satisfacer los requisitos de los técnicos y seguridad.</li><li>▪ Revisiones periódicas.</li></ul>
Contingencia	<ul style="list-style-type: none"><li>▪ Hacer el despliegue final en dos máquinas virtuales.</li></ul>

Riesgo R13	
Nombre	Incumplimiento RGPD
Descripción	El incumplimiento del RGPD puede dar lugar a cambiar bastantes cosas del proyecto.
Probabilidad	Bajo
Daño potencial	Alto
Mitigación	<ul style="list-style-type: none"><li>▪ Satisfacer los requisitos de seguridad y privacidad.</li><li>▪ Revisiones periódicas del cumplimiento.</li></ul>
Contingencia	<ul style="list-style-type: none"><li>▪ Cambiar lo necesario para el cumplimiento.</li></ul>



Riesgo R14	
Nombre	Calidad insuficiente
Descripción	Podría ocurrir que el producto final no esté considerado como de calidad.
Probabilidad	Bajo
Daño potencial	Alto
Mitigación	<ul style="list-style-type: none"> <li>■ Satisfacer los requisitos de todos los interesados.</li> <li>■ Revisiones periódicas.</li> <li>■ Uso de herramientas que comprueben la calidad.</li> </ul>
Contingencia	<ul style="list-style-type: none"> <li>■ Volver a etapas anteriores del proyecto para solucionar los fallos de calidad.</li> </ul>

## 2.5. Replanificación

Ha sido necesaria una replanificación del proyecto, ya que a finales del mes de julio de 2023 las circunstancias del alumno cambiaron. En un principio, la fecha límite de entrega se planificó para septiembre, ya que el alumno realizaría prácticas de empresa durante el verano y así tener la posibilidad de optar a la convocatoria extraordinaria para alumnos en estas circunstancias. Como se previó, el alumno empezó sus prácticas de empresa en el mes de julio. Sin embargo, las prácticas que se obtuvieron se trataban de una beca que le implicaban estar matriculado en la universidad durante el siguiente curso académico 2023-2024. Las nuevas circunstancias permitieron que la fecha de entrega inicialmente estimada para septiembre de 2023 pudiera ser aplazada, brindando al alumno la posibilidad de realizar el proyecto más espaciado en el tiempo y sin una fecha de entrega tan cercana. La fecha de entrega no se ha incluido en la planificación, ya que al ser entregado durante el curso se podrá entregar en cualquier momento, por lo que será en torno a la fecha de finalización del proyecto.

Aprovechando la situación, el alumno pudo realizar una ligera pausa en el proyecto entre los meses de agosto y septiembre.

Aunque el alumno desarrollará una actividad laboral al mismo tiempo que lleva a cabo este proyecto, la nueva replanificación será semejante a la planificación original, exceptuándose en las fechas de las tareas, algunas duraciones y algunas tareas adicionales que no se contemplaron al inicio.

En la planificación original las horas semanales dedicadas al proyecto eran 18 de media, ya que fluctuaban según las etapas e iteraciones. En la nueva replanificación se tendrán en cuenta de manera constante un total de 26 horas de trabajo semanales, la mayor parte de ellas serán dedicadas durante el fin de semana.

## 2.5. REPLANIFICACIÓN

---

La suma total de horas planificadas es 372, contando con las primeras fases realizadas con la planificación inicial y el resto de la replanificación. A continuación se muestra la replanificación global de cada fase, se puede comprobar que las fases de elaboración y construcción 1 se solapan, esto se debe a una tutoría que se realiza al final de la fase de elaboración, pero que no impide el inicio de construcción 1.

Ya que se realiza una replanificación se ha aprovechado la ocasión para añadir alguna tarea que no se había contemplado en la planificación inicial.

<b>Etapa</b>	<b>Duración</b>	<b>Fecha inicio</b>	<b>Fecha fin</b>
Fase de elaboración	18 horas	17/10/2023	27/10/2023
Fase de Construcción (Iteración 1)	173 horas	21/10/2023	10/12/2023
Fase de Construcción (Iteración 2)	41 horas	11/12/2023	23/12/2023
Fase de transición	31 horas	08/01/2024	18/01/2024

Tabla 2.7: Replanificación fases.

A continuación se detalla la planificación de todas las tareas de cada fase.

Tarea	Duración	Fecha inicio	Fecha fin	Anotaciones
Investigar arquitectura	3 horas	17/10/2023	19/10/2023	
Reunión tutoría	1 hora	18/10/2023	18/10/2023	
Revisión de prototipos de interfaz	1 hora	19/10/2023	19/10/2023	
Revisión de diseño de despliegue	1 hora	19/10/2023	19/10/2023	
Revisión de diseño de almacenamiento	1 hora	19/10/2023	19/10/2023	
Diseño arquitectónico	4 horas	19/10/2023	21/10/2023	
Rellenar memoria	1 hora	21/10/2023	21/10/2023	
Investigar funcionamiento SSO UVa	3 horas	21/10/2023	21/10/2023	
Reunión tutoría	1 hora	27/10/2023	27/10/2023	
Revisión de parte de diseño	2 horas	27/10/2023	27/10/2023	

Tabla 2.8: Replanificación elaboración.

## 2.5. REPLANIFICACIÓN

Tarea	Duración	Fecha inicio	Fecha fin	Anotaciones
Desarrollo Casos de uso 1/3	40 horas	21/10/2023	31/10/2023	CU1-CU5
Pruebas Casos de uso 1/3	5 horas	02/11/2023	03/11/2023	
Reunión semanal	1 hora	03/11/2023	03/11/2023	Reuniones semanales con tutores.
Revisión Casos de uso 1/3	5 horas	04/11/2023	04/11/2023	
Desarrollo Casos de uso 2/3	40 horas	04/11/2023	18/11/2023	CU6-CU10
Reunión semanal	1 hora	17/07/2023	17/07/2023	Reuniones semanales con tutores.
Pruebas Casos de uso 2/3	5 horas	18/11/2023	18/11/2023	
Revisión Casos de uso 2/3	5 horas	18/11/2023	19/11/2023	
Desarrollo Casos de uso 3/3	40 horas	18/11/2023	02/12/2023	CU11-CU15
Pruebas Casos de uso 3/3	5 horas	03/12/2023	03/12/2023	
Reunión semanal	1 hora	04/12/2023	04/12/2023	Reuniones semanales con tutores.
Revisión Casos de uso 3/3	7 horas	05/12/2023	08/12/2023	
Contenerización	4 horas	08/12/2023	09/12/2023	
Pruebas sistema contenerizado	5 horas	09/12/2023	09/12/2023	
Completar capítulos de la memoria	9 horas	10/12/2023	10/12/2023	

Tabla 2.9: Replanificación construcción 1.

<b>Tarea</b>	<b>Duración</b>	<b>Fecha inicio</b>	<b>Fecha fin</b>	<b>Anotaciones</b>
Reunión semanal	1 hora	11/12/2023	11/12/2023	Reuniones semanales con tutores.
Revisión de los casos de uso desarrollados	20 horas	11/12/2023	17/12/2023	
Pruebas generales	6 horas	17/12/2023	17/12/2023	
Completar capítulos de la memoria restantes	8 horas	19/12/2023	22/12/2023	
Preparar despliegue servidores UVa	6 horas	23/12/2023	23/12/2023	

Tabla 2.10: Replanificación construcción 2.

Tarea	Duración	Fecha inicio	Fecha fin	Anotaciones
Reunión semanal	1 hora	08/01/2024	08/01/2024	Reuniones semanales con tutores.
Despliegue servidores UVa	4 horas	09/01/2024	10/01/2024	
Pruebas sobre servidores UVa	5 horas	11/01/2024	12/01/2024	
Manuales de usuario y despliegue	4 horas	13/01/2024	13/01/2024	
Revisión de sistema	6 horas	13/01/2024	14/01/2024	
Revisión memoria	4 horas	14/01/2024	14/01/2024	
Reunión semanal	1 hora	15/01/2024	15/01/2024	Reuniones semanales con tutores.
Revisiones finales	6 horas	16/01/2024	18/01/2024	

Tabla 2.11: Replanificación transición.

## 2.6. Presupuesto

Se realizan dos presupuestos, uno simulado y el real. Al ser un TFG no se dispone de un presupuesto más allá de las pertenencias del estudiante; sin embargo, se calcula un presupuesto simulado para ver cuál sería el coste de un proyecto como este en un ámbito laboral en vez de uno académico.

Los recursos utilizados son los siguientes:

- Desarrollador Full-Stack [20], el sueldo bruto medio en España es de aproximadamente 30.000€, lo que equivalen a 15,6€/por hora.
- Oficina coworking [21], la tarifa individual para una disponibilidad completa es de 151,25€/mes.
- Ordenador portátil [30], el portátil es un *Acer Nitro 5 AN515-54-52G3*, a día de hoy ya no está en venta, su coste fue de 699€.
- Teléfono móvil [5], el teléfono móvil es un *iPhone 13 mini 128GB*, su coste es de 809€.
- Astah Profesional [6], tiene un coste de 7,5 €/mes.

- Balsamiq Wireframes [8], tiene un coste de 8,4 €/mes.

### 2.6.1. Presupuesto simulado

Recurso	Coste
Desarrollador Full-Stack	$15,6\text{€} \times 380\text{h} = 5.928\text{€}$
Seguridad social	La empresa [18] suele pagar alrededor de un 30 %. $5.928\text{€} \times 0.3 = 1.778,4\text{€}$
Oficina	$151,25\text{€} \times (21 \text{ semanas} / 4) = 794,06\text{€}$
Astah Professional	$7,5\text{€} \times (21 \text{ semanas} / 4) = 39,38\text{€}$
Balsamiq Wireframes	$8,4\text{€} \times (21 \text{ semanas} / 4) = 44,1\text{€}$
Ordenador portátil	699€
Teléfono móvil	809€
<b>Total</b>	<b>10.091,94€</b>

### 2.6.2. Presupuesto real

En el caso del presupuesto real no se tendrá en cuenta ningún coste, ya que como desarrollador, al ser un TFG en este caso no se cobra, la oficina de coworking no es necesaria, ya que el estudiante trabajará en su propio alojamiento, las licencias de Astah Professional y Balsamiq Wireframes las proporciona la UVa, y los dos dispositivos para el desarrollo ya han sido pagados en su momento independientemente del proyecto.

Recurso	Coste
Desarrollador Full-Stack	0€
Seguridad social	0€
Alojamiento	0€
Internet	0€
astah professional	0€
Balsamiq Wireframes	0
Ordenador portátil	0€
Teléfono móvil	0€
<b>Total</b>	<b>0€</b>

## 2.7. Metodologías para el desarrollo

Estas son las metodologías que se utilizarán de cara al desarrollo del proyecto.

### 2.7.1. Privacidad desde el diseño

Uno de los objetivos de este proyecto es garantizar la privacidad de los usuarios, para ello se usará la privacidad desde el diseño [1]. Antes de entrar en detalle en cada uno de sus principios, es necesario recalcar cuál es el ciclo de vida habitual y principal que sufren los datos: recogida, uso, conservación y difusión. Con ello, los principios de privacidad desde el diseño son los siguientes:

1. Proactivo, no reactivo; preventivo, no correctivo: Se trata de anticiparse a los sucesos que pueden afectar a la privacidad, buscando y analizando los posibles riesgos y minimizándolos. De esta forma se evita el peligro antes de que ocurra.
2. La privacidad como configuración predeterminada: Durante todas las etapas del tratamiento de datos se prioriza el tener los menos datos posibles.
3. Privacidad incorporada en la fase de diseño: La seguridad debe de estar integrada desde el inicio, empezando por los requisitos no funcionales.
4. Funcionalidad total: pensamiento “todos ganan”. En vez de que a costa de ganar privacidad se pierdan otras propiedades, hay que encontrar un equilibrio en el que todos ganen.
5. Aseguramiento de la privacidad en todo el ciclo de vida: En todas y cada una de las operaciones que sufren los datos en su ciclo de vida, se debe de implementar las medidas necesarias para garantizar su protección.



6. Visibilidad y transparencia: Para las personas cuyos datos son tratados, se debe de dejar claro que se están tratando y cómo se está haciendo.
7. Respeto por la privacidad de los usuarios: mantener un enfoque centrado en el usuario. El usuario es lo principal, por lo que se le tiene que permitir tener un papel activo sobre el control de sus datos.



## Capítulo 3

# Análisis

En esta sección se realiza un análisis con los requisitos detallados anteriormente y se realizan: un diagrama de casos de uso, modelo de dominio, diagrama de actividad y máquinas de estado.

### 3.1. Diagrama casos de uso

Se muestra el diagrama de casos de uso en la Figura 3.1, en él aparecen los actores que interactúan con el sistema, así como los casos de uso en los que están involucrados.

### 3.1. DIAGRAMA CASOS DE USO



Figura 3.1: Diagrama casos de uso.

## 3.2. Modelo de dominio

Se ha creado el modelo de dominio de la Figura 3.2. Este modelo tiene una clase Usuario, que contiene la información para identificarse en el sistema, la clase se trata de una generalización de las clases Solicitante y Gestora. Gestora es representativa, ya que no se almacena ningún dato más allá de los de usuario, en cambio, Solicitante guarda la información necesaria acerca de los solicitantes. Solicitante tiene relacionado una clase Paciente, que guardará la información del paciente, ambos están relacionados con la clase cita, ya que una todas las citas tendrán siempre un paciente y un solicitante. Se ha hecho una clase DiaDisponibilidad, para personalizar días de forma individual y para poder adjuntar las citas creadas a una fecha. DiaDisponibilidad está relacionado con HoraDisponibilidad, esta clase representa los huecos para las citas. Finalmente, la cita también está relacionada con esta HoraDisponibilidad para así poder adjudicar la cita a un hueco. Se ha creado una clase horarioPredeterminado para el horario que se mostrará por defecto. Finalmente, se ha hecho una clase contacto para guardar la información de contacto del gabinete. En cuanto a las enumeraciones se han creado las necesarias para satisfacer los requisitos de información. Para los datatype se han tenido que hacer cuatro para representar la fecha, hora, contacto y dirección postal.

Más tarde, este modelo de dominio se vio modificado para adaptarlo al diagrama de base de datos realizado durante la implementación. El modelo de dominio modificado es el de la Figura 3.3. Se ha añadido una clase HorarioDisponibilidad para definir un horario por defecto en el que se podrán recibir las citas; se eliminó la clase de semanaDisponibilidad, ya que no proporcionaba ninguna información útil, y se suprimió el datatype de DireccionPostal, ya que con el atributo localidad y dirección es suficiente.

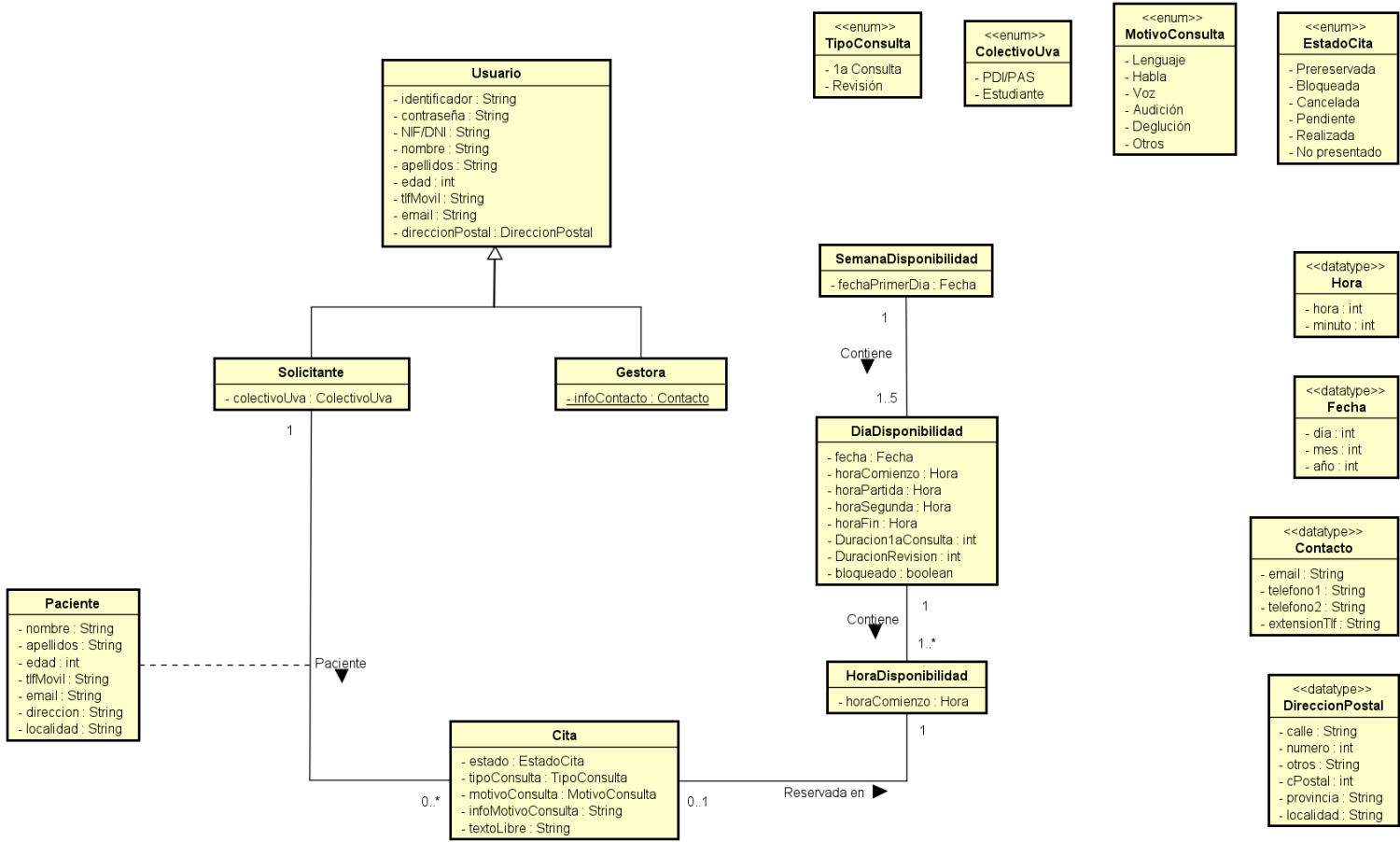


Figura 3.2: Modelo de dominio inicial.

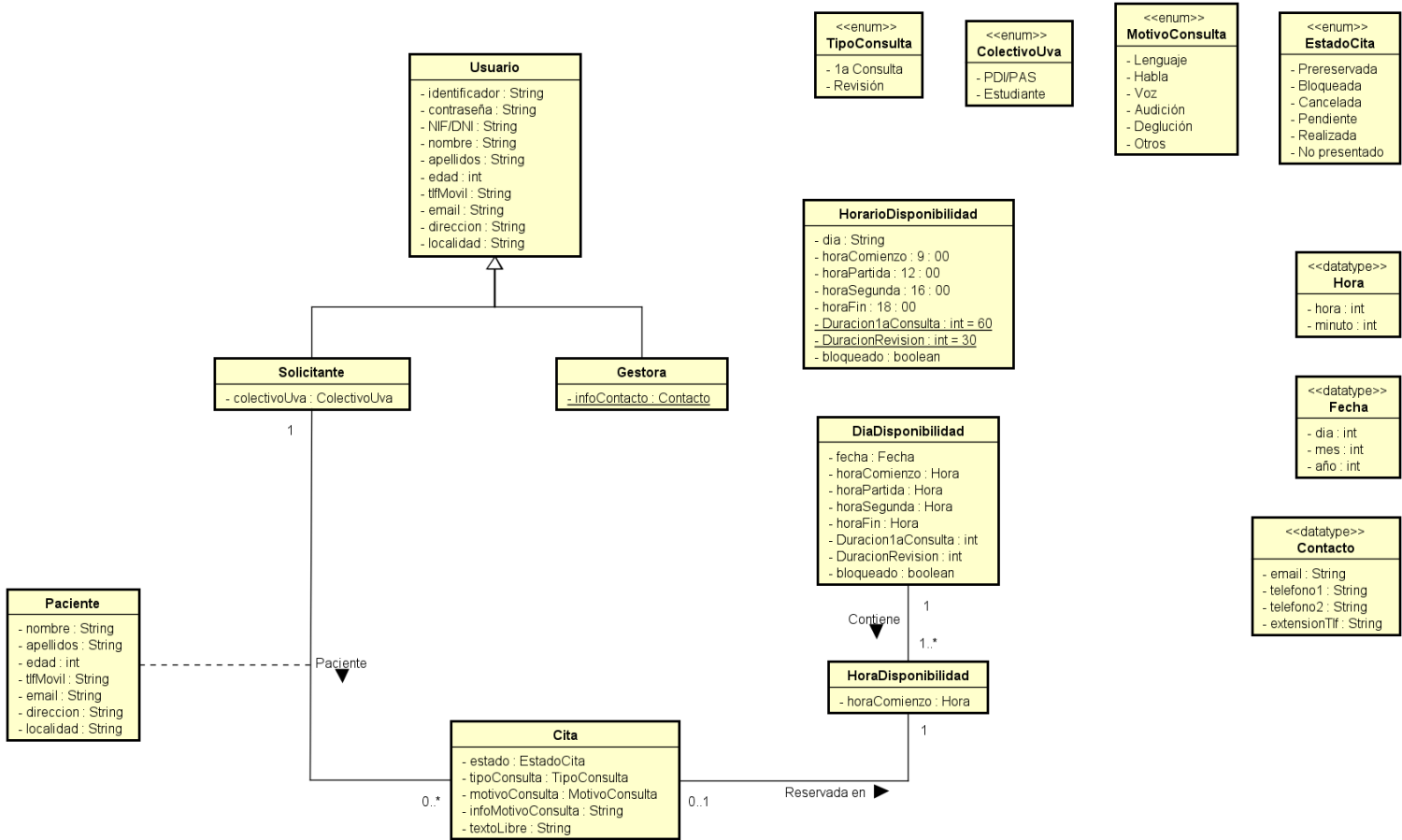


Figura 3.3: Modelo de dominio modificado.

### 3.3. Análisis de procesos

Los diagramas de actividad para el análisis de los procesos del “negocio” detallan las acciones que realizan los distintos actores para llevar a cabo su objetivo.

Se muestran dos diagramas de actividad, Figura 3.3, para las acciones realizadas por un solicitante y Figura 3.4 para las acciones realizadas por las gestoras.



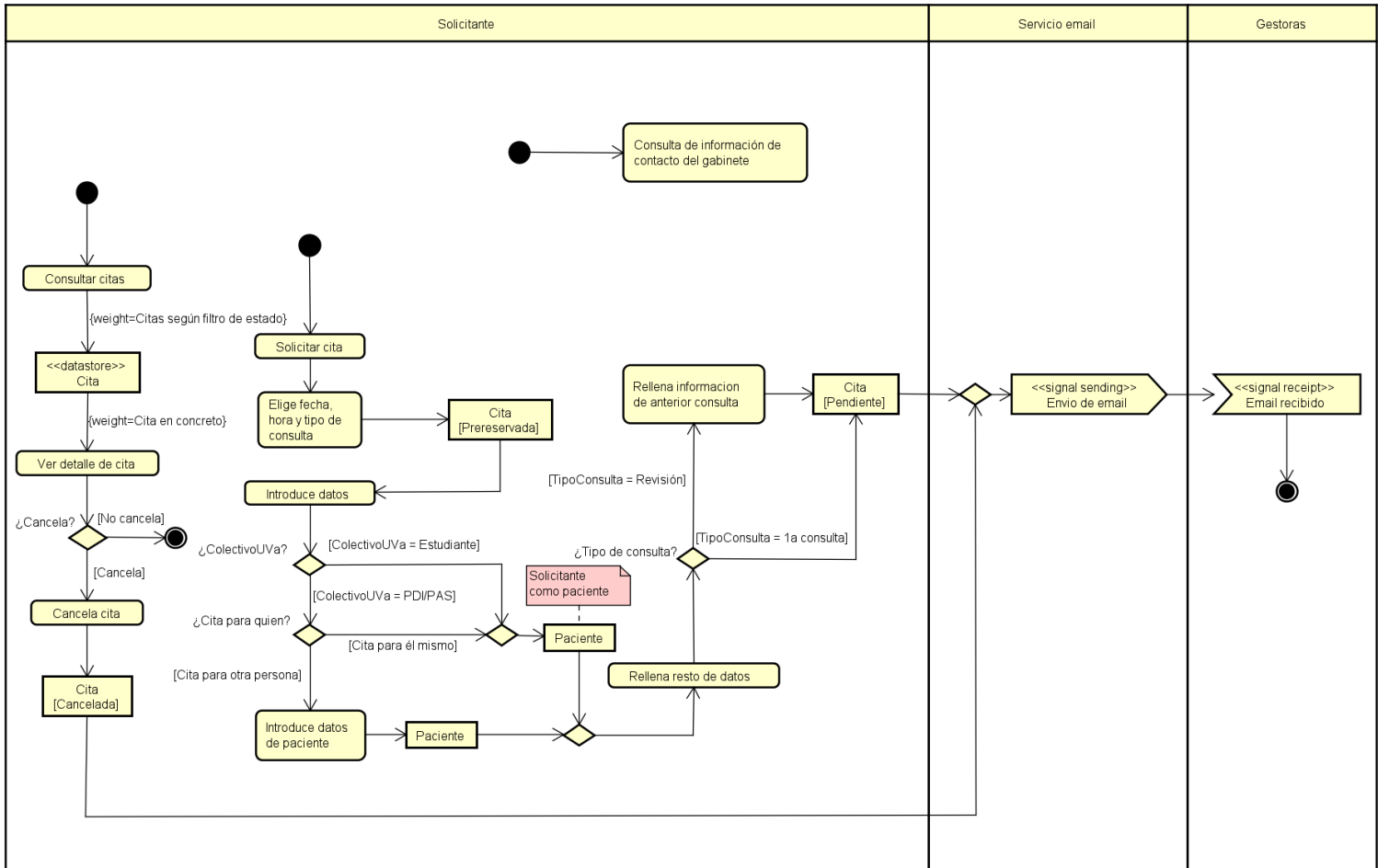


Figura 3.4: Diagrama de actividad del solicitante.

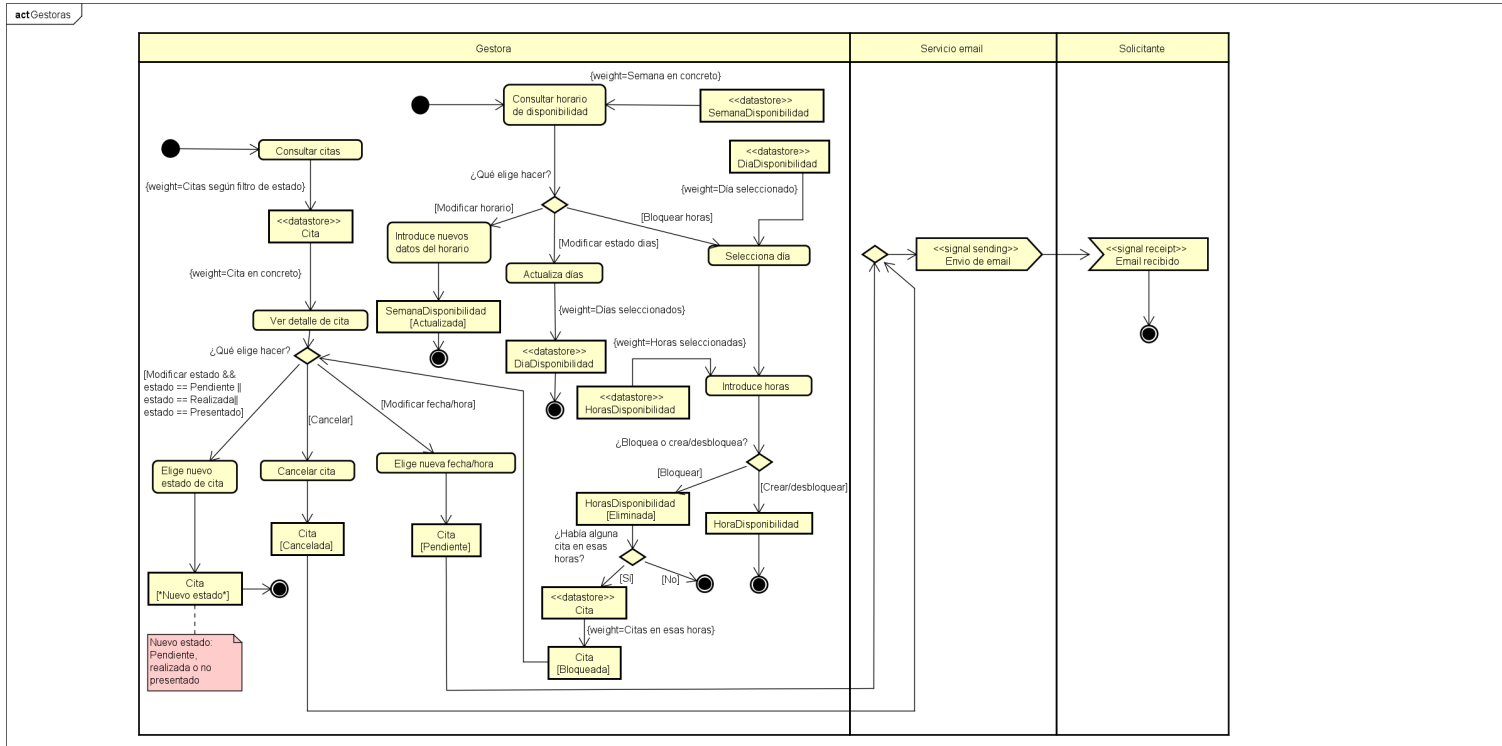


Figura 3.5: Diagrama de actividad de las gestoras.

### **3.4. Análisis de estados**

En esta sección se detalla el análisis de los objetos que se comportan como máquinas de estados. En primer lugar, la respectiva al estado de una cita (Figura 3.3). Cabe destacar que el estado de bloqueada se produce cuando se bloquea por parte de las gestoras una franja horaria en la que se encontraba dicha cita, cuando modifican su disponibilidad.

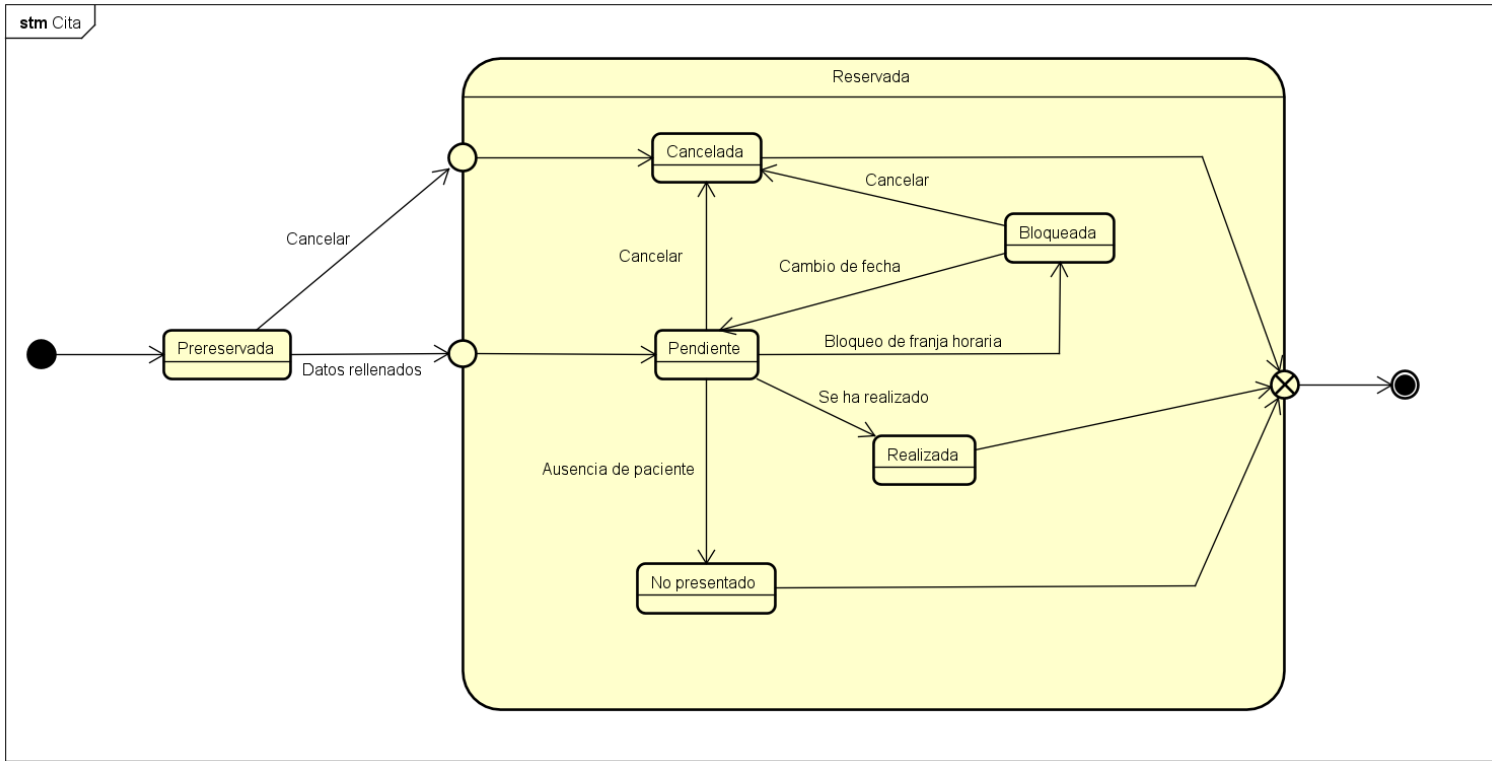


Figura 3.6: Máquina de estados de cita.

La siguiente máquina de estados (Figura 3.4) se refiere a los estados por los que pasa un día de disponibilidad.

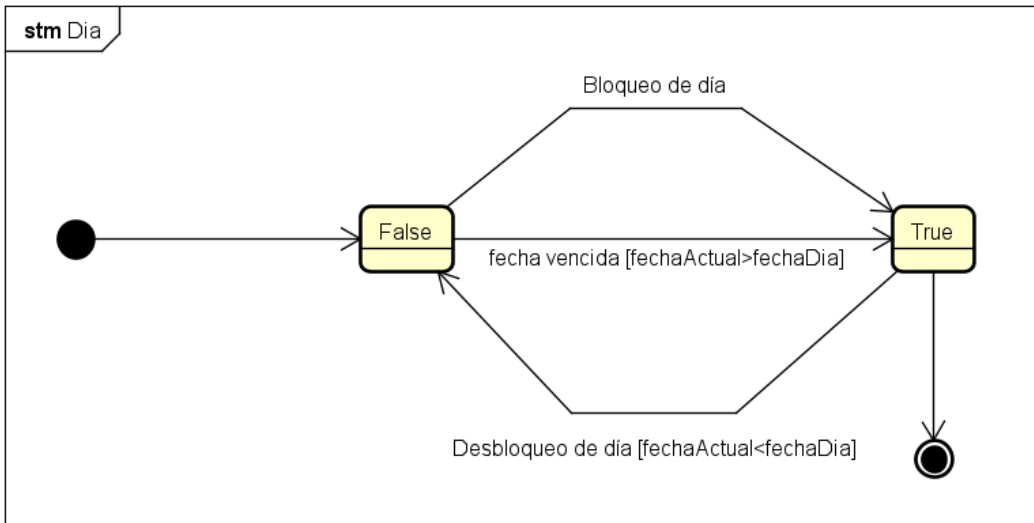


Figura 3.7: Máquina de estados de día.

### 3.5. Análisis de riesgos de seguridad de la información

La gestión de riesgos seguridad de la información fue uno de los requisitos impuestos por el responsable de privacidad de la UVa.

Se ha utilizado la herramienta PILAR [16], ya que se basa en la metodología MAGERIT [24]. Se trata de una metodología enfocada en la gestión de la seguridad basada en riesgos, elaborada directamente por el Gobierno de España.

Se realizó un análisis inicial, pero llegado a cierto punto se tenían algunas dudas acerca de la herramienta utilizada y como aplicar los resultados. Combinado también con la noticia de que este TFG no se implantaría en la UVa, se detuvo el análisis. De todos modos, como se invirtió cierto tiempo en ello, se ha añadido en el repositorio de la aplicación una carpeta con los recursos que se produjeron durante la realización de este análisis. Se trata de */docs/riesgos\_de\_seguridad\_de\_la\_informacion*.



## Capítulo 4

# Tecnologías utilizadas

En este apartado de la memoria se muestran cuáles han sido las tecnologías utilizadas en el proyecto, ya sea para su planificación, desarrollo y cualquiera necesaria para completarlo.

### 4.1. Herramientas de documentación, análisis y diseño

Se han utilizado las siguientes herramientas para escribir la documentación del proyecto

#### 4.1.1. Overleaf

Overleaf [53] es un editor colaborativo de LaTeX en la nube. Se utiliza para escribir, editar y publicar documentos de manera sencilla desde un navegador, permite escribir texto en LaTeX y a la vez visualizar el texto compilado en la misma pantalla. En este proyecto se ha utilizado para escribir la presente memoria, al ser colaborativo permite a los tutores ver el documento en cualquier momento.

#### 4.1.2. Astah Professional

Astah Professional [7] es una herramienta que permite crear diagramas UML de forma rápida, ya que proporciona los recursos y expresiones necesarias para los diagramas UML. En este proyecto se ha utilizado para hacer todos los diagramas de análisis y diseño.

### 4.1.3. Balsamiq Wireframes

Balsamiq Wireframes [9] es una herramienta de diseño de interfaces, permite plasmar las ideas iniciales del diseño de interfaz de un producto de software antes de escribir nada de código. En este proyecto se ha utilizado para hacer los bocetos iniciales de la interfaz.

### 4.1.4. PILAR Basic

PILAR Basic [16] es una herramienta que permite el análisis y la gestión de riesgos de seguridad de un sistema informático que se basa en la metodología MAGERIT. En este proyecto se ha utilizado para el análisis de riesgos de seguridad del proyecto.

## 4.2. Herramientas de comunicación

Herramientas utilizadas para comunicarse, ya sea con los tutores del proyecto, tanto con las logopedas del gabinete de logopedia.

### 4.2.1. Microsoft Teams

Microsoft Teams [22] es una herramienta de mensajería que permite una colaboración y comunicación en tiempo real, permitiendo tanto mensajería por chat, hasta reuniones con cámara y audio. En este proyecto se ha utilizado para comunicarse con las logopedas del Gabinete de Logopedia.

### 4.2.2. Outlook

Outlook [23] es una herramienta de organización de email, permitiendo gestionar los emails según los intereses del cliente, organizarse con un calendario y por supuesto enviar emails. En este proyecto se ha utilizado para comunicarse con los tutores del TFG.

## 4.3. Herramientas de desarrollo

Herramientas que han sido utilizadas para llevar a cabo la implementación de la aplicación web.



### 4.3.1. Visual Studio Code

Visual Studio Code [60] es un editor de código fuente que facilita su desarrollo, debido a que integra distintas funcionalidades. Posee soporte para depurar, control de versiones de Git, resalta sintaxis y errores según el lenguaje de programación y diversas herramientas más, gracias a su biblioteca de extensiones.

### 4.3.2. Git

Git [46] es un software libre de control de versiones, permite llevar el registro de los cambios en cada uno de los archivos y que varias personas coordinen su trabajo al realizar cambios sobre los archivos de un repositorio común.

### 4.3.3. GitLab

GitLab [47] se trata de un software que ofrece un servicio web colaborativo, con control de versiones y DevOps basado en Git. Permite alojar un repositorio con los archivos de un proyecto, manejar y visualizar el control de versiones.

### 4.3.4. git-crypt

git-crypt permite cifrar y descifrar archivos seleccionados dentro de un repositorio de git. Los archivos elegidos se cifran al hacer “commit” y se descifran al recuperar el repositorio. Todo mediante una llave secreta.

### 4.3.5. Java

Java [49] es un lenguaje de programación multiplataforma y orientado a objetos. Las aplicaciones Java son compiladas y ejecutadas en una máquina virtual Java.

### 4.3.6. Spring Boot

Spring Boot [36] es un framework que permite desarrollar aplicaciones Java. Facilita el desarrollo de aplicaciones, ya que pone a disposición del desarrollador sus librerías.

### 4.3.7. Nginx

Nginx [52] es un servidor web ligero de alto rendimiento, programado en C. Es de software libre y de código abierto.

#### 4.3.8. MySQL

MySQL [51] es un sistema de gestión de base de datos relacionales. Está basado en SQL (Structured Query Language), es de código abierto y está desarrollado por Oracle.

#### 4.3.9. HTML

HTML (HyperText Markup Language) [48] es un lenguaje de marcado para crear páginas web. Ofrece gran adaptabilidad, estructura lógica y una fácil interpretación para los desarrolladores.

#### 4.3.10. CSS

CSS (Cascading Style Sheets) [45] es un lenguaje de diseño para definir la presentación de un documento que ha sido desarrollado con base a un lenguaje de tipado, como puede ser HTML.

#### 4.3.11. JavaScript

JavaScript [50] es un lenguaje de programación interpretado, orientado a objetos, basado en clases e imperativo. En este proyecto, se utiliza como soporte para los documentos creados con HTML, proporcionándoles cierta funcionalidad.

#### 4.3.12. Thymeleaf

Thymeleaf [38] es un procesador de plantillas en la parte del servidor con el lenguaje de Java. Permite procesar plantillas HTML en la parte del servidor, para posteriormente ser mostrarlas en el navegador.

#### 4.3.13. Docker

Docker [15] es una herramienta que permite automatizar el despliegue de aplicaciones de manera sencilla y rápida. Independientemente de la máquina.

#### 4.3.14. Maven

Maven [4] es un gestor de proyectos java. Gestiona su construcción, tests y documentación.

## Capítulo 5

# Diseño

En este capítulo, se presentarán las decisiones de diseño fundamentales para el desarrollo de la aplicación.

### 5.1. Diseño de interfaz de usuario

Para el diseño de la interfaz se ha utilizado la herramienta *Balsamiq Wireframes*, la cual nos permite hacer unos bocetos sencillos que sirven para dejar claro cuál es la idea principal de la interfaz. Se han creado los diseños de la aplicación para las vistas de escritorio y móvil.

Los bocetos se utilizaron, en cierta parte, para que las logopedas del Gabinete de Logopedia de la UVa diesen su opinión y aprobación acerca de la interfaz y flujos de la aplicación. Tras dos iteraciones de estos bocetos, se pudo obtener su aprobación.

La explicación de cada interfaz y los flujos de navegación se realizan de la siguiente forma: Explicación del escenario correspondiente, su flujo habitual a través de cada interfaz y finalmente el conjunto de todas las imágenes de las interfaces de ese escenario ordenadas.

En primer lugar, la interfaz de inicio de sesión, siendo común tanto para solicitante como para gestora:



Figura 5.1: Boceto de inicio de sesión.



Figura 5.2: Boceto móvil de inicio de sesión.

### 5.1.1. Interfaz gestora

En cuanto a la parte de la interfaz que verán las gestoras al identificarse, esta es la interfaz inicial; en ella se ven las opciones que se pueden realizar.

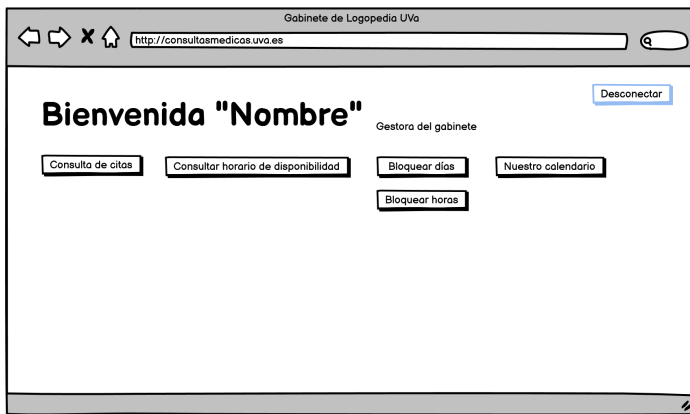


Figura 5.3: Boceto de gestora - inicio.

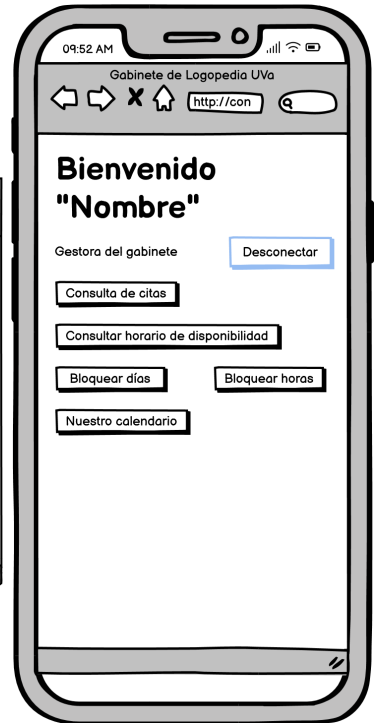


Figura 5.4: Boceto móvil de gestora - inicio.

Si pinchamos en la opción de Consulta de citas, iremos a la interfaz (figura 5.5) donde nos muestra las próximas citas, dos filtros para ordenar por fecha y por estado. En cada cita se puede seleccionar la opción de ver detalle. Cuando entramos en “ver detalle” la interfaz se adapta según la información que contiene la cita, se muestran dos ejemplos, el primero (figura 5.7) de una cita de revisión, donde el paciente es el solicitante y la segunda (figura 5.9), donde es una cita de 1a consulta y el solicitante es distinto del paciente. En cada cita, además de ver su información, se puede cancelar, cambiar el estado de la cita y modificar fecha y hora. En el caso de cancelar la cita se muestra un mensaje de confirmación (figura 5.11); en el caso de modificar el estado, al pulsar continuar también muestra un mensaje de confirmación (figura 5.13); al elegir la opción de modificar fecha y hora se muestra una interfaz (figura 5.15) con un calendario y un selector de hora; al elegir los valores y continuar se muestra un mensaje de confirmación (figura 5.17).

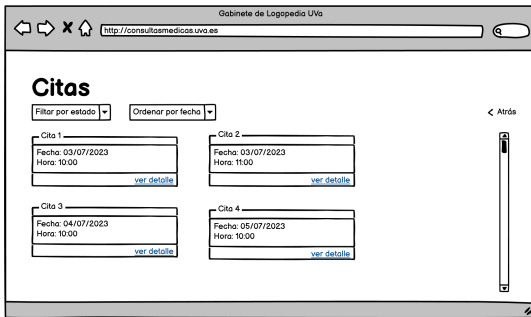


Figura 5.5: Boceto de gestora - consulta de citas.

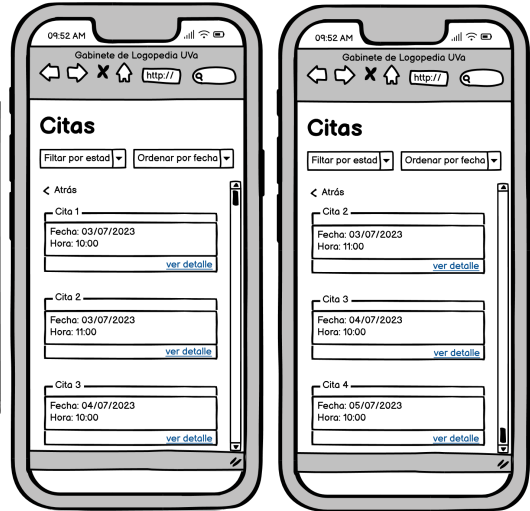


Figura 5.6: Boceto móvil de gestora - consulta de citas.

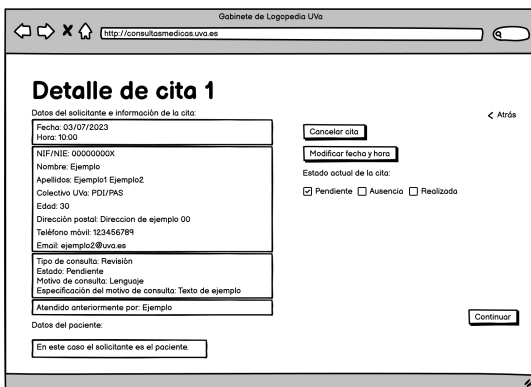


Figura 5.7: Boceto de gestora - detalle de cita 1.



Figura 5.8: Boceto móvil de gestora - detalle de cita 1.

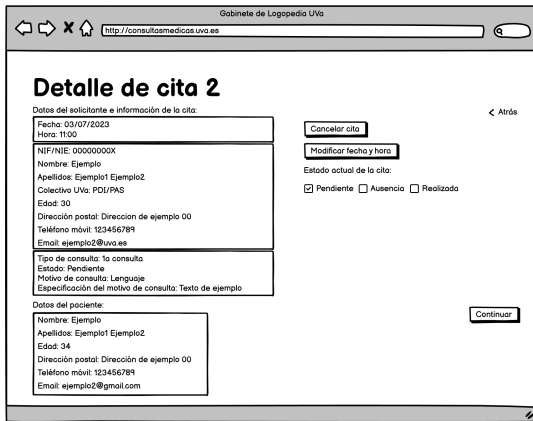


Figura 5.9: Boceto de gestora - detalle de cita 2.



Figura 5.10: Boceto móvil de gestora - detalle de cita 2.

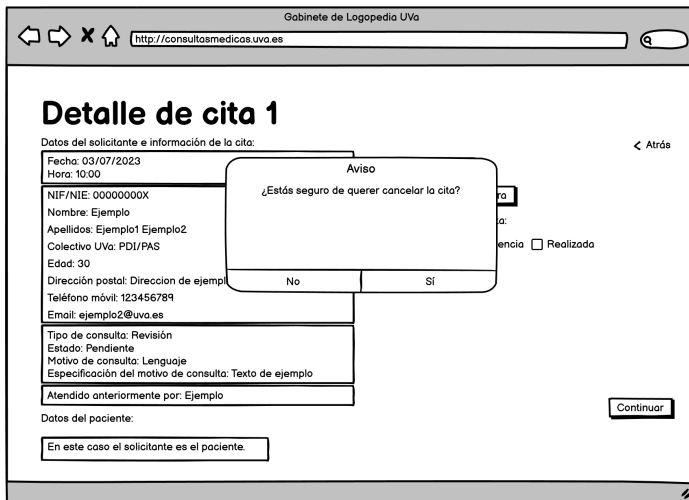


Figura 5.11: Boceto de gestora - confirmación cancelar cita.

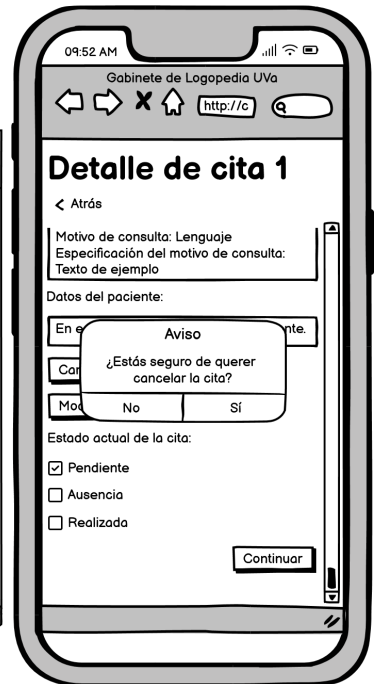


Figura 5.12: Boceto móvil de gestora - confirmación cancelar cita.

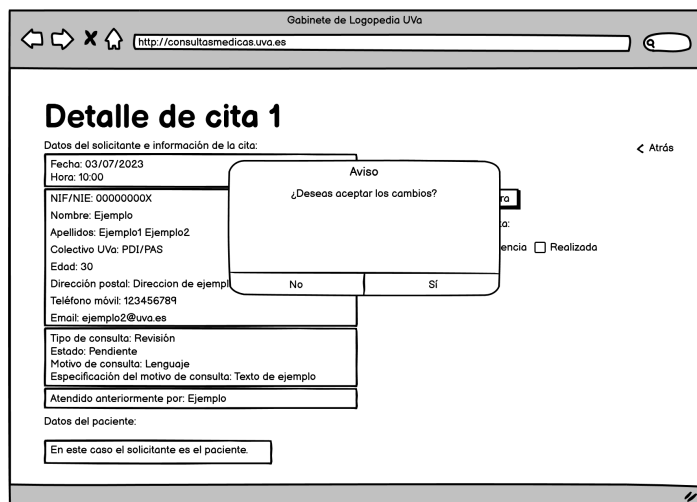


Figura 5.13: Boceto de gestora - confirmación, modificar estado de cita.

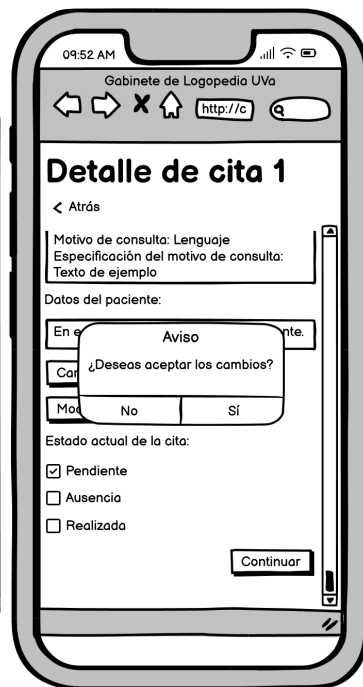


Figura 5.14: Boceto móvil de gestora - confirmación, modificar estado de cita.



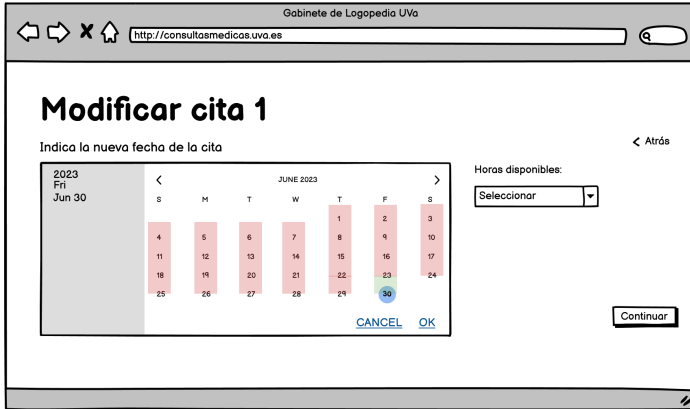


Figura 5.15: Boceto de gestora - modificar fecha y hora de cita.

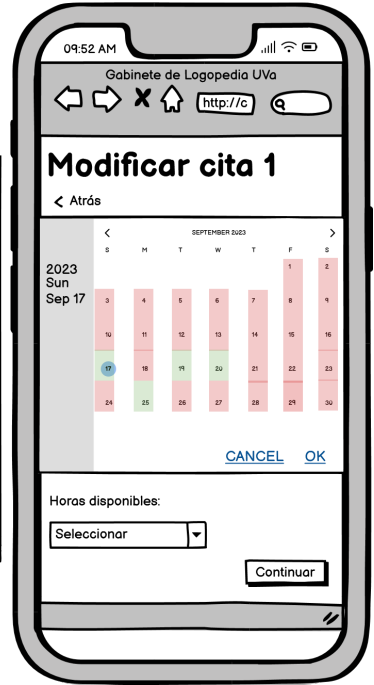


Figura 5.16: Boceto móvil de gestora - modificar fecha y hora de cita.

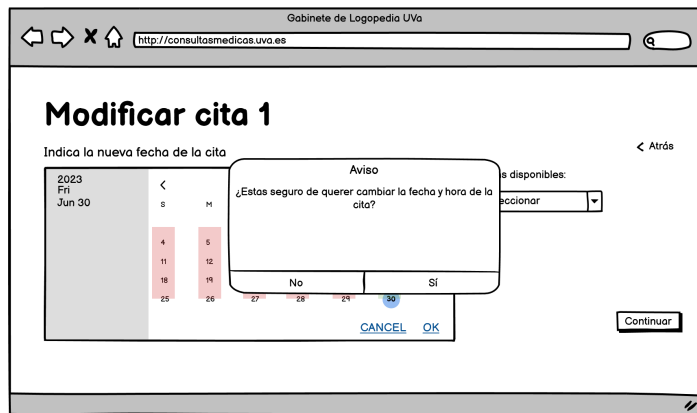


Figura 5.17: Boceto de gestora - confirmación, modificar fecha y hora de cita.



Figura 5.18: Boceto móvil de gestora - confirmación, modificar fecha y hora de cita.

Desde la interfaz de inicio, si pinchamos en la opción de consultar horario de disponibilidad, vemos la interfaz donde se muestra dicho horario de disponibilidad (figura 5.19), el cual tiene la opción de ser modificado, al pinchar para modificarlo; se muestra la interfaz de edición (figura 5.21); en ella se pueden cambiar las horas de inicio y fin de los distintos días, al pinchar en aceptar se muestra un mensaje de confirmación (figura 5.23).

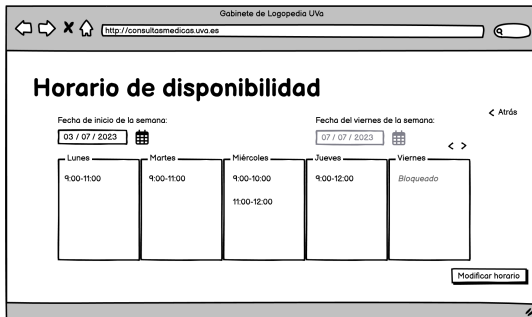


Figura 5.19: Boceto de gestora - horario de disponibilidad.



Figura 5.20: Boceto móvil de gestora - horario de disponibilidad.

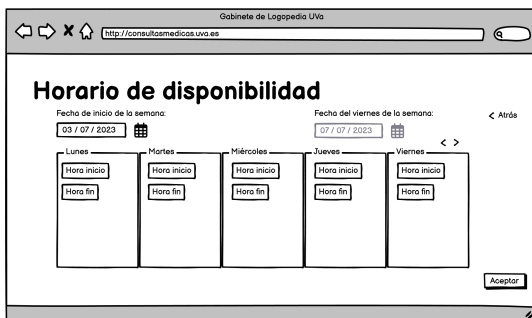


Figura 5.21: Boceto de gestora - modificar horario de disponibilidad.



Figura 5.22: Boceto móvil de gestora - modificar horario de disponibilidad.

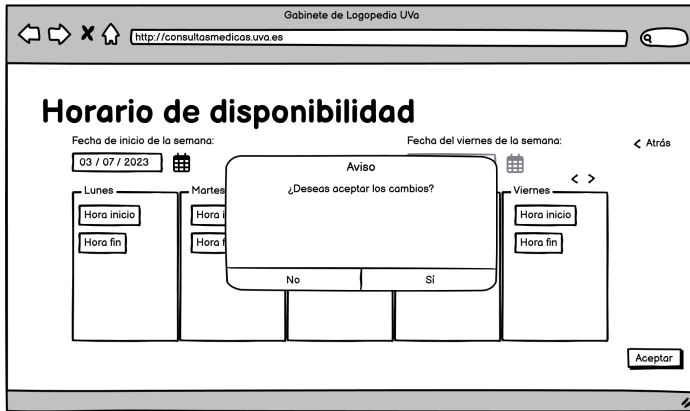


Figura 5.23: Boceto de gestora - confirmación de modificar horario de disponibilidad.



Figura 5.24: Boceto móvil de gestora - confirmación de modificar horario de disponibilidad.

Desde la interfaz de inicio, si pinchamos en la opción de bloquear días, vemos la interfaz de bloqueo de días (figura 5.25), donde hay dos campos en los que se introduce una fecha de inicio y otra de fin; al elegir las fechas se muestra un mensaje de confirmación (figura 5.27) y finalmente al aceptar, se muestra un calendario con el resultado (figura 5.29).

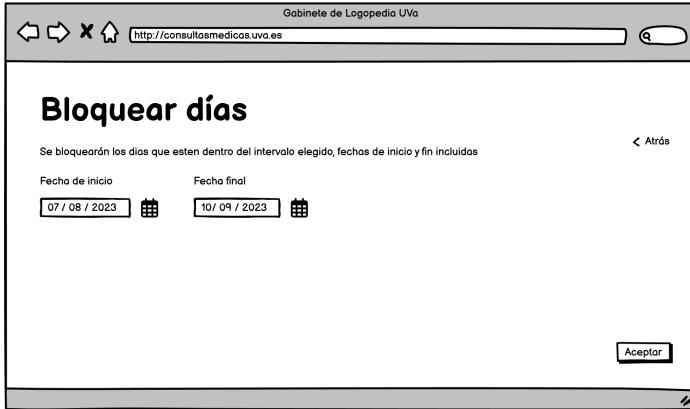


Figura 5.25: Boceto de gestora - bloqueo de días.

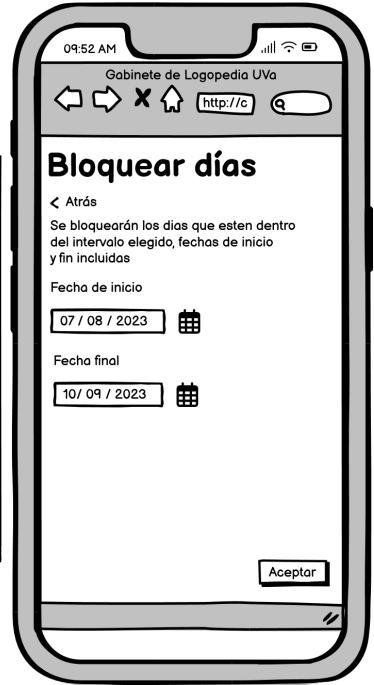


Figura 5.26: Boceto móvil de gestora - bloqueo de días.

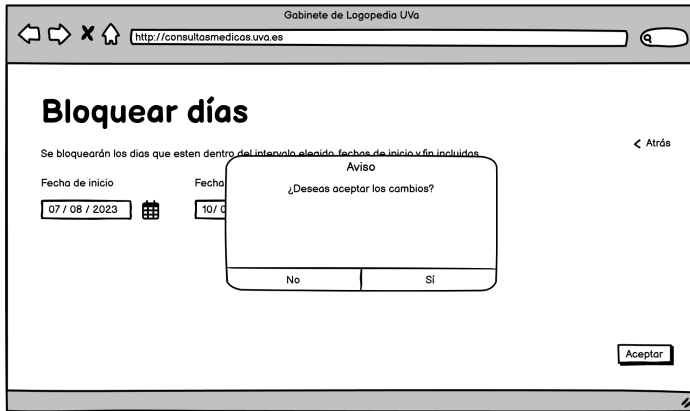


Figura 5.27: Boceto de gestora - confirmación de bloqueo de días.

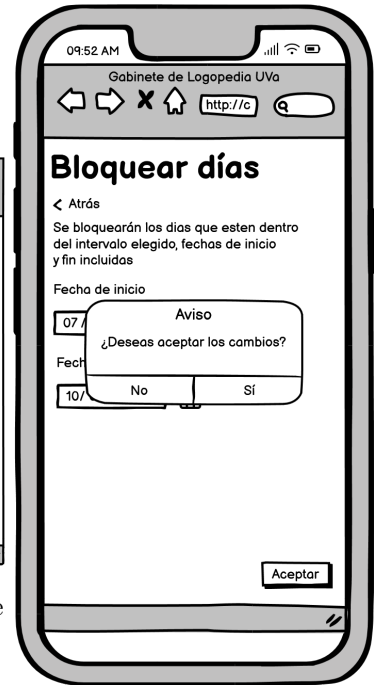


Figura 5.28: Boceto móvil de gestora - confirmación de bloqueo de días.

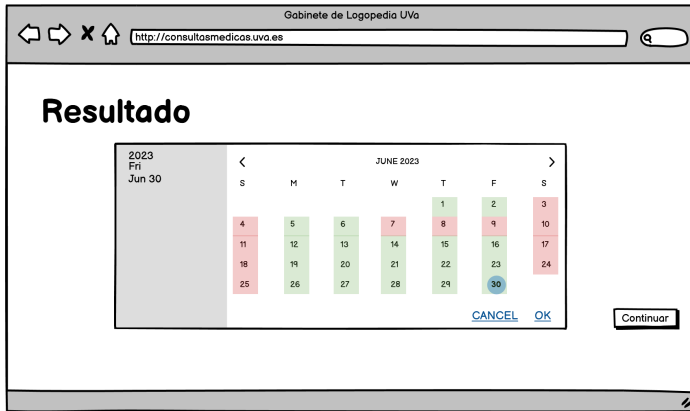


Figura 5.29: Boceto de gestora - resultado del bloqueo de días.



Figura 5.30: Boceto móvil de gestora - resultado del bloqueo de días.

Desde la interfaz de inicio, si pinchamos en la opción de bloquear horas vemos, la interfaz de bloqueo de horas (figura 5.31), donde se introduce una fecha, se muestran las horas actuales y dos campos (el segundo opcional) para poner los nuevos intervalos del horario de disponibilidad; al pulsar en aceptar se muestra un mensaje de confirmación (figura 5.33).

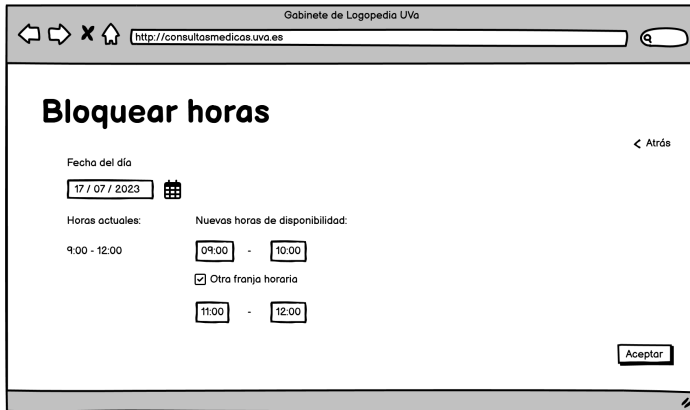


Figura 5.31: Boceto de gestora - bloqueo de horas.



Figura 5.32: Boceto móvil de gestora - bloqueo de horas.



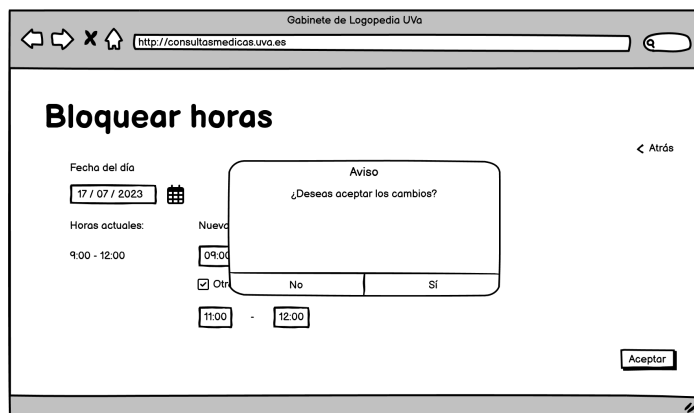


Figura 5.33: Boceto de gestora - confirmación de bloqueo de horas.

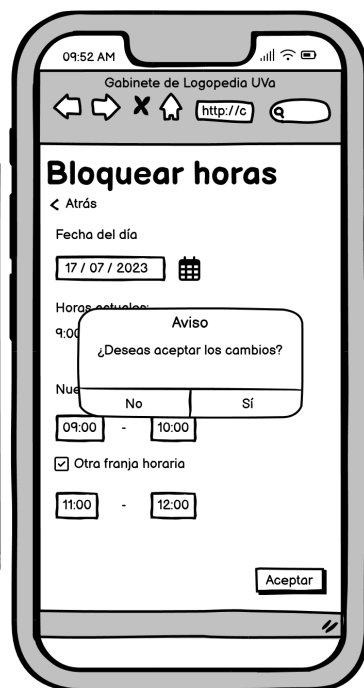


Figura 5.34: Boceto móvil de gestora - confirmación de bloqueo de horas.

Por último, desde la interfaz de inicio, si pinchamos en la opción de nuestro calendario, vemos la interfaz del calendario actual (figura 5.35); en él aparece un calendario con los días de disponibilidad en verde y los bloqueados en rojo. Se puede elegir entre vista mensual (por defecto) y anual (figura 5.37).

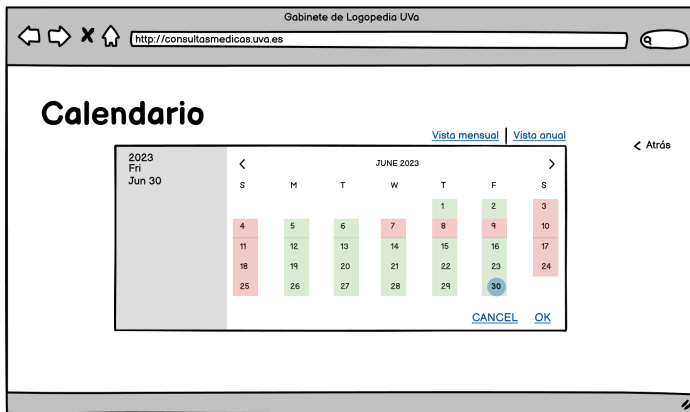


Figura 5.35: Boceto de gestora - calendario mensual.



Figura 5.36: Boceto móvil de gestora - calendario mensual.

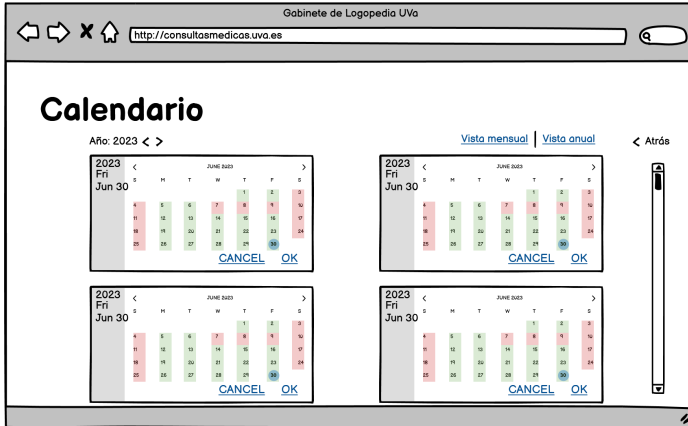


Figura 5.37: Boceto de gestora - calendario anual.



Figura 5.38: Boceto móvil de gestora - calendario anual.

### 5.1.2. Interfaz solicitante

En cuanto a la parte de la interfaz que verán los solicitantes al identificarse, esta es la interfaz inicial; en ella se ven las opciones que se pueden realizar.

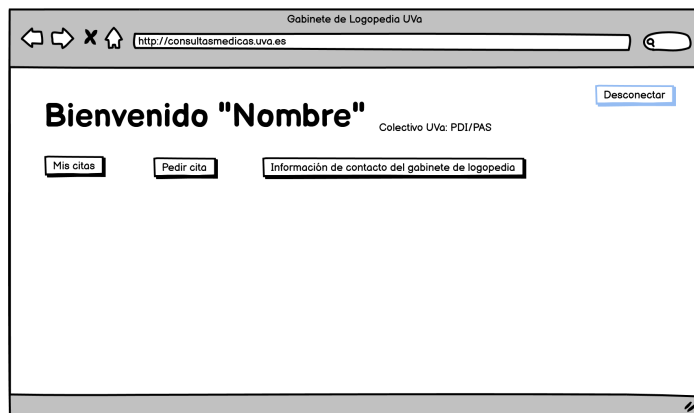


Figura 5.39: Boceto de solicitante - inicio.

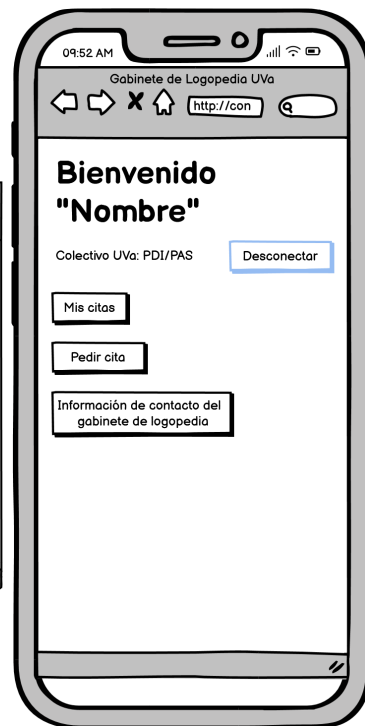


Figura 5.40: Boceto móvil de solicitante - inicio.

Si pinchamos en la opción de mis citas, se muestra la interfaz con las citas pendientes que tiene el solicitante (figura 5.41), la cual es igual que la de ver citas de las gestoras; al pinchar en ver detalles se muestra la interfaz con la información de la cita. Al igual que en las gestoras, la información mostrada está acorde a la información de la cita, se muestran dos ejemplos, uno con una cita de tipo revisión y el solicitante como paciente (figura 5.43) y otra de tipo 1a consulta y el solicitante distinto del paciente (figura 5.45). En este caso la única acción que se puede realizar es cancelar la cita; al pinchar sobre ello aparecerá un mensaje de confirmación (figura 5.47).

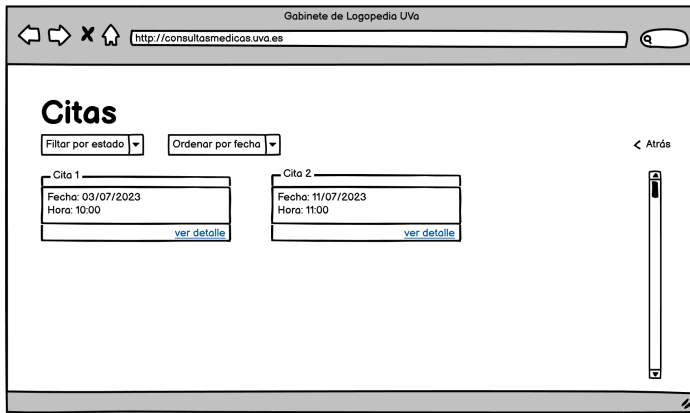


Figura 5.41: Boceto de solicitante - mis citas.



Figura 5.42: Boceto móvil de solicitante - mis citas.

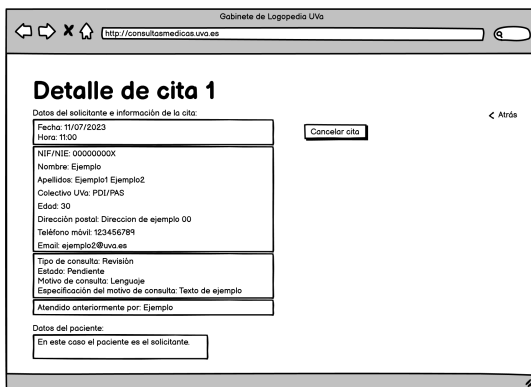


Figura 5.43: Boceto de solicitante - detalle de cita 1.



Figura 5.44: Boceto móvil de solicitante - detalle de cita 1.

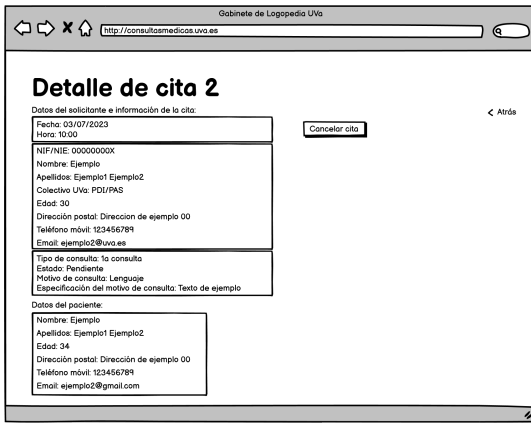


Figura 5.45: Boceto de solicitante - detalle de cita 2.



Figura 5.46: Boceto móvil de solicitante - detalle de cita 2.

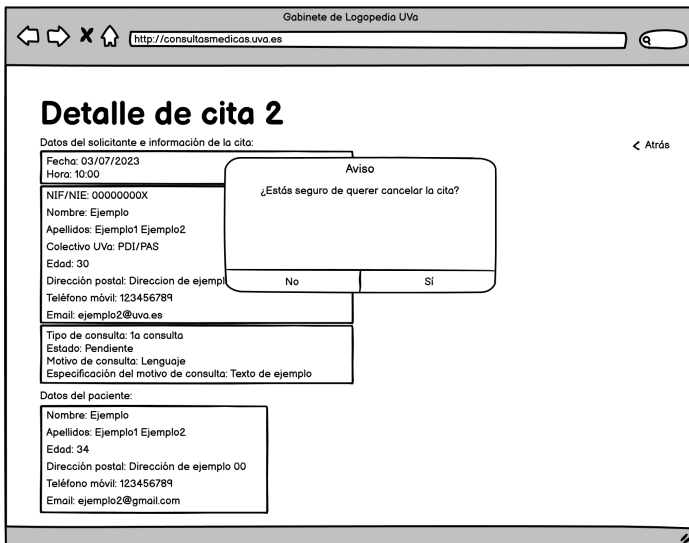


Figura 5.47: Boceto de solicitante - confirmación de cancelar cita.

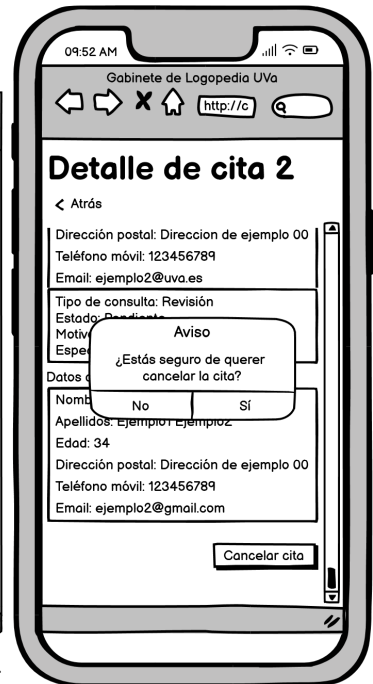


Figura 5.48: Boceto móvil de solicitante - confirmación de cancelar cita.

Desde la interfaz de inicio, si pinchamos en la opción de pedir cita, vemos la interfaz para pedir una cita (figura 5.49); en ella se muestra un calendario con los días disponibles y un selector para seleccionar la hora de la cita. Al pinchar en continuar pueden ocurrir dos escenarios distintos:

1. Solicitante de tipo estudiante (figura 5.51): se muestra la interfaz con los datos del solicitante y un selector con el tipo de consulta.
2. Solicitante de tipo PDI/PAS (figura 5.53): se muestra la interfaz con los datos del solicitante, un selector del tipo de consulta y la opción de añadir un paciente; al marcar la opción aparecen los datos a rellenar sobre el paciente

Sea cual sea el escenario, la interfaz que se muestra al continuar depende del tipo de consulta seleccionado. En el caso de 1a consulta se muestra una interfaz con más datos a rellenar (figura 5.55) y en el caso de revisión se muestra la misma, pero con un campo adicional a rellenar (figura 5.57).

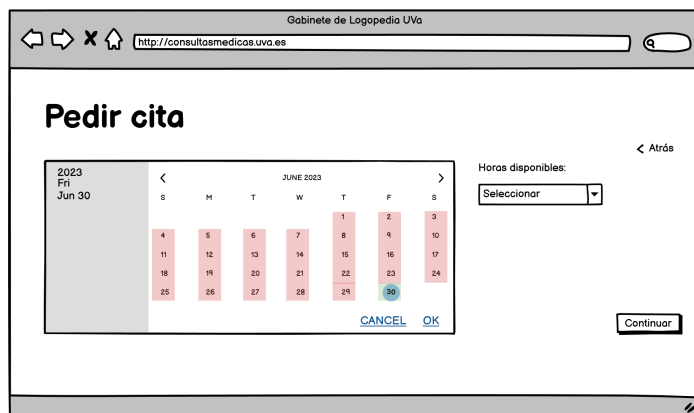


Figura 5.49: Boceto de solicitante - pedir cita, calendario.



Figura 5.50: Boceto móvil de solicitante - pedir cita, calendario.

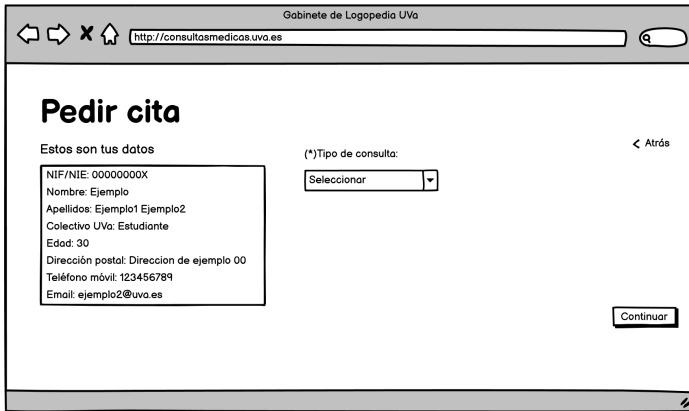


Figura 5.51: Boceto de solicitante - pedir cita para estudiante.

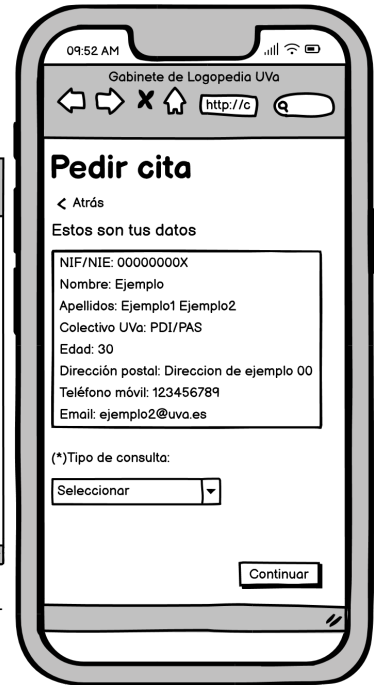


Figura 5.52: Boceto móvil de solicitante - pedir cita para estudiante.

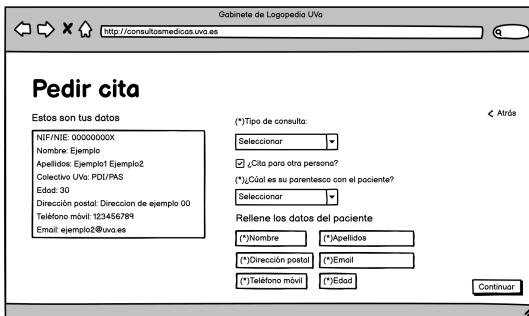


Figura 5.53: Boceto de solicitante - pedir cita para PDI/PAS.



Figura 5.54: Boceto móvil de solicitante - pedir cita para PDI/PAS.



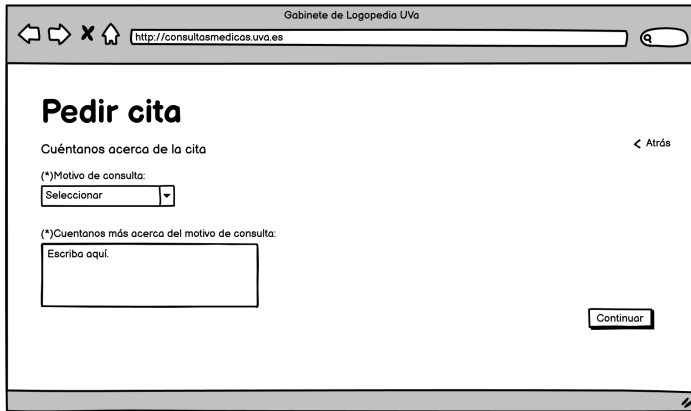


Figura 5.55: Boceto de solicitante - pedir cita 1a consulta.

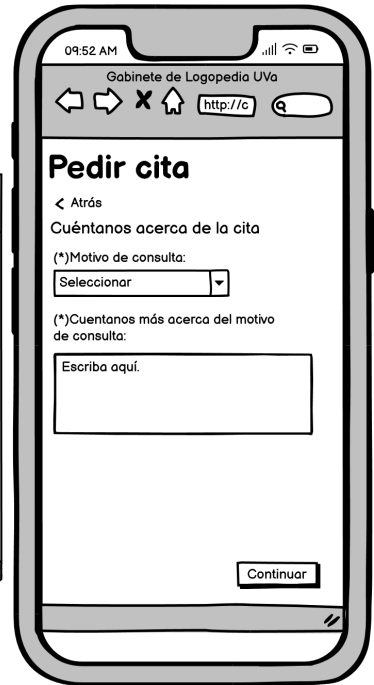


Figura 5.56: Boceto móvil de solicitante - pedir cita 1a consulta.

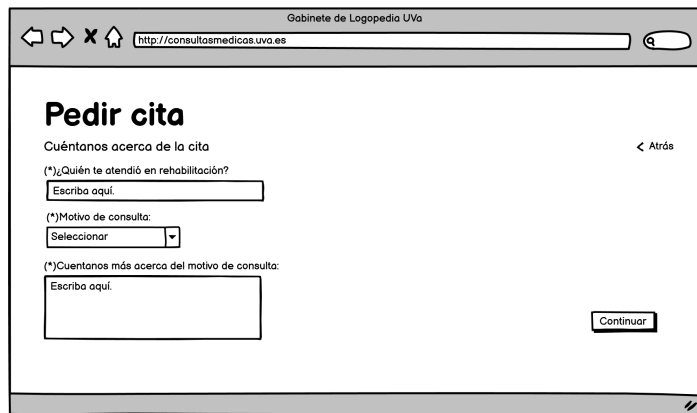


Figura 5.57: Boceto de solicitante - pedir cita revisión.

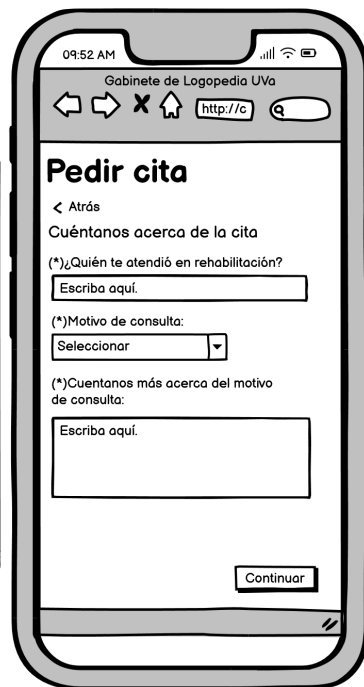


Figura 5.58: Boceto móvil de solicitante - pedir cita revisión.

Finalmente, desde la interfaz de inicio, si pinchamos en la opción de información de contacto del Gabinete de Logopedia, se muestra la interfaz con dicha información.

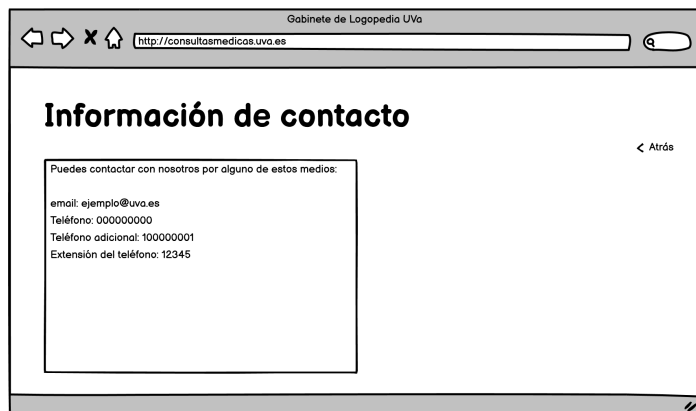


Figura 5.59: Boceto de solicitante - información de contacto del Gabinete de Logopedia.

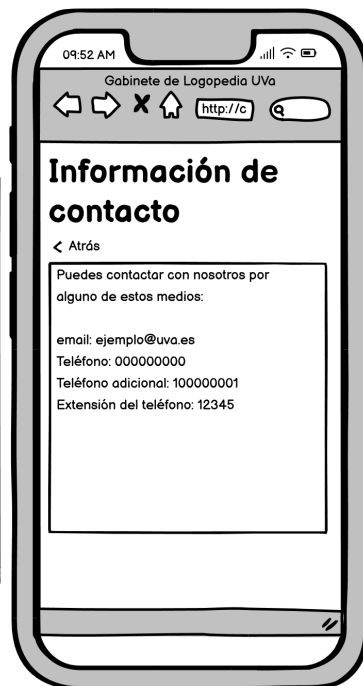


Figura 5.60: Boceto móvil de solicitante - información de contacto del Gabinete de Logopedia.

## 5.2. Arquitectura del sistema

Se ha optado por implementar la "Clean Architecture" de Robert C. Martin [33][37]. Esta elección se basa en las ventajas que ofrece en términos de mantenibilidad y escalabilidad del proyecto.

Esta arquitectura se caracteriza por cumplir los cinco principios SOLID:

1. Single Responsibility [56]: Cada clase debe de tener una única responsabilidad o funcionalidad.
2. Open/Closed Principle [54]: Las clases deben de estar abiertas para la extensión pero cerradas para la modificación.
3. Liskov Substitution [58]: Los objetos de una superclase deben poder ser sustituidos por objetos de sus subclases sin alterar el funcionamiento.
4. Interface Segregation [57]: Los clientes de un programa solo deben de conocer los métodos que utilizan, no los que no necesitan.

5. Dependency inversion [55]: Los módulos de alto nivel no deben depender de los de bajo nivel; ambos deben depender de las abstracciones. Las abstracciones no deben depender de las implementaciones, sino al revés.

La arquitectura se organiza en capas (Figura 5.61), cada una con una responsabilidad específica. Estas capas están dispuestas de manera que las superiores dependen de las inferiores, pero nunca al revés:

1. “Entities”: La capa más interna y por ello, la única independiente. Encapsula las reglas de negocio y puede estar compuesta por objetos con métodos o estructuras de datos y funciones.
2. “Use Cases”: Contiene reglas de negocio específicas para la aplicación. Dentro de ella se encapsulan e implementan los casos de uso.
3. “Interface Adapters”: Se encarga de adaptar los formatos de los datos de las capas superiores y los agentes externos, y utilizarlos tanto en la base de datos como en las vistas.
4. “Frameworks & Drivers”: La capa más externa, está compuesta de frameworks y herramientas como la configuración de la base de datos y el framework web.

La interacción entre las capas se hace a través de “Adapters”. Se tratan de interfaces ubicadas dentro de cada capa, y sirven tanto de entrada, como de salida. Las interfaces están implementadas en la capa que realiza las actividades deseadas. En cuanto a los datos que se envían a través de los “Adapters” se tratan de tipos de datos simples u objetos creados específicamente para cruzar las capas.

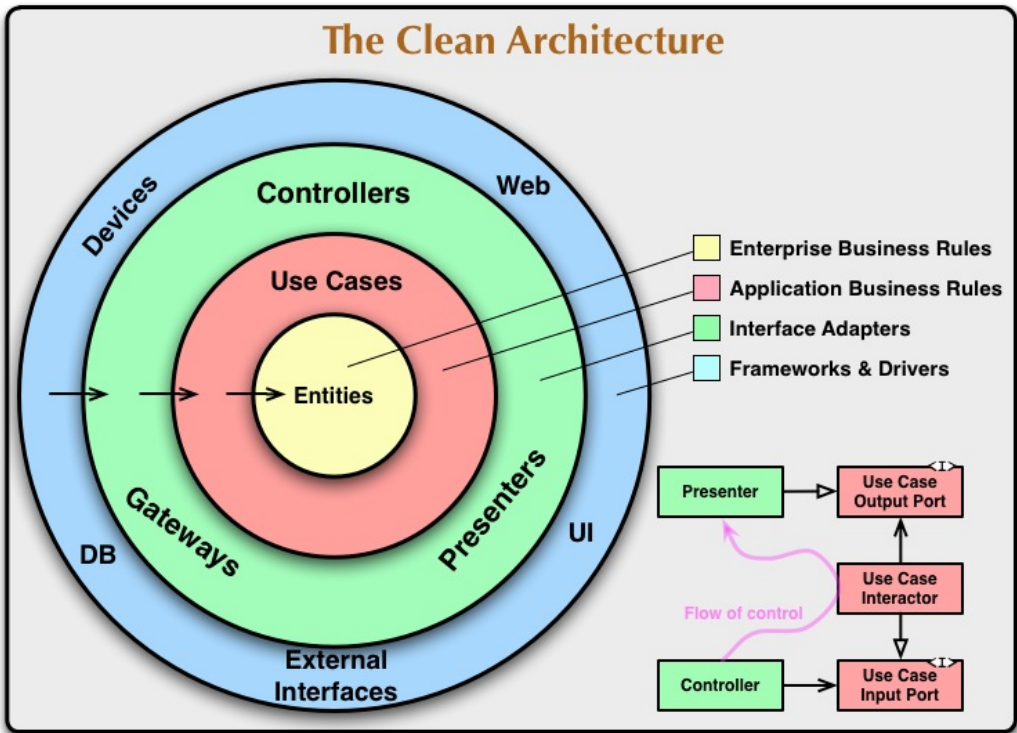


Figura 5.61: Capas de la "Clean Architecture". Tomada de [37]

A continuación se muestra el diagrama de dependencias (Figura 5.62). En él se pueden apreciar los distintos *packages* que utilizará la aplicación. En el caso de la capa *Frameworks/Drivers*, la más exterior, no depende de ninguna otra; la siguiente capa, *Interface Adapters*, depende de la capa superior e implementa algunos de sus *packages*; la siguiente capa, *Use cases*, depende de la superior; finalmente la capa central, *Entities*, no depende de ninguna otra. Para cruzar entre capas se utilizan los *boundaries* de la capa *Use Cases*. Junto con los modelos, se encargan de llevar la información de una capa a otra. Estos *boundaries* son suficientes porque la interacción entre capas es entre *Interface Adapters* y *Use Cases*.

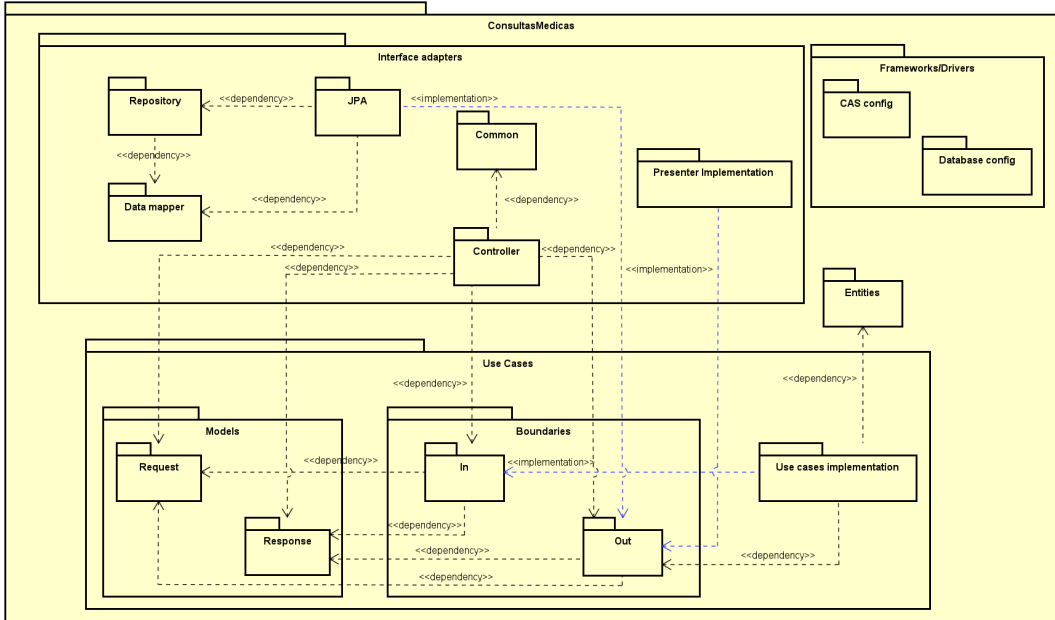


Figura 5.62: Diagrama de dependencias de la arquitectura.

El proceso habitual de una solicitud se muestra en la Figura 5.63. La solicitud se introduce directamente en el *controller* junto con un modelo de datos (*RequestModel*), donde se realiza una revisión inicial de los datos de entrada. Luego, la solicitud se dirige a la implementación del caso de uso mediante el *input boundary*. Aquí se aplican las reglas de negocio del caso de uso y de los modelos (*entities*), y se llevan a cabo las acciones necesarias en la base de datos utilizando el *JPA* a través del *out boundary*. Finalmente, se prepara la vista de respuesta con el *Presenter Implementation* a través del *out boundary*.

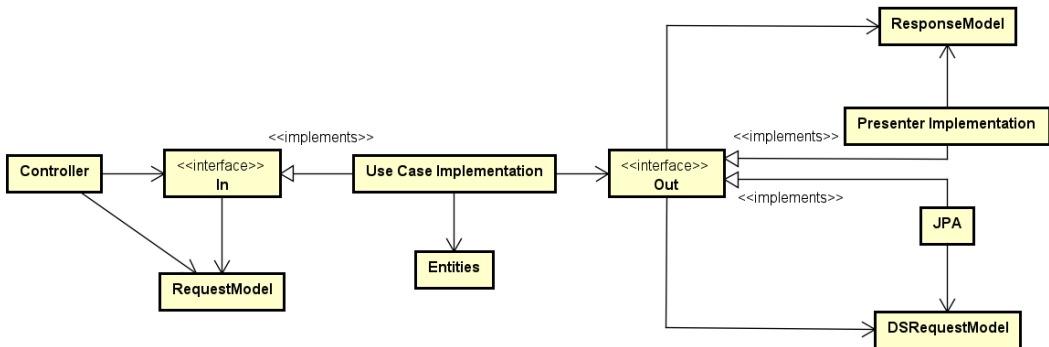


Figura 5.63: Clases tipo participantes en un flujo de trabajo habitual en la arquitectura.

### 5.3. Diagrama de despliegue

Se ha diseñado la arquitectura de despliegue siguiendo una interfaz API REST.

El frontend está compuesto solamente de los archivos que forman la interfaz gráfica con la que interactúa el usuario.

Se ha introducido una API Gateway. La API Gateway [39] es un gestor que actúa de intermediario en las peticiones entre frontend y backend aplicando políticas, verificación y control de acceso. De esta forma se aporta una capa más de seguridad.

La parte backend está compuesta de 4 componentes para cada uno de los servicios de la aplicación: gestión de reservas, solicitantes, pacientes y autenticación. El servicio de autenticación proveerá de tokens que permiten el acceso a los demás componentes. Se han introducido 3 bases de datos para almacenar la información de reservas, solicitantes, pacientes.

En la Figura 5.64 se muestra el respectivo diagrama de despliegue. Cabe destacar que este diagrama de despliegue sufrió cambios durante la implementación. En el capítulo de implementación puede verse el diagrama final y la justificación de los cambios (Sección 6.4).

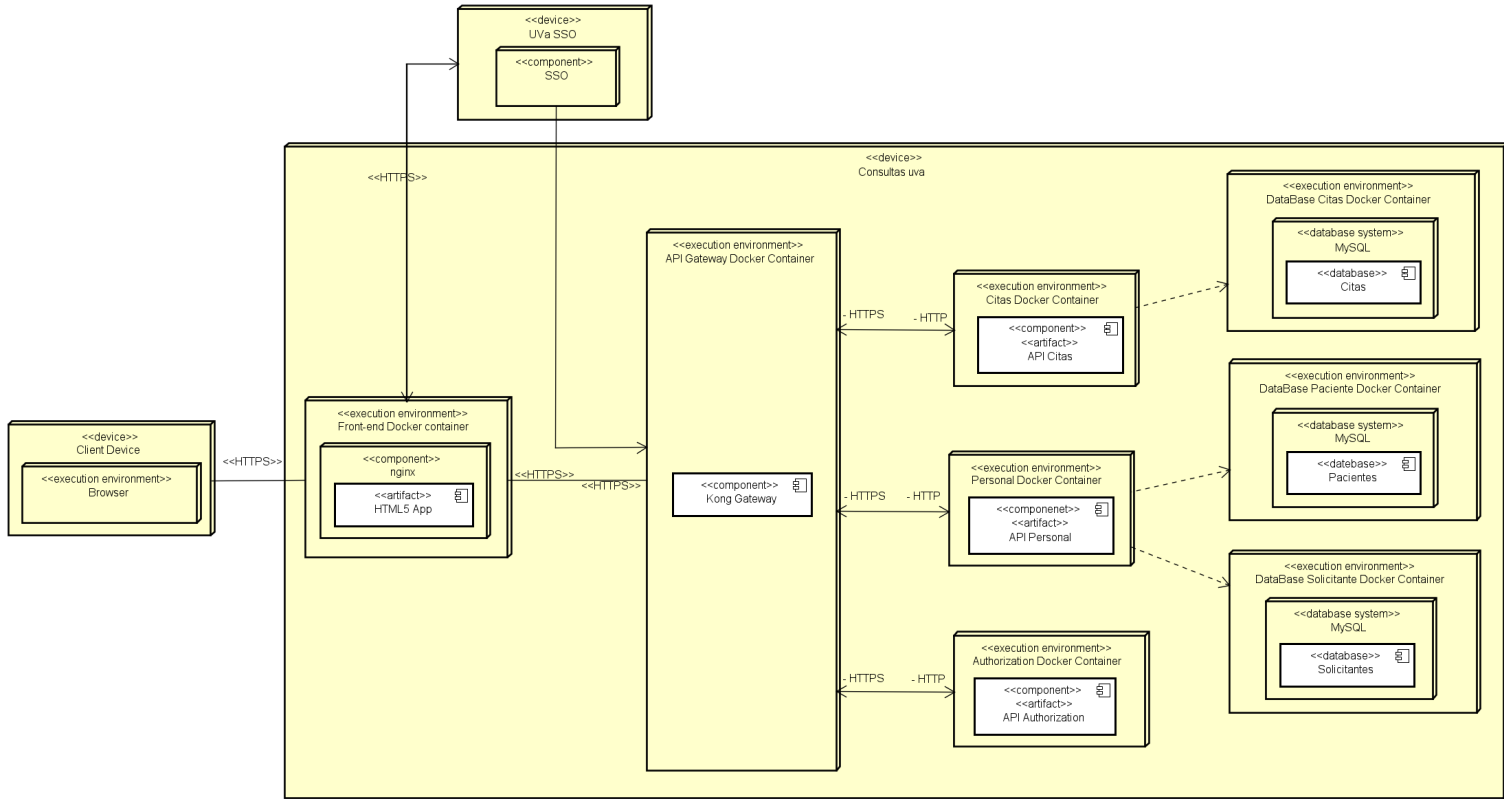


Figura 5.64: Diagrama de despliegue inicial, para la fase de diseño.



## 5.4. Modelo de bases de datos

En esta sección se muestra el modelo de bases de datos (Figura 5.65) obtenido a partir del modelo de dominio (Figura ??), adaptándolo a unas bases de datos relacionales. Se ha distribuido en tres bases de datos: Solicitantes, pacientes y citas. Para relacionar tablas entre bases de datos distintas; se hace de manera simbólica, añadiendo el id de la tupla objetivo como atributo. Por ejemplo, las tablas “Pacientes” y “Usuarios” se encuentran en bases de datos distintas, para relacionarlas se añade un atributo “idUsuario” en la tabla “Paciente”.

Cabe destacar que este modelo sufrió cambios durante la implementación. En el respectivo capítulo de implementación puede verse el modelo final y la justificación de los cambios(Sección 6.4).

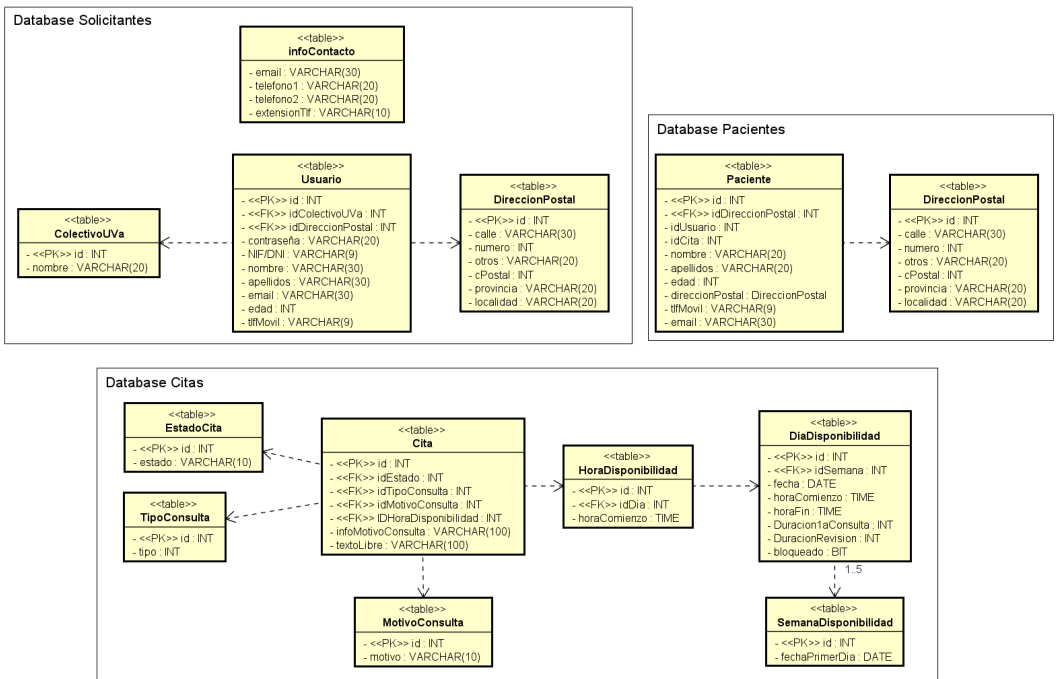


Figura 5.65: Modelo de bases de datos inicial, para la fase de diseño.



## Capítulo 6

# Implementación y pruebas

La aplicación final se ha desarrollado utilizando un frontend con HTML5, CSS, JavaScript y Thymeleaf; un backend con Java 11 (Spring Boot) y Thymeleaf, bases de datos con MySQL y un despliegue con Docker en una máquina virtual de la UVa. En este capítulo se detallarán estas implementaciones y las pruebas realizadas.

### 6.1. Bases de datos

Las bases de datos MySQL que se crearon para este proyecto son Citas, Usuarios y Pacientes, y cada una se aloja en un contenedor diferente.

Para su implementación, surgieron algunas complicaciones. Spring Boot, por defecto, utiliza la base de datos definida en el archivo “application.properties” para realizar todas las consultas. Por lo tanto, añadir dos bases de datos adicionales y personalizar cuál utilizar según la circunstancia, requería de una configuración especial [17].

Se creó un directorio “database” dentro de la capa de frameworks & drivers. En este directorio, se añadieron tres archivos de configuración, uno para cada base de datos.

Cada configuración contiene: un método que produce un “Entity Manager” para escanear el paquete donde están situadas las clases “Entity” que le corresponden, un método que produce un “DataSource” que obtiene la URL y las credenciales de acceso a la base de datos, un método para establecer un vínculo entre el primer “Entity Manager” con el “DataSource”. Adicionalmente, las bases de datos que requieren datos predefinidos (Appointments y Users), cuentan con un método adicional llamado “Initializer”. Este método se encarga de inicializar los datos a partir de un archivo *.sql* al desplegar la aplicación.

Para llevar a cabo las consultas, inserciones, actualizaciones y eliminaciones en las bases de datos, se ha optado por utilizar “JPA repository”, ya que es una especificación ampliamente

utilizada y facilita estas operaciones. Al utilizar varias bases de datos, para que cada “repository” apuntara hacia la base de datos que le corresponde, se añadió la etiqueta “Qualifier” con el nombre del “DataSource” de la base de datos.

El modelo de la base de datos cambió drásticamente respecto del diseñado inicialmente, el diagrama final está detallado en la sección de dificultades y decisiones 6.4.

## 6.2. Seguridad

Dado que la parte de seguridad tiene relevancia en este proyecto, en este apartado se detallarán las medidas de seguridad implementadas.

### 6.2.1. Single Sign-On

El inicio de sesión en la aplicación se realiza a partir del Single Sign-On (SSO) de la UVa. Esto permite a los usuarios acceder utilizando sus credenciales de la UVa, al tiempo que obtenemos datos relevantes sobre el usuario para nuestra aplicación.

Como solución más específica de SSO, se utilizará CAS (Central Authentication Service) [19], que es el utilizado por la UVa. Este tipo de tecnología permite un inicio de sesión centralizado, lo que permite a los usuarios acceder a múltiples aplicaciones y gestionar sus accesos de manera independiente. Su funcionamiento está compuesto por tres componentes: navegador cliente, aplicación web y servidor CAS. El funcionamiento habitual es el siguiente: un cliente visita una web que requiere de autenticación, la web redirige al usuario al servidor CAS, el usuario introduce sus credenciales, el servidor CAS las valida frente a su base de datos y redirige al usuario a la web original junto con un ticket de seguridad, la aplicación web valida el ticket contra el servidor CAS y, si es válido devuelve la información del usuario y da el usuario como autenticado.

Para que la UVa nos permitiera utilizarlo, fue necesario que habilitaran en su servidor CAS de reproducción: la dirección principal del proyecto (<http://157.88.125.198:8080/consultasmedicas>) los datos que serán devueltos para esta aplicación y los usuarios para los que se permitía realizar pruebas: el estudiante, la tutora y una gestora.

Para su implementación, el STIC de la UVa nos proporcionó un pequeño proyecto con la implementación y esta ha sido adaptada para nuestro TFG. La implementación se basa en la herramienta Spring Security CAS [34]. La implementación por capas es la siguiente:

- En la capa “frameworksDrivers”, se añadió un archivo de configuración del CAS y un archivo que establece las condiciones de acceso a las distintas rutas:
  - `CasAdasConfig.java`: Establece la información de contacto con el servidor CAS de la UVa. Entre ellos, las URL del servidor de la UVa: inicio de sesión, validación de tickets y desconexión de sesiones. Además de las URL de nuestra aplicación: retorno después del iniciar sesión, y desconexión de sesión.

- `WebSecurity.java`: Establece los filtros para acceder a las direcciones de la aplicación. Estos filtros controlan los accesos en función de que los usuarios estén identificados o no, y sus roles.
- En la capa “`interfaceAdapters`” se añadió un controlador para recibir la respuesta del servidor CAS de la UVa con el ticket de la sesión. Este controlador maneja el endpoint “`/secured`”.
- En la capa “`useCases`” se añadió un servicio “`CASLoginCU`”. Este servicio implementa la interfaz de Spring Boot “`AuthenticationUserDetailsService`” para validar el ticket en el servidor CAS de la UVa y cargar la información del usuario en la sesión de nuestra aplicación.
- En la capa “`entities`”, se añadió la clase “`MyUser`”. Esta clase extiende la clase estándar de Spring Boot “`User`” y trabaja en conjunto con “`AuthenticationUserDetailsService`” para cargar la información básica del usuario en el “`SecurityContextHolder`” de la sesión.

Una vez implementado el inicio de sesión, el funcionamiento se desarrolla de la siguiente manera:

1. Usuario sin autenticar: Al pulsarse el inicio de sesión el sistema detecta que no hay una sesión iniciada y realiza una redirección al SSO en <https://idpre.uva.es/CAS/login?service=http://157.88.125.198:8080/consultasmedicas/secured>; en ella el usuario introduce sus credenciales de la UVa, pulsa aceptar y se realiza una petición POST en la misma dirección con las credenciales como cuerpo.
2. Validación de credenciales por el servidor CAS: El servidor CAS de la UVa, al comprobar que las credenciales son correctas, realiza una redirección a la dirección del proyecto <http://157.88.125.198:8080/consultasmedicas/secured?ticket=ST-XXXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX>, donde incluye un ticket como parámetro de la petición.
3. Inicio de sesión en la aplicación: Después, la aplicación hace una petición al servidor CAS para validar el ticket. El servidor CAS, al validar el ticket, guarda el valor de la sesión (también ubicado como cookie en el navegador) y responde con la información del usuario. Al recibir la respuesta, la aplicación guarda el ticket, inicia una sesión para el usuario con el valor de la cookie, guarda la información de forma temporal en el “`SecurityContextHolder`” de Spring Boot; el servidor envía al cliente un token csrf, que utilizará junto con el id de la sesión en las peticiones que lo requieran.

La desconexión de una sesión se desarrolla de la siguiente manera: la aplicación elimina cualquier información almacenada en la sesión activa del usuario; la aplicación envía una petición al servidor CAS indicando una desconexión junto con la sesión; el servidor CAS invalida la sesión.

El acceso a las distintas rutas de la aplicación, además de estar restringido para los usuarios autenticados, está restringido según los roles del usuario. Cuando el usuario inicia

sesión, se le asigna un rol según su colectivo UVa. Los roles son los siguientes: “ESTUDIANTE”, “PDIPAS” y “GESTOR”. Las rutas de la aplicación, partes mostradas de la interfaz del navegador y acciones que tienen permitido hacer cada usuario están definidas y protegidas según su rol.

#### 6.2.2. Diseño para evitar inyecciones de código

Se denominan inyecciones SQL [28] a consultas SQL introducidas a partir de los datos de entrada que aporta el cliente a la aplicación. Estas consultas podrían permitir al cliente leer datos sensibles, modificarlos, ejecutar acciones de administrador e incluso recuperar la información de un fichero presente en la base de datos. Como medida de defensa [29] para esta amenaza, se utilizan “Prepared Statements”. Teniendo en cuenta que las consultas a la base de datos se llevan a cabo con los repositorios de JPA, funcionan con “Prepared Statements”, lo que permite que los datos introducidos por el cliente sean tratados como una cadena de caracteres completa y no como un comando.

Los ataques XSS (Cross Site Scripting)[26] se tratan de inyecciones de scripts que realiza un atacante a través de una aplicación web hacia un usuario diferente. El usuario desconoce que está siendo atacado y ejecutará los scripts confiando en que provienen de la aplicación a la que accedió. Como medida de defensa para esta amenaza, se utiliza la herramienta de “OWASP Encode” [27]. Con esta herramienta se codifican los parámetros de entrada que pueden ser utilizados como scripts de Java, se codifican los datos de salida que pueden ser utilizados como scripts de HTML y JavaScript.

Se ha implementado una clase de uso común denominada *Sanitize*, donde se han añadido las medidas anteriores. Además de estas medidas, se ha implementado un método en la clase *Sanitize* para desinfectar todas las entradas y salidas suprimiendo caracteres sospechosos como: salto de línea, retroceso, tabulación, avance de página, comillas, punto y coma, etc.

#### 6.2.3. git-crypt

Para proteger los datos sensibles de la aplicación que se encuentran en el repositorio, se ha utilizado git-crypt [2]. Permite seleccionar un número de archivos y cifrarlos en *AES-256* dentro del repositorio. Para trabajar con el repositorio en local y descifrar los archivos, se utiliza una clave generada por el propio git-crypt.

### 6.3. Estructura del código

Durante el desarrollo y como lugar de almacenaje final, se ha utilizado un repositorio en el GitLab de la UVa (<https://gitlab.inf.uva.es/enrojo/tfg-consultasmedicas>).

En el nivel superior de la estructura se encuentran el directorio “src”, contiene el código de la aplicación, que será examinado con mayor detalle más adelante. Además, se identifican

diversos archivos individuales que conforman el resto del repositorio.

1. “.gitattributes”: Contiene los nombres de los archivos que están cifrados mediante git-crypt (6.2.3).
2. “.gitignore”: Enumera los nombres de los archivos locales que se excluyen dentro del repositorio.
3. “Dockerfile”: Destinado a la preparación del contenedor backend.
4. “Makefile”: Permite gestionar los contenedores Docker de manera sencilla y rápida.
5. “Readme.md”: Breve documentación del proyecto, escrita en formato Markdown.
6. “docker-compose.yml”: Fichero que orquesta las imágenes y contenedores de Docker.
7. “docker-compose.override.yml”: Complementa el “docker-compose.yml” con información adicional, en este caso, datos sensibles para el acceso a las bases de datos.
8. “mvnw” y “mvnw.cmd”: Archivos de Maven, utilizados para construir el proyecto.
9. “pom.xml”: Archivo Maven, incluye las dependencias y propiedades del backend.

Dentro del fichero “src” se encuentra el directorio “test/java/edu/uva/consultasmedicas”, normalmente contiene los test automatizados para java, pero debido a que en este proyecto no se han desarrollado pruebas automatizadas, no se tendrá en cuenta estos ficheros. El segundo fichero en “src” es “main”; dentro de él se encuentran dos directorios “java/edu/uva/consultasmedicas” y “resources”. Dado que ambos son el grosor del proyecto, se detallarán por separado, explicando como se han implementado y sus contenidos.

En primer lugar, “java/edu/uva/consultasmedicas”. Contiene el backend en java (Spring Boot) de la aplicación. Está organizado según la arquitectura *clean*. A continuación se detalla cada uno de los paquetes y su implementación.

1. “entities”: Se trata de la capa central, no depende de ninguna otra. Contiene los modelos de las entidades que precisan de reglas de negocio. De esta forma cada clase tiene funciones que validan las instancias.
2. “frameWorksDrivers”: Se trata de la capa exterior. En este caso no depende de ninguna otra. Contiene las clases de configuración de las bases de datos y el CAS (SSO).
  - a) “cas\_config”: Contiene las configuraciones del SSO con autenticación CAS, junto con los permisos de los roles de usuario.
  - b) “database\_config”: Contiene la configuración de las bases de datos.
3. “InterfaceAdapters”: La siguiente capa después de “frameWorksDrivers”. contiene los siguientes directorios:

- a) “JPA”: Implementa los “out boundaries” (capa “useCases”) de tipo “dataSource”. Permite transformar los modelos de la capa superior a los “dataMapper”, después interactúa con los repositorios para hacer las consultas SQL. Existe uno por cada “DataMapper”.
  - b) “common”: Contiene la clase “Sanitize” 6.2.2, utilizada por otras clases dentro de esta capa.
  - c) “controller”: Contiene los controladores de cada caso de uso, realizan una validación básica de los datos de entrada, interactúan con el “input boundary” (capa “useCases”) y finalmente reciben la respuesta del “presenter” para añadirla a la vista con Thymeleaf. Existe uno por cada caso de uso.
  - d) “dataMapper”: Contiene los modelos que residen en cada una de las tres bases de datos;
  - e) “presenterImplementation”: Implementa los “out boundaries” (capa “useCases”), transforma los datos recibidos de la capa “useCases” y los prepara para que sean añadidos con Thymeleaf.
  - f) “repository”: Contiene los repositorios “JPA” para hacer las consultas en las tres bases de datos.
4. “useCases”: La capa siguiente a “InterfaceAdapters” y previa a “entities”, contiene los “boundaries”, la implementación de los casos de uso y los modelos de solicitud y respuesta.
- a) “boundaries”: Se tratan de interfaces y son los puntos de entrada y salida de la capa.
    - 1) “in”: Punto de entrada a la capa. Existe uno por cada caso de uso.
    - 2) “out/dataSource”: Punto de salida de la capa, para las operaciones en base de datos. Existe uno por cada modelo de la base de datos.
    - 3) “out/presenter”: Punto de salida de la capa, para enviar los datos del resultado del caso de uso. Existe uno por cada caso de uso.
  - b) “models”: Contiene los modelos utilizados para la salida y entrada de la capa.
    - 1) “request”: Modelos de solicitud de datos, utilizados tanto para la entrada de datos en los casos de uso (“in”), como para solicitar operaciones en base de datos (“out/dataSource”).
    - 2) “response”: Modelos de respuesta de datos, utilizados en su mayoría por los “presenter”.
  - c) “useCaseImplementation”: Implementa los “in boundaries”, realizan las comprobaciones de las reglas de negocio, tanto del caso de uso como de la capa “entities”, y orquestan las llamadas a los “out boundaries”.
5. “ConsultasMedicasApplication.java”: Se trata de la clase main del proyecto; dado que se utiliza Spring Boot, esta clase solo realiza un *SpringApplication.run()* para poner en marcha la aplicación.

En segundo lugar, “resources”. Contiene el frontend y algunas configuraciones del backend.



1. “static”: Contiene los archivos CSS, JavaScript y las imágenes utilizadas.
2. “templates”: Contiene las plantillas HTML preparadas ser procesadas por Thymeleaf.
3. “application.properties”: Contiene las propiedades de Spring Boot. Propiedades de bases de datos, SSO, hibernate y JPA.
4. “data-appointments.sql”: Contiene consultas SQL para insertar datos cuando la aplicación se inicia. Específicas para la base de datos de citas.
5. “data-users.sql”: Contiene consultas SQL para insertar datos cuando la aplicación se inicia. Específicas para la base de datos de usuarios.

## 6.4. Dificultades y decisiones

Durante la implementación acontecieron varias dificultades y se tomaron decisiones que difieren del diseño previamente realizado. Aquí se detallan:

1. Despliegue: Se tomaron varias decisiones que afectaron al despliegue (despliegue final en Figura 6.1).
  - a) Frontend con Nginx: En un principio se planificó alojar el frontend de la aplicación en un Nginx, separado del backend para una mayor seguridad. Al comienzo de la implementación se realizó así, pero, al conocer que este proyecto finalmente no se implantaría en la UVa, durante la implementación del SSO, se decidió unificarlo con Spring Boot. El proyecto cedido por el STIC de la UVa que incluía la integración del SSO, unificaba el frontend con Spring Boot, permitiendo también la integración con Thymeleaf, por lo que se tomó como referencia y se implementó de esta manera.
  - b) API Gateway con Kong: En este caso no se llegó a implementar ninguna primera versión de una API Gateway con Kong. Al implementar el SSO, se observó que su integración en Spring Boot y sus dependencias de seguridad [35] ya validaban las peticiones de entrada aplicando los criterios necesarios (archivo de configuración “src/main/java/edu/uva/consultasmedicas/frameWorksDrivers/cas\_config/WebSecurity.java”).
  - c) Microservicios: En el diagrama de despliegue original se representaron tres API (citas, personal y authorization). Finalmente, se unificaron en una sola. En el caso de citas y personal dependían demasiado entre ellos, y separados, se consideró que no eran lo suficientemente grandes como para estar divididos. “Authorization” se eliminó porque con el servidor del SSO de la UVa y las configuraciones en el único servicio ya era suficiente para suplir ese apartado de seguridad de la aplicación.
  - d) Dockerización: Con las decisiones anteriores se adaptó la dockerización para estar compuesta de cuatro contenedores, uno con la aplicación completa de “Consultas-Medicas” (frontend + backend), y tres para las tres bases de datos.

2. Thymeleaf: Al comienzo de la implementación y tras la eliminación de Nginx, se meditó acerca de añadir Thymeleaf al proyecto. Thymeleaf posee una gran integración con Spring Boot, facilitaba las vistas HTML y reducía el uso de JavaScript.
3. Modelo de bases de datos (Figura 6.2): Todos los nombres se tradujeron al inglés. La base de datos de solicitantes se pasó a llamar “users” en vez de “applicant”, dado que es un nombre más acorde con el contenido. Se eliminaron las tablas de dirección postal, ya que sustituirlo por los atributos “address” y “town” era suficiente. La información de contacto se movió a la base de datos de citas, ya que está más relacionada con ella que con los usuarios. Se eliminó la tabla “semanaDisponibilidad”, ya que no aportaba ninguna información útil. Se añadió la tabla “blockedHour” para representar las franjas horarias que están bloqueadas. Se añadió la tabla “schedule” para determinar el horario de disponibilidad por defecto.
4. Creación de horas y días: La aplicación muestra los días y horas libres según el horario de disponibilidad por defecto. Por lo tanto, la creación de los días y horas solo ocurre cuando se reserva una cita en dicho día y hora, cuando se modifica el horario de disponibilidad o cuando se bloquean horas/días.
5. Reglas de negocio: En el bloqueo de horas y días se ha establecido que el bloqueo debe de realizarse con un mínimo de tres días desde el día actual. Se omitió la regla de negocio RN3 para facilitar la implementación, por lo que se ha permitido que los solicitantes puedan tener varias citas a la vez, aunque sean del mismo paciente.
6. Casos de uso: Se enriquecieron los casos de uso “Gestora - Modificar horario de disponibilidad” y “Gestora - Bloquear horas”. En “Gestora - Modificar horario de disponibilidad” se añadieron las horas bloqueadas y la posibilidad de desbloquearlas. En “Gestora - Bloquear horas” se añadió la visualización de las horas disponibles del día seleccionado, la posibilidad de desbloquear las horas bloqueadas y las nuevas horas bloqueadas se agrupan lo máximo posible con las que ya había previamente bloqueadas.
7. Máquina virtual: La aplicación se ha ejecutado en todo momento en una máquina virtual dentro de la red de la UVa. Dado que los permisos de acceso del SSO estaban vinculados a esta máquina virtual, la funcionalidad completa de la aplicación solo se conseguía si la aplicación estaba desplegada dentro. Esto dificultó el desarrollo porque la comprobación del correcto funcionamiento del código precisó de utilizar los *logs* introducidos en la aplicación y hacer las comprobaciones en la máquina virtual, siempre vía una terminal *bash*.

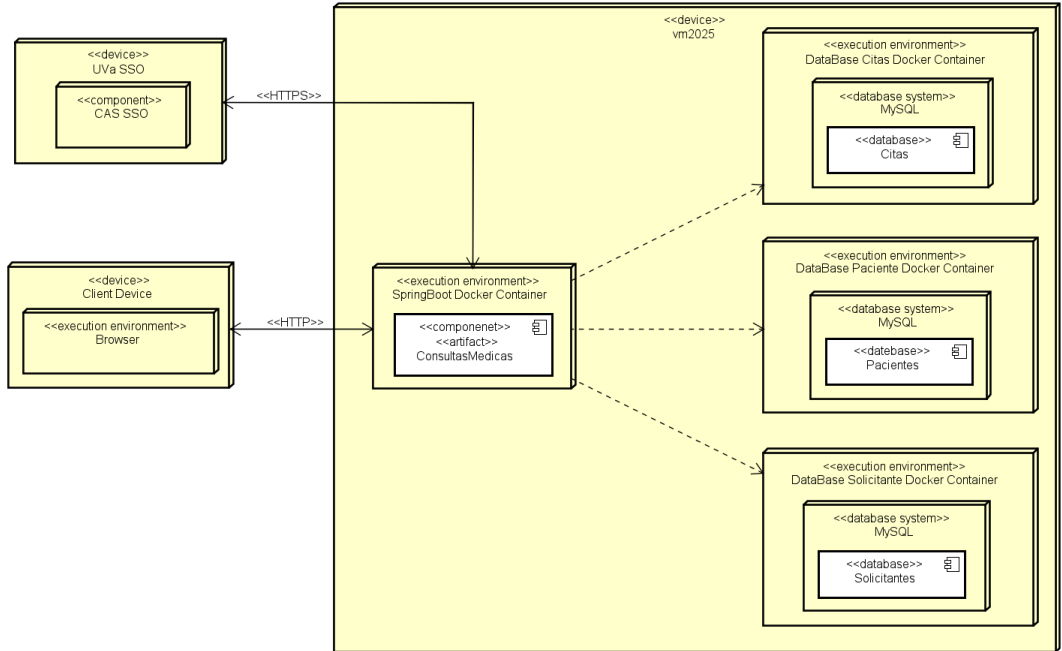


Figura 6.1: Diagrama de despliegue final, decidido en la implementación.

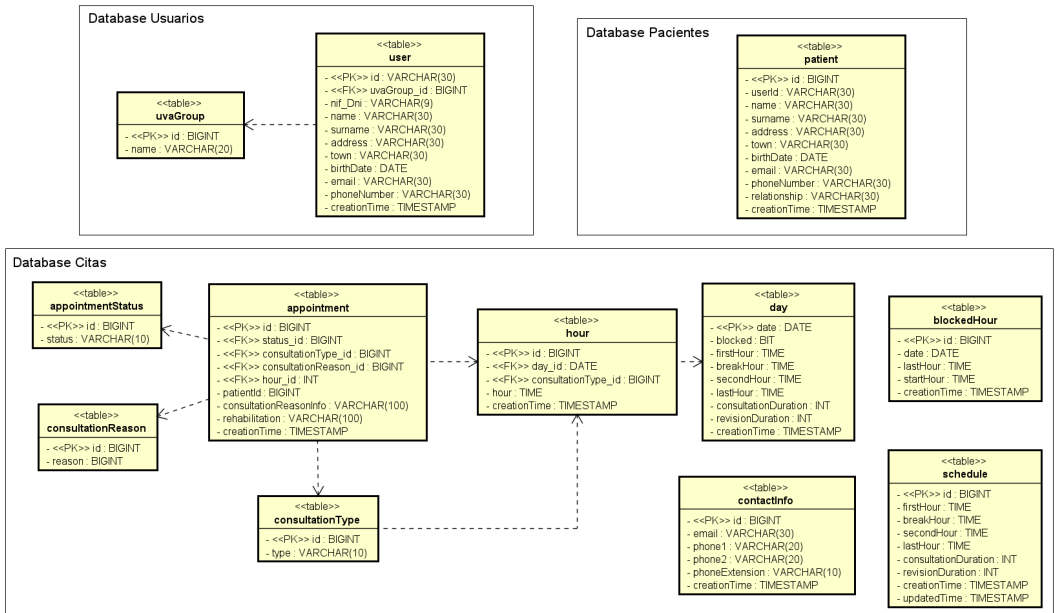


Figura 6.2: Modelo de bases de datos, decidido en la implementación.

## 6.5. Organización a la hora de desarrollar

El desarrollo del proyecto se realizó utilizando git y gitlab. Se creó un repositorio para alojar el proyecto y tener un control de versiones.

El trabajo se realizó creando ramas y *Merge Request*. Las ramas utilizan la sintaxis “prefix/TFG-00-breve\_resumen\_de\_los\_cambios” y las *Merge request* “prefijo 00: breve resumen de los cambios”. Se crea una *Merge Request* por cada tarea de la planificación, se desarrollaba la tarea y una vez finalizada, se aprueba la *Merge request* y se mezclaba en la rama principal (*master*). Dentro de cada *Merge Request* se hacen commits siguiendo la sintaxis “prefijo: breve resumen de los cambios”.

Los prefijos utilizados han sido los siguientes: *feat* (nueva funcionalidad añadida), *fix* (arreglo añadido), *test* (commit utilizado para probar una funcionalidad) y *refactor* (reorganización de código).

## 6.6. Pruebas

Para comprobar la correcta funcionalidad de la aplicación, se han realizado pruebas manuales desde el navegador sobre los distintos casos de uso. La ejecución de todas las pruebas se ha realizado desde un ordenador de escritorio, con el navegador Mozilla Firefox y apuntando a la dirección de la aplicación <http://157.88.125.198:8080/consultasmedicas/>.

Valores preestablecidos para todas las pruebas:

- Usuarios con permiso para iniciar sesión en el SSO de preproducción de la UVa: alumno de este TFG (usuario: \*\*\*\* contraseña:\*\*\*\*) y tutora del TFG (usuario: \*\*\*\* contraseña:\*\*\*\*)
  - Para las pruebas se utilizará el usuario del alumno (NIF/NIE: 00000000X; Nombre: ENRIQUE; Apellidos: ROJO ALVAREZ; Colectivo UVa: **estudiante, PDI/-PAS, gestor**; Fecha de nacimiento: dd/mm/aaa; Dirección postal: XXXXXXXXXXXX; Teléfono móvil: XXXXXXXXXXXX; Email: XXXXXXXXXXXX@estudiantes.uva.es), su colectivo UVa originalmente es de “estudiante” pero para poder realizar las pruebas se modificará para hacer pruebas con distintos roles.
- URL de SSO: <https://idpre.uva.es/CAS/login?service=http%3A%2F%2F157.88.125.198%3A8080%2Fconsultasmedicas%2Fsecured>
- Base de datos de citas:
  - Tipo de consultas: Primera consulta y Revisión.
  - Motivo de consulta: Lenguaje, Habla, Voz, Audición, Deglución y Otros.
  - Estado de cita: Prereservada, Bloqueada, Cancelada, Pendiente, Realizada, No presentado.

- Horario de disponibilidad por defecto: Lunes, jueves, viernes, sábado y domingo bloqueados; martes y miércoles disponibles (Duración de consulta de 60 minutos, duración de revisión de 30 minutos, hora de inicio 9:00, hora de partida 12:00, segunda hora de inicio 16:00 y hora de finalización 18:00).
- Base de datos usuarios:
  - Colectivo UVa: Estudiante, PDI/PAS y Gestor.

Se muestran los distintos escenarios que se han llevado a cabo, y al final de cada uno de ellos su resultado: “Éxito” o “Fallido”. Todas las pruebas se han realizado tras finalizar la tarea correspondiente a su implementación, por lo que en el caso de un test “Fallido”, se arreglará el error correspondiente en la tarea de revisión de los respectivos casos de uso y se probará de nuevo, añadiendo el nuevo resultado junto al de la primera ejecución. Existen dos tipos de escenarios: los únicos, que se ejecutan una sola vez, y los múltiples, se ejecutan varias veces cambiando las variables/opciones que están en **negrita**. Cabe destacar que después de cada ejecución, se eliminan todos los elementos creados por la misma, así, las siguientes ejecuciones no se ven afectadas por las anteriores. Cada prueba está identificada con un código: capítulo.sección.CU.prueba, por ejemplo, las pruebas de iniciar sesión están identificadas como 6.6.1.1, 6.6.1.2, 6.6.1.3...

### 6.6.1. Inicio de sesión

Se han hecho distintas pruebas para comprobar el correcto funcionamiento:

1. **Comprobación de que la aplicación redirige al SSO cuando el acceso a diferentes rutas requiere de autenticación.** Escenario múltiple: Sin estar autenticado. Acceso a la ruta “/ **citas**, / **citas**/ **calendario**, / **citas**/ **paciente** y / **citas**/ **datos**”, la aplicación redirige al SSO, se introducen las credenciales de un usuario permitido, se pulsa aceptar y el SSO redirige a “/inicio”. **Éxito**
2. **Comprobación de que se puede iniciar sesión en el SSO como estudiante o PDI/PAS y se redirige a su pantalla de inicio.** Escenario múltiple: Sin estar autenticado. Desde “/” se pulsa iniciar sesión, la aplicación redirige al SSO, se introducen las credenciales del usuario del alumno con rol de **estudiante/PDIPAS**, se pulsa aceptar y el SSO redirige a “/inicio” mostrando la pantalla de inicio de solicitante junto con el nombre y colectivo de usuario autenticado. **Éxito**.
3. **Comprobación de que se puede iniciar sesión en el SSO como gestor y se redirige a su pantalla de inicio.** Escenario único: Sin estar autenticado. Desde “/” se pulsa iniciar sesión, la aplicación redirige al SSO, se introducen las credenciales del usuario del alumno con rol de gestor, se pulsa aceptar y el SSO redirige a “/inicio” mostrando la pantalla de inicio de gestor junto con el nombre y colectivo de usuario autenticado. **Fallido**. Tras la tarea de revisión: **Éxito**
4. **Comprobación de que no se puede iniciar sesión en el SSO con un usuario incorrecto.** Escenario único: Sin estar autenticado. Desde “/” se pulsa iniciar sesión,

la aplicación redirige al SSO, se introducen las credenciales de un usuario incorrecto, se pulsa aceptar, el SSO muestra un mensaje de error y no redirige a la aplicación. **Éxito**

5. **Comprobación de que estando autenticado, se puede desconectar la sesión.** Escenario múltiple: Autenticado con el usuario del alumno con rol de **estudiante/P-DIPAS/gestor**. Desde “/inicio” se pulsa Desconectar, la aplicación elimina la sesión y redirige a “/”, se accede a “/citas”, la aplicación redirige al SSO para iniciar sesión. **Fallido** para el rol de gestor. Tras la tarea de revisión: **Éxito**

### 6.6.2. Solicitante - Pedir cita

Se han hecho distintas pruebas para comprobar el correcto funcionamiento:

1. **Comprobación de que los días disponibles/bloqueados y horas disponibles están acordes al horario de disponibilidad.** Escenario múltiple: Autenticado con el usuario del alumno con rol de **estudiante/PDIPAS**. Desde “/inicio” se pulsa en Pedir cita, se muestra la pantalla con el calendario de días libres, se comprueba que los días libres y bloqueados están acordes al horario de disponibilidad predefinido, se selecciona un día libre y el tipo de consulta “**Primera consulta/Revisión**”, se comprueba que las horas disponibles que aparecen están acordes al horario de disponibilidad predefinido. **Éxito**
2. **Comprobación de que no se puede seleccionar un día bloqueado.** Escenario múltiple: Autenticado con el usuario del alumno con rol de **estudiante/PDIPAS**. Desde “/inicio” se pulsa en Pedir cita, se muestra la pantalla con el calendario de días libres, se comprueba que los días libres y bloqueados están acordes al horario de disponibilidad predefinido, se comprueba que la aplicación no permite seleccionar un día bloqueado. **Éxito**
3. **Comprobación de que un usuario con rol estudiante o PDIPAS puede solicitar una primera consulta.** Escenario múltiple: Autenticado con el usuario del alumno con rol de **estudiante/PDIPAS**. Desde “/inicio” se pulsa en Pedir cita, se muestra la pantalla con el calendario de días libres y tipo de consulta, se selecciona un día libre y el tipo de consulta “Primera consulta”, se selecciona una hora disponible, se pulsa continuar, se comprueba que en la siguiente pantalla aparecen los datos correctos del usuario (NIF/NIE, Nombre, Apellidos, Colectivo UVa, Edad, Dirección postal, Teléfono móvil y Email), se pulsa continuar, se comprueba que aparece el selector de “Motivo de consulta” y el texto libre de “Cuéntanos más”, se comprueba que no aparece la opción de “añadir quién te atendió en la anterior consulta”, se selecciona el motivo de consulta “**Audición, Deglución, Habla, Lenguaje, Voz y Otros**” y se introduce en el texto libre “Esta es una prueba de pedir una cita como estudiante o PDIPAS...”, se pulsa continuar, la aplicación redirige a la pantalla de inicio, se pulsa en Pedir cita, se muestra la pantalla con el calendario de días libres y tipo de consulta, se comprueba que la hora en el día en que se pidió la cita ya no está disponible. Se comprueba vía logs del backend que se ha creado la cita con estado “Pendiente”, la hora, el día y el solicitante como paciente. **Éxito**

4. **Comprobación de que un usuario con rol estudiante o PDIPAS puede solicitar una revisión.** Escenario múltiple: Autenticado con el usuario del alumno con rol de **estudiante/PDIPAS**. Desde “/inicio” se pulsa en Pedir cita, se muestra la pantalla con el calendario de días libres y tipo de consulta, se selecciona un día libre y el tipo de consulta “Revisión”, se selecciona una hora disponible, se pulsa continuar, se comprueba que en la siguiente pantalla aparecen los datos correctos del usuario (NI-F/NIE, Nombre, Apellidos, Colectivo UVa, Edad, Dirección postal, Teléfono móvil y Email), se pulsa continuar, aparece un mensaje de advertencia indicando que no existe una “Primera consulta” previa, se pulsa continuar, se comprueba que aparece el selector de “Motivo de consulta” y el texto libre de “Cuéntanos más”, se introduce en “añadir quién te atendió en la anterior consulta” el texto: “Me atendió alguien” se selecciona el motivo de consulta “**Audición, Deglución, Habla, Lenguaje, Voz y Otros**” y se introduce en el texto libre “Esta es una prueba de pedir una cita como estudiante o PDIPAS...”, se pulsa continuar, la aplicación redirige a la pantalla de inicio, se pulsa en Pedir cita, se muestra la pantalla con el calendario de días libres y tipo de consulta, se comprueba que la hora en el día en que se pidió la cita ya no está disponible. Se comprueba vía logs del backend que se ha creado la cita con estado “Pendiente”, la hora, el día y el solicitante como paciente. **Éxito**
5. **Comprobación de que la hora de una cita en estado prereservada está bloqueada.** Escenario múltiple: Autenticado con el usuario del alumno con rol de **estudiante/PDIPAS**. Desde “/inicio” se pulsa en Pedir cita, se muestra la pantalla con el calendario de días libres y tipo de consulta, se selecciona un día libre y el tipo de consulta “Primera consulta”, se selecciona una hora disponible, se pulsa continuar, en otra pestaña del navegador nos dirigimos a “citas/calendario”, se comprueba que la hora de la cita está bloqueada. **Éxito**
6. **Comprobación de que una cita puede ser cancelada durante el proceso de solicitud.** Escenario múltiple: Autenticado con el usuario del alumno con rol de **estudiante/PDIPAS**. Desde “/inicio” se pulsa en Pedir cita, se muestra la pantalla con el calendario de días libres y tipo de consulta, se selecciona un día libre y el tipo de consulta “Primera consulta”, se selecciona una hora disponible, se pulsa continuar, en la siguiente pantalla se pulsa cancelar, aparece un mensaje de advertencia, se pulsa aceptar, la aplicación nos redirige a “/inicio”, nos dirigimos a “citas/calendario”, se comprueba que la hora de la cita está disponible. **Éxito**
7. **Comprobación de una cita en estado prereservada es eliminada a los 10 minutos.** Escenario múltiple: Autenticado con el usuario del alumno con rol de **estudiante/PDIPAS**. Desde “/inicio” se pulsa en Pedir cita, se muestra la pantalla con el calendario de días libres y tipo de consulta, se selecciona un día libre y el tipo de consulta “Primera consulta”, se selecciona una hora disponible, se pulsa continuar, se esperan 10 minutos, se pulsa continuar, aparece un mensaje de error “El tiempo para pedir una cita ha expirado (10 minutos). Inténtelo de nuevo”, nos dirigimos a “citas/calendario”, se comprueba que la hora de la cita está disponible, se comprueba vía logs del backend que la cita se ha eliminado. **Éxito**
8. **Comprobación de que no se puede hacer una petición POST directamente en el backend.** Escenario único: Desde un terminal bash se hace la petición `curl 'http://157.88.125.198:8080/consultasmedicas/citas/calendario'`

```
-X POST -H 'User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
x64; rv:123.0) Gecko/20100101 Firefox/123.0' -H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8'
-H 'Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3' -H 'Accept-Encoding: gzip, deflate' -H 'Content-Type: application/x-www-form-urlencoded'
-H 'Origin: http://157.88.125.198:8080' -H 'Connection: keep-alive' -H 'Referer: http://157.88.125.198:8080/consultasmedicas/citas/calendario'
-H 'Upgrade-Insecure-Requests: 1' -H 'DNT: 1' -H 'Sec-GPC: 1' -data-raw '_csrf=dconsultationType=Primera_consulta&startHour=09%3A00&day=2024-03-13^03-13', el servidor nos devuelve la plantilla HTML de que se ha producido un error, se comprueba vía logs del backend que la petición ha quedado rechazada. Éxito
```

9. **Comprobación de que no se puede hacer una petición POST directamente en el backend con un csrf inválido.**

Escenario único: Desde el navegador se hace una petición válida para crear una cita, se extrae la petición y se altera ligeramente el token csrf, desde un terminal bash se hace la petición alterada `curl 'http://157.88.125.198:8080/consultasmedicas/citas/calendario'`

```
-X POST -H 'User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
x64; rv:123.0) Gecko/20100101 Firefox/123.0' -H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8'
-H 'Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3' -H 'Accept-Encoding: gzip, deflate' -H 'Content-Type: application/x-www-form-urlencoded' -H 'Origin: http://157.88.125.198:8080' -H 'Connection: keep-alive' -H 'Referer: http://157.88.125.198:8080/consultasmedicas/citas/calendario' -H 'Cookie: JSESSIONID=5E8D2CDF96FFD7254ED4519F5822B46C' -H 'Upgrade-Insecure-Requests: 1' -H 'DNT: 1' -H 'Sec-GPC: 1' -data-raw '_csrf=3e1151af-72fb-41c5-b123-46ed1d1fa214&consultationType=Primera_consulta&startHour=11%3A00&day=2024-03-05^03-05', el servidor nos devuelve la plantilla HTML de que se ha producido un error, se comprueba vía logs del backend que la petición ha quedado rechazada. Éxito
```

10. **Comprobación de que no se puede hacer una petición POST directamente en el backend con una sesión inválida.**

Escenario único: Desde el navegador se hace una petición válida para crear una cita, se extrae la petición y se altera ligeramente el id de la sesión, desde un terminal bash se hace la petición alterada `curl 'http://157.88.125.198:8080/consultasmedicas/citas/calendario'`

```
-X POST -H 'User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
x64; rv:123.0) Gecko/20100101 Firefox/123.0' -H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8'
-H 'Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3' -H 'Accept-Encoding: gzip, deflate' -H 'Content-Type: application/x-www-form-urlencoded' -H 'Origin: http://157.88.125.198:8080' -H 'Connection: keep-alive' -H 'Referer: http://157.88.125.198:8080/consultasmedicas/citas/calendario' -H 'Cookie: JSESSIONID=5E8D2CDF96FFD7254ED4519F5822B26C' -H 'Upgrade-Insecure-Requests: 1' -H 'DNT: 1' -H 'Sec-GPC: 1' -data-raw '_csrf=4e1150af-78fb-47c5-b823-46ed1d4fa274&consultationType=Primera_consulta&startHour=16%3A00&day=2024-03-05', el servidor nos devuelve la plantilla HTML de que se ha producido un error, se comprueba vía logs del backend que la petición ha quedado rechazada. Éxito
```



### 6.6.3. Solicitante - Añadir información de paciente

Se han hecho distintas pruebas para comprobar el correcto funcionamiento:

1. **Comprobación de que un usuario con rol estudiante no tiene la opción de añadir un paciente.** Escenario múltiple: Autenticado con el usuario del alumno con rol de estudiante. Desde “/inicio” se pulsa en Pedir cita, se muestra la pantalla con el calendario de días libres y tipo de consulta, se selecciona un día libre y el tipo de consulta “**Primera consulta/Revisión**”, se selecciona una hora disponible, se pulsa continuar, se comprueba que en la siguiente pantalla aparecen los datos correctos del usuario (NIF/NIE, Nombre, Apellidos, Colectivo UVa, Edad, Dirección postal, Teléfono móvil y Email), se comprueba que no aparece la opción de añadir un paciente. **Éxito**
2. **Comprobación de que un usuario con rol PDIPAS puede añadir un paciente para una primera consulta.** Escenario múltiple: Autenticado con el usuario del alumno con rol de PDIPAS. Desde “/inicio” se pulsa en Pedir cita, se muestra la pantalla con el calendario de días libres y tipo de consulta, se selecciona un día libre y el tipo de consulta “Primera consulta”, se selecciona una hora disponible, se pulsa continuar, se comprueba que en la siguiente pantalla aparecen los datos correctos del usuario (NIF/NIE, Nombre, Apellidos, Colectivo UVa, Edad, Dirección postal, Teléfono móvil y Email), se comprueba que aparece la opción de añadir un paciente, se marca la opción, aparece un formulario para añadir los datos del paciente (Parentesco, Nombre, Apellidos, Dirección postal, Localidad, Email, Teléfono móvil y Fecha de nacimiento), se introducen los datos (**Padre/Madre, Hijo/Hija, Cónyuge**, Prueba, Prueba, Prueba, Prueba, prueba@gmail.com, 666666666, 01/01/1980), se pulsa continuar, se comprueba vía logs del backend que se ha creado el paciente y se ha asignado a la cita. **Éxito**
3. **Comprobación de que un usuario con rol PDIPAS puede añadir un paciente para una revisión.** Escenario múltiple: Autenticado con el usuario del alumno con rol de PDIPAS. Desde “/inicio” se pulsa en Pedir cita, se muestra la pantalla con el calendario de días libres y tipo de consulta, se selecciona un día libre y el tipo de consulta “Revisión”, se selecciona una hora disponible, se pulsa continuar, se comprueba que en la siguiente pantalla aparecen los datos correctos del usuario (NIF/NIE, Nombre, Apellidos, Colectivo UVa, Edad, Dirección postal, Teléfono móvil y Email), se comprueba que aparece la opción de añadir un paciente, se marca la opción, aparece un formulario para añadir los datos del paciente (Parentesco, Nombre, Apellidos, Dirección postal, Localidad, Email, Teléfono móvil y Fecha de nacimiento), se introducen los datos (**Padre/Madre, Hijo/Hija, Cónyuge**, Prueba, Prueba, Prueba, Prueba, prueba@gmail.com, 666666666, 01/01/1980), se pulsa continuar, aparece un mensaje de advertencia indicando que no existe una “Primera consulta” previa, se pulsa continuar, se comprueba vía logs del backend que se ha creado el paciente y se le ha asignado a la cita. **Éxito**
4. **Comprobación de que un usuario con rol PDIPAS puede reutilizar un paciente creado.** Escenario único: Autenticado con el usuario del alumno con rol de PDIPAS y con un paciente creado previamente. Desde “/inicio” se pulsa en Pedir cita, se muestra la pantalla con el calendario de días libres y tipo de consulta, se selecciona un día libre y el tipo de consulta “Revisión”, se selecciona una hora disponible, se pulsa

continuar, se comprueba que en la siguiente pantalla aparecen los datos correctos del usuario (NIF/NIE, Nombre, Apellidos, Colectivo UVa, Edad, Dirección postal, Teléfono móvil y Email), se comprueba que aparece la opción de añadir un paciente, se marca la opción, aparece una tarjeta con la información de los pacientes del usuario y un formulario para añadir los datos del paciente (Parentesco, Nombre, Apellidos, Dirección postal, Localidad, Email, Teléfono móvil y Fecha de nacimiento), se comprueba que los datos del paciente son correctos, se pulsa seleccionar, la aplicación rellena automáticamente el formulario con los datos del paciente, se pulsa continuar, se comprueba vía logs del backend que se ha reutilizado el paciente y se le ha asignado a la cita.

**Éxito**

5. **Comprobación de que un usuario con rol PDIPAS puede editar un paciente creado.** Escenario único: Autenticado con el usuario del alumno con rol de PDIPAS y con un paciente creado previamente. Desde “/inicio” se pulsa en Pedir cita, se muestra la pantalla con el calendario de días libres y tipo de consulta, se selecciona un día libre y el tipo de consulta “Primera consulta”, se selecciona una hora disponible, se pulsa continuar, se comprueba que en la siguiente pantalla aparecen los datos correctos del usuario (NIF/NIE, Nombre, Apellidos, Colectivo UVa, Edad, Dirección postal, Teléfono móvil y Email), se comprueba que aparece la opción de añadir un paciente, se marca la opción, aparece una tarjeta con la información de los pacientes del usuario y un formulario para añadir los datos del paciente (Parentesco, Nombre, Apellidos, Dirección postal, Localidad, Email, Teléfono móvil y Fecha de nacimiento), se comprueba que los datos del paciente son correctos, se pulsa seleccionar, la aplicación rellena automáticamente el formulario con los datos del paciente, se cambia el nombre a “Nuevo nombre”, se pulsa continuar, se introduce en “añadir quién te atendió en la anterior consulta” el texto: “Me atendió alguien” se selecciona el motivo de consulta “Audición” y se introduce en el texto libre “Esta es una prueba de pedir una cita como estudiante o PDIPAS...”, se pulsa continuar, la aplicación redirige a la pantalla de inicio, se pulsa en Pedir cita, se muestra la pantalla con el calendario de días libres y tipo de consulta, se selecciona un día libre y el tipo de consulta “Primera consulta”, se selecciona una hora disponible, se pulsa continuar, se marca la opción de pedir cita para otra persona, se comprueba que el paciente que aparece tiene la nueva información editada. Se comprueba vía logs del backend que el paciente ha sido editado y asignado a la cita. **Éxito**

### 6.6.4. Solicitante - Consultar citas

Debido a que estas pruebas dependen de tener una cita creada, se definen algunos ejemplos de citas, usuarios y paciente creados que se utilizaran en los escenarios:

1. cita1: (Fecha: 20/03/2024; Hora: 09:00; Estado: Pendiente; Tipo de consulta: Primera consulta; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el solicitante como paciente)
2. cita2: (Fecha: 20/03/2024; Hora: 10:00; Estado: Pendiente; Tipo de consulta: Revisión; Atendido anteriormente por: Me atendió alguien; Estado: Pendiente; Motivo de consul-

ta: Audición; Especificación del motivo de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el solicitante como paciente)

3. cita3: (Fecha: 20/03/2024; Hora: 09:00; Estado: Pendiente; Tipo de consulta: Primera consulta; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el paciente es el paciente1)
4. cita4: (Fecha: 20/03/2024; Hora: 10:00; Estado: Pendiente; Tipo de consulta: Revisión; Atendido anteriormente por: Me atendió alguien; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el paciente es el paciente1)
5. paciente1: (Nombre: PRUEBA; Apellidos: PRUEBA; Fecha de nacimiento: 01/01/1980; Relación con solicitante: Padre/Madre; Dirección postal: Prueba; Teléfono móvil: 666666666; Email: prueba@gmail.com)

Se han hecho distintas pruebas para comprobar el correcto funcionamiento:

1. **Comprobación de que un usuario con rol estudiante o PDIPAS puede consultar citas aunque no tenga ninguna.** Escenario múltiple: Autenticado con el usuario del alumno con rol de **estudiante/PDIPAS**. Desde “/inicio” se pulsa en Mis citas, se muestra la pantalla de mis citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, no aparece ninguna cita, se ordena por estado **Cancelada, Realizada, No presentado** y no aparece ninguna cita. **Éxito**
2. **Comprobación de que un usuario con rol estudiante puede consultar citas existentes.** Escenario único: Autenticado con el usuario del alumno con rol de **estudiante/PDIPAS** y con las citas: cita1, cita2 creadas previamente. Desde “/inicio” se pulsa en Mis citas, se muestra la pantalla de mis citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, aparecen las citas creadas con las fechas y horas correctas, se cambia el filtro de ordenar por fecha a “Descendente”, las citas se ordenan correctamente. **Éxito**
3. **Comprobación de que un usuario con rol PDIPAS puede consultar citas existentes.** Escenario único: Autenticado con el usuario del alumno con rol de **PDI-PAS**, con las citas: cita1, cita2, cita3, cita4 y el paciente1 creados previamente. Desde “/inicio” se pulsa en Mis citas, se muestra la pantalla de mis citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, aparecen las citas creadas con las fechas y horas correctas, se cambia el filtro de ordenar por fecha a “Descendente”, las citas se ordenan correctamente. **Éxito**
4. **Comprobación de que un usuario con rol estudiante puede consultar el detalle de una cita existente de tipo “Primera consulta” y con el solicitante como paciente.** Escenario único: Autenticado con el usuario del alumno con rol de **estudiante** y con la cita1 creada previamente. Desde “/inicio” se pulsa en Mis citas, se muestra la pantalla de mis citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, aparece la cita creada con la fecha y hora correctas, se pulsa en ver detalle, aparecen los datos del solicitante (NIF/NIE, Nombre, Apellidos,

Colectivo UVa, Edad, Dirección Postal, Teléfono móvil, Email), información de la cita (Fecha, Hora, Tipo de consulta, Estado, Motivo de consulta, Especificación del motivo de consulta) y los datos del paciente como “En este caso el paciente es el solicitante.”, se comprueba que los datos son correctos. **Éxito**

5. **Comprobación de que un usuario con rol estudiante puede consultar el detalle de una cita existente de tipo “Revisión” y con el solicitante como paciente.** Escenario único: Autenticado con el usuario del alumno con rol de estudiante y con la cita2 creada previamente. Desde “/inicio” se pulsa en Mis citas, se muestra la pantalla de mis citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, aparece la cita creada con la fecha y hora correctas, se pulsa en ver detalle, aparecen los datos del solicitante (NIF/NIE, Nombre, Apellidos, Colectivo UVa, Edad, Dirección Postal, Teléfono móvil, Email), información de la cita (Fecha, Hora, Tipo de consulta, Atendido anteriormente por, Estado, Motivo de consulta, Especificación del motivo de consulta) y los datos del paciente como “En este caso el paciente es el solicitante.”, se comprueba que los datos son correctos. **Éxito**
6. **Comprobación de que un usuario con rol PDIPAS puede consultar el detalle de una cita existente de tipo “Primera consulta” y con un paciente distinto del solicitante.** Escenario único: Autenticado con el usuario del alumno con rol de PDIPAS, con la cita3 y paciente1 creados previamente. Desde “/inicio” se pulsa en Mis citas, se muestra la pantalla de mis citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, aparece la cita creada con la fecha y hora correctas, se pulsa en ver detalle, aparecen los datos del solicitante (NIF/NIE, Nombre, Apellidos, Colectivo UVa, Edad, Dirección Postal, Teléfono móvil, Email), información de la cita (Fecha, Hora, Tipo de consulta, Estado, Motivo de consulta, Especificación del motivo de consulta) y los datos del paciente (Parentesco, Nombre, Apellidos, Edad, Dirección postal, Localidad, Email, Teléfono móvil), se comprueba que los datos son correctos. **Éxito**
7. **Comprobación de que un usuario con rol PDIPAS puede consultar el detalle de una cita existente de tipo “Revisión” y con un paciente distinto del solicitante.** Escenario único: Autenticado con el usuario del alumno con rol de PDIPAS, con la cita4 y paciente1 creados previamente. Desde “/inicio” se pulsa en Mis citas, se muestra la pantalla de mis citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, aparece la cita creada con la fecha y hora correctas, se pulsa en ver detalle, aparecen los datos del solicitante (NIF/NIE, Nombre, Apellidos, Colectivo UVa, Edad, Dirección Postal, Teléfono móvil, Email), información de la cita (Fecha, Hora, Tipo de consulta, Atendido anteriormente por, Estado, Motivo de consulta, Especificación del motivo de consulta) y los datos del paciente (Parentesco, Nombre, Apellidos, Edad, Dirección postal, Localidad, Email, Teléfono móvil), se comprueba que los datos son correctos. **Éxito**
8. **Comprobación de que un usuario con rol Estudiante puede filtrar por los estados Realizada/No presentado/Cancelada.** Escenario múltiple: Autenticado con el usuario del alumno con rol de Gestor, con la cita1 creada previamente. Desde “/inicio” se pulsa en Mis citas, se muestra la pantalla de mis citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, aparece la cita creada con la fecha y hora correctas, se accede al contenedor donde está desplegada la base de datos

de citas `docker exec -it tfg-consultasmedicas_database-mysql-appointments_1 bash -l`, se selecciona la base de datos `USE ConsultasMedicas-appointments`; se ejecuta la consulta `UPDATE appointment SET status_id=5/6/3 WHERE id=1`; en el navegador, se cambia el filtro de estado a **Realizada/No presentado/Cancelada**, se comprueba que la cita modificada aparece. **Éxito**

### 6.6.5. Solicitante - Cancelar cita

Debido a que estas pruebas dependen de tener una cita creada, se definen algunos ejemplos de citas, usuarios y paciente creados que se utilizaran en los escenarios:

1. cita1: (Fecha: 20/03/2024; Hora: 09:00; Estado: Pendiente; Tipo de consulta: Primera consulta; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el solicitante como paciente)
2. cita2: (Fecha: 20/03/2024; Hora: 10:00; Estado: Pendiente; Tipo de consulta: Revisión; Atendido anteriormente por: Me atendió alguien; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el solicitante como paciente)
3. cita3: (Fecha: 20/03/2024; Hora: 09:00; Estado: Pendiente; Tipo de consulta: Primera consulta; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el paciente es el paciente1)
4. cita4: (Fecha: 20/03/2024; Hora: 10:00; Estado: Pendiente; Tipo de consulta: Revisión; Atendido anteriormente por: Me atendió alguien; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el paciente es el paciente1)
5. paciente1: (Nombre: PRUEBA; Apellidos: PRUEBA; Fecha de nacimiento: 01/01/1980; Relación con solicitante: Padre/Madre; Dirección postal: Prueba; Teléfono móvil: 666666666; Email: prueba@gmail.com)

Se han hecho distintas pruebas para comprobar el correcto funcionamiento:

1. **Comprobación de que un usuario con rol estudiante puede cancelar una cita.**  
Escenario múltiple: Autenticado con el usuario del alumno con rol de estudiante, con la **cita1**, **cita2** creada previamente. Desde `/inicio` se pulsa en Mis citas, se muestra la pantalla de mis citas con los filtros: estado `Pendiente` y ordenar por fecha `Ascendente` por defecto, aparece la cita creada con la fecha y hora correctas, se pulsa en ver detalle, se pulsa en `Cancelar cita`, aparece un mensaje de confirmación, se pulsa `aceptar`, la aplicación redirige a `/citas` con el filtro de estado `Pendiente`, se comprueba que la cita cancelada ya no aparece, se cambia el filtro de estado a `Cancelada`, se comprueba que la cita cancelada aparece, se pulsa en `Ver detalle`, se comprueba que el estado de

la cita es “Cancelada”, se accede a “citas/calendario”, se comprueba que la hora en el día de la cita cancelada está disponible. **Éxito.**

### 2. **Comprobación de que un usuario con rol PDIPAS puede cancelar una cita.**

Escenario múltiple: Autenticado con el usuario del alumno con rol de PDIPAS, con la **cita1, cita2, cita3, cita4** y paciente1 creados previamente. Desde “/inicio” se pulsa en Mis citas, se muestra la pantalla de mis citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, aparece la cita creada con la fecha y hora correctas, se pulsa en ver detalle, se pulsa en “Cancelar cita”, aparece un mensaje de confirmación, se pulsa “aceptar”, la aplicación redirige a “/citas” con el filtro de estado “Pendiente”, se comprueba que la cita cancelada ya no aparece, se cambia el filtro de estado a “Cancelada”, se comprueba que la cita cancelada aparece, se pulsa en “Ver detalle”, se comprueba que el estado de la cita es “Cancelada”, se accede a “citas/calendario”, se comprueba que la hora en el día de la cita cancelada está disponible. **Éxito.**

### 3. **Comprobación de que un usuario con rol estudiante puede cancelar una cita, reservarla de nuevo y cancelarla de nuevo.**

Escenario único: Autenticado con el usuario del alumno con rol de estudiante, con la cita1 creada previamente. Desde “/inicio” se pulsa en Mis citas, se muestra la pantalla de mis citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, aparece la cita creada con la fecha y hora correctas, se pulsa en ver detalle, se pulsa en “Cancelar cita”, aparece un mensaje de confirmación, se pulsa “aceptar”, la aplicación redirige a “/citas” con el filtro de estado “Pendiente”, se accede a “citas/calendario” se selecciona la fecha 20/03/2024, tipo de consulta “Primera consulta” y hora 09:00, se pulsa continuar, se pulsa continuar, se selecciona motivo de consulta “Audición” y se introduce en el texto libre “Esta es una prueba de pedir una cita como estudiante o PDIPAS...”, se pulsa en finalizar, se accede a /citas, se pulsa en “ver detalle de la cita con fecha 20/03/2024 y hora 09:00, se comprueba que su estado es “Pendiente”, se pulsa en “Cancelar cita”, se pulsa aceptar, la aplicación redirige a “/citas” con el filtro de estado “Pendiente”, se comprueba que la cita cancelada ya no aparece, se cambia el filtro de estado a “Cancelada”, se comprueba que aparecen dos citas con la misma fecha y hora como canceladas, se accede a “citas/calendario”, se comprueba que la hora en el día de la cita cancelada está disponible. **Éxito.**

## 6.6.6. Solicitante - Consultar información de contacto del gabinete

Para este caso de uso se ha inicializado en base de datos la información de contacto con los siguientes datos:

1. email: unidad.logopedia@uva.es
2. phone1: 983423021
3. phone2: 608838683
4. phoneExtension: null

Debido a que este caso de uso es muy simple, solo se ha hecho un escenario para comprobar el correcto funcionamiento:

1. **Comprobación de que un usuario con rol estudiante o PDIPAS puede consultar la información de contacto.** Escenario múltiple: Autenticado con el usuario del alumno con rol de **estudiante/PDIPAS**. Desde “/inicio” se pulsa en “Información de contacto”, se muestra la pantalla de información de contacto mostrando correctamente los datos que han sido previamente inicializados. **Éxito**.

### 6.6.7. Gestora - Consultar citas de todas las gestoras

Debido a que estas pruebas dependen de tener una cita creada, se definen algunos ejemplos de citas, usuarios y paciente creados que se utilizaran en los escenarios:

1. cita1: (Fecha: 20/03/2024; Hora: 09:00; Estado: Pendiente; Tipo de consulta: Primera consulta; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el solicitante como paciente)
2. cita2: (Fecha: 20/03/2024; Hora: 10:00; Estado: Pendiente; Tipo de consulta: Revisión; Atendido anteriormente por: Me atendió alguien; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el solicitante como paciente)
3. cita3: (Fecha: 20/03/2024; Hora: 09:00; Estado: Pendiente; Tipo de consulta: Primera consulta; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el paciente es el paciente1)
4. cita4: (Fecha: 20/03/2024; Hora: 10:00; Estado: Pendiente; Tipo de consulta: Revisión; Atendido anteriormente por: Me atendió alguien; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el paciente es el paciente1)
5. paciente1: (Nombre: PRUEBA; Apellidos: PRUEBA; Fecha de nacimiento: 01/01/1980; Relación con solicitante: Padre/Madre; Dirección postal: Prueba; Teléfono móvil: 666666666; Email: prueba@gmail.com)

Se han hecho distintas pruebas para comprobar el correcto funcionamiento:

1. **Comprobación de que un usuario con rol Gestor puede consultar citas, aunque no haya ninguna.** Escenario múltiple: Autenticado con el usuario del alumno con rol de Gestor. Desde “/inicio” se pulsa en Consulta de citas, se muestra la pantalla de las citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, no aparece ninguna cita, se ordena por estado **Cancelada, Realizada, No presentado y Bloqueada** y no aparece ninguna cita. **Éxito**.

2. **Comprobación de que un usuario con rol Gestor puede consultar citas existentes de usuarios de tipo Estudiante.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor; las citas: cita1, cita2 creadas previamente y vinculadas a un usuario de tipo Estudiante. Desde “/inicio” se pulsa en Consulta de citas, se muestra la pantalla de las citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, aparecen las citas creadas con las fechas y horas correctas, se cambia el filtro de ordenar por fecha a “Descendente”, las citas se ordenan correctamente. **Éxito.**
3. **Comprobación de que un usuario con rol Gestor puede consultar citas existentes de usuarios de tipo PDIPAS.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor; las citas: cita3, cita4 creadas previamente y vinculadas a un usuario de tipo PDIPAS. Desde “/inicio” se pulsa en Consulta de citas, se muestra la pantalla de las citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, aparecen las citas creadas con las fechas y horas correctas, se cambia el filtro de ordenar por fecha a “Descendente”, las citas se ordenan correctamente. **Éxito.**
4. **Comprobación de que un usuario con rol Gestor puede consultar el detalle de una cita vinculada a un usuario Estudiante, de tipo “Primera consulta” y con el solicitante como paciente.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor; la cita1 creada previamente y vinculada a un usuario de tipo Estudiante. Desde “/inicio” se pulsa en Consulta de citas, se muestra la pantalla de las citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, aparece la cita creada con la fecha y hora correctas, se pulsa en ver detalle, aparecen los datos del solicitante (NIF/NIE, Nombre, Apellidos, Colectivo UVa, Edad, Dirección Postal, Teléfono móvil, Email), información de la cita (Fecha, Hora, Tipo de consulta, Estado, Motivo de consulta, Especificación del motivo de consulta) y los datos del paciente como “En este caso el paciente es el solicitante.”, se comprueba que los datos son correctos. **Éxito.**
5. **Comprobación de que un usuario con rol Gestor puede consultar el detalle de una cita vinculada a un usuario Estudiante, de tipo “Revisión” y con el solicitante como paciente.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor; la cita2 creada previamente y vinculada a un usuario de tipo Estudiante. Desde “/inicio” se pulsa en Consulta de citas, se muestra la pantalla de las citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, aparece la cita creada con la fecha y hora correctas, se pulsa en ver detalle, aparecen los datos del solicitante (NIF/NIE, Nombre, Apellidos, Colectivo UVa, Edad, Dirección Postal, Teléfono móvil, Email), información de la cita (Fecha, Hora, Tipo de consulta, Atendido anteriormente por, Estado, Motivo de consulta, Especificación del motivo de consulta) y los datos del paciente como “En este caso el paciente es el solicitante.”, se comprueba que los datos son correctos. **Éxito.**
6. **Comprobación de que un usuario con rol Gestor puede consultar el detalle de una cita vinculada a un usuario PDIPAS, de tipo “Primera consulta” y con un paciente distinto del solicitante.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor; la cita3 y paciente1 creados previamente, y la cita vinculada a un usuario de tipo PDIPAS. Desde “/inicio” se pulsa en Consulta de citas, se muestra la pantalla de las citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, aparece la cita creada con la fecha y hora correctas, se



pulsa en ver detalle, aparecen los datos del solicitante (NIF/NIE, Nombre, Apellidos, Colectivo UVa, Edad, Dirección Postal, Teléfono móvil, Email), información de la cita (Fecha, Hora, Tipo de consulta, Estado, Motivo de consulta, Especificación del motivo de consulta) y los datos del paciente (Parentesco, Nombre, Apellidos, Edad, Dirección postal, Localidad, Email, Teléfono móvil), se comprueba que los datos son correctos. **Éxito.**

7. **Comprobación de que un usuario con rol Gestor puede consultar el detalle de una cita vinculada a un usuario PDIPAS, de tipo “Revisión” y con un paciente distinto del solicitante.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor, con la cita4 y paciente1 creados previamente, y la cita vinculada a un usuario de tipo PDIPAS. Desde “/inicio” se pulsa en Consulta de citas, se muestra la pantalla de las citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, aparece la cita creada con la fecha y hora correctas, se pulsa en ver detalle, aparecen los datos del solicitante (NIF/NIE, Nombre, Apellidos, Colectivo UVa, Edad, Dirección Postal, Teléfono móvil, Email), información de la cita (Fecha, Hora, Tipo de consulta, Atendido anteriormente por, Estado, Motivo de consulta, Especificación del motivo de consulta) y los datos del paciente (Parentesco, Nombre, Apellidos, Edad, Dirección postal, Localidad, Email, Teléfono móvil), se comprueba que los datos son correctos. **Éxito.**
  
8. **Comprobación de que un usuario con rol Gestor puede filtrar por los estados Realizada/No presentado/Cancelada/Bloqueada.** Escenario múltiple: Autenticado con el usuario del alumno con rol de Gestor, con la cita1 creada previamente. Desde “/inicio” se pulsa en Consulta de citas, se muestra la pantalla de mis citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, aparece la cita creada con la fecha y hora correctas, se accede al contenedor donde está desplegada la base de datos de citas `docker exec -it tfg-consultasmedicas_database-mysql-appointments_1 bash -l`, se selecciona la base de datos `USE ConsultasMedicas-appointments;`, se ejecuta la consulta `UPDATE appointment SET status_id=5/6/3/2 WHERE id=1;`, en el navegador, se cambia el filtro de estado a **Realizada/No presentado/Cancelada/Bloqueada**, se comprueba que la cita modificada aparece. **Éxito.**

### 6.6.8. Gestora - Cancelar cita de solicitante

Debido a que estas pruebas dependen de tener una cita creada, se definen algunos ejemplos de citas, usuarios y paciente creados que se utilizaran en los escenarios:

1. cita1: (Fecha: 20/03/2024; Hora: 09:00; Estado: Pendiente; Tipo de consulta: Primera consulta; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el solicitante como paciente)

Se han hecho distintas pruebas para comprobar el correcto funcionamiento:

1. **Comprobación de que un usuario con rol Gestor puede cancelar una cita.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor, con la cita1 creada previamente. Desde “/inicio” se pulsa en Consulta de citas, se muestra la pantalla de mis citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, aparece la cita creada con la fecha y hora correctas, se pulsa en ver detalle, se pulsa en “Cancelar cita”, aparece un mensaje de confirmación, se pulsa “aceptar”, la aplicación redirige a “/citas” con el filtro de estado “Pendiente”, se comprueba que la cita cancelada ya no aparece, se cambia el filtro de estado a “Cancelada”, se comprueba que la cita cancelada aparece, se pulsa en “Ver detalle”, se comprueba que el estado de la cita es “Cancelada”. Autenticándose con un usuario de tipo Estudiante, se accede a “citas/calendario”, se comprueba que la hora en el día de la cita cancelada está disponible. **Éxito.**
2. **Comprobación de que un usuario con rol Gestor no puede cancelar una cita con un estado que no sea “Pendiente” o “Bloqueada”.** Escenario múltiple: Autenticado con el usuario del alumno con rol de Gestor, con la cita1 creada previamente. Se accede al contenedor donde está desplegada la base de datos de citas `docker exec -it tfg-consultasmedicas_database-mysql-appointments_1 bash -l`, se selecciona la base de datos `USE ConsultasMedicas-appointments;`, se ejecuta la consulta `UPDATE appointment SET status_id=5/6/3 WHERE id=1;`, en el navegador, desde “/inicio” se pulsa en Consulta de citas, se muestra la pantalla de mis citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, se filtra por el estado **Realizada/No presentado/Cancelada**, aparece la cita creada con la fecha y hora correctas, se pulsa en ver detalle, se comprueba que no aparece la opción de “Cancelar cita”. **Éxito.**

### 6.6.9. Gestora - Modificar fecha y/u hora de una cita

Debido a que estas pruebas dependen de tener una cita creada, se definen algunos ejemplos de citas, usuarios y paciente creados que se utilizaran en los escenarios:

1. cita1: (Fecha: 20/03/2024; Hora: 09:00; Estado: Pendiente; Tipo de consulta: Primera consulta; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el solicitante como paciente)
2. cita2: (Fecha: 20/03/2024; Hora: 09:00; Estado: Bloqueado; Tipo de consulta: Primera consulta; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el solicitante como paciente)

Se han hecho distintas pruebas para comprobar el correcto funcionamiento:

1. **Comprobación de que un usuario con rol Gestor puede modificar la fecha y hora de una cita en estado “Pendiente”.** Escenario único: Autenticado con el

usuario del alumno con rol de Gestor y con la cita1 creada previamente. Desde “/inicio” se pulsa en Consulta de citas, se muestra la pantalla de mis citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, aparece la cita creada con la fecha y hora correctas, se pulsa en ver detalle, se pulsa en “Modificar fecha y hora”, aparece un mensaje de confirmación, se pulsa “aceptar”, la aplicación redirige a “/consultar/citas/id/dia-hora”, se muestra un calendario y un selector de hora, se selecciona un día y hora disponible, se pulsa en “Aceptar”, la aplicación redirige a “/consultar/citas”, se comprueba que la cita que aparece es la modificada con la nueva fecha y hora. Se inicia sesión con un usuario de tipo Estudiante, se accede a “citas/calendario”, se comprueba que la hora en el día de la cita modificada está disponible, se comprueba que la nueva hora y día no está disponible. **Éxito**.

2. **Comprobación de que un usuario con rol Gestor puede modificar la fecha y hora de una cita en estado “Bloqueada”**. Escenario único: Autenticado con el usuario del alumno con rol de Gestor y con la cita2 creada previamente. Desde “/inicio” se pulsa en Consulta de citas, se muestra la pantalla de mis citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, aparece la cita creada con la fecha y hora correctas, cambia el filtro de estado a “Bloqueada”, aparece la cita creada, se pulsa en ver detalle, se pulsa en “Modificar fecha y hora”, aparece un mensaje de confirmación, se pulsa “aceptar”, la aplicación redirige a “/consultar/citas/id/dia-hora”, se muestra un calendario y un selector de hora, se selecciona un día y hora disponible, se pulsa en “Aceptar”, la aplicación redirige a “/consultar/citas”, se comprueba que la cita que aparece es la modificada con la nueva fecha y hora, se pulsa en “Ver detalle”, se comprueba que el estado de la cita ha cambiado a “Pendiente”. **Fallido**, el estado de la cita no cambia a “Pendiente”. Tras la revisión de casos de uso: **Éxito**.
3. **Comprobación de que un usuario con rol Gestor no puede modificar la fecha y hora de una cita con un estado que no sea “Pendiente” o “Bloqueado”**. Escenario múltiple: Autenticado con el usuario del alumno con rol de Gestor y con la cita1 creada previamente. Se accede al contenedor donde está desplegada la base de datos de citas `docker exec -it tfg-consultasmedicas_database-mysql-appointments_1 bash -l`, se selecciona la base de datos `USE ConsultasMedicas-appointments;`, se ejecuta la consulta `UPDATE appointment SET status_id=5/6/3 WHERE id=1;`, en el navegador, desde “/inicio” se pulsa en Consulta de citas, se muestra la pantalla de mis citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, se filtra por el estado **Realizada/No presentado/Cancelada**, aparece la cita creada con la fecha y hora correctas, se pulsa en ver detalle, se comprueba que no aparece la opción de “Modificar fecha y hora”. **Éxito**.

### 6.6.10. Gestora - Cambiar estado de cita

Debido a que estas pruebas dependen de tener una cita creada, se definen algunos ejemplos de citas, usuarios y paciente creados que se utilizaran en los escenarios:

1. cita1: (Fecha: 20/03/2024; Hora: 09:00; Estado: Pendiente; Tipo de consulta: Primera consulta; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo

de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el solicitante como paciente)

2. cita2: (Fecha: 20/03/2024; Hora: 09:00; Estado: Bloqueada; Tipo de consulta: Primera consulta; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el solicitante como paciente)
3. cita3: (Fecha: 20/03/2024; Hora: 09:00; Estado: Cancelada; Tipo de consulta: Primera consulta; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el solicitante como paciente)
4. cita4: (Fecha: 20/03/2024; Hora: 09:00; Estado: Realizada; Tipo de consulta: Primera consulta; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el solicitante como paciente)
5. cita5: (Fecha: 20/03/2024; Hora: 09:00; Estado: No presentado; Tipo de consulta: Primera consulta; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el solicitante como paciente)

Se han hecho distintas pruebas para comprobar el correcto funcionamiento:

1. **Comprobación de que un usuario con rol Gestor puede cambiar el estado de una cita en estado “Pendiente” cuando el día de la cita coincide con el actual o es posterior.** Escenario múltiple: Autenticado con el usuario del alumno con rol de Gestor y con la cita1 creada previamente. Desde “/inicio” se pulsa en Consulta de citas, se muestra la pantalla de mis citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, aparece la cita creada con la fecha y hora correctas, se pulsa en ver detalle, se muestran los detalles de la cita junto con un selector de estado, se comprueba que los estados seleccionables son Pendiente, Realizada y No presentado, se comprueba que el estado seleccionado por defecto es “Pendiente”, se selecciona **Realizada, No presentado**, se pulsa en confirmar estado, aparece un mensaje de confirmación, se pulsa “aceptar”, la aplicación redirige a “/consultar/citas”, se muestran las citas en estado “Pendiente”, se comprueba que la cita modificada no aparece, se filtra por estado **Realizada, No presentado**, se comprueba que la cita modificada aparece con el nuevo estado. **Éxito.**
2. **Comprobación de que un usuario con rol Gestor no puede cambiar el estado de una cita en estado “Pendiente” cuando el día de la cita es posterior al actual .** Escenario múltiple: Autenticado con el usuario del alumno con rol de Gestor y con la cita1 creada previamente. Desde “/inicio” se pulsa en Consulta de citas, se muestra la pantalla de mis citas con los filtros: estado “Pendiente” y ordenar por fecha “Ascendente” por defecto, aparece la cita creada con la fecha y hora correctas, se pulsa en ver detalle, se muestran los detalles de la cita junto con un selector de estado,

se comprueba que los estados seleccionables son Pendiente, Realizada y No presentado, se comprueba que el estado seleccionado por defecto es “Pendiente”, se selecciona **Realizada**, **No presentado**, se pulsa en confirmar estado, aparece un mensaje de confirmación, se pulsa “aceptar”, la aplicación muestra un mensaje de error alertando de que la fecha actual debe de ser la de la cita o posterior. **Éxito**.

3. **Comprobación de que un usuario con rol Gestor no puede cambiar el estado de una cita en estado “Bloqueada”, “Cancelada”, “Realizada” y “No presentado”.** Escenario múltiple: Autenticado con el usuario del alumno con rol de Gestor, con la **cita2**, **cita3**, **cita4**, **cita5** creada previamente. Desde “/inicio” se pulsa en Consulta de citas, se cambia el filtro de estado a **Bloqueada/Cancelada/Realizada/No presentado**, aparece la cita creada, se pulsa en ver detalle, se comprueba que no aparece ninguna opción para seleccionar otro estado. **Fallido**, una cita con estado “Bloqueada” muestra la opción de cambiar el estado. Tras la revisión de casos de uso: **Éxito**.
4. **Comprobación de que no es posible realizar un SQL injection para eliminar las bases de datos.**

Escenario múltiple: Se realiza un curl desde la consola de comandos con un token válido: `curl 'http://157.88.125.198:8080/consultasmedicas/consultar/citas/13' -X POST -H 'User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:123.0) Gecko/20100101 Firefox/123.0' -H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8' -H 'Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3' -H 'Accept-Encoding: gzip, deflate' -H 'Content-Type: application/x-www-form-urlencoded' -H 'Origin: http://157.88.125.198:8080' -H 'Connection: keep-alive' -H 'Referer: http://157.88.125.198:8080/consultasmedicas/consultar/citas/13' -H 'Cookie: JSESSIONID=7C63A29B412C66CC23D8324F15ECA2C8' -H 'Upgrade-Insecure-Requests: 1' -H 'DNT: 1' -H 'Sec-GPC: 1' -data-raw '_csrf=5be28ffa-6638-4707-bbf0-87e33e19675a&newStatus=Pendiente; DROP ALL DATABASE;/DROP ALL DATABASE;'`, la petición no tiene ningún efecto. **Éxito**.

### 6.6.11. Gestora - Consultar su horario de disponibilidad

Se ha añadido un horario de disponibilidad por defecto para todas las pruebas:

1. Lunes: (Duración de primera consulta 0; duración revisión: 0; primera hora: null; segunda hora: null; tercera hora: null; última hora: null)
2. Martes: (Duración de primera consulta 60; duración revisión: 30; primera hora: 09:00; segunda hora: 12:00; tercera hora: 16:00; última hora: 18:00)
3. Miércoles: (Duración de primera consulta 60; duración revisión: 30; primera hora: 09:00; segunda hora: 12:00; tercera hora: 16:00; última hora: 18:00)
4. Jueves: (Duración de primera consulta 0; duración revisión: 0; primera hora: null; segunda hora: null; tercera hora: null; última hora: null)

5. Viernes: (Duración de primera consulta 0; duración revisión: 0; primera hora: null; segunda hora: null; tercera hora: null; última hora: null)

Se han hecho distintas pruebas para comprobar el correcto funcionamiento:

1. **Comprobación de que un usuario con rol Gestor puede consultar el horario de disponibilidad.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor. Desde “/inicio” se pulsa en Consultar horario de disponibilidad, se muestran los días lunes, martes, miércoles, jueves y viernes, y un selector para el primer día de la semana. Se comprueba que el día seleccionado de por defecto es el lunes de la semana actual. Se comprueba que la disponibilidad mostrada está acorde con los datos predefinidos. **Éxito.**
2. **Comprobación de que un usuario con rol Gestor puede consultar el horario de disponibilidad de una semana en concreto.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor. Desde “/inicio” se pulsa en Consultar horario de disponibilidad, se muestran los días lunes, martes, miércoles, jueves y viernes, y un selector para el primer día de la semana. Se selecciona la fecha 22/04/2024. Se comprueba que la disponibilidad mostrada está acorde con los datos predefinidos. **Éxito.**
3. **Comprobación de que un usuario con rol Gestor, al consultar el horario de disponibilidad, no puede elegir otra fecha que no sea un lunes.** Escenario múltiple: Autenticado con el usuario del alumno con rol de Gestor. Desde “/inicio” se pulsa en Consultar horario de disponibilidad, se muestran los días lunes, martes, miércoles, jueves y viernes, y un selector para el primer día de la semana. Se selecciona la fecha **23/04/2024, 24/04/2024, 25/04/2024, 26/04/2024, 27/04/2024, 28/04/2024**. Se muestra un mensaje de error indicando que la fecha debe de ser un lunes. **Éxito.**
4. **Comprobación de que un usuario con rol Gestor, al consultar el horario de disponibilidad, puede utilizar el botón de “Atrás”.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor. Desde “/inicio” se pulsa en Consultar horario de disponibilidad, se muestran los días lunes, martes, miércoles, jueves y viernes, y un selector para el primer día de la semana. Se pulsa sobre el botón de “Atrás”, la aplicación redirige a la pantalla de inicio. **Éxito.**
5. **Comprobación de que un usuario con rol Estudiante o PDI/PAS no puede consultar el horario de disponibilidad.** Escenario único: Autenticado con el usuario del alumno con rol de **Estudiante, PDI/PAS**. Desde “/inicio” se introduce la ruta “/consultar/horario”. La aplicación muestra un mensaje de error, impidiendo el acceso a la información. **Éxito.**

### 6.6.12. Gestora - Modificar el horario de disponibilidad

Se ha añadido un horario de disponibilidad por defecto para todas las pruebas:

1. Lunes: (Duración de primera consulta 0; duración revisión: 0; primera hora: null; segunda hora: null; tercera hora: null; última hora: null)
2. Martes: (Duración de primera consulta 60; duración revisión: 30; primera hora: 09:00; segunda hora: 12:00; tercera hora: 16:00; última hora: 18:00)
3. Miércoles: (Duración de primera consulta 60; duración revisión: 30; primera hora: 09:00; segunda hora: 12:00; tercera hora: 16:00; última hora: 18:00)
4. Jueves: (Duración de primera consulta 0; duración revisión: 0; primera hora: null; segunda hora: null; tercera hora: null; última hora: null)
5. Viernes: (Duración de primera consulta 0; duración revisión: 0; primera hora: null; segunda hora: null; tercera hora: null; última hora: null)

Se han añadido varias citas que se utilizarán en algunos escenarios:

1. cita1: (Fecha: 09/04/2024; Hora: 11:00; Estado: Pendiente; Tipo de consulta: Primera consulta; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el solicitante como paciente)
2. cita2: (Fecha: 10/04/2024; Hora: 11:00; Estado: Pendiente; Tipo de consulta: Revisión; Atendido anteriormente por: Me atendió alguien; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el solicitante como paciente)

Las pruebas se han realizado sobre la semana en la que se encuentra creada la cita (08/04/2024), se han hecho distintas pruebas para comprobar el correcto funcionamiento:

1. **Comprobación de que un usuario con rol Gestor puede modificar el horario de disponibilidad sin cambiar ningún valor.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor y con la “cita1” creada. Desde “/inicio” se pulsa en Consultar horario de disponibilidad, se muestran los días lunes, martes, miércoles, jueves y viernes, y un selector para el primer día de la semana. Se comprueba que el día seleccionado de por defecto es el lunes de la semana actual. Se pulsa sobre “Modificar horario”. Se muestra el mismo horario, pero con campos modificables, los días sin horario se muestran como desactivados. Se pulsa sobre “Modificar horario”. Aparece un mensaje de confirmación y se pulsa en aceptar. Se muestra un mensaje de éxito junto con el aviso de que no se ha bloqueado ninguna cita. Se comprueba que el horario de disponibilidad es el mismo. **Éxito.**
2. **Comprobación de que un usuario con rol Gestor puede modificar el horario de disponibilidad con una nueva franja horaria.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor y con la “cita1” creada. Se accede a la ruta “/consultar/horario/modificar?monday=2024-04-08&friday=2024-04-12”. En el

martes, se introduce la nueva franja horaria “10:00-12:00 y 13:00-16:00”. Se pulsa sobre “Modificar horario”. Aparece un mensaje de confirmación y se pulsa en aceptar. Se muestra un mensaje de éxito junto con el aviso de que no se ha bloqueado ninguna cita. Se comprueba que el horario de disponibilidad ha cambiado. Se comprueba que al solicitar una cita el horario de disponibilidad ha cambiado. **Éxito.**

3. **Comprobación de que un usuario con rol Gestor puede modificar el horario de disponibilidad, con nuevas duraciones de consulta.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor y con la “cita1” creada. Se accede a la ruta “/consultar/horario/modificar?monday=2024-04-08&friday=2024-04-12”. En el martes, se introduce las duraciones de consulta “30 para primera consulta y 15 para revisión”. Se pulsa sobre “Modificar horario”. Aparece un mensaje de confirmación y se pulsa en aceptar. Se muestra un mensaje de éxito junto con el aviso de que no se ha bloqueado ninguna cita. Se comprueba que el horario de disponibilidad ha cambiado. Se comprueba que al solicitar una cita el horario de disponibilidad ha cambiado. **Éxito.**

4. **Comprobación de que un usuario con rol Gestor puede modificar el horario de disponibilidad bloqueando/desbloqueando días.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor y con la “cita1” creada. Se accede a la ruta “/consultar/horario/modificar?monday=2024-04-08&friday=2024-04-12”. Se pulsa sobre la “checkbox” del lunes y permite rellenar los campos de texto; se introduce la nueva franja horaria “10:00-12:00 y 13:00-16:00” y las duraciones de consulta “60 para primera consulta y 30 para revisión”. Se pulsa sobre la “checkbox” del miércoles, impidiendo la modificación de los campos de texto. Se pulsa sobre “Modificar horario”. Aparece un mensaje de confirmación y se pulsa en aceptar. Se muestra un mensaje de éxito junto con el aviso de que no se ha bloqueado ninguna cita. Se comprueba que el horario de disponibilidad ha cambiado. Se comprueba que al solicitar una cita el horario de disponibilidad ha cambiado. **Éxito.**

5. **Comprobación de que un usuario con rol Gestor puede modificar el horario de disponibilidad, bloqueando un día que tiene una cita pendiente, la cita pasa a estado bloqueado.**

Escenario único: Autenticado con el usuario del alumno con rol de Gestor y con la “cita1” creada. Se accede a la ruta “/consultar/horario/modificar?monday=2024-04-08&friday=2024-04-12”. Se pulsa sobre la “checkbox” del miércoles, impidiendo la modificación de los campos de texto. Se pulsa sobre “Modificar horario”. Aparece un mensaje de confirmación y se pulsa en aceptar. Se muestra un mensaje de éxito junto con el aviso de que se ha bloqueado una cita. Se comprueba que la cita tiene el estado de bloqueada. Se comprueba que el horario de disponibilidad ha cambiado. Se comprueba que al solicitar una cita el horario de disponibilidad ha cambiado. **Éxito.**

6. **Comprobación de que un usuario con rol Gestor puede modificar el horario de disponibilidad, dejando una cita pendiente de tipo primera consulta fuera de la franja horaria, la cita pasa a estado bloqueado.** Escenario múltiple: Autenticado con el usuario del alumno con rol de Gestor y con la “cita1” creada. Se accede a la ruta “/consultar/horario/modificar?monday=2024-04-08&friday=2024-04-12”. En el martes, se introduce la nueva franja horaria: “09:00-10:00”, “09:00-11:00”, “12:00-13:00”. Se pulsa sobre “Modificar horario”. Aparece un mensaje de confirmación y se pulsa en aceptar. Se muestra un mensaje de éxito junto con el aviso de que



se ha bloqueado una cita. Se comprueba que la cita tiene el estado de bloqueada. Se comprueba que el horario de disponibilidad ha cambiado. Se comprueba que al solicitar una cita el horario de disponibilidad ha cambiado. **Éxito.**

7. **Comprobación de que un usuario con rol Gestor puede modificar el horario de disponibilidad, dejando una cita pendiente de tipo revisión fuera de la franja horaria, la cita pasa a estado bloqueado.** Escenario múltiple: Autenticado con el usuario del alumno con rol de Gestor y con la “cita2” creada. Se accede a la ruta “/consultar/horario/modificar?monday=2024-04-08&friday=2024-04-12”. En el miércoles, se introduce la nueva franja horaria: “09:00-10:00”, “09:00-11:00”, “12:30-13:30”. Se pulsa sobre “Modificar horario”. Aparece un mensaje de confirmación y se pulsa en aceptar. Se muestra un mensaje de éxito junto con el aviso de que se ha bloqueado una cita. Se comprueba que la cita tiene el estado de bloqueada. Se comprueba que el horario de disponibilidad ha cambiado. Se comprueba que al solicitar una cita el horario de disponibilidad ha cambiado. **Éxito.**
8. **Comprobación de que un usuario con rol Gestor puede modificar el horario de disponibilidad, dejando una cita pendiente de tipo primera consulta fuera del horario porque su duración no cabe en el horario, la cita pasa a estado bloqueado.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor y con la “cita1” creada. Se accede a la ruta “/consultar/horario/modificar?monday=2024-04-08&friday=2024-04-12”. En el martes, se introduce los nuevos valores de franja horaria y duración de primera consulta: 10:30-11:30 y 60”. Se pulsa sobre “Modificar horario”. Aparece un mensaje de confirmación y se pulsa en aceptar. Se muestra un mensaje de éxito junto con el aviso de que se ha bloqueado una cita. Se comprueba que la cita tiene el estado de bloqueada. Se comprueba que el horario de disponibilidad ha cambiado. Se comprueba que al solicitar una cita el horario de disponibilidad ha cambiado. **Éxito.**
9. **Comprobación de que un usuario con rol Gestor puede modificar el horario de disponibilidad, dejando una cita pendiente de tipo revisión fuera del horario porque su duración no cabe en el horario, la cita pasa a estado bloqueado.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor y con la “cita2” creada. Se accede a la ruta “/consultar/horario/modificar?monday=2024-04-08&friday=2024-04-12”. En el miércoles, se introduce los nuevos valores de franja horaria y duración de revisión: 10:30-11:30 y 60”. Se pulsa sobre “Modificar horario”. Aparece un mensaje de confirmación y se pulsa en aceptar. Se muestra un mensaje de éxito junto con el aviso de que se ha bloqueado una cita. Se comprueba que la cita tiene el estado de bloqueada. Se comprueba que el horario de disponibilidad ha cambiado. Se comprueba que al solicitar una cita el horario de disponibilidad ha cambiado. **Éxito.**
10. **Comprobación de que un usuario con rol Gestor puede modificar el horario de disponibilidad, dejando una cita bloqueada y al poner de nuevo el anterior horario, la cita pasa a estado pendiente.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor y con la “cita1” creada. Se accede a la ruta “/consultar/horario/modificar?monday=2024-04-08&friday=2024-04-12”. En el martes, se introduce la nueva franja horaria: “09:00-10:00”. Se pulsa sobre “Modificar horario”. Aparece un mensaje de confirmación y se pulsa en aceptar. Se muestra un mensaje de éxito junto con el aviso de que se ha bloqueado una cita. Se comprueba que la cita tiene el estado de bloqueada. Se accede a la ruta

“/consultar/horario/modificar?monday=2024-04-08&friday=2024-04-12”. En el martes, se introduce la nueva franja horaria: “09:00-14:00”. Se pulsa sobre “Modificar horario”. Aparece un mensaje de confirmación y se pulsa en aceptar. Se muestra un mensaje de éxito junto con el aviso de que se ha desbloqueado una cita. Se comprueba que la cita tiene el estado de pendiente. **Éxito.**

11. **Comprobación de que un usuario con rol Gestor, con horas bloqueadas previamente en el intervalo en el que se encuentra una cita, se modifica el horario de disponibilidad dejando una cita fuera del horario, luego se pone de nuevo el anterior horario, la cita no se desbloquea porque las horas siguen bloqueadas.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor y con la “cita1” creada. Se ejecuta el escenario 6.6.12.10. Se accede a la ruta “/consultar/horario/modificar?monday=2024-04-08&friday=2024-04-12”. En el martes, se introduce la nueva franja horaria: “09:00-10:00”. Se pulsa sobre “Modificar horario”. Aparece un mensaje de confirmación y se pulsa en aceptar. Se muestra un mensaje de éxito junto con el aviso de que no se ha bloqueado ninguna cita. Se comprueba que la cita tiene el estado de bloqueada. Se accede a la ruta “/consultar/horario/modificar?monday=2024-04-08&friday=2024-04-12”. En el martes, se introduce la nueva franja horaria: “09:00-14:00”. Se pulsa sobre “Modificar horario”. Aparece un mensaje de confirmación y se pulsa en aceptar. Se muestra un mensaje de éxito junto con el aviso de que no se ha desbloqueado ninguna cita. Se comprueba que la cita tiene el estado de bloqueada. **Fallido.** Tras la revisión de casos de uso: **Éxito.**

### 6.6.13. Gestora - Bloquear días

Se ha añadido un horario de disponibilidad por defecto para todas las pruebas:

1. Lunes: (Duración de primera consulta 0; duración revisión: 0; primera hora: null; segunda hora: null; tercera hora: null; última hora: null)
2. Martes: (Duración de primera consulta 60; duración revisión: 30; primera hora: 09:00; segunda hora: 12:00; tercera hora: 16:00; última hora: 18:00)
3. Miércoles: (Duración de primera consulta 60; duración revisión: 30; primera hora: 09:00; segunda hora: 12:00; tercera hora: 16:00; última hora: 18:00)
4. Jueves: (Duración de primera consulta 0; duración revisión: 0; primera hora: null; segunda hora: null; tercera hora: null; última hora: null)
5. Viernes: (Duración de primera consulta 0; duración revisión: 0; primera hora: null; segunda hora: null; tercera hora: null; última hora: null)

Se ha añadido una cita que se utilizará en algunos escenarios:

1. cita1: (Fecha: 16/04/2024; Hora: 11:00; Estado: Pendiente; Tipo de consulta: Primera consulta; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo

de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el solicitante como paciente)

Ya que este caso de uso tiene pocas variaciones, solo se han realizado tres pruebas para comprobar el correcto funcionamiento:

1. **Comprobación de que un usuario con rol Gestor puede bloquear días.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor. Desde “/inicio” se pulsa en Bloquear días. Se muestran dos selectores de fecha (inicio y fin). Se seleccionan las fechas 16/04/2024 y 17/04/2024. Se pulsa sobre “Aceptar cambios”. Aparece un mensaje de confirmación y se pulsa en aceptar. Se muestra un mensaje de éxito junto con el aviso de que no se ha bloqueado ninguna cita y una vista del calendario. Se pulsa sobre continuar, la aplicación redirige a la pantalla de inicio Se comprueba que los días están bloqueados. **Éxito.**
2. **Comprobación de que un usuario con rol Gestor puede bloquear días en los que hay citas pendientes.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor y con la “cita1” creada. Desde “/inicio” se pulsa en Bloquear días. Se muestran dos selectores de fecha (inicio y fin). Se seleccionan las fechas 16/04/2024 y 17/04/2024. Se pulsa sobre “Aceptar cambios”. Aparece un mensaje de confirmación y se pulsa en aceptar. Se muestra un mensaje de éxito junto con el aviso de que se han bloqueado dos citas y una vista del calendario. Se pulsa sobre continuar, la aplicación redirige a la pantalla de inicio Se comprueba que los días están bloqueados. Se comprueba que los días están bloqueados. Se comprueban que las dos citas han pasado a estado bloqueado. **Éxito.**
3. **Comprobación de que un usuario con rol Estudiante o PDI/PAS no puede bloquear días.** Escenario múltiple: Autenticado con el usuario del alumno con rol de **Estudiante, PDI/PAS.** Desde “/inicio” se introduce la ruta “/bloqueo/dias”. La aplicación muestra un mensaje de error, impidiendo el acceso a la pantalla de bloqueo de días. **Éxito.**

### 6.6.14. Gestora - Bloquear horas

Se ha añadido un horario de disponibilidad por defecto para todas las pruebas:

1. Lunes: (Duración de primera consulta 0; duración revisión: 0; primera hora: null; segunda hora: null; tercera hora: null; última hora: null)
2. Martes: (Duración de primera consulta 60; duración revisión: 30; primera hora: 09:00; segunda hora: 12:00; tercera hora: 16:00; última hora: 18:00)
3. Miércoles: (Duración de primera consulta 60; duración revisión: 30; primera hora: 09:00; segunda hora: 12:00; tercera hora: 16:00; última hora: 18:00)

4. Jueves: (Duración de primera consulta 0; duración revisión: 0; primera hora: null; segunda hora: null; tercera hora: null; última hora: null)
5. Viernes: (Duración de primera consulta 0; duración revisión: 0; primera hora: null; segunda hora: null; tercera hora: null; última hora: null)

Se ha añadido una cita que se utilizará en algunos escenarios:

1. cita1: (Fecha: 16/04/2024; Hora: 11:00; Estado: Pendiente; Tipo de consulta: Primera consulta; Estado: Pendiente; Motivo de consulta: Audición; Especificación del motivo de consulta: Esta es una prueba de pedir una cita como estudiante o PDIPAS...; el solicitante como paciente)

Se han hecho distintas pruebas para comprobar el correcto funcionamiento:

1. **Comprobación de que un usuario con rol Gestor puede bloquear y desbloquear horas.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor. Desde “/inicio” se pulsa en Bloquear horas. Se muestra un selector de fecha, dos campos de texto para introducir el intervalo de horas, la franja horaria y las horas bloqueadas del día seleccionado. Se selecciona la fecha 16/04/2024. Se muestra correctamente el horario y ninguna hora bloqueada en el día seleccionado. Se introducen las horas “09:00 y 12:00”. Se pulsa sobre “Aceptar cambios”. Aparece un mensaje de confirmación y se pulsa en aceptar. Se muestra un mensaje de éxito junto con el aviso de que no se ha bloqueado ninguna cita. Se comprueba que las horas están bloqueadas en el día correspondiente. Se accede a “/bloqueo/horas”. Se selecciona la fecha 16/04/2024. Se muestra la franja horaria bloqueada junto con un botón de desbloquear horas. Se pulsa sobre desbloquear horas. Aparece un mensaje de confirmación y se pulsa en aceptar. Se muestra un mensaje de éxito junto con el aviso de que no se ha desbloqueado ninguna cita. Se comprueba que las horas ya no están bloqueadas en el día correspondiente. **Éxito.**
2. **Comprobación de que un usuario con rol Gestor puede bloquear y desbloquear horas en las que se encuentra una cita pendiente, la cita pasa al estado bloqueado.** Escenario múltiple: Autenticado con el usuario del alumno con rol de Gestor y con la “cita1” creada. Desde “/inicio” se pulsa en Bloquear horas. Se muestra un selector de fecha, dos campos de texto para introducir el intervalo de horas, la franja horaria y las horas bloqueadas del día seleccionado. Se selecciona la fecha 16/04/2024. Se muestra correctamente el horario y ninguna hora bloqueada en el día seleccionado. Se introducen las horas “09:00 y 12:00”, “11:00 y 12:00” y “11:30 y 12:30”. Se pulsa sobre “Aceptar cambios”. Aparece un mensaje de confirmación y se pulsa en aceptar. Se muestra un mensaje de éxito junto con el aviso de que se ha bloqueado una cita. Se comprueba que la cita ha pasado al estado de bloqueada. Se accede a “/bloqueo/horas”. Se selecciona la fecha 16/04/2024. Se muestra la franja horaria bloqueada junto con un botón de desbloquear horas. Se pulsa sobre desbloquear horas. Aparece un mensaje de confirmación y se pulsa en aceptar. Se muestra un mensaje de éxito junto con el aviso de que se ha desbloqueado una cita. Se comprueba que la cita ha pasado al estado pendiente. **Éxito.**

3. **Comprobación de que un usuario con rol Gestor puede bloquear horas varias veces y la franja bloqueada se agrupa.** Escenario múltiple: Autenticado con el usuario del alumno con rol de Gestor. Desde “/inicio” se pulsa en Bloquear horas. Se muestra un selector de fecha, dos campos de texto para introducir el intervalo de horas, la franja horaria y las horas bloqueadas del día seleccionado. Se selecciona la fecha 16/04/2024. Se muestra correctamente el horario y ninguna hora bloqueada en el día seleccionado. Se introducen las horas “09:00 y 12:00”. Se pulsa sobre “Aceptar cambios”. Aparece un mensaje de confirmación y se pulsa en aceptar. Se muestra un mensaje de éxito junto con el aviso de que no se ha bloqueado ninguna cita. Se comprueba que las horas están bloqueadas en el día correspondiente. Se accede a “/bloqueo/horas”. Se selecciona la fecha 16/04/2024. Se muestra la franja horaria bloqueada “09:00-12:00”. Se introducen las horas “10:00 y 14:00”, “09:00 y 16:00”. Se pulsa sobre “Aceptar cambios”. Aparece un mensaje de confirmación y se pulsa en aceptar. Se muestra un mensaje de éxito junto con el aviso de que no se ha bloqueado ninguna cita. Se comprueba que las horas están bloqueadas en el día correspondiente. Se accede a “/bloqueo/horas”. Se selecciona la fecha 16/04/2024. Se muestra la franja horaria bloqueada “09:00-14:00”, “09:00-16:00”. **Fallido**, en el segundo caso, las horas bloqueadas no se agruparon. Tras la revisión de casos de uso: **Éxito**.
4. **Comprobación de que un usuario con rol Estudiante o PDI/PAS no puede bloquear horas.** Escenario único: Autenticado con el usuario del alumno con rol de **Estudiante, PDI/PAS**. Desde “/inicio” se introduce la ruta “/bloqueo/horas”. La aplicación muestra un mensaje de error, impidiendo el acceso a la pantalla de bloqueo de días. **Éxito**.

### 6.6.15. Gestora - Consultar calendario

Se ha añadido un horario de disponibilidad por defecto para todas las pruebas:

1. Lunes: (Duración de primera consulta 0; duración revisión: 0; primera hora: null; segunda hora: null; tercera hora: null; última hora: null)
2. Martes: (Duración de primera consulta 60; duración revisión: 30; primera hora: 09:00; segunda hora: 12:00; tercera hora: 16:00; última hora: 18:00)
3. Miércoles: (Duración de primera consulta 60; duración revisión: 30; primera hora: 09:00; segunda hora: 12:00; tercera hora: 16:00; última hora: 18:00)
4. Jueves: (Duración de primera consulta 0; duración revisión: 0; primera hora: null; segunda hora: null; tercera hora: null; última hora: null)
5. Viernes: (Duración de primera consulta 0; duración revisión: 0; primera hora: null; segunda hora: null; tercera hora: null; última hora: null)

Se han hecho distintas pruebas para comprobar el correcto funcionamiento:

1. **Comprobación de que un usuario con rol Gestor puede consultar el horario mensual y anual.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor. Desde “/inicio” se pulsa en Nuestro calendario. Se muestra un calendario donde aparecen los días libres y bloqueados, y un botón de “Vista anual”. Se presionan el botón de avance de mes y el calendario se actualiza con la información correspondiente. Se presiona el botón de “Vista anual”. Se muestran nueve calendarios correspondientes a los meses de abril-diciembre del año actual (2024). Se presiona sobre el botón de siguiente año. Se muestran doce calendarios correspondientes a los meses de enero-diciembre del año siguiente (2025). **Éxito.**
2. **Comprobación de que un usuario con rol Gestor puede consultar el horario mensual y anual.** Escenario único: Autenticado con el usuario del alumno con rol de Gestor. Se ejecuta previamente la prueba 6.6.13.2. Desde “/inicio” se pulsa en Nuestro calendario. Se muestra un calendario donde aparecen los días libres y bloqueados, y un botón de “Vista anual”. Se comprueba que aparecen los días bloqueados. **Éxito.**
3. **Comprobación de que un usuario con rol Estudiante o PDI/PAS no puede consultar el horario.** Escenario único: Autenticado con el usuario del alumno con rol de **Estudiante, PDI/PAS**. Desde “/inicio” se introduce la ruta “/consultar/calendario/mes”, “/consultar/calendario/anual. La aplicación muestra un mensaje de error, impidiendo el acceso a la pantalla del calendario. **Éxito.**

### 6.6.16. Valoración tras finalizar las pruebas

Pese a que, finalmente todas las pruebas pasaron con éxito, cabe destacar algunas de ellas que fueron de gran utilidad para detectar errores y fue necesario repetir varias veces. Aquí se detallan:

- Prueba 6.6.1.3: Ayudó a comprobar que, en ese momento, la vista de inicio para el rol de gestor aún no estaba implementada.
- Prueba 6.6.1.5: Al igual que el anterior, dado que no existía una vista de inicio para gestoras, no era posible desconectar la sesión.
- Prueba 6.6.9.2: Con esta prueba se pudo detectar que tras actualizar la fecha y hora de una cita bloqueada, su estado no volvía a “Pendiente”, continuando como “Bloqueada”.
- Prueba 6.6.10.3: Este escenario se repitió en varias ocasiones, ya que las combinaciones de estado inicial y nuevo estado son numerosas. Se pudo detectar que las citas en estado “Bloqueado” tenían la opción de cambiar el estado, cuando las únicas opciones disponibles deben de ser “Modificar fecha y hora” y “Cancelar cita”.
- Prueba 6.6.12.11: Este escenario ayudó a detectar un caso concreto y que podría haber pasado desapercibido. Dado que al modificar el horario de disponibilidad, las citas en estado “Bloqueado” que tienen hueco en el nuevo horario su estado pasa automáticamente a “Pendiente”. El error sucedía al llevarse a cabo esta automatización, no se tenía en cuenta las horas que podían estar bloqueadas, dando como resultado el desbloqueo de citas en horarios bloqueados.

- Prueba 6.6.14.3: Se pudo detectar un error muy concreto en la función que reordenaba las franjas de horas bloqueadas. Cuando existía una franja de horas y se bloqueaba otra franja donde la hora de inicio coincidía con la ya existente, no se agrupaban.





## Capítulo 7

# Seguimiento del proyecto

En esta sección se detalla como ha sido el seguimiento real del proyecto, así como la aparición de riesgos y cómo se han gestionado.

### 7.1. Fase de inicio

Durante la fase de inicio se produjeron los riegos R8 y R9; en cuanto al R8 no se organizaron correctamente las horas de estudio y dedicación al TFG, el R9 también se produjo y se tuvo que ir a convocatoria extraordinaria, esto retrasó la planificación 3 semanas y se anuló la reunión semanal del 07/06/2023, dando así tiempo para el estudio de los exámenes. Se produjo el R10, con ello se tuvo que añadir y mover el orden de algunas tareas: se añadió otra tarea más para especificar los casos de uso después de la reunión semanal del 10/05/2023, se añadió una tarea para revisar el documento de la memoria, se añadió una tarea para revisar el contexto y motivación, otra tarea con una descripción de la aplicación actual y finalmente se tuvieron que mover los órdenes de varias tareas.

Debido al retraso de 3 semanas se añadieron dos reuniones semanales y sus respectivas tareas de revisar requisitos y casos de uso. A continuación se muestra la tabla de la fase de inicio con las horas y fechas reales de cada tarea:

Tarea	Duración real	Duración estimada	Fecha inicio	Fecha fin	Anotaciones
Primera reunión	1 hora	1 hora	13/04/2023	13/04/2023	Reunión con tutores y las logopedas del Gabinete de Logopedia.

## 7.1. FASE DE INICIO

Contexto, alcance y objetivos	5 horas	5 horas	14/04/2023	15/04/2023	
Identificar interesados	2 horas	2 horas	15/04/2023	15/04/2023	
Elicitación de requisitos	5 horas	8 horas	15/04/2023	16/04/2023	
Reunión semanal	1 hora	1 hora	10/05/2023	10/05/2023	Reuniones semanales con tutores.
Revisión de requisitos	4 horas	3 horas	11/05/2023	12/05/2023	
Especificación de casos de uso	4 horas	8 horas	14/05/2023	15/05/2023	
Diagrama de casos de uso	2 horas	2 horas	15/05/2023	15/05/2023	
Identificación de riesgos	2 horas	2 horas	15/05/2023	16/05/2023	
Presupuestos	2 horas	2 horas	16/05/2023	16/05/2023	
Investigar Privacidad desde el diseño	1 hora	1 hora	16/05/2023	16/05/2023	
Reunión semanal	1 hora	1 hora	17/05/2023	17/05/2023	Reunión con tutores y gabinete.
Revisar requisitos y casos de uso	6 horas	3 horas	17/05/2023	18/05/2023	
Especificación de casos de uso	5 horas	- horas	19/05/2023	20/05/2023	
Reunión semanal	1 hora	1 hora	23/05/2023	23/05/2023	Reunión semanales con tutores.
Primera reunión	1 hora	-	13/04/2023	13/04/2023	Reunión con tutores y las logopedas del Gabinete de Logopedia.
Revisar requisitos y casos de uso	4 horas	-	25/05/2023	26/05/2023	

Revisar memoria en general	3 horas	-	27/05/2023	27/05/2023	
Reunión semanal	1 hora	-	31/05/2023	31/05/2023	Reunión semanales con tutores.
Revisar contexto y motivación	1 horas	-	09/06/2023	09/06/2023	
Aplicación actual	1 hora	-	09/06/2023	09/06/2023	
Revisar requisitos y casos de uso	2 horas	-	10/05/2023	10/05/2023	
Investigar RUP	1 hora	1 hora	12/06/2023	12/06/2023	
Planificación	4 horas	4 horas	12/06/2023	13/06/2023	
Seguimiento	2 horas	4 horas	13/06/2023	13/06/2023	

Tabla 7.1: Seguimiento planificación inicio.

## 7.2. Fase de elaboración

Durante la fase de elaboración se produjo el R10 y se tuvo que reorganizar el orden de algunas tareas.

En la segunda semana, no se pudo acabar el diagrama de actividad por fechas de exámenes, así que se decidió añadir una tarea más para completarlo. Aun así se realizó igualmente la iteración de revisión de diagramas, y por la misma razón se retrasó la segunda reunión semanal dos días.

En la tercera semana, se dividió la tarea de investigación del análisis de riesgos de seguridad, ya que se requería más tiempo del planificado. Después de la reunión semanal del 28/06/2023 se acordó enseñar los bocetos a las logopedas del gabinete para hacer una iteración con ellas. Por lo tanto, se realizaron dos iteraciones seguidas, la primera con las indicaciones de los tutores y la segunda con las trabajadoras del gabinete.

En la cuarta semana, se añadió una tarea nueva para rellenar algunos apartados de la memoria, se eliminó la reunión por problemas de disponibilidad de los tutores y se añadió una tercera iteración de los diagramas de análisis.

En la quinta semana, se añadió una iteración más para la revisión de los bocetos de la interfaz.

Tarea	Duración real	Duración estimada	Fecha inicio	Fecha fin	Anotaciones
Reunión semanal	1 hora	1 hora	14/06/2023	14/06/2023	Reuniones semanales con tutores.
Modelo de dominio	3 horas	4 horas	15/06/2023	15/06/2023	
Diagramas de máquinas de estados	3 horas	5 horas	17/06/2023	17/06/2023	
Diagrama de actividad 1/2	2 horas	2 horas	19/06/2023	19/06/2023	
Reunión semanal	1 hora	1 horas	23/06/2023	23/06/2023	Reuniones semanales con tutores.
Revisión de diagramas de análisis	5 horas	5 horas	24/06/2023	25/06/2023	
Diagrama de actividad 2/2	3 horas	2 horas	26/06/2023	26/06/2023	

Prototipos de interfaz escritorio	3 horas	- horas	26/06/2023	27/06/2023	
Investigar análisis de riesgos de seguridad 1/2	3 horas	4 horas	27/06/2023	27/06/2023	
Reunión semanal	1 hora	1 horas	28/06/2023	28/06/2023	Reuniones semanales con tutores.
Revisión de diagramas de análisis	3 horas	3 horas	29/06/2023	30/06/2023	
Revisión de prototipos de interfaz	6 horas	3 horas	30/06/2023	04/07/2023	
Investigar análisis de riesgos de seguridad 2/2	4 horas	4 horas	06/07/2023	06/07/2023	
Análisis de riesgos de seguridad	4 horas	8 horas	07/07/2023	08/07/2023	
Rellenar memoria	3 horas	4 horas	09/07/2023	09/07/2023	
Reunión semanal	1 hora	1 horas	13/07/2023	13/07/2023	Reuniones semanales con tutores.
Revisión de prototipos de interfaz	1 hora	3 horas	15/07/2023	15/07/2023	

Tabla 7.2: Seguimiento planificación elaboración 1/3.

## 7.2. FASE DE ELABORACIÓN

Tras lo explicado en el capítulo de la replanificación, a partir de aquí se llevó a cabo una pausa en el proyecto, por lo que se dejó de seguir la planificación original y en la vuelta de verano se replanificaría.

Aunque se realizara esta pausa, antes de agosto tuvieron lugar algunas reuniones con tutores y con el responsable de privacidad y se realizaron las siguientes tareas (Tabla 7.3). Durante este periodo se dio el riesgo R10, en la planificación original había algunas tareas necesarias que no estaban definidas, así que algunas de ellas se realizaron en este intervalo de tiempo, el resto se añadirían en la replanificación.

A finales de agosto, al retomar las reuniones semanales con los tutores, éstas dejaron de ser periódicas debido a distintas circunstancias ajenas al estudiante y profesores. Se planificarían reuniones según disponibilidad y avances en el proyecto.

Tarea	Duración real	Duración estimada	Fecha inicio	Fecha fin	Anotaciones
Reunión semanal	1 hora	1 hora	18/07/2023	18/07/2023	Reunión con Yania y responsable de privacidad.
Reunión semanal	1 hora	1 hora	27/07/2023	27/07/2023	Reunión con Yania.
Rellenar memoria	4 horas	- horas	01/08/2023	09/08/2023	Reescritura y corrección de algunas secciones.
Diseño de despliegue	4 horas	- horas	12/08/2023	20/08/2023	
Reunión semanal	1 hora	1 hora	07/09/2023	07/09/2023	Reunión con Yania.
Revisión diseño de despliegue	- horas	4 horas	11/09/2023	11/09/2023	
Prototipos de interfaz móvil	5 horas	- horas	13/09/2023	17/09/2023	
Diseño de almacenamiento	3 horas	- horas	20/09/2023	22/09/2023	
Preparación de entorno de desarrollo	3 horas	- horas	25/09/2023	29/09/2023	
Replanificación	6 horas	- horas	09/10/2023	17/10/2023	

Tabla 7.3: Seguimiento planificación elaboración 2/3.

Una vez realizada la replanificación, se comienza a llevar a cabo.

En la primera semana, se suprimió la tarea de revisión de prototipos de interfaz, debido a que después de la tutoría no se sugirió ningún cambio, y se añadió una tarea para rellenar la memoria.

Tarea	Duración real	Duración estimada	Fecha inicio	Fecha fin	Anotaciones
Investigar arquitectura	5 horas	3 horas	17/10/2023	21/10/2023	
Reunión tutoría	1 hora	1 hora	18/10/2023	18/10/2023	Reunión con Yania.
Revisión de diseño de despliegue	1 hora	1 hora	19/10/2023	19/10/2023	
Revisión de diseño de almacenamiento	1 hora	1 hora	19/10/2023	19/10/2023	
Diseño arquitectónico	5 horas	4 horas	19/10/2023	21/10/2023	
Rellenar memoria	1 horas	- horas	21/10/2023	21/10/2023	
Investigar funcionamiento SSO UVa	4 horas	3 horas	21/10/2023	21/10/2023	
Reunión tutoría	1 horas	1 hora	27/10/2023	27/10/2023	Reunión con Yania.
Revisión de parte de diseño	2 horas	2 horas	27/10/2023	27/10/2023	

Tabla 7.4: Seguimiento planificación elaboración 3/3.

### 7.3. Fase de construcción

#### 7.3.1. Primera iteración

Una vez finalizada la fase de elaboración, da comienzo la primera iteración de la fase de construcción.

En la primera semana (23/10/2023), se dedicó al desarrollo de casos de uso 1/3. Se hicieron los cimientos de la aplicación con el frontend de Nginx y el backend de Spring Boot, este último con la arquitectura “clean”.

En la segunda semana (30/10/2023), se adelantó la tarea de contenerización, ya que se vio más conveniente integrarlo desde el principio e ir escalándolo según la evolución del proyecto, de esta forma facilita los levantamientos de la aplicación a la hora de desarrollar. Se empezó a desarrollar el backend del caso de uso de pedir cita. Se cambió la fecha de la primera reunión semanal, a causa de disponibilidades entre los participantes.

En la tercera semana (06/11/2023), por ineficiencia en la organización, se dedicaron menos horas de las previstas. Se añadió la herramienta de git-crypt en el repositorio, se hizo la interfaz de la pantalla de inicio y se continuó con el desarrollo del caso de pedir cita. Además, se llevó a cabo una reunión con la tutora.

En la cuarta semana (13/11/2023), se creó la estructura de la mayoría de clases de la arquitectura clean, se dedicó tiempo a intentar añadir varias bases de datos a Spring Boot, lo que retrasó el desarrollo del caso de uso. Además, se llevó a cabo una reunión con la tutora.

En la quinta semana (20/11/2023), se añadieron con éxito las dos bases adicionales de Spring Boot siguiendo una guía de Baeldung [17], se implementó el endpoint para la creación de una cita y se empezó a desarrollar la primera interfaz del caso de uso de pedir cita con un calendario. Además, se llevó a cabo una reunión con la tutora.

En la sexta semana (27/11/2023), se añadió la interfaz con el calendario dinámico [25], se decidió eliminar el frontend con Nginx y unificarlo con Spring Boot, de esta forma se puede utilizar la herramienta Thymeleaf, se implementó el caso de uso de inicio de sesión integrándolo con el SSO de la UVa y se realizaron pruebas en cuanto a su correcta funcionalidad. Además, se llevó a cabo una reunión con la tutora.

En la séptima semana (04/12/2023), no se realizó ningún avance debido a la festividad del puente de la constitución.

En la octava semana (11/12/2023), se añadió más funcionalidad a la interfaz del calendario y al flujo del backend en cuanto a pedir una cita. Además, se llevó a cabo una reunión con la tutora.

En la novena semana (18/12/2023), se dedicaron menos horas de las planificadas debido a las festividades de Navidad. En cuanto a la base de datos, se decidió añadir una tabla adicional para guardar el horario de disponibilidad semanal que la aplicación utilizará por defecto. En cuanto al caso de uso de pedir cita, se enriqueció la funcionalidad para que el



calendario muestre los días disponibles según el horario por defecto o los días bloqueados. Además, se llevó a cabo una reunión con la tutora.

En la décima semana (25/12/2023), se dedicó muy poco tiempo de trabajo debido a las festividades de Navidad. En cuanto al caso de uso de pedir cita: Al elegir un día del calendario y el tipo de consulta, se muestran las horas libres considerando el horario de disponibilidad predeterminado, las citas existentes y los días bloqueados.

En la undécima semana (01/01/2024), se dedicó menos tiempo de trabajo debido a las festividades. Se realizó una revisión y actualización de la memoria para reflejar los cambios que se han producido desde el inicio hasta el momento: Se añadió un requisito de información para el nuevo horario de disponibilidad predeterminado, se eliminó la regla de negocio "Los horarios de disponibilidad no podrán superponerse entre las gestoras", ya que solo habrá un único horario de disponibilidad para todas las gestoras y se modificó el caso de uso de "Solicitar una cita" para que el usuario elija el tipo de consulta al seleccionar la fecha y la hora, en lugar de hacerlo al introducir los datos de la consulta. Se continuó el desarrollo del caso de uso de pedir cita: Se crea una cita pre reservada y se muestra el formulario para rellenar los datos del paciente.

En la duodécima semana (08/01/2024), se actualizó el diagrama de actividad de solicitante con el cambio de la semana anterior y el modelo de dominio añadiendo la clase para el horario de disponibilidad predeterminado, se eliminó la clase `SemanaDisponibilidad` que estaba asociada con `DiaDisponibilidad` y se ha eliminado el datatype `direccionPostal` para reemplazarlo con los atributos dirección y localidad. Se realizaron algunas refactorizaciones de elementos que estaban desordenados en el código. Además, se llevó a cabo una reunión con la tutora.

En la decimotercera semana (15/01/2024), se añadió la creación de paciente del caso de uso "Añadir información de pacientes", se añadió la auto-eliminación de las citas que se mantienen pre reservadas durante 10 minutos, se realizaron más refactorizaciones de código para mantener la arquitectura clean. Además, se llevó a cabo una reunión con la tutora.

En la decimocuarta semana (22/01/2024), se añadió la comprobación de solicitud de cita de primera consulta antes que revisión, el frontend de la solicitud de pacientes, la creación del usuario como paciente en el caso de que lo sea y se añadió la desconexión de la sesión (logout). Además, se llevó a cabo una reunión con la tutora.

En la decimoquinta semana (29/01/2024), se realizaron menos horas de las planificadas debido a una caída en enfermedad por parte del estudiante. En el caso de uso "Añadir información de pacientes" se añadieron tarjetas con los pacientes actuales y la posibilidad de editarlos/seleccionarlos, se añadieron controles de seguridad a todos los parámetros de entrada para prevenir inyecciones de comandos SQL, se añadieron validaciones a la clase paciente para validar los formularios de entrada acorde a los caracteres permitidos y lógica.

En la decimosexta semana (05/02/2024), se realizaron correcciones que fueron detectadas en el caso de uso "Añadir paciente", se refactorizó gran parte del front-end para que su organización fuese homogénea, las vistas se hicieron responsive. Se añadió una tarea para escribir en la memoria, donde, se documentó la información correspondiente a la "Clean architecture". Además, se llevó a cabo una reunión con la tutora.

En la decimoséptima semana (12/02/2024), se solucionó un “bug” que se identificó en el caso de uso “Añadir información de pacientes”, donde, tras mostrar una alerta de revisión sin primera consulta previa, el paciente ya se había creado. Se documentó en la memoria: “Clean architecture”, bases de datos, SSO y defensa contra inyecciones. Además, se llevó a cabo una reunión con la tutora.

En la decimoctava semana (19/02/2024), se arregló un bug en el calendario de solicitar cita, donde el último día del mes no estaba siendo calculado correctamente, se implementaron los casos de uso “Solicitante - consultar citas” y “Solicitante - cancelar cita”. Además, se llevó a cabo una reunión con la tutora.

En la decimonovena semana (26/02/2024), se dio como finalizada la tarea de “Casos de uso 1/3”, se realizaron las tareas de “Pruebas Casos de uso 1/3” y Revisión Casos de uso 1/3, y se implementó el caso de uso “Solicitante - Consultar información de contacto del gabinete”. Además, se llevó a cabo una reunión con la tutora.

En la vigésima semana (04/03/2024), se realizaron los casos de uso “Gestora - Consultar citas de todas las gestoras”, “Gestora - Cancelar cita de solicitante” y “Gestora - Modificar fecha u/y hora de una cita”. Además, se llevó a cabo una reunión con la tutora.

En la vigesimoprimer semana (11/03/2024), se añadió un mensaje de confirmación en la finalización de los casos de uso necesarios, se realizó el caso de uso “ Gestora - Cambiar estado de cita”, dando como finalizada la tarea de “Desarrollo Casos de uso 2/3”, se realizaron las tareas de “Pruebas Casos de uso 2/3” y “Revisión Casos de uso 2/3”. Además, se llevó a cabo una reunión con la tutora.

En la vigesimotercera semana (18/03/2024), se dedicaron menos horas debido a un viaje que realizó el estudiante. Se añadió el caso de uso “Gestora - Consultar su horario de disponibilidad”. Además, se llevó a cabo una reunión con la tutora.

En la vigesimocuarta semana (25/03/2024), se añadió el caso de uso “Gestora - Modificar el horario de disponibilidad”.

En la vigesimoquinta semana (01/04/2024), se añadieron los casos de uso “Gestora - Bloquear días”, “Gestora - Bloquear horas” “Gestora - Consultar calendario”. Además, se llevó a cabo una reunión con la tutora.

En la vigesimosexta semana (08/04/2024), se realizaron las tareas de probar los casos de uso 3/3, revisión de casos de uso 3/3 y se inició la de rellenar memoria. Se omitió la tarea de “Pruebas sistema contenerizado”, ya que el sistema se contenerizó desde el principio y ya se fue probando en cada iteración de casos de uso. Además, se llevó a cabo una reunión con la tutora.

En la vigesimoséptima semana (15/04/2024), se finalizó la tarea de rellenar memoria.

CAPÍTULO 7. SEGUIMIENTO DEL PROYECTO

Tarea	Duración real	Duración estimada	Fecha inicio	Fecha fin	Anotaciones
Desarrollo Casos de uso 1/3	125 horas	40 horas	21/10/2023	27/02/2024	CU1-CU5, los casos de uso se desarrollaron simultáneamente con la arquitectura y los cimientos de la aplicación a medida que se iban necesitando. Por lo tanto, se incluyen dentro de esta tarea.
Contenerización	5 horas	4 horas	29/10/2023	01/11/2023	
Escribir memoria	6 horas	- horas	02/01/2024	10/01/2024	
Reunión semanal	1 hora	1 hora	02/11/2023	02/11/2023	Reunión con Yania.
Reunión semanal	1 hora	1 hora	09/11/2023	09/11/2023	Reunión con Yania.
Reunión semanal	1 hora	1 hora	16/11/2023	16/11/2023	Reunión con Yania.
Reunión semanal	1 hora	1 hora	23/11/2023	23/11/2023	Reunión con Yania.
Reunión semanal	1 hora	1 hora	28/11/2023	28/11/2023	Reunión con Yania.
Reunión semanal	1 hora	1 hora	11/12/2023	11/12/2023	Reunión con Yania.
Reunión semanal	1 hora	1 hora	18/12/2023	18/12/2023	Reunión con Yania.
Reunión semanal	1 hora	1 hora	11/01/2024	11/01/2024	Reunión con Yania.
Reunión semanal	1 hora	1 hora	18/01/2024	18/01/2024	Reunión con Yania.
Reunión semanal	1 hora	1 hora	25/01/2024	25/01/2024	Reunión con Yania.
Reunión semanal	1 hora	1 hora	08/02/2024	08/02/2024	Reunión con Yania.

### 7.3. FASE DE CONSTRUCCIÓN

Escribir memoria	9 horas	- horas	11/02/2024	17/02/2024	
Reunión semanal	1 hora	1 hora	15/02/2024	15/02/2024	Reunión con Yania.
Reunión semanal	1 hora	1 hora	22/02/2024	22/02/2024	Reunión con Yania.
Reunión semanal	1 hora	1 hora	27/02/2024	27/02/2024	Reunión con Yania.
Pruebas Casos de uso 1/3	9 horas	5 horas	28/02/2024	02/03/2024	Realización de pruebas y su documentación en la memoria
Revisión Casos de uso 1/3	2 horas	5 horas	02/03/2024	03/03/2024	Arreglo de bugs encontrados en la tarea de pruebas
Desarrollo Casos de uso 2/3	21 horas	40 horas	03/03/2024	16/03/2024	CU6-CU10
Reunión semanal	1 hora	1 hora	05/03/2024	05/03/2024	Reunión con Yania.
Reunión semanal	1 hora	1 hora	12/03/2024	12/03/2024	Reunión con Yania.
Pruebas Casos de uso 2/3	4 horas	5 horas	16/03/2024	17/03/2024	Realización de pruebas y documentarlas en la memoria
Revisión Casos de uso 2/3	2 horas	5 horas	17/03/2024	17/03/2024	Arreglo de bugs encontrados en la tarea de pruebas
Desarrollo Casos de uso 3/3	45 horas	40 horas	17/03/2024	07/04/2024	CU11-CU15
Reunión semanal	1 hora	1 hora	19/03/2024	19/03/2024	Reunión con Yania.
Reunión semanal	1 hora	1 hora	02/04/2024	02/04/2024	Reunión con Yania.

Pruebas Casos de uso 3/3	4 horas	5 horas	07/04/2024	09/04/2024	Realización de pruebas y documentarlas en la memoria
Reunión semanal	1 hora	1 hora	09/04/2024	09/04/2024	Reunión con Yania.
Revisión Casos de uso 3/3	4 horas	7 horas	09/04/2024	10/04/2024	Arreglo de bugs encontrados en la tarea de pruebas
Completar capítulos de la memoria	6 horas	7 horas	10/04/2024	15/04/2024	

Tabla 7.5: Seguimiento planificación construcción 1.

### 7.3.2. Segunda iteración

La segunda iteración no fue necesaria. Dado el buen resultado de la primera iteración y una menor exigencia por la no implantación del proyecto en la UVa.

## 7.4. Fase de transición

Una vez finalizada la fase de construcción, comenzó la fase de transición.

Dado que el proyecto no se implantará en la UVa, se suprimieron las tareas “Despliegue servidores UVa”, “Pruebas sobre servidores UVa”, “Revisión de sistema” y “Revisiones finales”.

En la primera semana (15/04/2024), se realizaron las tareas “Manuales de usuario y despliegue” y “Revisión memoria”, dando el proyecto como finalizado.

Tarea	Duración real	Duración estimada	Fecha inicio	Fecha fin	Anotaciones
Manuales de usuario y despliegue	3 horas	4 horas	15/04/2024	16/04/2024	

Revisión memoria	7 horas	4 horas	16/04/2024	18/04/2024	
------------------	---------	---------	------------	------------	--

Tabla 7.6: Seguimiento planificación transición.

## 7.5. Resumen de la ejecución del proyecto

Con la planificación inicial, la estimación de la dedicación total era de 380 horas y la fecha de finalización del proyecto era el 06/09/2023. El alumno realizaría las prácticas de empresa durante el verano y así poder utilizar la convocatoria para alumnos que realizan prácticas curriculares durante el verano. En el mes de julio, el alumno comenzó las prácticas de empresa con una beca de una duración de seis meses, pero que le requería seguir matriculado en la universidad. Con esta novedad se decidió aplazar la entrega durante el siguiente curso, por lo que se tomó un descanso durante los meses de verano y se realizó una replanificación.

En la replanificación, la nueva estimación de la dedicación total era de 372 horas y la fecha de finalización del proyecto era el 18/01/2024. Durante esta replanificación se sufrieron grandes desviaciones, debido a la renovación de la beca de las prácticas del alumno (por seis meses más), interfiriendo en el desarrollo del proyecto, y algunas dificultades que ocurrieron durante la implementación de la aplicación.

La dedicación total del proyecto ha sido de 432 horas. La fase de inicio duró 62 horas, la de elaboración 96 horas, la de construcción 265 horas y la de transición 10 horas. Se aprecia que las horas estimadas y las reales han sufrido de una desviación de 52 horas sobre la planificación inicial y de 60 para la replanificación. Esta desviación se debe, sobre todo, a la tarea de “Desarrollo Casos de uso 1/3”, ya que duró 125 horas sobre las 40 horas estimadas. Por otra parte, se pudo compensar un poco, suprimiendo la segunda iteración de la fase de construcción y la corta duración de la fase de transición, debido en gran medida a la no implantación de este proyecto en la UVa.

La fecha de finalización del proyecto conseguida ha sido el 18/04/2024, que ha sufrido una desviación de siete meses respecto de la planificación inicial y de tres meses respecto de la replanificación.

## Capítulo 8

# Conclusiones

Este capítulo está dedicado a las conclusiones obtenidas tras dar el proyecto como finalizado y para establecer unas líneas de trabajo futuras. Teniendo en cuenta los objetivos que quedarían por cumplir o mejoras que añadir.

### 8.1. Conclusiones

La idea inicial era sustituir la aplicación web de gestión de citas del Gabinete de Logopedia de la UVa por la creada en este proyecto. Durante el desarrollo del proyecto se recibió la noticia de que no podría ser implantado en la UVa. La UVa sustituyó la aplicación original por una creada con *Microsoft Bookings*, ubicada en <https://outlook.office365.com/owa/calendar/UNIDADLOGOPEDIA@uvaes.onmicrosoft.com/bookings/>.

Una vez finalizado el proyecto, se considera que ha sido completado con éxito, cumpliendo los objetivos propuestos. Pese a que finalmente no podrá ser implantado en la UVa, se considera un proyecto académicamente completo.

Las conclusiones finales en cuanto a trabajo realizado y aprendizaje adquirido son las siguientes:

- Se ha desarrollado una aplicación completamente funcional, que permite gestionar citas del Gabinete de Logopedia de la UVa.
- Se ha integrado en la aplicación web el SSO de la UVa, permitiendo iniciar sesión con los usuarios de la UVa.
- Se ha podido realizar una captación de requisitos y revisiones del proyecto con clientes reales. Lo que ha significado un aprendizaje en cuanto a tratar con personas reales que solicitan un proyecto.

- Se ha hecho un desarrollo de frontend, un campo donde el estudiante tenía poca experiencia y no estaba muy familiarizado. En la parte del backend, el alumno estaba más familiarizado, aunque se han utilizado las herramientas de Spring Boot y Thymeleaf, siendo en gran parte nuevas para el estudiante. Teniendo como resultado final un aprendizaje en estas herramientas.
- Dado que el proyecto tenía un cierto enfoque en la seguridad, se han aprendido nuevos conceptos e implementaciones para mejorar la seguridad de una aplicación web.
- La planificación del proyecto ha sido un punto débil, aunque ocurrieron algunos sucesos que no estaban en la mano del estudiante, ha ayudado a aprender cuál es el ritmo de trabajo y adaptación del estudiante, y poder tenerlo en cuenta para próximos proyectos.

## 8.2. Líneas de trabajo futuras

Existen ciertos puntos que serían muy positivos para el proyecto si se llegaran a implementar. Las líneas de trabajo futuras son las siguientes:

- Implantación del proyecto en el sistema de la UVa, fue uno de los desencadenantes del proyecto y se considera un objetivo pleno.
- Añadir *DevSecOps*, para lograr un desarrollo más ágil y tener en cuenta la seguridad.
- Añadir un servicio de email para notificar a los solicitantes y gestoras los cambios de citas u horarios.
- Añadir el bloqueo automático de fechas festivas.



# Bibliografía

- [1] Agencia española de protección de datos. Guía de privacidad desde el diseño. <https://www.aepd.es/es/documento/guia-privacidad-desde-diseno.pdf>. Accedido: 16-05-2023.
- [2] AGWA. git-crypt. <https://github.com/AGWA/git-crypt>. Accedido: 24-02-2024.
- [3] Amazon Web Services. ¿qué es el sdlc? <https://aws.amazon.com/es/what-is/sdlc/>. Accedido: 16-05-2023.
- [4] Apache Maven. Welcome to apache maven. <https://maven.apache.org/>. Accedido: 27-04-2024.
- [5] Apple. Compra un iphone 13. <https://www.apple.com/es/shop/buy-iphone/iphone-13/pantalla-de-5,4%E2%80%B3-128gb-medianoche>. Accedido: 16-05-2023.
- [6] astah. Individual licensing options. <https://astah.net/pricing/individual/>. Accedido: 16-05-2023.
- [7] Astah. Software design tool. <https://astah.net/products/astah-professional/>. Accedido: 08-07-2023.
- [8] Balsamiq. Balsamiq plans pricing. <https://balsamiq.com/buy/>. Accedido: 17-04-2024.
- [9] Balsamiq. Introduction to balsamiq wireframes for desktop. <https://balsamiq.com/wireframes/desktop/docs/intro/>. Accedido: 08-07-2023.
- [10] Bob Hughes, Mike Cotterell. Software project management, 5th edition, mcgraw hill / europe, middle east and africa.
- [11] BOE. Ley orgánica 3/2018, de 5 de diciembre, de protección de datos personales y garantía de los derechos digitales. <https://www.boe.es/boe/dias/2018/12/06/pdfs/BOE-A-2018-16673.pdf>. Accedido: 20-08-2023.
- [12] BOE. Real decreto 311/2022, de 3 de mayo, por el que se regula el esquema nacional de seguridad. <https://www.boe.es/boe/dias/2022/05/04/pdfs/BOE-A-2022-7191.pdf>. Accedido: 20-08-2023.

- [13] BOE. Reglamento (ue) 2016/679 del parlamento europeo y del consejo, de 27 de abril de 2016, relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos y por el que se deroga la directiva 95/46/ce (reglamento general de protección de datos). <https://www.boe.es/doue/2016/119/L00001-00088.pdf>. Accedido: 20-08-2023.
- [14] Carlos Martín Alonso. Aplicación web para el Área de cooperación internacional para el desarrollo de la uva. <https://uvadoc.uva.es/handle/10324/27524>. Accedido: 12-06-2023.
- [15] Docker. Build. <https://www.docker.com/#build>. Accedido: 27-04-2024.
- [16] EAR/PILAR. Entorno de análisis de riesgos. <https://www.pilar-tools.com/es/index.html>. Accedido: 08-07-2023.
- [17] Eugen Paraschiv. Spring jpa – multiple databases. <https://www.baeldung.com/spring-data-jpa-multiple-databases>. Accedido: 23-11-2023.
- [18] Expansión. ¿cómo se calculan las cotizaciones a la seguridad social? <https://www.expansion.com/economia-para-todos/impuestos/como-se-calculan-las-cotizaciones-a-la-seguridad-social.html#:~:text=El%20trabajador%20abona%20alrededor%20del,del%20sector%20y%20del%20convenio>. Accedido: 16-05-2023.
- [19] geeksforgeeks. Central authentication service. <https://www.geeksforgeeks.org/central-authentication-service/>. Accedido: 14-02-2024.
- [20] Glassdoor. Sueldos para el puesto de full stack developer en españa. [https://www.glassdoor.es/Sueldos/full-stack-developer-sueldo-SRCH\\_K00,20.htm](https://www.glassdoor.es/Sueldos/full-stack-developer-sueldo-SRCH_K00,20.htm). Accedido: 16-05-2023.
- [21] Lebuero. Tarifas de coworking en valladolid. <https://leburowork.es/precio-coworking-valladolid>. Accedido: 03-01-2024.
- [22] Microsoft. What is microsoft teams? <https://support.microsoft.com/en-us/topic/what-is-microsoft-teams-3de4d369-0167-8def-b93b-0eb5286d7a29>. Accedido: 08-07-2023.
- [23] Microsoft. What is outlook? <https://support.microsoft.com/en-au/office/what-is-outlook-10f1fa35-f33a-4cb7-838c-a7f3e6228b20>. Accedido: 08-07-2023.
- [24] Ministerio para la Transformación Digital y de la Función Pública. Magerit v.3 : Metodología de análisis y gestión de riesgos de los sistemas de información. [https://administracionelectronica.gob.es/pae\\_Home/pae\\_Documentacion/pae\\_Metodolog/pae\\_Magerit.html](https://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Magerit.html). Accedido: 15-04-2024.
- [25] opensource-coding. Dynamic calendar in html css javascript. <https://github.com/opensource-coding/Dynamic-Calendar-in-HTML-CSS-JavaScript>. Accedido: 27-11-2023.
- [26] OWASP. Cross site scripting (xss). <https://owasp.org/www-community/attacks/xss/>. Accedido: 14-02-2024.

- [27] OWASP. Owasp java encoder. <https://owasp.org/www-project-java-encoder/>. Accedido: 14-02-2024.
- [28] OWASP. Sql injection. [https://owasp.org/www-community/attacks/SQL\\_Injection](https://owasp.org/www-community/attacks/SQL_Injection). Accedido: 14-02-2024.
- [29] OWASP. Sql injection prevention cheat sheet. [https://cheatsheetseries.owasp.org/cheatsheets/SQL\\_Injection\\_Prevention\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html). Accedido: 03-02-2024.
- [30] PCcomponentes. Acer nitro 5 an515-54-52g3 intel core i5-9300h/8gb/256gb ssd/gtx 1050/15.6". <https://www.pccomponentes.com/acer-nitro-5-an515-54-52g3-intel-core-i5-9300h-8gb-256gb-ssd-gtx-1050-156>. Accedido: 16-05-2023.
- [31] Peter Eeles. Captura de requisitos arquitectonicos. <https://web.archive.org/web/2020112020231/http://www.ibm.com/developerworks/rational/library/4706.html#N100A7>. Accedido: 15-05-2023.
- [32] Rebeca Hernando Brecht. Aplicación web como adaptación de un juego educativo sobre la gestión de la deuda técnica. <https://uvadoc.uva.es/handle/10324/50091>. Accedido: 12-06-2023.
- [33] Robert C. Martin. Clean architecture: A craftsman's guide to software structure and design. pearson education.
- [34] Spring. Cas authentication. <https://docs.spring.io/spring-security/reference/servlet/authentication/cas.html#servlet-cas>. Accedido: 14-02-2024.
- [35] spring. Securing a web application. <https://spring.io/guides/gs/securing-web>. Accedido: 13-04-2024.
- [36] Spring. Spring boot. <https://spring.io/projects/spring-boot#overview>. Accedido: 21-10-2023.
- [37] The Clean Code Blog. The clean architecture. <https://blog.cleancoder.com/uncl-e-bob/2012/08/13/the-clean-architecture.html>. Accedido: 10-02-2024.
- [38] Thymeleaf. Thymeleaf. <https://www.thymeleaf.org/>. Accedido: 24-02-2024.
- [39] TIBCO. ¿qué es api gateway? <https://www.tibco.com/es/reference-center/what-is-an-api-gateway>. Accedido: 06-09-2023.
- [40] Universidad de Valladolid. Consultas médicas. <http://consultasmedicas.uva.es/index.php/app/inicio>. Accedido: 09-06-2023.
- [41] Universidad de Valladolid. Guía docente de la asignatura trabajo de fin de grado (mención ingeniería del software). [https://albergueweb1.uva.es/guia\\_docente/uploads/2022/545/46976/1/Documento.pdf](https://albergueweb1.uva.es/guia_docente/uploads/2022/545/46976/1/Documento.pdf). Accedido: 16-05-2023.
- [42] Universidad de Valladolid. Logopedia para PDI, PAS y estudiantes. <https://med.uva.es/logopedia-para-pdi-pas-y-estudiantes/>. Accedido: 14-04-2023.

- [43] Universidad de Valladolid. Sede electrónica de la universidad de valladolid. <https://portal.sede.uva.es/>. Accedido: 27-05-2023.
- [44] Universidad de Valladolid. Trabajos fin de grado, calendario de depósito y defensa, primera y segunda convocatoria curso 2022-2023. [https://www.inf.uva.es/wp-content/uploads/2023/03/Cal\\_TFG\\_22-23.pdf](https://www.inf.uva.es/wp-content/uploads/2023/03/Cal_TFG_22-23.pdf). Accedido: 16-05-2023.
- [45] Wikipedia. Css. <https://es.wikipedia.org/wiki/CSS>. Accedido: 21-10-2023.
- [46] Wikipedia. Git. <https://es.wikipedia.org/wiki/Git>. Accedido: 21-10-2023.
- [47] Wikipedia. Gitlab. <https://es.wikipedia.org/wiki/GitLab>. Accedido: 21-10-2023.
- [48] Wikipedia. Html. <https://es.wikipedia.org/wiki/HTML>. Accedido: 21-10-2023.
- [49] Wikipedia. Java. [https://es.wikipedia.org/wiki/Java\\_\(lenguaje\\_de\\_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)). Accedido: 21-10-2023.
- [50] Wikipedia. Javascript. <https://es.wikipedia.org/wiki/JavaScript>. Accedido: 21-10-2023.
- [51] Wikipedia. Mysql. <https://es.wikipedia.org/wiki/MySQL>. Accedido: 21-10-2023.
- [52] Wikipedia. Nginx. <https://es.wikipedia.org/wiki/Nginx>. Accedido: 21-10-2023.
- [53] Wikipedia. Overleaf. <https://en.wikipedia.org/wiki/Overleaf>. Accedido: 08-07-2023.
- [54] Wikipedia. Principio de abierto/cerrado. [https://es.wikipedia.org/wiki/Principio\\_de\\_abierto/cerrado](https://es.wikipedia.org/wiki/Principio_de_abierto/cerrado). Accedido: 10-02-2024.
- [55] Wikipedia. Principio de inversión de la dependencia. [https://es.wikipedia.org/wiki/Principio\\_de\\_inversi%C3%B3n\\_de\\_la\\_dependencia](https://es.wikipedia.org/wiki/Principio_de_inversi%C3%B3n_de_la_dependencia). Accedido: 10-02-2024.
- [56] Wikipedia. Principio de responsabilidad única. [https://es.wikipedia.org/wiki/Principio\\_de\\_responsabilidad\\_%C3%BAnica](https://es.wikipedia.org/wiki/Principio_de_responsabilidad_%C3%BAnica). Accedido: 10-02-2024.
- [57] Wikipedia. Principio de segregación de la interfaz. [https://es.wikipedia.org/wiki/Principio\\_de\\_segregaci%C3%B3n\\_de\\_la\\_interfaz](https://es.wikipedia.org/wiki/Principio_de_segregaci%C3%B3n_de_la_interfaz). Accedido: 10-02-2024.
- [58] Wikipedia. Principio de sustitución de liskov. [https://es.wikipedia.org/wiki/Principio\\_de\\_sustituci%C3%B3n\\_de\\_Liskov](https://es.wikipedia.org/wiki/Principio_de_sustituci%C3%B3n_de_Liskov). Accedido: 10-02-2024.
- [59] Wikipedia. Upedu. [https://es.wikipedia.org/wiki/Proceso\\_Unificado\\_de\\_Racional](https://es.wikipedia.org/wiki/Proceso_Unificado_de_Racional). Accedido: 12-06-2023.
- [60] Wikipedia. Visual studio code. [https://es.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://es.wikipedia.org/wiki/Visual_Studio_Code). Accedido: 21-10-2023.

## Apéndice A

# Manuales

### A.1. Manual de despliegue e instalación

En esta sección se detallan los requisitos necesarios para desplegar el proyecto y que pueda ser utilizado.

#### A.1.1. Requisitos de despliegue

Para poder desplegar la aplicación se necesita tener instalado:

- Docker.
- Java 11.
- git-crypt.

El proyecto se despliega en Docker, por lo que la instalación de programas es mínima. Se requiere de una construcción del proyecto con Maven, pero dado que el repositorio incluye el archivo *mvnw*, no será necesario instalarlo.

El proyecto contiene archivos cifrados con la herramienta *git-crypt* que requieren ser descifrados para poder desplegar el proyecto. Situado en el directorio raíz del proyecto, se ejecuta el comando *git-crypt unlock /dirección/llave*. Para obtener la llave póngase en contacto con el autor del proyecto.

El proyecto utiliza un SSO con un servidor CAS, por lo tanto, para que el proyecto sea completamente funcional se deben de configurar una serie de parámetros en el archivo *application.properties*, ubicado en *src/main/resources*:

1. *cas.service.url*: Dirección a la que el servidor CAS redirigirá una vez haya autenticado al usuario.
2. *cas.entrypoint*: Dirección del servidor CAS para la gestión de los inicios de sesión.
3. *cas.ticket.validator*: Dirección del servidor CAS para la validación de tickets.
4. *cas.logoutFilter.processUrl*: Ruta de entrada que redirigirá las peticiones de desconexión de sesión hacia el servidor CAS.
5. *cas.logoutFilter.success.url*: Dirección del servidor CAS que manejará la desconexión de sesión.
6. *cas.singleSignOutFilter.callbackPath*: Ruta que interceptará la respuesta del servidor CAS en la desconexión de sesión.

### A.1.2. Construcción y despliegue

El proyecto posee un *Makefile* que facilita la construcción y despliegue del proyecto.

Situado en el directorio raíz del proyecto, se debe de ejecutar *make build app-up*, el sistema construirá el *.jar* de java y se desplegará en la dirección *localhost:8080*. Cabe destacar que en algunos sistemas es necesario añadir el comando *sudo* previo al *make*.

## A.2. Manual de mantenimiento

En cuanto al backend, el código del proyecto está organizado por casos de uso, por lo que la integración de uno nuevo requiere de añadir el controlador en *src/main/java/edu/uva/-consultasmedicas/interfaceAdapters/controller*, los *boundaries* en *src/main/java/edu/uva/-consultasmedicas/useCases/boundaries*, los modelos de solicitud y respuesta de datos en *src/main/java/edu/uva/consultasmedicas/useCases/models*, la implementación de la lógica del caso de uso en *src/main/java/edu/uva/consultasmedicas/useCases/useCasesImplementation* y la implementación del *presenter* en *src/main/java/edu/uva/consultasmedicas/interfaceAdapters/presenterImplementation*. En el frontend, se deben de añadir los HTML, CSS y JavaScript necesarios en *src/main/resources*.

En el caso de querer modificar un caso de uso ya existente, remítase a los directorios anteriormente mencionados y modifique los archivos con el nombre del caso de uso.

## A.3. Manual de usuario

Dado que el proyecto tiene diferentes casos de uso y roles, en esta sección solo se mostrará el manual de usuario para iniciar sesión y pedir una cita como solicitante de tipo PDI/PAS.

Para el resto de casos de uso, tome como referencia los bocetos del proyecto (Sección 5.1), ya que en ellos se explican los flujos, y los bocetos son fieles al resultado de la aplicación final.

Desde la dirección <http://157.88.125.198:8080/consultasmedicas/> se ve la vista inicial sin un usuario autenticado (Figura A.1), se presiona en el botón Iniciar sesión.



---

Figura A.1: Vista de inicio sin usuario autenticado.

Se muestra el inicio de sesión de la UVa a partir del SSO (A.2), donde se introducen las credenciales del usuario UVa y se presiona en *aceptar*.

### A.3. MANUAL DE USUARIO

---



Figura A.2: Vista de inicio de sesión.

La aplicación redirige a la vista de inicio acorde al rol del usuario autenticado (Figura A.3), en este ejemplo *PDI/PAS*. Se presiona en el botón *Pedir Cita*.

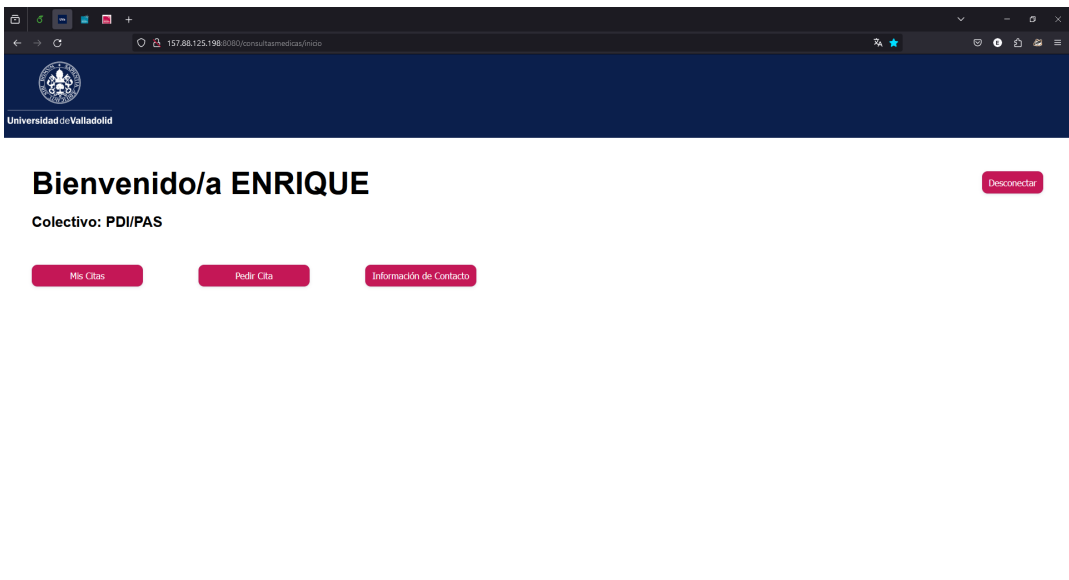
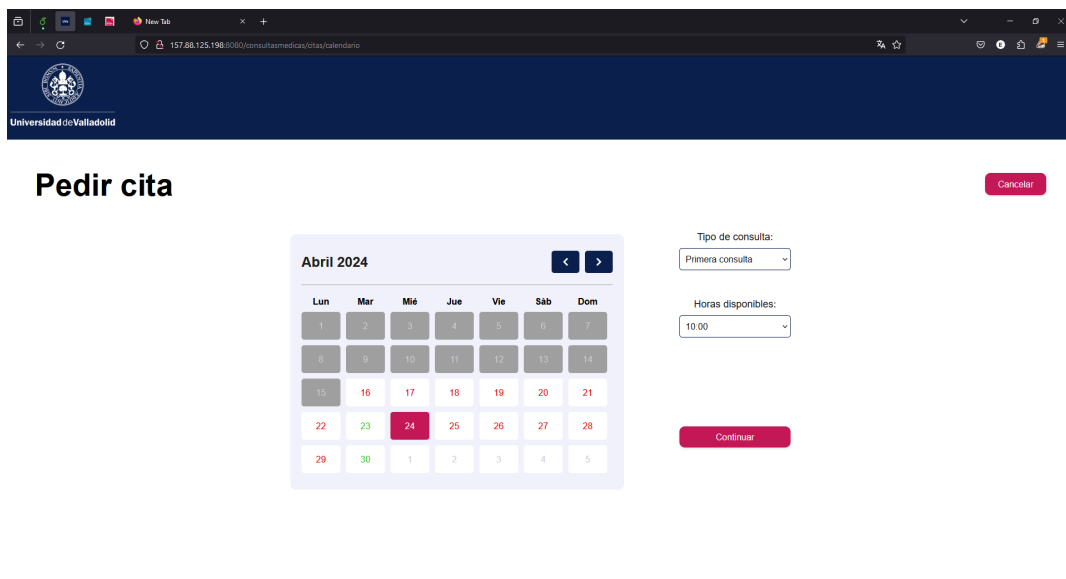


Figura A.3: Vista de inicio con usuario autenticado.

Se muestra una vista con un calendario y dos desplegables (Figura A.4). Se presiona en



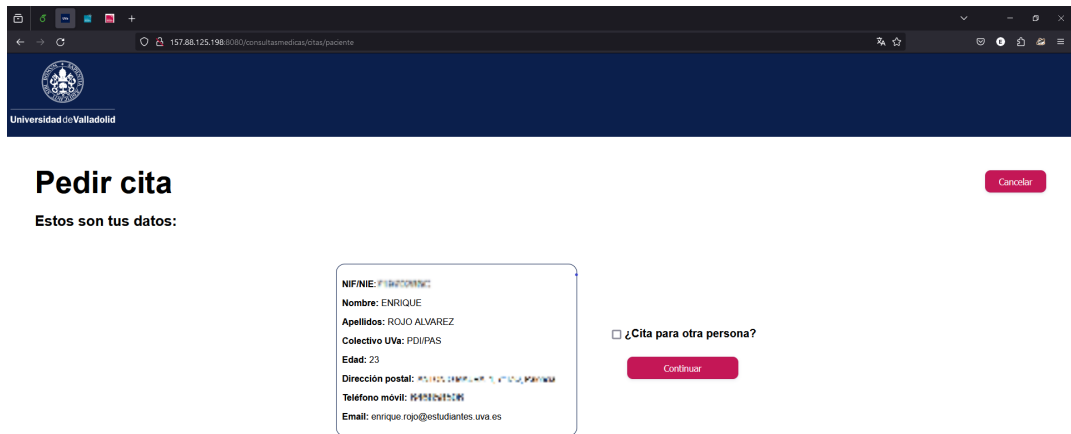
el día deseado, se selecciona el tipo de consulta y la hora disponible en la que se celebrará la cita. Se presiona en el botón de *Continuar*.



---

Figura A.4: Vista de calendario al solicitar cita.

Se muestra la información del usuario y una casilla marcap que permite solicitar la cita para otra persona (Figura A.5); en este ejemplo se presiona sobre la casilla.



---

Figura A.5: Vista de información del usuario al solicitar cita.

Con la casilla marcada, se muestra una tarjeta con los pacientes que se han añadido previamente y un formulario para añadir uno nuevo (Figura A.6). Bajo la tarjeta de pacientes, hay tres botones, *Anterior* y *Siguiente* permiten desplazarse entre los pacientes, y *Seleccionar* rellena automáticamente el formulario con los datos de ese paciente, permitiendo la modificación de alguno de los datos. En este caso se rellena el formulario con la información de un paciente nuevo (Figura A.7). Se presiona sobre el botón de *Continuar*.

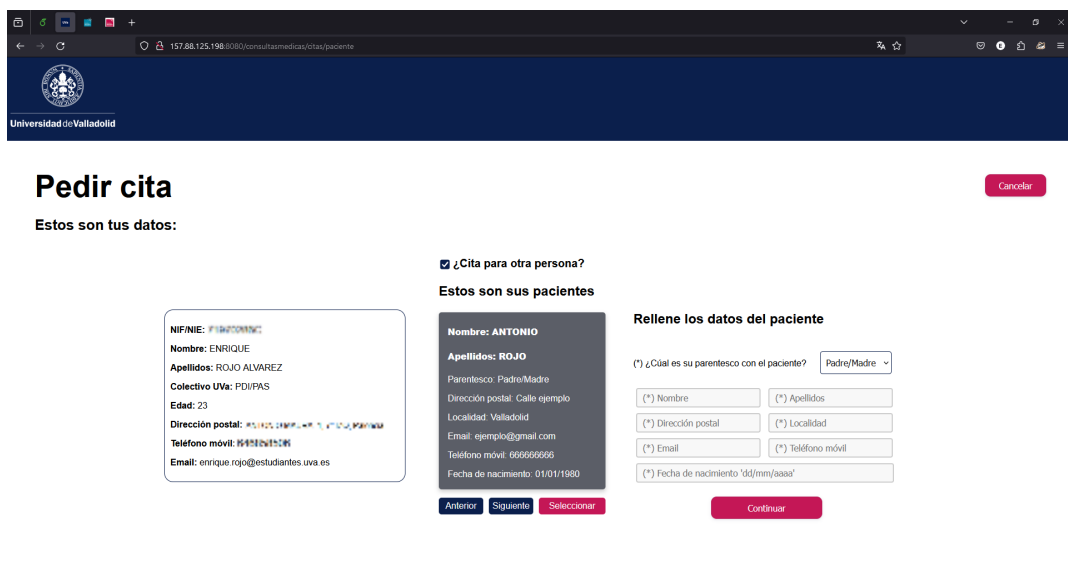


Figura A.6: Vista de pacientes al solicitar cita.

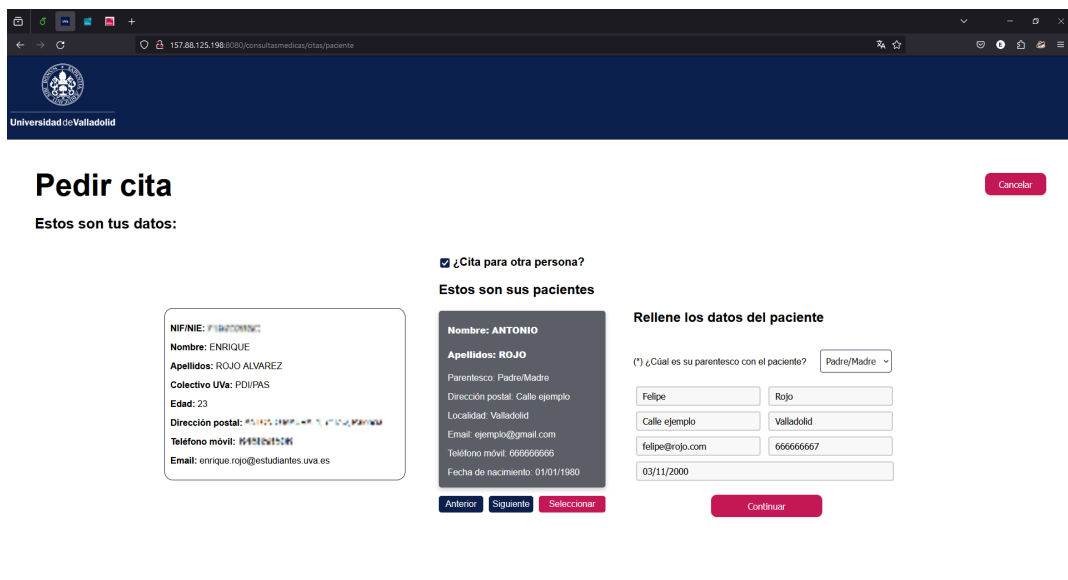


Figura A.7: Vista de nuevo paciente al solicitar cita.

La aplicación muestra una vista para dar información sobre la cita, con un desplegable sobre el motivo de la consulta y una caja de texto para añadir información que puede ser relevante para la cita (Figura A.8). Se introducen los datos y se presiona en *Finalizar*.

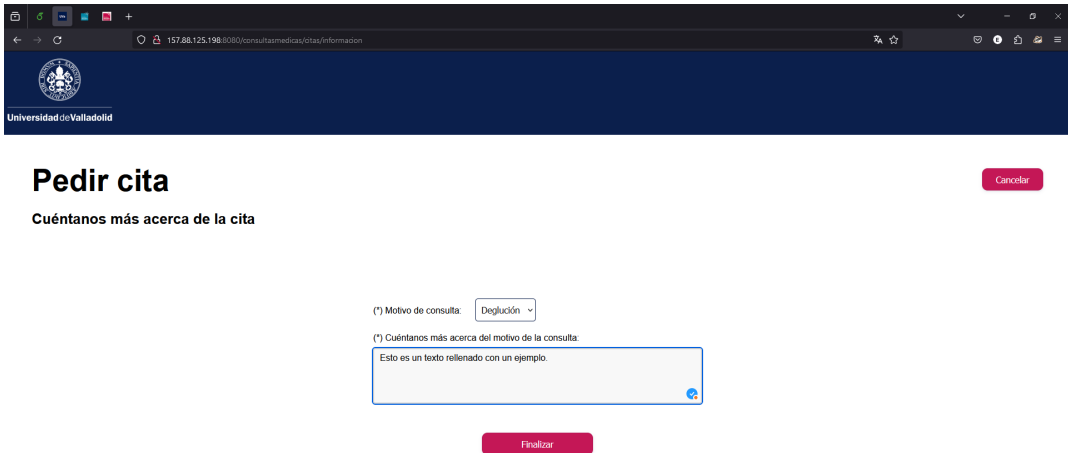


Figura A.8: Vista de información al solicitar cita.

Finalmente, se muestra de nuevo la vista del inicio, pero con un mensaje indicando que se ha creado la cita con éxito e indicando la simulación de una notificación a las gestoras vía e-mail (Figura A.9) (envío real de e-mail para líneas futuras).

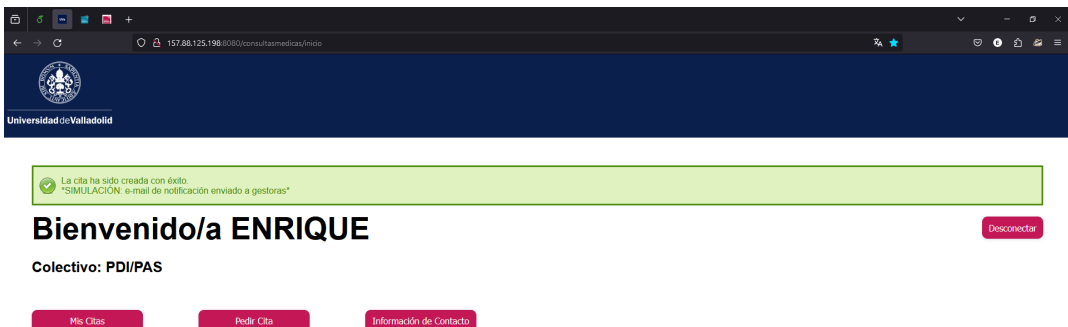


Figura A.9: Vista de éxito al solicitar cita.

Cabe destacar que durante todo el proceso de la solicitud de cita, se puede presionar sobre el botón de *Cancelar* para abortar la solicitud.



## Apéndice B

# Resumen de enlaces adicionales

Los enlaces útiles de interés en este Trabajo Fin de Grado son:

- Repositorio del código: <https://gitlab.inf.uva.es/enrojo/tfg-consultasmedicas>.
- Dirección donde el proyecto está desplegado para la entrega: <http://157.88.125.198:8080/consultasmedicas>

La máquina virtual de despliegue no se garantiza que permanezca operativa después de la presentación del Trabajo de Fin de Grado.