



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA
Mención en Ingeniería del Software

**Tutorías InfUVa: una web y bot de Telegram
para facilitar la gestión de tutorías para
alumnos y profesores de la Escuela de
Ingeniería Informática**

Alumno: Richard Albán Fernández

Tutor: Yania Crespo González-Carvajal

A mi familia, que ha estado ahí en todo momento

Agradecimientos

A mi familia, que siempre ha intentado animarme en los momentos más duros.

A mis mascotas, en especial a mi perra Kira, que si no fuese por ella no se que hubiera sido de mi.

A mis compañeros, que han hecho que este viaje sea más llevadero.

A mi tutora Yania, que gracias a su ayuda y guía he podido llevar a cabo un trabajo satisfactorio, además de ser todo un ejemplo en cuanto a trabajo y dedicación.

Gracias a todos.

Resumen

El objetivo de este trabajo de final de grado es desarrollar una aplicación web que permita facilitar tanto a profesores como a alumnos de la Escuela de Ingeniería Informática de la Universidad de Valladolid, la gestión de tutorías y, además, servir de directorio para que los alumnos puedan encontrar fácilmente la información que necesiten de un profesor.

Adicionalmente, esta aplicación web se complementa con un bot de Telegram, que es el sistema de mensajería más utilizado entre alumnos de la escuela, y se encargará de facilitar la información de los profesores y/o acceso a la web para solicitar tutorías.

El proyecto se ha desarrollado a partir del framework Angular, HTML5, SCSS y el lenguaje TypeScript para la parte frontend y el framework Spring con el lenguaje Java para la parte backend, siguiendo el marco de trabajo ágil Scrum.

Abstract

The objective of this end-of-degree project is to develop a web application that allows both teachers and students of the School of Computer Engineering of the University of Valladolid to facilitate the management of tutorials and, in addition, to serve as a directory so that students can easily find the information they need from a teacher. Additionally, this web application is complemented by a Telegram bot, which is the most widely used messaging system among students at the school, and will be responsible for providing information from teachers and/or access to the web to request tutorials.

The project has been developed from the Angular framework, HTML5, SCSS and the TypeScript language for the frontend part and the Spring framework with the Java language for the backend part, following the Scrum agile framework.

Índice general

Agradecimientos	III
Resumen	V
Abstract	VII
Lista de figuras	XV
Lista de tablas	XXI
1. Introducción	1
1.1. Contexto	1
1.2. Motivación	1
1.3. Objetivos	2
1.3.1. Objetivos del proyecto	2
1.3.2. Objetivos personales	2
1.4. Estructura de la memoria	2
2. Requisitos y planificación	5
2.1. Scrum	5
2.1.1. ¿Qué es Scrum?	5
2.1.2. Scrum Team	6

IX

2.1.3. Artefactos Scrum	7
2.1.4. Eventos Scrum	7
2.2. Roles y responsabilidades en el proyecto	8
2.3. Planificación inicial	9
2.4. Plan de riesgos	10
2.4.1. Introducción	10
2.4.2. Gestión inicial de riesgos	11
2.5. Estimación de costes	17
2.5.1. Estimación de costes simulado	17
2.5.2. Estimación de costes real	18
2.6. Product Backlog inicial	18
2.7. Product Backlog final	19
3. Tecnologías utilizadas	21
3.1. Herramientas para la comunicación	21
3.1.1. Rocket.Chat	21
3.1.2. Webex	21
3.2. Herramientas para el análisis, diseño y documentación	22
3.2.1. Astah	22
3.2.2. Balsamiq Cloud	22
3.2.3. Overleaf	22
3.3. Herramientas para la gestión y control del proyecto	22
3.3.1. Git	22
3.3.2. GitLab	23
3.4. Tecnologías para el desarrollo del proyecto	24
3.4.1. Visual Studio Code	24
3.4.2. Eclipse IDE	25

3.4.3. Node.js	25
3.4.4. NPM	26
3.4.5. Angular	26
3.4.6. PrimeNG	26
3.4.7. Nginx	26
3.4.8. Spring Framework	27
3.4.9. Apache Maven	27
3.4.10. Apache Tomcat	27
3.4.11. PostgreSQL	27
3.4.12. H2 Database	27
3.4.13. Telegram	28
3.4.14. Docker	28
4. Análisis	29
4.1. Requisitos funcionales como historias de usuario	29
4.2. Modelo de dominio inicial	33
4.3. Modelo de dominio final	33
4.4. Modelo de proceso de negocio	34
4.5. Modelado dinámico	37
5. Diseño	39
5.1. Diseño de la interfaz de usuario	39
5.2. Diseño arquitectónico	59
5.2.1. Arquitectura en el Cliente (MVVM)	59
5.2.2. Arquitectura en el Servidor	60
5.2.3. Arquitectura en el Bot de Telegram	61
5.3. Patrones de diseño empleados	63

5.3.1. Patrón DTO	63
5.3.2. Patrón DAO	63
5.3.3. Patrón Comando	63
5.3.4. Patrón Thread pool	64
5.4. Diseño de la API REST	65
5.5. Diseño basado en componentes en el frontend de la aplicación	67
5.5.1. Componentes creados en el desarrollo del proyecto	67
5.6. Diseño del bot de Telegram	71
5.7. Diseño de la comunicación entre objetos	72
5.8. Diseño del modelo de datos	72
5.9. Despliegue de la aplicación web y del bot de Telegram	73
6. Implementación y pruebas	79
6.1. Implementación	79
6.1.1. Organización del código	79
6.1.2. Dockerización del proyecto	83
6.1.3. Principales dificultades y retos	85
6.1.4. Licencia	86
6.2. Plan de pruebas y evaluación	87
6.2.1. Casos de prueba	88
6.2.2. Resultados de las pruebas	128
7. Seguimiento del proyecto	133
7.1. Seguimiento de los sprints realizados	133
7.1.1. Sprint 0 (08/02/2022 - 23/02/2022)	133
7.1.2. Sprint 1 (23/02/2022 - 09/03/2022)	133
7.1.3. Sprint 2 (09/03/2022 - 23/03/2022)	134

7.1.4. Sprint 3 (23/03/2022 - 06/04/2022)	135
7.1.5. Sprint 4 (06/04/2022 - 27/04/2022)	137
7.1.6. Sprint 5 (27/04/2022 - 11/05/2022)	139
7.1.7. Sprint 6 (11/05/2022 - 25/05/2022)	140
7.1.8. Sprint de refuerzo 1 (25/05/2022 - 08/06/2022)	142
7.1.9. Sprint de refuerzo 2 (08/06/2022 - 29/06/2022)	144
7.1.10. Momentos finales del proyecto (29/06/2022 - 14/07/2022)	148
7.2. Resumen del desarrollo del proyecto (08/02/2022 - 14/07/2022)	149
7.2.1. Tiempo estimado y empleado	149
7.2.2. Costes finales	149
8. Conclusiones y trabajo futuro	151
8.1. Conclusiones	151
8.2. Líneas de trabajo futuras	151
Bibliografía	153
A. Manuales	157
A.1. Manual de despliegue local	157
A.2. Manual de usuario	158
A.2.1. Manual de usuario de la aplicación web	158
A.2.2. Manual de usuario del bot de Telegram	164
B. Resumen de enlaces adicionales	177

Lista de Figuras

2.1. Funcionamiento de Scrum. Tomado de [45]	6
2.2. Marco de trabajo de riesgos de Lyytinen-Mathiassen-Ropponen. Tomado de [27]	11
2.3. Épicas y su subdivisión en historias de usuario. Tomado de [32]	19
3.1. Git - Ramificación. Tomado de [20]	23
3.2. GitLab Issue Board (parte 1)	24
3.3. GitLab Issue Board (parte 2)	25
3.4. Aplicaciones “dockerizadas”	28
4.1. Modelo de dominio inicial	33
4.2. Modelo de dominio final	34
4.3. Modelo de proceso de negocio - Solicitar tutoría	36
4.4. Máquina de estados - Solicitud Tutoría	37
5.1. Vista de Inicio de sesión - Usuario no identificado	40
5.2. Vista de Inicio - Profesor	40
5.3. Vista de Inicio - Alumno	41
5.4. Vista de Buscar profesor por nombre - Alumno	41
5.5. Vista de Buscar profesor por asignatura - Alumno	42
5.6. Vista de Información profesor: visualización de asignaturas - Alumno	42
5.7. Vista de Información profesor: visualización de tutorías programadas - Alumno	43

5.8. Vista de Solicitar tutoría - Alumno	43
5.9. Vista de Solicitar tutoría: ver horario de tutorías del profesor seleccionado - Alumno	44
5.10. Vista de Consultar solicitudes - Alumno	44
5.11. Vista de Solicitud de tutoría: en estado pendiente - Alumno	45
5.12. Vista de Solicitud de tutoría: en estado aceptada - Alumno	45
5.13. Vista de Solicitud de tutoría: en estado confirmada - Alumno	46
5.14. Vista de Solicitud de tutoría: en estado rechazada - Alumno	46
5.15. Vista de Consultar solicitudes - Profesor	47
5.16. Vista de Solicitud de tutoría: en estado pendiente - Profesor	47
5.17. Vista de Solicitud de tutoría: aceptar solicitud - Profesor	48
5.18. Vista de Solicitud de tutoría: aceptar solicitud con propuesta de nuevo horario - Profesor	48
5.19. Vista de Solicitud de tutoría: rechazar solicitud - Profesor	49
5.20. Vista de Solicitud de tutoría: en estado aceptada o confirmada - Profesor	49
5.21. Vista de Calendario tutorías - Usuario	50
5.22. Vista de Calendario tutorías: ver detalle tutoría - Usuario	50
5.23. Vista de Calendario tutorías: ver detalle solicitud aceptada (pendiente de confirmar) - Profesor	51
5.24. Vista de Consultar tutorías: mediante filtros - Alumno	51
5.25. Vista de Consultar tutorías: próximamente - Alumno	52
5.26. Vista de Consultar tutorías: hoy - Alumno	52
5.27. Vista de Consultar tutorías: mediante filtros - Profesor	53
5.28. Vista de Consultar tutorías: próximamente - Profesor	53
5.29. Vista de Consultar tutorías: hoy - Profesor	54
5.30. Vista de Tutoría - Alumno	54
5.31. Vista de Tutoría: cancelar tutoría - Usuario	55
5.32. Vista de Tutoría (cancelada) - Alumno	55

5.33. Vista de Tutoría - Profesor	56
5.34. Vista de Tutoría (cancelada) - Profesor	56
5.35. Vista de Perfil - Alumno	57
5.36. Vista de Perfil - Profesor	57
5.37. Vista de Perfil: cargar nuevo horario - Profesor	58
5.38. Funcionamiento del patrón arquitectónico MVVM. Tomado de [10]	59
5.39. Arquitectura general del frontend	60
5.40. Arquitectura general de la API	61
5.41. Arquitectura general del bot de Telegram	62
5.42. Diseño detallado: DAOs API REST	64
5.43. Diseño detallado: DAOs del bot de Telegram	64
5.44. Patrón Comando: Estructura	65
5.45. Patrón Comando en el bot de Telegram: Diseño de las relaciones entre clases	65
5.46. Patrón Comando en el bot de Telegram: Diseño detallado de las clases	66
5.47. Ejemplo de un Thread pool. Tomado de [64]	67
5.48. Diseño API REST parte 1: Solicitudes y Tutorías	68
5.49. Diseño API REST parte 2: resto de recursos	68
5.50. Componente en Angular. Tomado de [7]	69
5.51. Diagrama de Componentes	71
5.52. Secuencia principal - HU05: Solicitar tutoría	72
5.53. Subdiagrama <i>Interaction Use</i> “Hacer solicitud tutoría” - HU05: Solicitar tutoría	73
5.54. Subdiagrama <i>Interaction Use</i> “Procesar petición solicitar tutoría en api” - HU05: Solicitar tutoría	74
5.55. Subdiagrama <i>Interaction Use</i> “Nueva solicitud tutoría” parte 1 - HU05: Soli- citar tutoría	75
5.56. Subdiagrama <i>Interaction Use</i> “Nueva solicitud tutoría” parte 2 - HU05: Soli- citar tutoría	76
5.57. Modelo de Datos: modelo relacional	77

5.58. Diagrama de despliegue	77
6.1. Estructura del repositorio en GitLab	80
6.2. Estructura del código en el frontend	81
6.3. Estructura del código de la API REST	83
6.4. Estructura del código en el bot de Telegram	84
6.5. Dockerfile del frontend	84
6.6. Dockerfile de la API REST	85
6.7. Dockerfile del bot de Telegram	85
A.1. Pantalla de Inicio de sesión de la web Tutorías InfUVa	159
A.2. Pantalla de Inicio de la web Tutorías InfUVa - Alumno	160
A.3. Pantalla de Perfil de la web Tutorías InfUVa - Alumno	161
A.4. Pantalla de Buscar profesor: búsqueda por profesor, de la web Tutorías InfUVa - Alumno	162
A.5. Pantalla de Buscar profesor: búsqueda por asignatura, de la web Tutorías InfUVa - Alumno	163
A.6. Pantalla de Información profesor, de la web Tutorías InfUVa - Alumno	164
A.7. Pantalla de Solicitar tutoría de la web Tutorías InfUVa - Alumno	165
A.8. Pantalla de Solicitar tutoría: horarios, de la web Tutorías InfUVa - Alumno	166
A.9. Pantalla de Solicitud tutoría: pendiente, de la web Tutorías InfUVa - Alumno	166
A.10. Pantalla de Solicitud tutoría: confirmada, de la web Tutorías InfUVa - Alumno	167
A.11. Pantalla de Consultar solicitudes de la web Tutorías InfUVa - Alumno	167
A.12. Pantalla de Calendario tutorías: detalle tutoría, de la web Tutorías InfUVa - Alumno	168
A.13. Pantalla de Consultar tutorías de la web Tutorías InfUVa - Alumno	169
A.14. Pantalla de Tutoría de la web Tutorías InfUVa - Alumno	169
A.15. Pantalla de Inicio de la web Tutorías InfUVa - Profesor	170
A.16. Pantalla de Consultar solicitudes de la web Tutorías InfUVa - Profesor	170

A.17.Pantalla de Solicitud tutoría: aceptar solicitud, de la web Tutorías InfUVa - Profesor	171
A.18.Pantalla de Calendario tutorías: detalle solicitud pendiente de confirmación, de la web Tutorías InfUVa - Profesor	171
A.19.Pantalla de Consultar tutorías de la web Tutorías InfUVa - Profesor	172
A.20.Pantalla de Tutoría de la web Tutorías InfUVa - Profesor	172
A.21.Pantalla de Perfil de la web Tutorías InfUVa - Profesor	173
A.22.Pantalla de Perfil: cargar horario tutorías, de la web Tutorías InfUVa - Profesor	173
A.23.Inicio	174
A.24. <i>Iniciar bot</i>	174
A.25.Comandos	174
A.26.Iniciar sesión en bot	174
A.27.Buscar profesor - tipo búsqueda	174
A.28.Buscar profesor - titulaciones	174
A.29.Buscar profesor - cursos	175
A.30.Buscar profesos - menciones	175
A.31.Buscar profesor - asignaturas	175
A.32.Buscar profesor - profesores	175
A.33.Buscar profesor - detalle profesor	175
A.34.Buscar profesor - por nombre	175
A.35.Acceso a la web	176
A.36.Terminar sesión	176

Lista de Tablas

2.1. Planificación inicial	10
2.2. Matriz de riesgos	12
2.3. Riesgo R01: Falta de formación del equipo de desarrollo	12
2.4. Riesgo R02: Planificación de la duración del proyecto poco realista	13
2.5. Riesgo R03: Retraso en el desarrollo de las tareas	13
2.6. Riesgo R04: Gold plating	14
2.7. Riesgo R05: Baja médica por enfermedad u otro problema del equipo de desarrollo	14
2.8. Riesgo R06: Ausencia por consulta médica u otra actividad	15
2.9. Riesgo R07: Mala comunicación con el tutor	15
2.10. Riesgo R08: Cambios en los requisitos	16
2.11. Estimación de costes simulado	18
2.12. Estimación de costes real	18
2.13. Product Backlog inicial	19
2.14. Product Backlog final ordenado por prioridad	20
4.1. Historias de usuario de la épica EP01	29
4.2. Historias de usuario de la épica EP02	30
4.3. Historias de usuario de la épica EP03	30
4.4. Historias de usuario de la épica EP04	31

4.5. Historias de usuario de la épica EP05	31
4.6. Historias de usuario de la épica EP06	31
4.7. Historias de usuario de la épica EP07	32
6.1. CP-01: Buscar profesor por nombre vacío	88
6.2. CP-02: Buscar profesor por cadena de texto coincidente con nombre o apellidos	88
6.3. CP-03: Buscar profesor por nombre y apellidos exactos	88
6.4. CP-04: Buscar profesor por cadena de texto no coincidente	89
6.5. CP-05: Buscar profesor por asignatura a partir de la titulación	89
6.6. CP-06: Buscar profesor por asignatura a partir de la titulación y cadena de texto en el nombre de la asignatura	89
6.7. CP-07: Buscar profesor por asignatura a partir de la titulación y curso académico	90
6.8. CP-08: Buscar profesor por asignatura a partir de la titulación, curso acadé- mico y cadena de texto en el nombre de la asignatura	90
6.9. CP-09: Buscar profesor por asignatura a partir de la titulación, curso acadé- mico y mención	91
6.10. CP-10: Buscar profesor por asignatura a partir de la titulación, curso acadé- mico, mención y cadena de texto en el nombre de la asignatura	91
6.11. CP-11: Buscar profesor por asignatura a partir de una cadena de texto en el nombre de la asignatura no coincidente	92
6.12. CP-12: Se muestra la información del profesor cuyo horario de tutorías está dividido por cuatrimestres	93
6.13. CP-13: Buscar profesor por asignatura a partir de una cadena de texto en el nombre de la asignatura no coincidente	93
6.14. CP-14: Solicitar tutoría dentro de horario de tutorías del profesor	94
6.15. CP-15: Solicitar tutoría fuera del horario de tutorías del profesor	95
6.16. CP-16: Solicitar tutoría para un intervalo de tiempo en el que ya hay tutorías establecidas para un profesor	96
6.17. CP-17: Solicitar tutoría para un intervalo de tiempo en el que ya hay tutorías establecidas para el alumno solicitante	97
6.18. CP-18: Solicitar tutoría con algún campo obligatorio sin rellenar	98

6.19. CP-19: Ver horario de tutorías del profesor al que se le quiere solicitar la tutoría	98
6.20. CP-20: Ver horario de tutorías del profesor al que se le quiere solicitar la tutoría	98
6.21. CP-21: Solicitar tutoría sin seleccionar asignatura hace que el campo “Comentario” sea obligatorio	99
6.22. CP-22: Solicitar tutoría en un espacio de tiempo menor a un día	100
6.23. CP-23: Solicitar tutoría un viernes, sábado o domingo para el lunes siguiente	101
6.24. CP-24: Solicitar tutoría en una fecha e intervalo de horas que se solape con otras solicitudes pendientes que hayamos realizado	102
6.25. CP-25: Cancelar solicitud en estado pendiente o aceptada desde el listado de solicitudes	103
6.26. CP-26: Cancelar solicitud en estado pendiente o aceptada desde la vista en detalle de la solicitud	103
6.27. CP-27: Iniciar sesión como alumno	104
6.28. CP-28: Iniciar sesión como profesor	104
6.29. CP-29: Iniciar sesión con usuario no existente	104
6.30. CP-30: Iniciar sesión con contraseña incorrecta	105
6.31. CP-31: Iniciar sesión sin introducir credenciales	105
6.32. CP-32: Acceder a funcionalidades sin haber iniciado sesión	105
6.33. CP-33: Cerrar sesión	106
6.34. CP-34: Aceptar solicitud pendiente	106
6.35. CP-35: Aceptar solicitud pendiente para horario en el que el profesor ya tiene tutorías o solicitudes pendientes de confirmación	107
6.36. CP-36: Aceptar solicitud pendiente proponiendo horario en el que existen solicitudes por confirmar o tutorías	108
6.37. CP-37: Rechazar solicitud pendiente	109
6.38. CP-38: Rechazar solicitud pendiente sin indicar motivo de rechazo	109
6.39. CP-39: Rechazar solicitud pendiente indicando un motivo de rechazo con menos de 10 caracteres	109
6.40. CP-40: Confirmar solicitud de tutoría	110
6.41. CP-41: Consultar solicitudes mediante filtros (Profesor)	110

6.42. CP-42: Consultar solicitudes mediante filtros (Alumno)	111
6.43. CP-43: Ver detalle de tutoría	111
6.44. CP-44: Ver detalle de solicitud tutoría aceptada	111
6.45. CP-45: Ver más información de una tutoría mediante el calendario de tutorías	112
6.46. CP-46: Ver más información de una solicitud pendiente de confirmación me- diante el calendario de tutorías	112
6.47. CP-47: Consultar próximas tutorías	113
6.48. CP-48: Consultar tutorías que tienen lugar en el día de hoy	113
6.49. CP-49: Consultar tutorías mediante filtros (Alumno)	113
6.50. CP-50: Consultar tutorías mediante filtros (Profesor)	114
6.51. CP-51: Consultar tutoría	114
6.52. CP-52: Cancelar tutoría	115
6.53. CP-53: Consultar perfil	115
6.54. CP-54: Consultar solicitudes pendientes	116
6.55. CP-55: Consultar solicitudes pendientes	116
6.56. CP-56: Consultar próximas tutorías	116
6.57. CP-57: Rechazar automáticamente las solicitudes pendientes caducadas . . .	117
6.58. CP-58: Cargar un fichero que cumple con el formato adecuado	118
6.59. CP-59: Cargar un fichero csv sin contenido	119
6.60. CP-60: Cargar un fichero csv con una cabecera incorrecta	119
6.61. CP-61: Cargar un fichero csv con un tipo de horario anual y cuatrimestral . .	120
6.62. CP-62: Cargar un fichero csv con alguna hora de inicio mayor que la hora de fin	120
6.63. CP-63: Cargar un fichero csv con alguna hora de inicio mayor que la hora de fin	121
6.64. CP-64: Cargar un fichero csv con cadena que no representa un día en la co- lumna 'dia'	121
6.65. CP-65: Consultar información de un profesor sin identificarse	122
6.66. CP-66: Iniciar sesión en el bot de Telegram con credenciales válidas	122

6.67. CP-67: Acceder a la información de un profesor mediante un tipo de búsqueda general	123
6.68. CP-68: Acceder a la información de un profesor buscando a partir de mis asignaturas	124
6.69. CP-69: Acceder a la información de un profesor buscando a partir del nombre de una asignatura	125
6.70. CP-70: Acceder a la información de un profesor buscando a partir del nombre de una asignatura	125
6.71. CP-71: Buscar asignatura mediante el tipo de búsqueda por nombre de asignatura e introducir una cadena de texto que no coincida con ninguna asignatura	126
6.72. CP-72: Terminar la sesión con el bot	126
6.73. CP-73: Intentar buscar un profesor cuando la API de TutoriasInfUVa no está disponible	127
6.74. CP-74: Acceder a la web mediante el bot de Telegram sin haberse identificado previamente	127
6.75. CP-75: Acceder a la web mediante el bot de Telegram habiendo iniciado sesión previamente	128
6.76. Resultados de los casos de prueba y sus respectivas soluciones	131
7.1. Tareas del sprint 0	134
7.2. Tareas del sprint 1	135
7.3. Tareas del sprint 2	136
7.4. Riesgo R09: Análisis inicial incorrecto	137
7.5. Tareas del sprint 3	138
7.6. Tareas del sprint 4	140
7.7. Tareas del sprint 5	141
7.8. Tareas del sprint 6	143
7.9. Tareas del sprint de refuerzo 1	145
7.10. Tareas del sprint de refuerzo 2	147
7.11. Tareas de los momentos finales del proyecto	148
7.12. Coste simulado final	150

7.13. Coste real final 150

Capítulo 1

Introducción

1.1. Contexto

En la actualidad, tenemos acceso a casi cualquier información a partir de unos pocos clicks, si queremos comprar algo, incluso con un solo click podemos tenerlo al día siguiente en nuestro hogar [2] y así, podríamos enumerar infinidad de casos en los que la inmediatez es la protagonista. En la Sección 1.3.1 se definen los objetivos del proyecto y cómo se pretende aportar un granito de arena a esta "sociedad del click" facilitando la solicitud de tutorías y el acceso a la información de los profesores.

1.2. Motivación

Durante estos años cursando la titulación, siempre me he fijado como en los grupos de chat de las aplicaciones de mensajería que utilizamos para comunicarnos entre los estudiantes de la escuela, concretamente Telegram, existe un tipo de mensaje recurrente: "alguien sabe el correo de X profesor?". Siempre me preguntaba por qué la gente no era capaz de buscar la información de contacto de un profesor por sí misma en lugar de preguntar por estos grupos. "¿Tanto les cuesta mirar en la web de la escuela?", me preguntaba cada vez que uno de estos mensajes llegaba a mi bandeja. Finalmente, me di cuenta de que si el mensaje era tan recurrente es porque la información no era tan accesible como podía imaginar, por lo que es de aquí donde nace mi idea, la idea de tener un lugar (aplicación web) en el que los alumnos puedan encontrar con facilidad la información de los profesores y de poder, además, solicitar tutorías a los profesores y gestionarlas, ya que este era habitualmente el objetivo final de esa pregunta recurrente.

Además, mi tutora me sugirió la posibilidad de crear un bot de Telegram, idea que me pareció magnífica como complemento de la web ya que el alumno tendría la información de los profesores y el acceso a la web al alcance de su mano.

1.3. Objetivos

1.3.1. Objetivos del proyecto

El objetivo principal que persigue el desarrollo de este proyecto es elaborar una aplicación web en la que los usuarios, profesores y alumnos de la escuela de Ingeniería Informática de la Universidad de Valladolid, dispongan de las siguientes características:

- Gestionar la solicitud de tutorías.
- Visualizar mediante un calendario las tutorías planificadas.
- Buscar información de profesores.

Además, esta aplicación web se complementará con un bot de Telegram, que es el sistema de mensajería más utilizado entre alumnos de la escuela, y se encargará de facilitar la información de los profesores y/o acceso a la web para solicitar tutorías.

1.3.2. Objetivos personales

Este proyecto, al tratarse de un proyecto en un contexto académico, tiene una serie de objetivos para el alumno de cara a formarse como profesional. Estos objetivos son los siguientes:

- Adquirir conocimientos en el desarrollo web, tanto en la parte Backend como Frontend.
- Familiarizarse en el uso de tecnologías modernas de alta demanda, como son los framework Angular y Spring.
- Fortalecer conocimientos en las metodologías y marcos de trabajos para la gestión de proyectos, en este caso Scrum.

1.4. Estructura de la memoria

En esta sección se presenta la estructura seguida en el presente documento:

Capítulo 1 Introducción. En este capítulo se describe el porqué del proyecto y los objetivos del mismo.

Capítulo 2 Requisitos y planificación. En este capítulo se describe el marco de trabajo empleado para la gestión del proyecto, así como su adaptación a las condiciones del mismo. También se presenta la planificación inicial y los requisitos del proyecto.

Capítulo 3 Tecnologías utilizadas. En este capítulo se presentan las herramientas y tecnologías empleadas en todos los aspectos del proyecto, desde la comunicación con la tutora hasta el desarrollo del mismo.

Capítulo 4 Análisis. En este capítulo se describe la fase de análisis del proyecto, donde se plasman los diferentes diagramas utilizados para modelar el sistema tras la elicitación de requisitos.

Capítulo 5 Diseño. En este capítulo se muestran las decisiones de diseño empleadas para la posterior implementación. Se presenta desde los prototipos ideados para la interfaz de usuario hasta el diseño del despliegue a seguir.

Capítulo 6 Implementación y pruebas. En este capítulo se describe la organización seguida en la implementación del proyecto, la dockerización del mismo y las pruebas elaboradas para probar las aplicaciones desarrolladas.

Capítulo 7 Seguimiento del proyecto. En este capítulo se presenta como se ha avanzado el proyecto, así como las dificultades encontradas y como se han resuelto. También se compara la planificación del proyecto con la realidad.

Capítulo 8 Conclusiones y trabajo futuro. En este capítulo se exponen las conclusiones a las que se ha llegado al finalizar el proyecto y las propuestas como posible trabajo futuro.

Apéndice A Manuales. En este apéndice se describe el despliegue local de las aplicaciones y el manual de usuario para aprender a utilizarlas.

Apéndice B Resumen de enlaces adicionales. En este apéndice se incluyen los enlaces de interés del proyecto: el repositorio de código, el enlace de la web desplegada y el enlace al bot de Telegram.

Capítulo 2

Requisitos y planificación

2.1. Scrum

Para la realización de este proyecto se utilizará la metodología ágil Scrum adaptada a las limitaciones que supone el trabajo de fin de grado. Es por ello que conviene antes saber qué es Scrum.

2.1.1. ¿Qué es Scrum?

Scrum, más que una metodología, es un marco de trabajo mediante el que se puede abordar problemas de difícil adaptación, mientras que al mismo tiempo se entregan de forma eficiente y creativa productos con el máximo valor posible [45]. En este marco de trabajo, un único equipo desarrolla, entrega y mantiene productos complejos, que son entregados de manera incremental a partir de un enfoque iterativo, cuyas iteraciones se conocen como Sprints. Con este marco, se busca un proceso de trabajo ágil mediante el que es posible controlar la incertidumbre y por lo tanto, el riesgo. Para lograr esto, se apoya en 3 pilares:

- **Transparencia.** Se debe dar visibilidad a los aspectos significativos del proceso a todos los responsables del resultado.
- **Inspección.** Se promueve la inspección frecuente para la identificación y corrección de variaciones no deseadas.
- **Adaptación.** Se deben realizar ajustes en los diferentes procesos para evitar variaciones no deseadas.

SCRUM FRAMEWORK

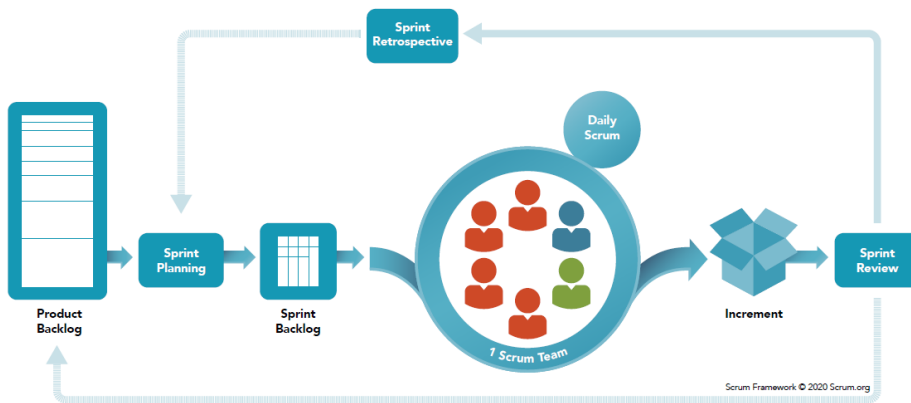


Figura 2.1: Funcionamiento de Scrum. Tomado de [45]

2.1.2. Scrum Team

Un Scrum Team o Equipo Scrum consta de 3 roles [31], que son el Product Owner o Dueño de Producto, el Scrum Master y el Equipo de Desarrollo. Este equipo es auto-organizado y multifuncional, de manera que es el propio equipo quien elige la mejor forma de realizar su trabajo y no depende de personas externas al equipo para realizar el trabajo.

- **Product Owner:** Es quien toma las decisiones, y quien realmente conoce el negocio del cliente y la visión que tiene del producto. Se encarga de asegurar que el Scrum Team trabaje adecuadamente en lo referente a la perspectiva de negocio. Además, es el encargado del artefacto Product Backlog (ver Sección 2.1.3), donde tiene como función añadir y ordenar, según prioridad, la lista de historias de usuario.
- **Scrum Master:** Se asegura de que las reglas establecidas por el marco de trabajo se entiendan, se cumplan, y funcionen. Se encarga de eliminar aquellos impedimentos que no permitan que el proceso fluya y sirve como punto de unión entre el Product Owner y el Equipo de Desarrollo, además de interactuar con el cliente y los diferentes gestores. También es el encargado de seleccionar un subconjunto de elementos del Product Backlog para que sean abordados en el Sprint en lo que se conoce como Sprint Backlog (ver Sección 2.1.3).
- **Equipo de Desarrollo:** Está formado, generalmente, por un equipo de 5 a 9 personas, que están auto-gerenciados, es decir, que tienen la autoridad suficiente para organizar y tomar las decisiones que consideren necesarias para lograr sus objetivos. Se encargan

de producir los incrementos de producto (ver Sección 2.1.3), que deben ajustarse a la planificación del Sprint, y trabajan sobre el Sprint Backlog.

2.1.3. Artefactos Scrum

Los artefactos de Scrum existen a modo de favorecer la transparencia, inspección y adaptación. Gracias a estos artefactos se maximiza la transparencia, favoreciendo así que todos los miembros del Scrum Team dispongan del mismo entendimiento de lo que se está realizando a partir de los artefactos [45]. Estos artefactos son los siguientes:

- **Product Backlog.** Consiste en una lista de los requisitos ordenados por prioridad de lo que se necesita para mejorar el producto desde el punto de vista del cliente. Estos requisitos pueden crecer y evolucionar a lo largo del proyecto. En Scrum a estos requisitos se los conoce como historias de usuario, y es el Product Owner quien se encarga de mantener actualizada esta lista, además de refinarla.
- **Sprint Backlog.** Es un subconjunto de las historias de usuario del Product Backlog que selecciona el equipo durante la planificación del Sprint actual y que prevén poder completarlo en dicho Sprint. Esta lista no se puede modificar durante la ejecución del Sprint pues puede comprometer la coordinación y cohesión con el desarrollo de otras tareas. Sin embargo, si no se ha realizado un buen pronóstico, las tareas no finalizadas pueden pasarse al siguiente Sprint o incluso se pueden añadir o modificar historias de usuario a partir de la nueva información obtenida o aprendida durante el Sprint.
- **Incremento.** Es el resultado de cada Sprint, es decir, los elementos del Sprint Backlog terminados y utilizables. Cada incremento es el valor del resultado de un Sprint acumulado con los incrementos anteriores.

2.1.4. Eventos Scrum

Scrum consta de unos eventos prefijados con el objetivo de crear una regularidad y evitar la necesidad de realizar reuniones no definidas en el marco de trabajo [45]. Dichos eventos son los siguientes:

- **Sprint.** Hacen referencia al evento más importante de Scrum, donde las ideas se convierten en valor. Estos eventos tienen una duración fija de 1 a 4 semanas. Cada nuevo Sprint comienza inmediatamente después de terminar el anterior. Este evento es el más fundamental y característico de este marco, pues el resto de eventos ocurren dentro de los Sprints. Es importante conocer que durante un Sprint no se pueden realizar cambios que pongan en peligro el Sprint Goal, la calidad no debe disminuir, el *Product Backlog* (ver Sección 2.1.3) se refina según se necesite, y el alcance puede aclararse y volver a negociarse con el Product Owner. El resultado de este evento es un *Incremento* (ver Sección 2.1.3).

- **Sprint Planning.** En este evento se lleva a cabo la planificación del Sprint y da comienzo al mismo. En esta reunión participa todo el *Scrum Team* (ver Sección 2.1.2). En este evento el Product Owner propone cómo el producto puede maximizar su valor y utilidad en el Sprint actual, para lo que se define un Sprint Goal. En esta planificación, el equipo de desarrollo selecciona los elementos del Product Backlog que prevé que puede finalizar durante el Sprint, dando lugar al *Sprint Backlog* (ver Sección 2.1.3). Además, el Scrum Team puede aprovechar esta reunión para refinar los elementos seleccionados con el objetivo de aumentar su entendimiento y aumentar la confianza del equipo. Por último, los desarrolladores planifican el trabajo necesario para crear un Incremento a partir del Sprint Backlog, para lo que se suelen descomponer los elementos de esta lista en elementos más pequeños con un coste de entre 4 a 16 horas, aunque la manera de realizar esta planificación queda al criterio exclusivo de los desarrolladores.
- **Sprint Goal.** Se trata del único objetivo del Sprint. Está en mente de los desarrolladores mientras estos trabajan en el Sprint y les sirve de guía acerca de por qué se está llevando a cabo un incremento. Proporciona flexibilidad en cuanto al trabajo necesario para lograr el objetivo y crea coherencia entre los miembros del Scrum Team.
- **Daily Scrum.** Consiste en un evento de unos 15 minutos de duración, que tiene como objetivo inspeccionar el progreso. Esta breve reunión diaria permite reducir la complejidad y los impedimentos que pueden poner en riesgo el progreso hacia el Sprint Goal, además de mejorar la comunicación entre el equipo.
- **Sprint Review.** Se realiza al finalizar el Sprint y tiene como objetivo inspeccionar el resultado del Sprint y determinar posibles adaptaciones de cara al futuro. En esta reunión se puede refinar el Product Backlog si se considera necesario. El *Product Owner* (ver Sección 2.1.2) se encarga de señalar qué elementos del Sprint Backlog se han hecho y cuáles no, a lo que los desarrolladores responden con lo que ha salido bien en el Sprint y con los problemas que se han encontrado y cómo los resolvieron si es que lo hicieron. El resultado de esta reunión de revisión del Sprint es un Product Backlog refinado para el próximo Sprint.
- **Sprint Retrospective.** En este evento el Scrum Team identifica los impedimentos que provocaron un desvío del Sprint Goal para identificar sus orígenes de manera que no vuelvan a ocurrir. También se analiza qué es lo que se hizo bien, qué es lo que se puede mejorar y qué es a lo que el equipo se compromete mejorar en el siguiente Sprint. En este evento el *Scrum Master* (ver Sección 2.1.2) se encarga de animar al resto del equipo con objetivo de mejorar el ambiente y confianza, además del proceso a seguir y las prácticas que pueden mejorar la efectividad. Este evento concluye el Sprint.

2.2. Roles y responsabilidades en el proyecto

Puesto que el marco de trabajo es Scrum, se debe definir quien se encargará de cada uno de los roles que marca dicho marco. Como se trata de un Trabajo de Fin de Grado la aplicación de Scrum no puede ser del todo ajustada debido a las limitaciones de este tipo de trabajo, donde normalmente suele participar únicamente un único alumno y un profesor

asignado que hace de tutor del trabajo. Por esta y otras limitaciones, se debe adaptar este marco de trabajo al contexto en el que nos encontramos.

En cuanto a los roles, el alumno será el encargado de los roles de Product Owner y del Equipo de Desarrollo, por lo que se encargará de definir el Product Backlog y del desarrollo de cada una de las historias de usuario, mientras que la tutora será quien se encargue del rol de Scrum Master, decidiendo así que historias de usuario se introducen en el Sprint Backlog y que el trabajo fluya adecuadamente.

En cuanto a los eventos, la duración de los sprints será de 2 semanas. Las reuniones de Sprint Planning, Sprint Review y Sprint Retrospective se mantienen, y tendrán lugar al final del sprint, momento en el que se comienza el siguiente sprint. Debido que realizar Daily Scrum es una tarea imposible para este proyecto, ya que tanto el alumno como la tutora tienen otras obligaciones que cumplir, se ha decidido que las reuniones pasarán a ser semanales.

2.3. Planificación inicial

A este proyecto, se le dedicarán más o menos 5 horas diarias sin contar fines de semana. Puesto que la asignatura correspondiente al Trabajo de Fin de Grado [53] consta de un total de 300 horas, se calculan 60 días para la realización del trabajo, que se corresponden a 12 semanas. Se ha establecido una duración de Sprint de 2 semanas (ver Sección 2.2), por lo que se cuenta con **6 sprints** para satisfacer las horas requeridas por la asignatura.

Sin embargo, se considera que se trabajará más tiempo que el planificado en la asignatura, pues se debe contar con un periodo de planificación y preparación de cara a la ejecución del proyecto. Para ello se emplean 2 semanas, en las que se realizará un sprint a mayores, el Sprint 0 [31], el cual se dedicará a la planificación inicial del proyecto, elicitación de requisitos (en este caso, historias de usuario), formación del alumno en las tecnologías con las que se desarrollará el proyecto, elaboración de un modelo de dominio inicial, selección de arquitectura y preparación del entorno de trabajo (instalación del software necesario, etc.).

Debido a los posibles riesgos de planificación del proyecto, se han añadido a mayores 2 sprints de refuerzo que servirán para completar tareas que no se hayan terminado a tiempo o trabajar en los detalles finales de la documentación a entregar. Teniendo, entonces, sprints de 50 horas, si el proyecto se lleva a cabo adecuadamente se realizarán un total de 6 sprints, además del Sprint 0, por lo que se estima un total de 320 horas. En caso de tener que realizarse uno de los sprints de refuerzo la estimación sería de 370 horas y en caso de utilizarse ambos sprints de refuerzo se tendría como estimación inicial un total de 420 horas.

Para medir la carga de trabajo se emplearán puntos de historia y **se establece que 1 punto de historia equivale a 5 horas de trabajo.**

En la Tabla 2.1 se puede observar la calendarización de los diferentes sprints, así como otros eventos del proyecto.

2.4. PLAN DE RIESGOS

Sprint	Comienzo	Finalización	Observaciones
Sprint 0	08/02/2022	23/02/2022	
Scrum semanal	15/02/2022		
Sprint Review, Sprint Retrospective y Sprint Planning	23/02/2022		
Sprint 1	23/02/2022	09/03/2022	
Scrum semanal	02/03/2022		
Sprint Review, Sprint Retrospective y Sprint Planning	09/03/2022		
Sprint 2	09/03/2022	23/03/2022	
Scrum semanal	16/03/2022		
Sprint Review, Sprint Retrospective y Sprint Planning	23/03/2022		
Sprint 3	23/03/2022	06/04/2022	
Scrum semanal	30/03/2022		
Sprint Review, Sprint Retrospective y Sprint Planning	06/04/2022		
Sprint 4	06/04/2022	27/04/2022	La semana del 11 al 15 de abril de 2022 se hará un descanso por vacaciones de Semana Santa, por ello el sprint termina más tarde.
Scrum semanal	20/04/2022		
Sprint Review, Sprint Retrospective y Sprint Planning	27/04/2022		
Sprint 5	27/04/2022	11/05/2022	
Scrum semanal	04/04/2022		
Sprint Review, Sprint Retrospective y Sprint Planning	11/05/2022		
Sprint 6	11/05/2022	25/05/2022	
Scrum semanal	18/05/2022		
Sprint Review, Sprint Retrospective y Sprint Planning	25/05/2022		En caso de no realizarse el siguiente sprint de refuerzo, el Sprint Planning no se llevará a cabo.
Sprint refuerzo 1	25/05/2022	08/06/2022	
Scrum semanal	01/06/2022		
Sprint Review, Sprint Retrospective y Sprint Planning	08/06/2022		En caso de no realizarse el siguiente sprint de refuerzo, el Sprint Planning no se llevará a cabo.
Sprint refuerzo 2	08/06/2022	22/06/2022	
Scrum semanal	15/06/2022		
Sprint Review, Sprint Retrospective y Sprint Planning	22/06/2022		
Límite solicitud defensa primera convocatoria	24/06/2022		
Límite solicitud defensa segunda convocatoria	15/07/2022		

Tabla 2.1: Planificación inicial

2.4. Plan de riesgos

2.4.1. Introducción

Un riesgo en un proyecto es aquello que evita que un proyecto sea completado con éxito [27]. Se caracterizan por estar relacionados con el futuro, que es incierto y algunos riesgos que son obvios cuando el proyecto está en marcha pueden no serlo a la hora de planificarlo, y por implicar una causa y un efecto. En un proyecto existen dos tipos de riesgo, los riesgos de negocio y los riesgos de proyecto.

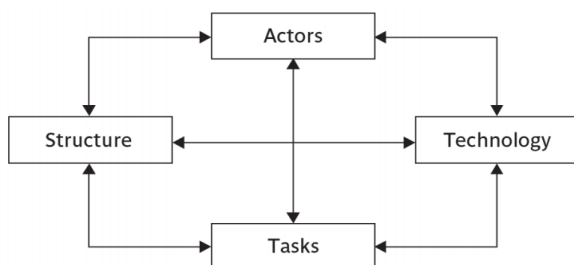


Figura 2.2: Marco de trabajo de riesgos de Lyttinen-Mathiassen-Ropponen. Tomado de [27]

Los riesgos de negocio son aquellos que afectan al modelo de negocio y provocan que el producto entregado no sea rentable. Por otro lado, los riesgos de proyecto son aquellos que afectan a la planificación del proyecto, poniendo en riesgo la calidad de este y su fecha de finalización. Este último tipo de riesgo es el que concierne a este trabajo y es en el que nos centraremos.

Los riesgos de proyecto se pueden categorizar en 4 aspectos según su naturaleza (ver Figura 2.2):

- **Actores:** son aquellos que hacen referencia a las personas involucradas en el desarrollo de la aplicación en cuestión.
- **Estructura:** son los referidos a la gestión de la estructuras y sistemas de gestión, e incluso afectan a la planificación y al control.
- **Tecnología:** Se refiere a las tecnología utilizadas para el desarrollo de la aplicación.
- **Tareas:** hace referencia a las tareas planeadas. Puede incluir retrasos por la complejidad de una tarea.

Para llevar a cabo un correcto plan de riesgos debemos seguir una series de pasos:

1. Identificación de riesgos.
2. Análisis y priorización de riesgos.
3. Planificación de riesgos.
4. Monitorización de riesgos.

2.4.2. Gestión inicial de riesgos

A continuación, se detallan los riesgos identificados en la fase de planificación del proyecto. Sin embargo, al tratarse de un proyecto gobernado por un marco de trabajo ágil e iterativo

2.4. PLAN DE RIESGOS

se pueden identificar nuevos riesgos a medida que avanza el proyecto. Los nuevos riesgos identificados durante la ejecución del proyecto, se detallan en el Capítulo 7, correspondiente al seguimiento del proyecto. Para realizar un análisis de los riesgos y poder priorizarlos, se elabora una matriz de riesgos en la que se muestran las acciones a seguir según la probabilidad e impacto de un riesgo. En la Tabla 2.2 se muestra la matriz de riesgos que se empleará en este proyecto.

Probabilidad / Impacto	Bajo	Medio	Alto
Bajo	Monitorizar	Monitorizar	Aplicar plan de mitigación
Medio	Monitorizar	Aplicar plan de mitigación	Mitigar lo posible y preparar la aplicación del plan de contingencia
Alto	Aplicar plan de mitigación	Mitigar lo posible y preparar la aplicación del plan de contingencia	Mitigar lo posible y preparar la aplicación del plan de contingencia

Tabla 2.2: Matriz de riesgos

Para la reducción de la probabilidad y reducción del impacto de los riesgos se establece un plan de mitigación y de contingencia, respectivamente. Los riesgos identificados en la planificación del proyecto se presentan de la Tabla 2.3 a la 2.10.

ID del riesgo	R01
Título	Falta de formación del equipo de desarrollo
Categoría del riesgo	Tecnología
Descripción del riesgo	El equipo de desarrollo no tiene la formación necesaria para el uso fluido de las tecnologías que se aplican en el proyecto.
Probabilidad del riesgo	Medio
Impacto del riesgo	Medio
Plan de mitigación	<ul style="list-style-type: none"> ■ Se han elegido tecnologías con las que el alumno está familiarizado. ■ Familiarizarse con aquellas tecnologías en las que se tenga menos conocimiento antes de la ejecución del proyecto.
Plan de contingencia	<ul style="list-style-type: none"> ■ Solicitar ayuda a tutora o personas con un conocimiento más avanzado en las tecnologías.

Tabla 2.3: Riesgo R01: Falta de formación del equipo de desarrollo

ID del riesgo	R02
Título	Planificación de la duración del proyecto poco realista
Categoría del riesgo	Estructura
Descripción del riesgo	La planificación del proyecto es demasiado optimista y no se dispone del tiempo necesario para cumplir con el trabajo diario establecido y la fecha de finalización del proyecto.
Probabilidad del riesgo	Medio
Impacto del riesgo	Alto
Plan de mitigación	<ul style="list-style-type: none"> ■ El uso del marco de trabajo Scrum favorece el correcto desarrollo del trabajo en proyectos en los que se dispone de un alto grado de incertidumbre.
Plan de contingencia	<ul style="list-style-type: none"> ■ Pedir consejo a tutor. ■ Aumentar el número de horas dedicadas. ■ Retrasar la finalización del proyecto.

Tabla 2.4: Riesgo R02: Planificación de la duración del proyecto poco realista

ID del riesgo	R03
Título	Retraso en el desarrollo de las tareas
Categoría del riesgo	Tareas
Descripción del riesgo	La complejidad de algunas tareas de un Sprint puede llevar más tiempo del esperado.
Probabilidad del riesgo	Medio
Impacto del riesgo	Medio
Plan de mitigación	<ul style="list-style-type: none"> ■ Planificar una duración adecuada para los sprints teniendo en cuenta el tamaño y complejidad de las historias de usuario. ■ Se han planificado dos sprints de refuerzo.
Plan de contingencia	<ul style="list-style-type: none"> ■ Emplear más horas en el desarrollo de las tareas. ■ Retrasar las tareas inacabadas al siguiente sprint.

Tabla 2.5: Riesgo R03: Retraso en el desarrollo de las tareas

ID del riesgo	R04
Título	Gold plating
Categoría del riesgo	Tareas
Descripción del riesgo	Se han desarrollado o se están desarrollando funciones que no son necesarias y no se han solicitado.
Probabilidad del riesgo	Bajo
Impacto del riesgo	Medio
Plan de mitigación	<ul style="list-style-type: none"> ■ Refinar el backlog todo lo posible.
Plan de contingencia	<ul style="list-style-type: none"> ■ No llevar a cabo el desarrollo de funcionalidad innecesaria. ■ Ponerse en contacto directo con los clientes e interesados fuera del Sprint Planning y preguntar sobre la funcionalidad.

Tabla 2.6: Riesgo R04: Gold plating

ID del riesgo	R05
Título	Baja médica por enfermedad u otro problema del equipo de desarrollo
Categoría del riesgo	Actores
Descripción del riesgo	Alguno de los miembros del equipo de desarrollo está enfermo o con otra patología y no puede desarrollar sus tareas a tiempo. Puesto que se trata de un proyecto desarrollado por una única persona, es un riesgo de un alto impacto.
Probabilidad del riesgo	Bajo
Impacto del riesgo	Alto
Plan de mitigación	<ul style="list-style-type: none"> ■ Se han planificado dos Sprints sprints de refuerzo. ■ Se cuenta con Sprints con una duración suficiente para que el actor pueda recuperarse y, aunque el Sprint se pueda ver afectado, no lo haga en su totalidad.
Plan de contingencia	<ul style="list-style-type: none"> ■ Retrasar tareas no acabadas al Sprint siguiente. ■ Se planifican dos sprints de refuerzo para actuar como colchón.

Tabla 2.7: Riesgo R05: Baja médica por enfermedad u otro problema del equipo de desarrollo

ID del riesgo	R06
Título	Ausencia por consulta médica u otra actividad
Categoría del riesgo	Actores
Descripción del riesgo	Alguno de los miembros del equipo de desarrollo falta a su puesto debido a que tiene que acudir a una consulta médica u otra actividad que requiere de su asistencia.
Probabilidad del riesgo	Bajo
Impacto del riesgo	Alto
Plan de mitigación	<ul style="list-style-type: none"> ■ Se dispone de algunas tardes para evitar que la ausencia provoque problemas. ■ Se cuenta con sprints con una duración suficiente para que el actor pueda recuperarse y, aunque el Sprint se pueda ver afectado, no lo haga en su totalidad.
Plan de contingencia	<ul style="list-style-type: none"> ■ Trabajar durante los fines de semana. Retrasar tareas no acabadas al Sprint siguiente. ■ Se planifican dos sprints de refuerzo para actuar como colchón.

Tabla 2.8: Riesgo R06: Ausencia por consulta médica u otra actividad

ID del riesgo	R07
Título	Mala comunicación con el tutor
Categoría del riesgo	Actores
Descripción del riesgo	Una mala comunicación con el tutor del proyecto afecta al correcto desarrollo del proyecto.
Probabilidad del riesgo	Bajo
Impacto del riesgo	Alto
Plan de mitigación	<ul style="list-style-type: none"> ■ Se utiliza el marco de trabajo Scrum, de manera que se establecen reuniones periódicas con el tutor para mejorar la comunicación y aclarar las dudas y problemas que surjan.
Plan de contingencia	<ul style="list-style-type: none"> ■ Establecer canales de comunicación directa. ■ Llevar a cabo reuniones fuera de las planificadas para tratar de resolver todos los problemas y dudas lo antes posible.

Tabla 2.9: Riesgo R07: Mala comunicación con el tutor

ID del riesgo	R08
Título	Cambios en los requisitos
Categoría del riesgo	Estructura
Descripción del riesgo	Durante el proyecto existe la posibilidad de que los requisitos cambien, siendo estos modificados o añadiendo nuevos o eliminando los que ya están.
Probabilidad del riesgo	Medio
Impacto del riesgo	Medio
Plan de mitigación	<ul style="list-style-type: none"> ■ La utilización del marco de trabajo Scrum favorece la gestión de la incertidumbre, por lo que es posible incluir nuevos requisitos o eliminarlos en cada Sprint Planning, de manera que se refine el Product Backlog sin afectar gravemente al desarrollo del proyecto.
Plan de contingencia	<ul style="list-style-type: none"> ■ Planificar los sprints ajustándose a los nuevos requisitos. ■ Retrasar la finalización del proyecto.

Tabla 2.10: Riesgo R08: Cambios en los requisitos

2.5. Estimación de costes

Para la elaboración de la estimación de costes se tendrán en cuenta los costes de recursos humanos, así como los costes del hardware y software que se utilizará en la elaboración del proyecto. En primer lugar, se contempla un caso simulado y después el caso real de este proyecto.

2.5.1. Estimación de costes simulado

El coste de recursos humanos en este proyecto está formado por el sueldo de un único desarrollador de software, que en España, el sueldo promedio a febrero de 2022 para este puesto de trabajo es de 29391€ anuales según la portal de empleo *indeed.com* [28], que obtiene el sueldo promedio a partir de los diferentes sueldos pagados en las diferentes ciudades españolas. Este salario debemos incrementarlo en un 30 %, correspondiente a la aportación de la empresa a la seguridad social por el trabajador, obteniendo entonces un coste por el trabajador de 38208'3€. Por lo tanto, si anualmente se realizan unas 1800 horas de trabajo, obtenemos que el coste por hora es de 21'23€, aproximadamente, por lo que teniendo planificadas 320 horas para la elaboración del trabajo si no hay complicaciones, el presupuesto bruto para contar con un desarrollador de software en el proyecto es de **6793'60€**.

En cuanto al hardware, como ordenador a emplear se escoge un portátil Lenovo Think-Book 15 G2 I7L, valorado en 658'55€ con una vida útil de 4 años, por lo que si dividimos el precio total del ordenador entre el número de meses de vida útil (48 meses), obtenemos un precio de 13'72€/mes. Tenemos, entonces, para los aproximadamente 4 meses de duración del proyecto un coste amortizado de **54'88€**.

Para el software se emplearán las licencias más económicas en cada uno de los servicios. Para la comunicación se utilizará Rocket.Chat, que cuenta con una **versión gratuita** que incluye unas prestaciones más que suficientes para el proyecto. Para la creación de los prototipos de la aplicación se va a emplear Balsamiq Cloud, cuya tarifa mínima es de 7'95€/mes, por lo tanto y puesto que es un servicio de compra mensual se tendrán que adquirir el servicio por 4 meses para satisfacer la duración del proyecto, teniendo, así, un coste de **31'80€**. En cuanto a la elaboración de los modelos de la aplicación se ha escogido como opción Astah Professional que vale 8'82€/mes, teniendo un coste para 4 meses de **35'28€**.

El coste total asciende, entonces, a 6915'56€. Sin embargo, para evitar que este presupuesto llegue a sobrepasarse debido a que el proyecto dure más de lo previsto o se utilicen herramientas no planeadas, el presupuesto total se elevará un 20 % teniendo como presupuesto total para el proyecto la cifra de **8298'67€**.

En la Tabla 2.11 se muestra el resumen de la estimación de costes simulado.

Concepto	Precio	Cantidad	Total
Hora de trabajo de desarrollador de software	21'23€/hora	320 horas	6793'60€
Equipo (Lenovo ThinkBook 15 G2 ITL)	13'72€/mes	4 meses	54'88€
Licencia Rocket.Chat	0€/mes	4 meses	0€
Licencia Balsamiq Cloud	7'95€/mes	4 meses	31'80€
Licencia Astah Professional	8'82€/mes	4 meses	35'28€
Total			6915'56€
Total normalizado (+20 %)			8298'67€

Tabla 2.11: Estimación de costes simulado

2.5.2. Estimación de costes real

Para el coste real, puesto que se trata de un proyecto con un fin académico, no se contará con un sueldo para el alumno. Además, el software a utilizar será gratuito o se contará con licencias que facilita la Escuela y por las que el alumno no tiene que pagar.

En cuanto al hardware, el alumno utilizará su propio equipo, un Asus ROG Strix G15 G513QR-HF010T, que tiene un coste total de 2099€. Teniendo como vida útil del equipo 4 años, se obtiene un coste aproximado de 43'73€/mes, por lo que para cubrir la duración del proyecto (4 meses), el coste amortizado será de aproximadamente 174'92€.

Puesto que el proyecto se realizará en casa del alumno o en la Universidad, los costes de luz y otros servicios están a cargo de entidades externas al proyecto. Por lo tanto, se tiene como presupuesto total únicamente el coste del equipo, es decir, que el presupuesto final es de 174'92€.

En la Tabla 2.12 se muestra el resumen de la estimación de costes real.

Concepto	Precio	Cantidad	Total
Equipo (Asus ROG Strix G15 G513QR-HF010T)	43'73€/mes	4 meses	174'92€
Total			174'92€

Tabla 2.12: Estimación de costes real

2.6. Product Backlog inicial

Como se ha indicado en la *Planificación inicial* (ver Sección 2.3), durante las 2 primeras semanas del proyecto, concretamente, durante el sprint 0, se realizan, entre otras cosas, la elicitación de requisitos. Sin embargo, en este proyecto nos encontramos en el marco de trabajo Scrum, por lo que no hablamos de requisitos sino de historias de usuario. Una historia de usuario [32] es una pequeña descripción con un lenguaje no técnico de las tareas de desarrollo que siguen el esquema *Como* -stakeholder- *quiero* -necesidad- *para* -propósito-, siendo el stakeholder cualquier entidad con interés en el proyecto.

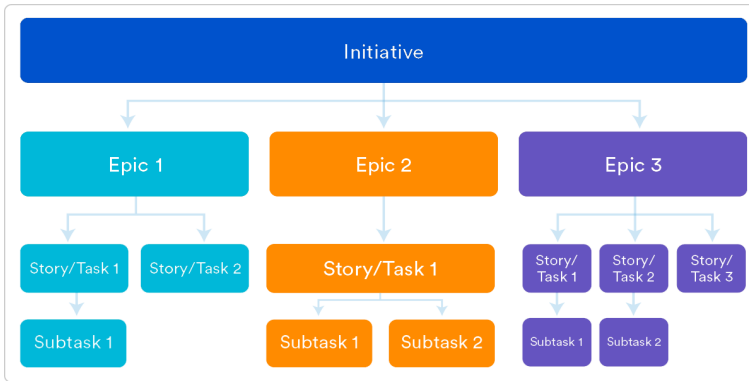


Figura 2.3: Épicas y su subdivisión en historias de usuario. Tomado de [32]

En una primera instancia, las historias de usuario son unas pocas frases que utilizan un lenguaje sencillo y no entran en demasiados detalles, quedando así una historia de usuario poco refinada que da lugar a una *historia épica* o *epic*. En la Tabla 2.13 se puede observar una primera versión del Product Backlog inicial de este proyecto.

Número	Épica
1	Como usuario quiero acceder al sistema para utilizar las funcionalidades de ésta de forma personalizada.
2	Como alumno quiero conocer la información de los profesores para tener acceso a ésta de manera sencilla y rápida.
3	Como usuario quiero gestionar la solicitud de tutorías para facilitar el proceso.
4	Como usuario quiero gestionar mi información para mantenerla actualizada.
5	Como usuario quiero gestionar las tutorías programadas para planificarme con facilidad.
6	Como alumno quiero que la funcionalidad de la aplicación se integre en un bot de Telegram para facilitar el acceso a sus funcionalidades.

Tabla 2.13: Product Backlog inicial

2.7. Product Backlog final

Durante el transcurso del proyecto hasta su finalización, el *Product Backlog* ha sufrido diferentes cambios, pues se han añadido nuevas historias de usuario, otras se han refinado y algunas se han eliminado. Para ser concreto, se han añadido ocho historias de usuario, de la *HU18* a la *HU25*, y se ha eliminado una, la *HU11*. Además, la historia de usuario *HU12* ha visto modificada su especificación y la historia *HU20* no se ha realizado por falta de tiempo, por lo que se ha bloqueado para un futuro desarrollo ya fuera del alcance de este proyecto. En

2.7. PRODUCT BACKLOG FINAL

el Capítulo 7, dedicado al seguimiento del proyecto, se detallan estos cambios en el *Product Backlog*.

En la Tabla 2.14 se muestra el *Product Backlog final* ordenado por prioridad, mostrándose en color rojo las historias de usuario eliminadas, en color naranja las que han sufrido un cambio en su especificación y en color morado las que han sido bloqueadas.

Identificador	Historia de usuario	Puntos de historia
HU03	Como alumno quiero buscar un profesor según diferentes filtros (por asignatura o por nombre) para poder acceder de manera más sencilla a la información del profesor.	5
HU04	Como alumno quiero ver la información detallada del profesor.	5
HU18	Como equipo de desarrollo quiero idear el formulario de solicitud de tutoría sin llegar a desarrollarlo	1
HU05	Como alumno quiero hacer una solicitud de tutoría a un profesor para facilitar la tarea de solicitar una tutoría.	4
HU06	Como alumno quiero cancelar una solicitud de tutoría para rectificar algunas condiciones de la solicitud o para simplemente cancelarla.	2
HU01	Como usuario quiero acceder a la pantalla de inicio de sesión y a partir de mis credenciales entrar en mi cuenta de profesor o alumno.	2
HU02	Como usuario quiero terminar mi sesión para salir del sistema.	1
HU07	Como profesor quiero aceptar una solicitud de tutoría recibida.	2
HU08	Como profesor quiero rechazar una solicitud de tutoría recibida.	2
HU19	Como alumno quiero confirmar una solicitud de tutoría aceptada.	1
HU09	Como usuario quiero consultar las solicitudes según su estado (pendiente, aceptada, modificada, confirmada o rechazada) u otros factores (asignatura, usuario, fecha o grupal) para conocer el estado en el que se encuentran.	3
HU13	Como usuario quiero consultar un calendario en el que se vean reflejadas las tutorías programadas.	4
HU14	Como usuario quiero consultar las tutorías programadas (para hoy o próximamente) en forma de lista y filtrar según diversos parámetros (cancelada o no, asignatura, usuario o fecha).	2
HU15	Como usuario quiero cancelar una tutoría para evitar que esta se lleve a cabo.	2
HU10	Como usuario quiero acceder a mi perfil para poder acceder a mi información.	2
HU11	Como profesor quiero establecer un horario de tutorías para que los alumnos puedan conocer el horario de disponibilidad de las mismas.	4
HU21	Como profesor quiero que las solicitudes pendientes se muestren en la pantalla de inicio para acceder a ellas con mayor facilidad.	1
HU22	Como alumno quiero que las solicitudes en estado aceptada se muestren en la pantalla de inicio para acceder a ellas más fácilmente.	1
HU24	Como usuario quiero que las próximas tutorías se muestren en la pantalla de inicio para facilitar su acceso.	1
HU23	Como profesor quiero que todas las solicitudes pendientes que no hayan sido atendidas a tiempo se rechacen automáticamente para automatizar la gestión.	2
HU16	Como alumno quiero conocer la información de un profesor a partir de un bot de Telegram para facilitar el acceso a la información del profesor.	6
HU17	Como alumno quiero acceder a la aplicación mediante un bot de Telegram para facilitar la navegación hasta la misma.	6
HU12	Como profesor quiero que la información sobre mi horario de tutorías se cargue mediante un fichero CSV.	3
HU20	Como miembro de la administración quiero que la población (titulaciones, asignaturas, usuarios y su información) de la aplicación se carguen mediante un fichero.	3
HU25	Como equipo de desarrollo quiero “dockerizar” las aplicaciones para poder ejecutarlas en diferentes entornos sin preocuparnos por las dependencias.	6

Tabla 2.14: Product Backlog final ordenado por prioridad

Capítulo 3

Tecnologías utilizadas

3.1. Herramientas para la comunicación

3.1.1. Rocket.Chat

Rocket.Chat [44] es una plataforma de comunicación de código abierto. Es la plataforma que se ha utilizado durante la ejecución del proyecto para mantener el contacto con la tutora en caso de tener que preguntar alguna duda o cuestión puntual. También ha sido la plataforma que ha servido para comunicar la cancelación o retraso de alguno de los *eventos Scrum* (ver Sección 2.1.4) por alguna razón externa al proyecto.

3.1.2. Webex

Webex [56] ha sido la plataforma de videoconferencia que se ha empleado en el proyecto para llevar a cabo las diferentes reuniones que marcan los eventos Scrum. Esta herramienta se caracteriza por ser *flexible*, pues se adapta a cualquier estilo de trabajo; *inclusivo*, pues se ofrece una experiencia igualitaria para todos independientemente de la ubicación geográfica, el idioma o el estilo de comunicación; y segura, ya que su diseño privado de forma predeterminada, nos protege sin importar el lugar en donde se trabaje.

3.2. Herramientas para el análisis, diseño y documentación

3.2.1. Astah

Astah [8] es una herramienta de modelado de software. Con ella podemos visualizar en forma de diagramas ideas y diseños de software. Gracias a su facilidad de uso y simplicidad, es la herramienta que se ha escogido para elaborar todos los diagramas de análisis (ver Capítulo 4) y diseño (ver Capítulo 5) del proyecto, además de que gracias a la licencia que posee la Escuela de Ingeniería Informática de la Universidad de Valladolid, la aplicación puede ser utilizada en su versión profesional obteniendo así todas sus ventajas.

3.2.2. Balsamiq Cloud

Balsamiq Cloud [11] es una herramienta online que sirve para diseñar interfaces usuarios y que ha sido la utilizada para elaborar todos los prototipos de la interfaz de usuario de la web creada. En el Capítulo 5, dedicado al diseño, se muestran los diferentes prototipos creados.

3.2.3. Overleaf

Overleaf [37] es un editor de $\text{L}^{\text{T}}\text{E}^{\text{X}}$ [43] que permite a varios colaboradores trabajar sobre un mismo proyecto sin importar en que lugar del mundo se encuentren. En este proyecto ha servido para elaborar el documento presente y, para que, además, la tutora tenga acceso en todo momento y poder ofrecer la retroalimentación necesaria para obtener, como resultado final, un documento de calidad en términos de un trabajo de fin de grado.

3.3. Herramientas para la gestión y control del proyecto

3.3.1. Git

Git [20] es una herramienta gratuita y de código abierto que nos permite tener un sistema de control de versiones distribuido para manejar todo tipo de proyectos con velocidad y eficiencia.

La diferencia de Git con otros sistemas de control de versiones es su sistema de ramas. Esto permite que se tengan diferentes ramas locales que pueden ser totalmente independientes entre sí, lo que permite crear ramas para probar ideas, corregir errores, etc., que pueden, después, combinarse con la rama original.



Figura 3.1: Git - Ramificación. Tomado de [20]

3.3.2. GitLab

GitLab [22] es un servicio web de desarrollo colaborativo de software, control de versiones y DevOps [58] basado en Git. GitLab tiene como objetivo reunir en una misma aplicación todas las herramientas necesarias para el desarrollo de un proyecto de software en equipo, desde la planificación del mismo hasta la producción.

En este proyecto, GitLab se ha utilizado para mantener un control de versiones de manera remota y para la gestión de tareas a partir de la herramienta GitLab Issue Board [21]. En esta herramienta de tableros se crearon distintas etiquetas para identificar el estado en la que se encontraba una tarea correspondiente a una historia de usuario. Las etiquetas que se crearon son las siguientes:

- **Product Backlog.** Representa el Product Backlog (ver Sección 2.1.3) y es donde se encuentran todas las historias de usuario identificadas inicialmente y las que son identificadas durante el transcurso del proyecto con sus respectivas *issues*.
- **Sprint Backlog.** Representa el Sprint Backlog (ver Sección 2.1.3). Aquí se encuentran las *issues* a realizar durante la ejecución de un Sprint y que son añadidas durante las reuniones de Sprint Planning.
- **Doing.** Aquí es donde se encuentran las *issues* que se están realizando y que provienen del Sprint Backlog.
- **Under Review.** Este es el estado en el que se encuentran las *issues* que se consideran que se han terminado pero que no han sido revisadas. Las revisiones se realizan durante las reuniones de Sprint Review y en ellas se determina si una *issue* pasa al siguiente estado o no. Proviene del estado “Doing”.
- **Done.** En este estado se encuentran las *issues* que ya se han terminado y han sido revisadas. Proviene del estado “Under Review”.

Además de estas etiquetas, GitLab incluye la columna “Closed”, que es donde se encuentran las *issues* ya finalizadas y revisadas que se han ido cerrando durante el proyecto.

3.4. TECNOLOGÍAS PARA EL DESARROLLO DEL PROYECTO

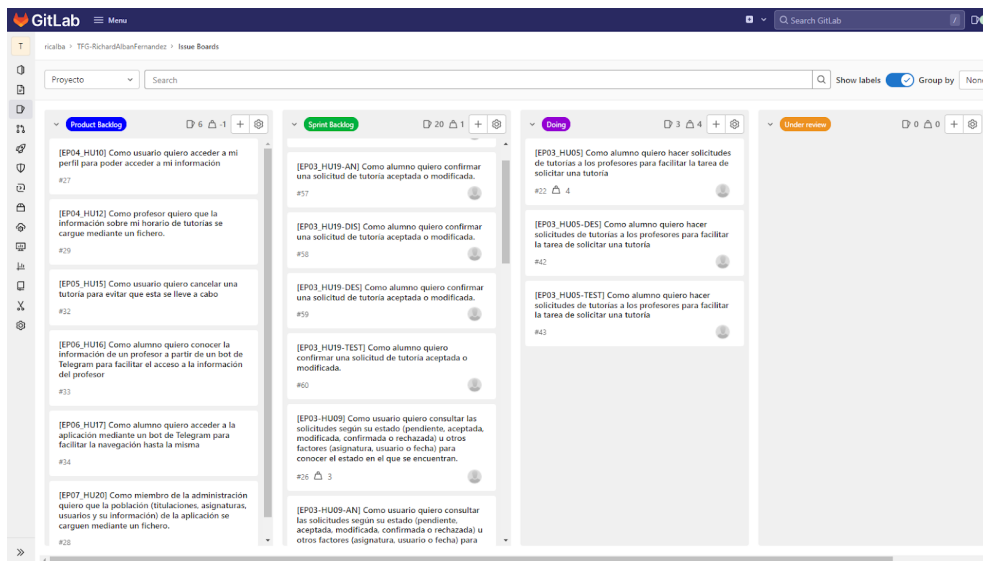


Figura 3.2: GitLab Issue Board (parte 1)

Para identificar la fase de cada una de las historias de usuario se ha seguido la siguiente nomenclatura para las *issues*: “<[<idEpic>_<idHistoriaDeUsuario>-<fase>] <descripción historia de usuario>”. Las fases que puede tener cada *issue* son las siguientes:

- **AN** - Correspondiente a una tarea de análisis.
- **DIS** - Correspondiente a una tarea de diseño.
- **DES** - Correspondiente a una tarea de desarrollo.
- **TEST** - Correspondiente a una tarea de pruebas.

Además, para cada conjunto de *issues* de una historia de usuario se incluye una *issue padre*, que no incluye la fase, pues hace referencia a la historia de usuario en su totalidad, y que además es donde se define el peso de la historia de usuario. También, cabe destacar que la *issue padre* no podrá pasar al siguiente estado hasta que todas sus *issues hijas* lo hayan hecho. En las Figuras 3.2 y 3.3 se muestra un ejemplo de como se encuentra el tablero en un momento determinado del sprint.

3.4. Tecnologías para el desarrollo del proyecto

3.4.1. Visual Studio Code

Visual Studio Code [12] es un editor de código desarrollado por Microsoft [66]. A diferencia de otro editores, Visual Studio Code se puede extender mediante plugins, que

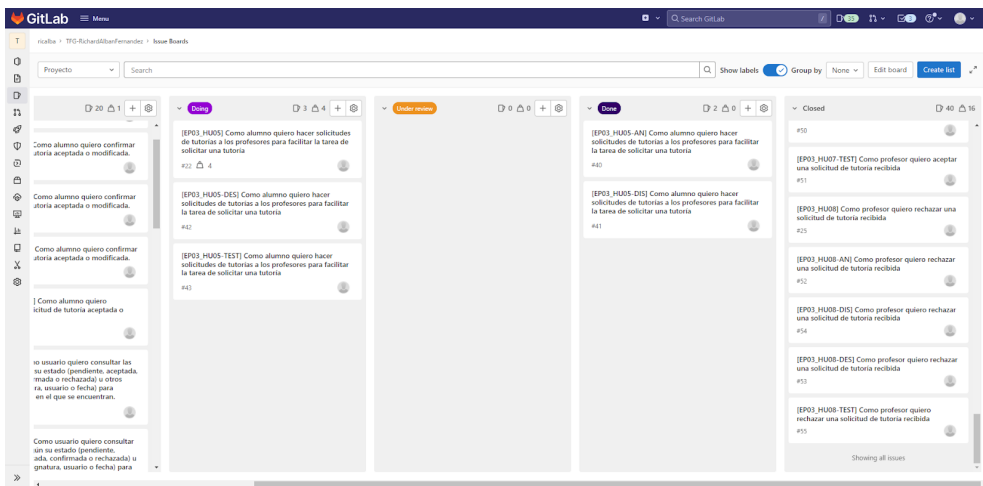


Figura 3.3: GitLab Issue Board (parte 2)

aportan nueva funcionalidad y/o soporte de tecnologías. De esta manera, VS Code se puede dotar de la funcionalidad que podríamos esperar de entorno de desarrollo integrado [59], convirtiéndose así en uno de los editores de código más populares en la actualidad.

En este proyecto, VS Code es empleado para la parte frontend y para la edición de ficheros de configuración, como pueden ser los ficheros de configuración Dockerfile [13], entre otros.

3.4.2. Eclipse IDE

Eclipse IDE [18], en concreto Eclipse IDE for Enterprise Java and Web Developers, es un entorno de desarrollo integrado [59] específico para el lenguaje de programación Java, con soporte para gran variedad de tecnologías, como gestores de dependencia o control de versiones. Además, cuenta con un Marketplace que permite personalizar y añadir nuevas funcionalidades en el entorno.

En este proyecto, Eclipse IDE se utiliza para la parte backend y el desarrollo del bot de Telegram.

3.4.3. Node.js

Node.js [33] es un entorno de ejecución para JavaScript orientado a eventos asíncronos y diseñado para la creación de aplicaciones escalables. En este proyecto se utiliza como soporte para el desarrollo en Angular.

3.4.4. NPM

Node Package Manager (NPM) [34] es un gestor de paquetes de código abierto desarrollado en JavaScript utilizado para añadir dependencias, librerías y módulos a las aplicaciones que desarrollemos. En este proyecto, se utiliza como una dependencia de Angular, pues lo necesita para muchas de sus características y funciones, y, además, nos permitirá añadir las dependencias que necesitemos en nuestro proyecto Angular.

3.4.5. Angular

Angular [5] es una plataforma de desarrollo, construida sobre TypeScript [52], pensada para el desarrollo de aplicaciones web escalables y de una sola página (SPA). Las Single Page Application [54], se caracterizan porque toda la aplicación tiene una única página, tal y como indica su nombre. En este tipo de páginas, todos los recursos se cargan de una sola vez o de manera dinámica según se acceda a ellos. Las características esenciales de Angular son:

- **Componentes.** Son los bloques que componen la aplicación. Estos componentes incluyen una clase TypeScript, una plantilla HTML, y estilos (CSS o SCSS, entre otros posibles).
- **Plantillas.** Cada plantilla HTML marca como se representa un componente.
- **Inyección de dependencias.** La inyección de dependencias nos permite declarar las dependencias de una clase TypeScript sin preocuparnos de la creación de las instancias, ya que es el propio framework quien se encarga de hacerlo.

Angular es el framework mediante el que se desarrollará la parte frontend de este proyecto.

3.4.6. PrimeNG

PrimeNG [39] es un conjunto de componentes de interfaz de usuario nativos de Angular y de código abierto. Gracias a PrimeNG, podemos desarrollar una aplicación de calidad y con un aspecto visual atractivo sin demorarnos en el uso avanzado de hojas de estilo, como CSS [55] o SCSS [17], favoreciendo así el desarrollo si no se está muy familiarizado con éstas, además de facilitar tareas como la paginación, el lazy loading [60] o el desarrollo de aplicaciones responsivas [1].

3.4.7. Nginx

Nginx [30] es un servidor web de código abierto. Se caracteriza por utilizar un enfoque asíncrono basado en eventos donde las solicitudes se manejan en un solo hilo. En este proyecto se utilizará como servidor web a la hora de desplegar el frontend.

3.4.8. Spring Framework

Spring [48] es el framework más popular del mundo para el desarrollo de aplicaciones en Java. Algunas claves para este éxito es que hace que programar en Java sea más rápido, fácil y seguro, convirtiéndolo en un framework muy productivo.

Para este proyecto haremos uso de Spring Boot [47], que es una versión más simple de Spring Framework que nos permite crear aplicaciones conocidas como “just run”. Lo emplearemos para el desarrollo del backend de nuestro proyecto, en el que crearemos una API REST que será consumida por el frontend. También lo emplearemos para desarrollar el bot de Telegram.

3.4.9. Apache Maven

Apache Maven [42] es una herramienta de software para la construcción y gestión de proyectos de software. Permite, a partir de un modelo de objeto del proyecto (POM), la construcción de un proyecto. En este modelo se describe el proyecto a construir, indicando sus dependencias y el orden en el que se construirán sus elementos, además de otras tareas como la compilación y su empaquetado. En este proyecto se empleará para la parte backend junto a Spring Boot, facilitando así la gestión de dependencias y el empaquetado de la aplicación dependiendo del profile que configuremos, entre otras cosas.

3.4.10. Apache Tomcat

Apache Tomcat [65] es un contenedor web con soporte de servlets (clases Java que amplían las capacidades de un servidor), que además puede funcionar como servidor web. En este proyecto lo emplearemos como servidor web en el despliegue del backend y del bot de Telegram.

3.4.11. PostgreSQL

PostgreSQL [38] es un sistema de base de datos relacional de código abierto. En este proyecto se creará una base de datos en *PostgreSQL* para la aplicación de tutorías, a la que se accederá mediante una API REST.

3.4.12. H2 Database

H2 Database [25] es un sistema de base de datos en memoria para Java. Normalmente, es utilizada como base de datos en los procesos de pruebas de integración, creándose a la hora de realizar los tests y destruyéndose al terminar de ejecutarlos. Para este proyecto se ha decidido usar como base de datos para el bot de Telegram, pues únicamente se creará una

entidad, cuyos registros serán temporales, pues se eliminarán una vez se termine la sesión con el bot.

3.4.13. Telegram

Telegram [51] es una de las aplicaciones de mensajería más populares en la actualidad, contando con más de 700 millones de usuarios. Tiene soporte para Android, iOS, Windows, Mac y Linux, además de disponer de versión web.

En este proyecto se utilizará su Bot API [50], que nos permitirá crear un bot a partir de otro bot, el *BotFather*, mediante unos pocos pasos. Una vez creamos el bot, *BotFather* nos dará acceso a un token de autenticación que nos permitirá profundizar en la creación de comandos del bot a partir de la API de Telegram o las librerías creadas por la comunidad y que la propia Telegram recoge en su web [49]. En este proyecto se ha escogido desarrollar la funcionalidad del bot mediante una de las librerías de Java, concretamente la librería *TelegramBots* [9], pues es con el lenguaje que se está más familiarizado.

3.4.14. Docker

Docker [16] nos permite empaquetar el software en imágenes de contenedor. Una imagen de contenedor es un paquete de software ligero, independiente y ejecutable que incluye todo lo necesario para que una aplicación pueda ser ejecutada: código, dependencias... Cuando una imagen de contenedor es ejecutada se convierte en un contenedor, el cual podremos utilizar en cualquier dispositivo que únicamente tenga instalado el *Docker Engine* [14], lo que facilita enormemente la ejecución de aplicaciones sin importar el sistema operativo y sin preocuparnos de las dependencias, pues todas estas quedan recogidas en una simple imagen. En la Figura 3.4 se puede ver de manera esquemática como quedan en un contenedor diferentes aplicaciones mediante Docker.

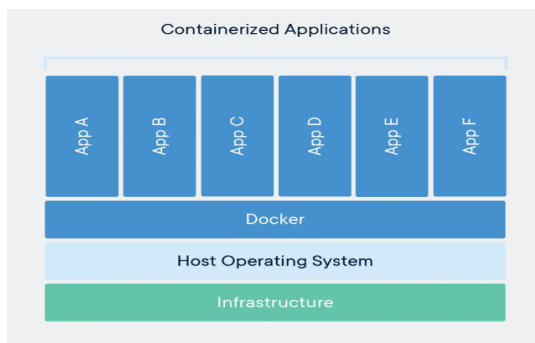


Figura 3.4: Aplicaciones “dockerizadas”

En este proyecto se utiliza Docker para el despliegue del frontend, backend y bot de Telegram.

Capítulo 4

Análisis

En este capítulo se detalla el ejercicio de análisis elaborado durante el sprint 0 y en cada una de las tareas de análisis de las historias usuario que se han desarrollado durante los diferentes sprints.

4.1. Requisitos funcionales como historias de usuario

A continuación, se muestran los requisitos funcionales identificados a lo largo del proyecto en forma de historia de usuario junto con sus puntos de historia. Además, irán desglosados según la épica a la que pertenece cada historia. Algunas historias de usuario no pertenecen a ninguna historia épica, pues nacieron de alguna necesidad específica durante el transcurso de un sprint y su descripción no se asemeja a ninguna de las historias épicas definidas en el *Product Backlog inicial*. Para que no queden sueltas se las agrupa en una historia épica creada específicamente para estas historias de usuario que han quedado sueltas, la *épica EP07*. Además, también se incluyen en esta sección, aquellas historias de usuario especificadas en algún momento del proyecto pero que después han sido eliminadas, mostrándose de un color rojo para diferenciarse del resto.

Historias de usuario de la épica EP01		
Identificador	Historia de usuario	Puntos de historia
HU01	Como usuario quiero acceder a la pantalla de inicio de sesión y a partir de mis credenciales entrar en mi cuenta de profesor o alumno.	2
HU02	Como usuario quiero terminar mi sesión para salir del sistema.	1

Tabla 4.1: Historias de usuario de la épica EP01

Historias de usuario de la épica EP02		
Identificador	Historia de usuario	Puntos de historia
HU03	Como alumno quiero buscar un profesor según diferentes filtros (por asignatura o por nombre) para poder acceder de manera más sencilla a la información del profesor.	5
HU04	Como alumno quiero ver la información detallada del profesor.	5

Tabla 4.2: Historias de usuario de la épica **EP02**

Historias de usuario de la épica EP03		
Identificador	Historia de usuario	Puntos de historia
HU18	Como equipo de desarrollo quiero idear el formulario de solicitud de tutoría sin llegar a desarrollarlo.	1
HU05	Como alumno quiero hacer una solicitud de tutoría a un profesor para facilitar la tarea de solicitar una tutoría.	5
HU06	Como alumno quiero cancelar una solicitud de tutoría para rectificar algunas condiciones de la solicitud o para simplemente cancelarla.	2
HU07	Como profesor quiero aceptar una solicitud de tutoría recibida.	2
HU08	Como profesor quiero rechazar una solicitud de tutoría recibida.	2
HU19	Como alumno quiero confirmar una solicitud de tutoría aceptada.	1
HU09	Como usuario quiero consultar las solicitudes según su estado (pendiente, aceptada, confirmada, cancelada o rechazada) u otros factores (asignatura, usuario, fecha o grupal) para conocer el estado en el que se encuentran.	3
HU21	Como profesor quiero que las solicitudes pendientes se muestren en la pantalla de inicio para acceder a ellas con mayor facilidad.	1
HU22	Como alumno quiero que las solicitudes en estado aceptada se muestren en la pantalla de inicio para acceder a ellas más fácilmente.	1
HU23	Como profesor quiero que todas las solicitudes pendientes que no hayan sido atendidas a tiempo se rechacen automáticamente para automatizar la gestión.	2

Tabla 4.3: Historias de usuario de la épica **EP03**

Historias de usuario de la época EP04		
Identificador	Historia de usuario	Puntos de historia
HU10	Como usuario quiero acceder a mi perfil para poder acceder a mi información.	2
HU11	Como profesor quiero establecer un horario de tutorías para que los alumnos puedan conocer el horario de disponibilidad de las mismas.	4
HU12	Como profesor quiero que la información sobre mi horario de tutorías se cargue mediante un fichero CSV.	3

Tabla 4.4: Historias de usuario de la época **EP04**

Historias de usuario de la época EP05		
Identificador	Historia de usuario	Puntos de historia
HU13	Como usuario quiero consultar un calendario en el que se vean reflejadas las tutorías programadas.	4
HU14	Como usuario quiero consultar las tutorías programadas (para hoy o próximamente) en forma de lista y filtrar según diversos parámetros (cancelada o no, asignatura, usuario o fecha).	2
HU15	Como usuario quiero cancelar una tutoría para evitar que esta se lleve a cabo.	2
HU24	Como usuario quiero que las próximas tutorías se muestren en la pantalla de inicio para facilitar su acceso.	1

Tabla 4.5: Historias de usuario de la época **EP05**

Historias de usuario de la época EP06		
Identificador	Historia de usuario	Puntos de historia
HU16	Como alumno quiero conocer la información de un profesor a partir de un bot de Telegram para facilitar el acceso a la información del profesor.	6
HU17	Como alumno quiero acceder a la aplicación mediante un bot de Telegram para facilitar la navegación hasta la misma.	6

Tabla 4.6: Historias de usuario de la época **EP06**

Historias de usuario de la épica EP07		
Identificador	Historia de usuario	Puntos de historia
HU20	Como miembro de la administración quiero que la población (titulaciones, asignaturas, usuarios y su información) de la aplicación se carguen mediante un fichero.	3
HU25	Como equipo de desarrollo quiero “dockerizar” las aplicaciones para poder ejecutarlas en diferentes entornos sin preocuparnos por las dependencias.	6

Tabla 4.7: Historias de usuario de la épica **EP07**

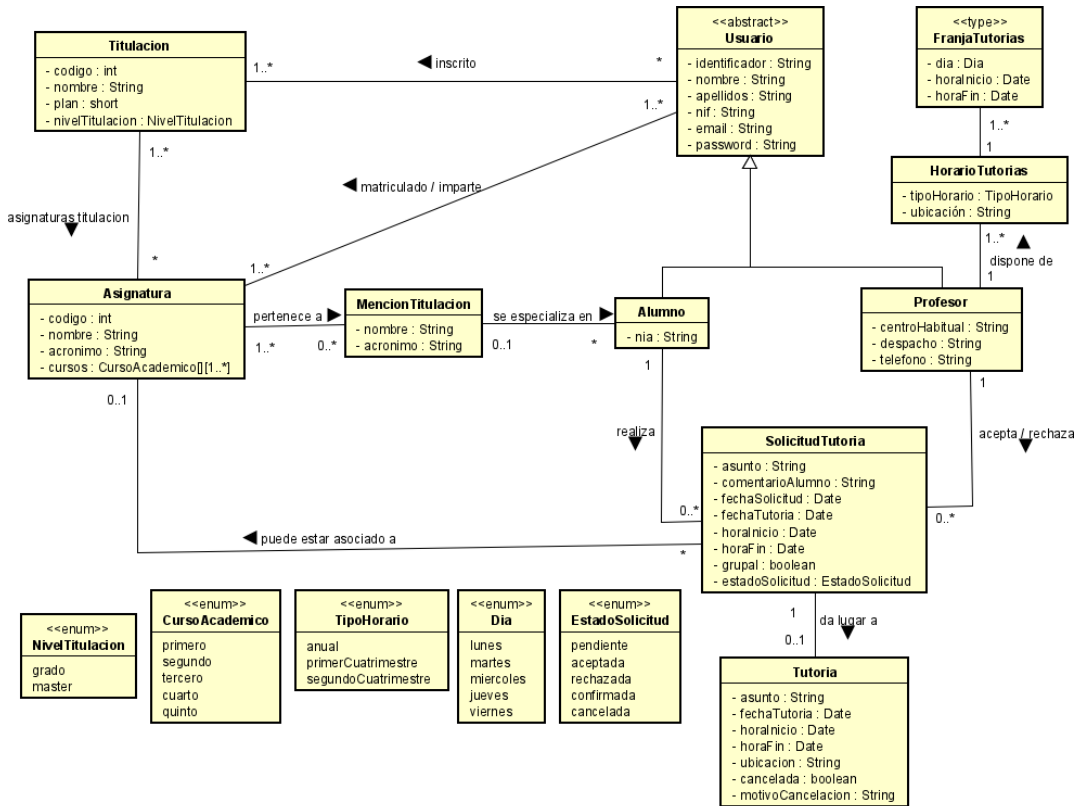


Figura 4.1: Modelo de dominio inicial

4.2. Modelo de dominio inicial

Durante el sprint 0 (ver Sección 7.1.1) se elabora el modelo de dominio inicial a partir de las historias de usuario identificadas en ese momento: de la *HU01* a la *HU17*. Este modelo será modificado posteriormente debido a que no se habían tenido en cuenta todos los detalles necesarios a la hora de modelarlo. La razón de estos cambios se explican en la Sección 7.1.4 del Capítulo 7, dedicado al seguimiento del proyecto. El modelo de dominio inicial puede observarse en la Figura 4.1.

4.3. Modelo de dominio final

Para corregir el problema en la elaboración del modelo de dominio inicial, cuya causa y consecuencias se detallan en la Sección 7.1.4 del capítulo dedicado al *seguimiento del proyecto*, se crea un nuevo modelo que corrige al modelo inicial. En este nuevo modelo de dominio final, el cual puede observarse en la Figura 4.2, se incluyen las novedades en un color azul. En cuanto a las clases, se añaden dos, la clase *CursoAcademico*, anteriormente especificada como una

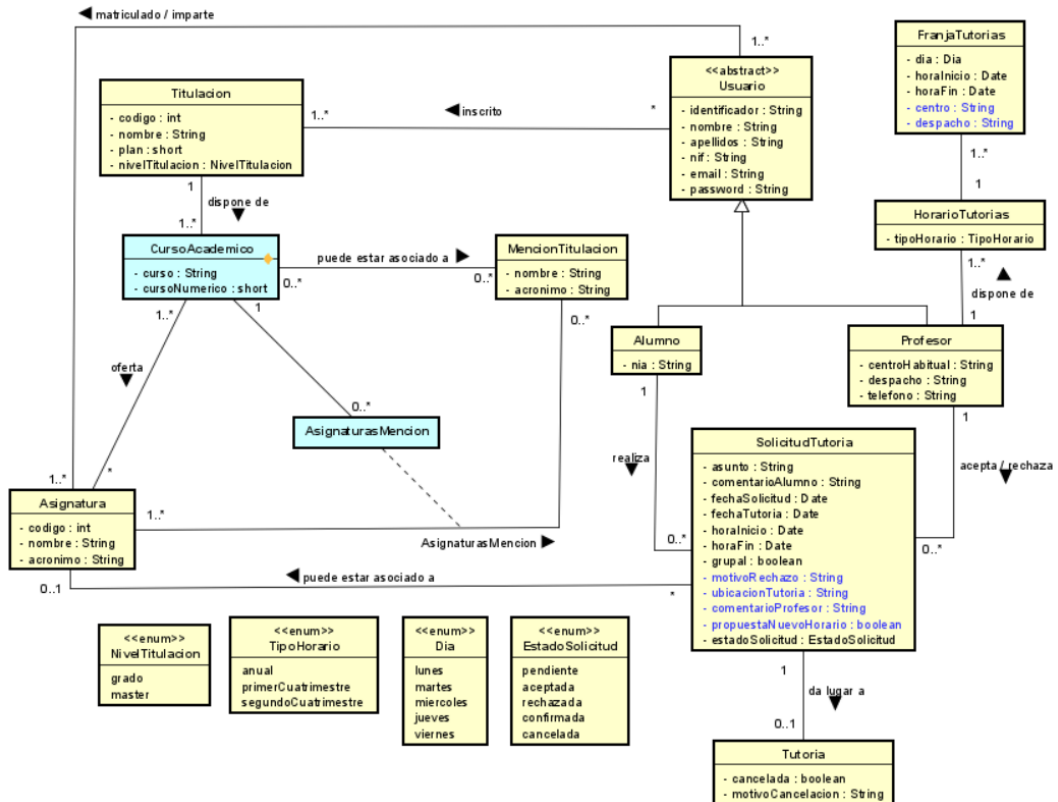


Figura 4.2: Modelo de dominio final

enumeración, y la clase asociativa *AsignaturasMencion*, dada por la asociación entre la clase *Asignatura* y la clase *MencionTitulacion*. La adición de la clase *CursoAcademico*, nos permite tener un dominio del problema mucho más amplio, incluyendo todo tipo de titulación con sus respectivos cursos, en lugar de pensar únicamente en el Grado en Ingeniería Informática y sus cuatro cursos, y su asociación con la clase asociativa *AsignaturasMencion* permitirá establecer si un curso está asociado a una mención académica o no. En cuanto a los atributos, se añaden nuevos campos para modelar nuevas posibilidades que no se habían tenido en cuenta en un primer momento, y, además, se eliminan los campos repetidos en la clase *Tutoria* respecto de la clase *SolicitudTutoria*, pues estos pueden derivarse. También se elimina la relación entre la clase *Alumno* y la clase *MencionTitulacion* pues están relacionadas por transitividad.

4.4. Modelo de proceso de negocio

En la Figura 4.3 se presenta el diagrama de actividades que muestra el proceso de negocio más representativo del proyecto, **el proceso de solicitar una tutoría**. El proceso se muestra en todas sus fases, desde el momento en el que el alumno realiza la solicitud hasta el momento

en el que el propio alumno confirma su asistencia. Tal y como se ve en la figura, en este proceso participa un alumno, que es el solicitante, y un profesor, que es quien recibe la solicitud y decide que hacer con ella.

En un primer lugar, el alumno selecciona el profesor destinatario y selecciona la fecha, hora y duración estimada de la tutoría. A continuación, introduciría los detalles de la solicitud (asignatura, si es grupal o no, asunto y/o comentario). Procede a tratar de realizar la solicitud, lugar donde se produce una bifurcación:

- si la solicitud está fuera del horario de tutorías del profesor, se muestra un mensaje de advertencia, donde el alumno puede decidir si continuar o cancelar el proceso.
- si la solicitud está dentro del horario de tutorías del profesor se continua con el proceso.

Continuando con el camino principal, vuelven a surgir dos posibles opciones:

- si la solicitud es para el día siguiente o si siendo fin de semana (incluyendo el viernes) se solicita una solicitud para el lunes siguiente, se muestra un mensaje de advertencia, pues la solicitud puede que no sea atendida a tiempo, por lo que es alumno quien decide si continuar o cancelar el proceso.
- por el contrario, si hay espacio suficiente para que la solicitud sea atendida, se continua con el proceso.

Realizadas estas comprobaciones, se procede a realizar la solicitud, volviendo a existir dos posibles casos:

- si el alumno tiene otras solicitudes en el espacio de tiempo de la nueva solicitud o si existen tutorías en el espacio de tiempo de la nueva solicitud para el alumno o el profesor, se muestra un mensaje de error y se cancela el proceso.
- por el contrario, si hay espacio para la nueva solicitud, se crea una nueva solicitud en el estado *pendiente*.

A continuación, es cuando entra en escena el rol del profesor, que mira sus solicitudes pendientes y selecciona una, teniendo, ahora, dos posibles opciones: aceptarla o rechazarla. Realmente, también existe la opción de no realizar ninguna acción, situación que provoca el nacimiento de la historia de usuario *HU23* (ver Sección 4.1).

En caso de que el profesor rechace la solicitud, tiene que escribir un motivo. Tras esto la solicitud pasa al estado *rechazada* y se notifica. En caso contrario, el profesor acepta la solicitud donde existen dos posibles opciones:

- si no hay espacio de tiempo para aceptar la nueva tutoría, se puede cancelar el proceso, manteniéndose la tutoría en estado *pendiente*, o se da la posibilidad de ofrecer al alumno un nuevo horario, con lo que se continuaría el proceso.

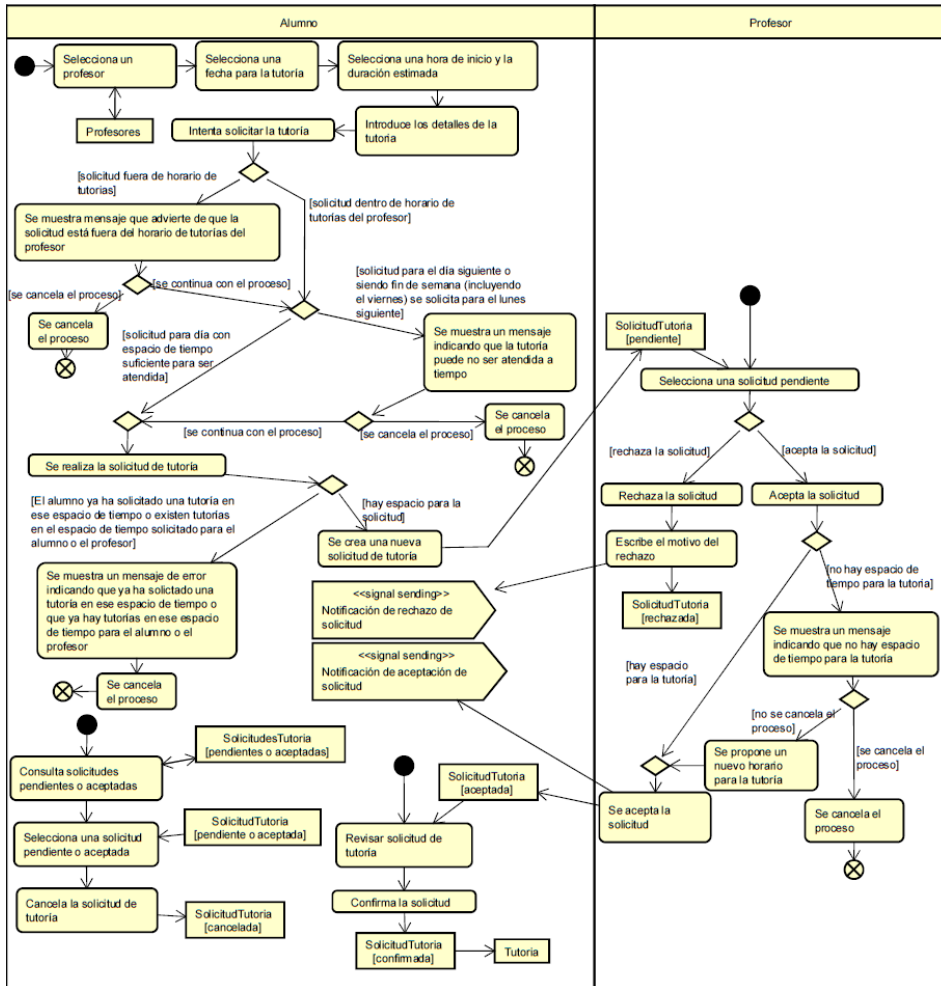


Figura 4.3: Modelo de proceso de negocio - Solicitar tutoría

- si hay espacio para aceptar la nueva tutoría, se continuaría con el proceso.

Volviendo al proceso principal, la solicitud se aceptaría pasando al estado *aceptada* y se notificaría al alumno. Por último, el alumno revisa sus solicitudes, seleccionando la solicitud en estado *aceptada*, donde confirma la asistencia a la tutoría, la cual pasa al estado *confirmada* y se crea una **nueva tutoría**.

También, existe el caso en el que el alumno revise sus solicitudes en estado *aceptada* o *pendiente* y que decida cancelar la solicitud, pasando ésta al último estado posible, *cancelada*.

4.5. Modelado dinámico

En esta sección se muestra el modelado de una *Solicitud de tutoría* como una máquina de estados. El resultado de los estados en los que puede encontrarse un *solicitud de tutoría* puede verse en la Figura 4.4.

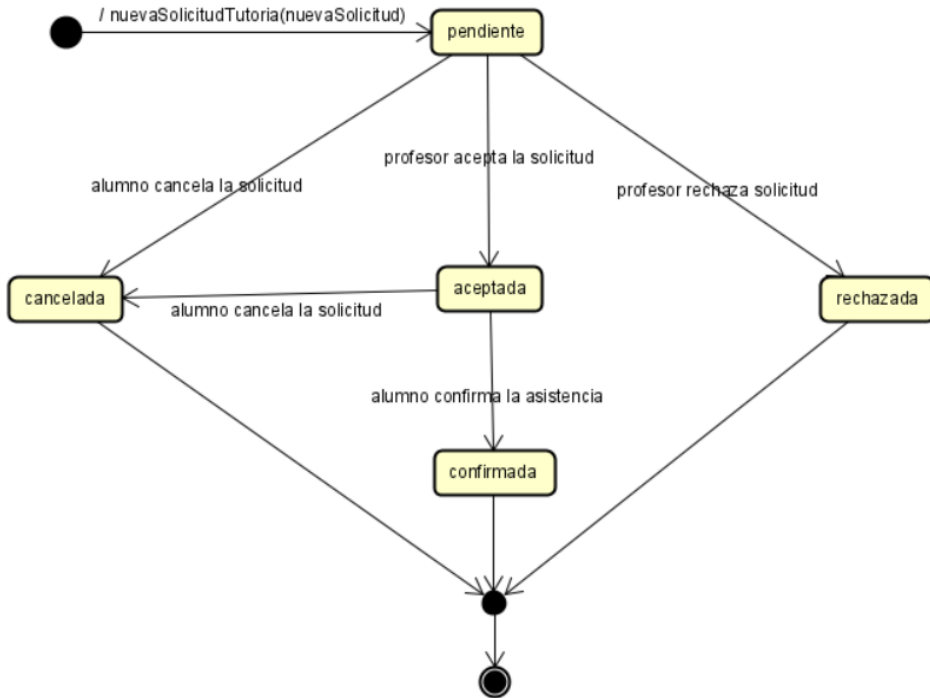


Figura 4.4: Máquina de estados - Solicitud Tutoría

Capítulo 5

Diseño

En este capítulo se presenta en detalle el proceso de diseño realizado a lo largo del proyecto, después de las tareas de análisis, tanto en el período de planificación como durante la elaboración de las tareas de diseño de los diferentes sprints.

5.1. Diseño de la interfaz de usuario

En esta sección se muestran los prototipos creados para las diferentes interfaces de usuario de la aplicación web. Dichas interfaces se han elaborado con la herramienta de wireframes *Balsamiq*. Un *wireframe* es un esquema de la estructura visual de una aplicación, en este caso, una web. Estos prototipos se han ido creando y completando durante las tareas de diseño de cada historia de usuario, dando como resultado final los prototipos que pueden verse desde la Figura 5.1 hasta la 5.37. Cada figura viene acompañada de un pie de imagen con el formato: “*Vista de <nombre-vista>:<posible-aclaración> - Rol de usuario*”.

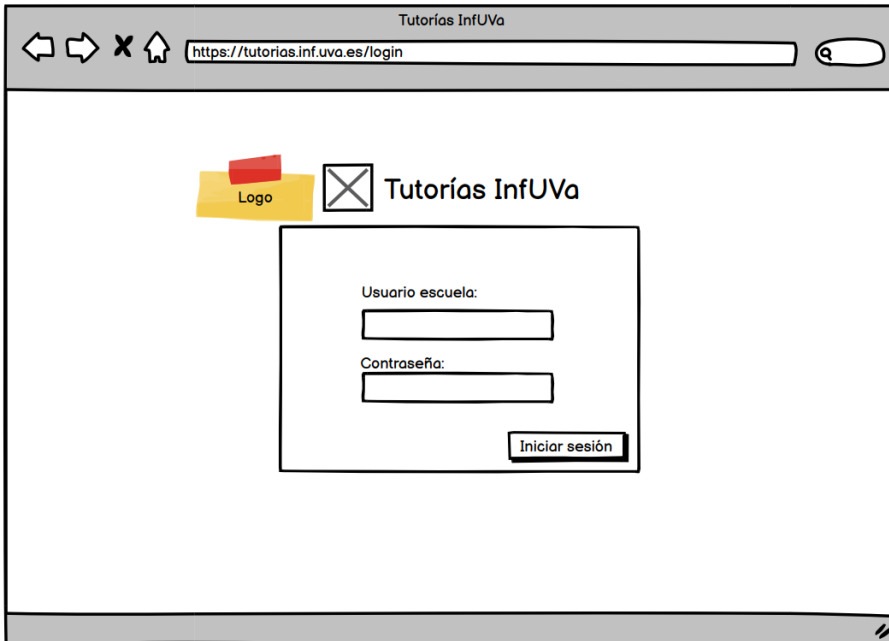


Figura 5.1: Vista de Inicio de sesión - Usuario no identificado

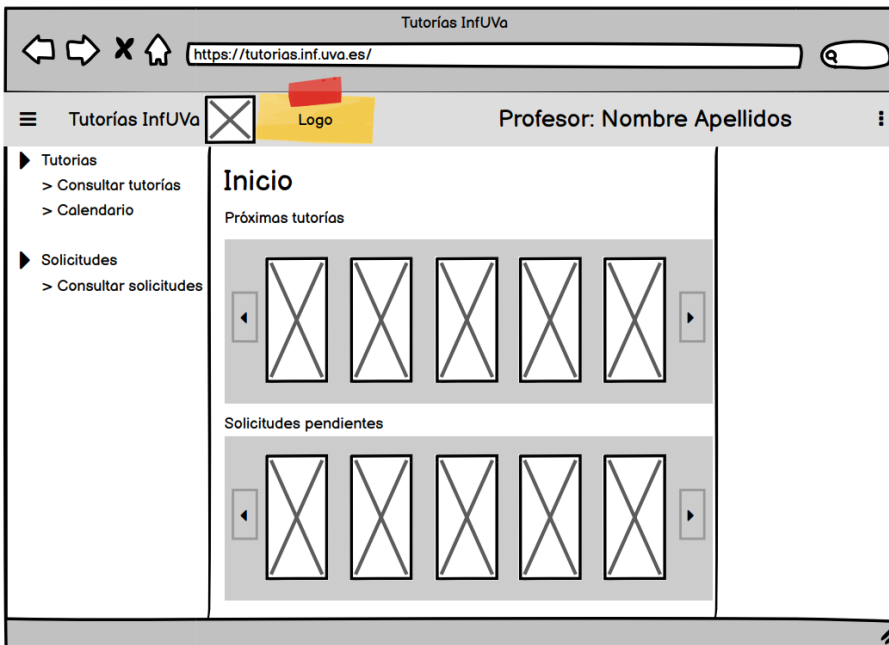


Figura 5.2: Vista de Inicio - Profesor

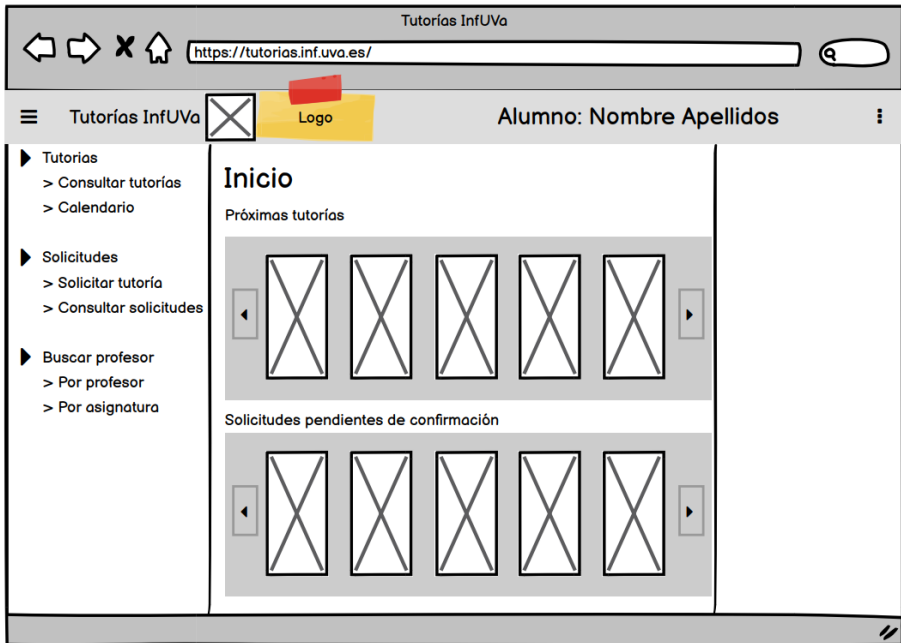


Figura 5.3: Vista de Inicio - Alumno

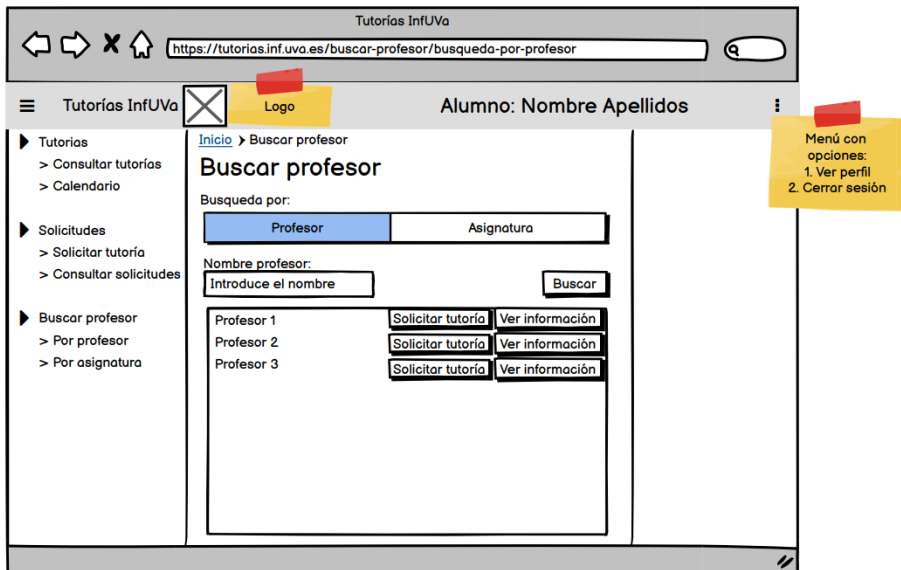


Figura 5.4: Vista de Buscar profesor por nombre - Alumno

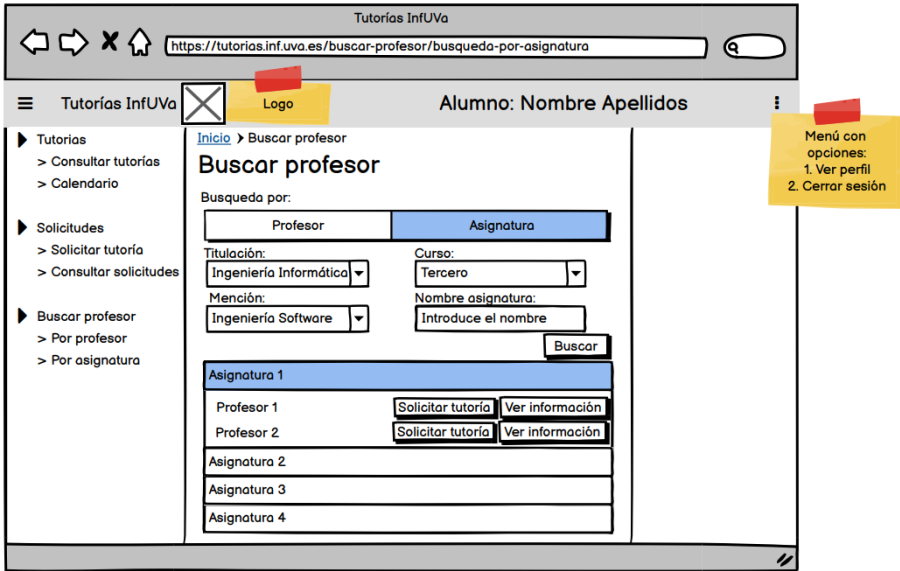


Figura 5.5: Vista de Buscar profesor por asignatura - Alumno

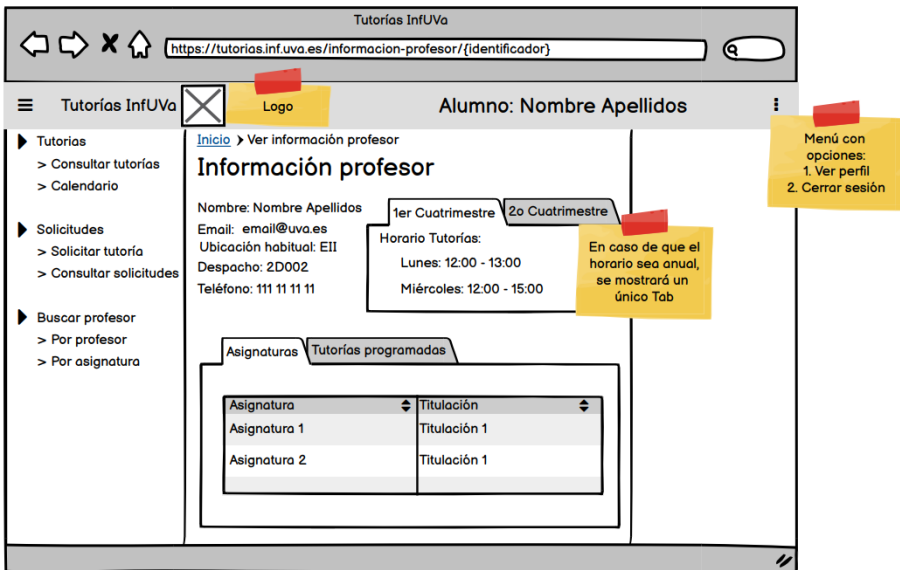


Figura 5.6: Vista de Información profesor: visualización de asignaturas - Alumno

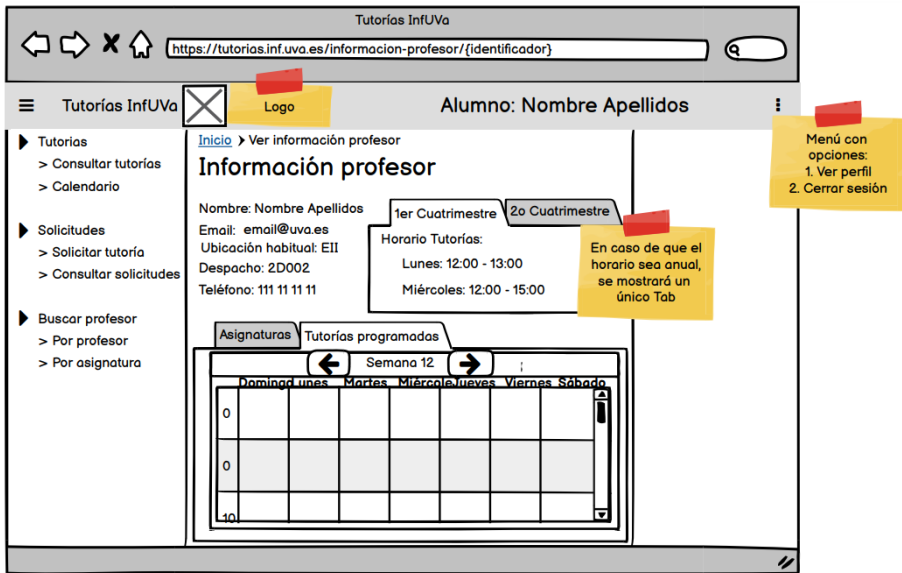


Figura 5.7: Vista de Información profesor: visualización de tutorías programadas - Alumno

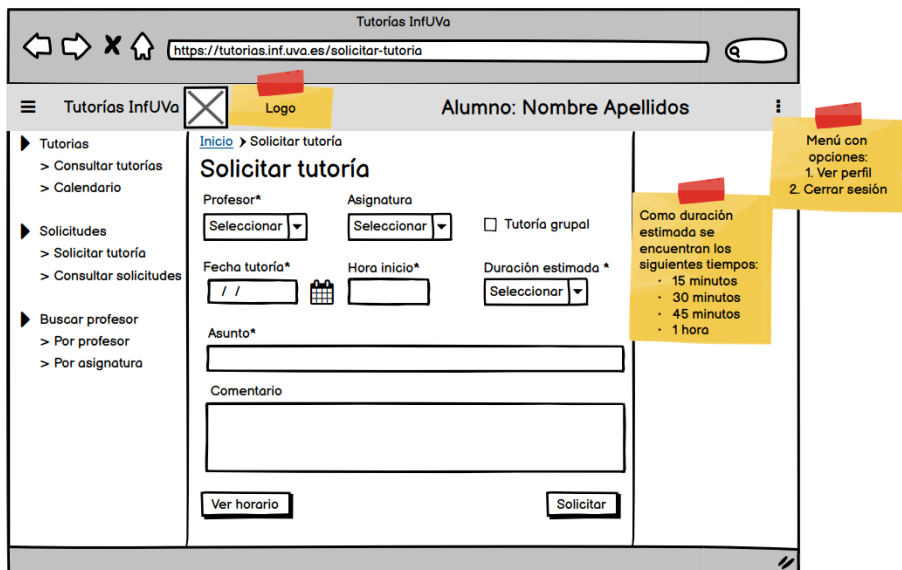


Figura 5.8: Vista de Solicitar tutoría - Alumno

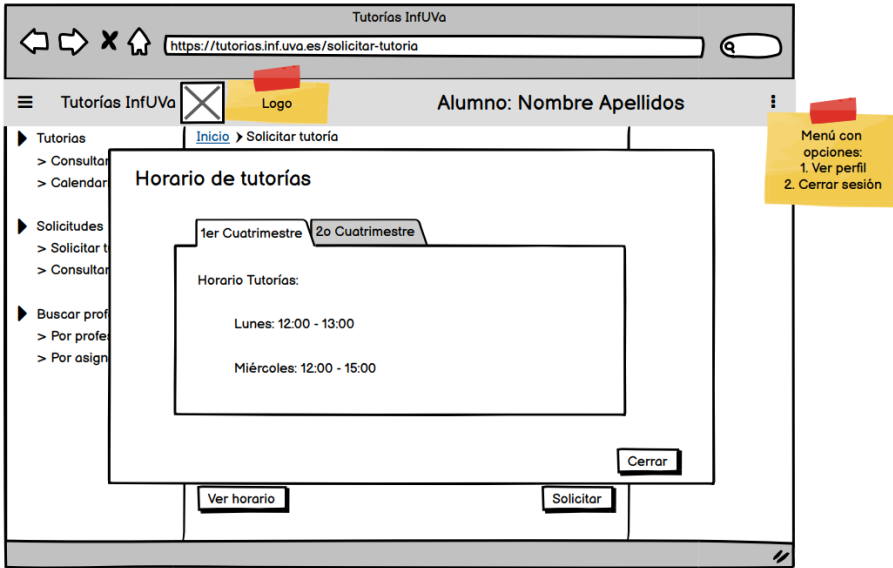


Figura 5.9: Vista de Solicitar tutoría: ver horario de tutorías del profesor seleccionado - Alumno

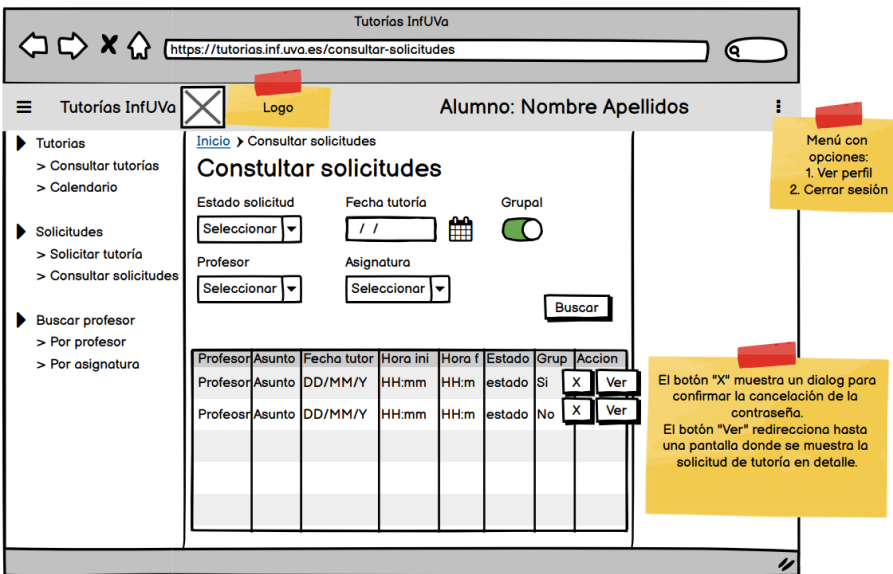


Figura 5.10: Vista de Consultar solicitudes - Alumno

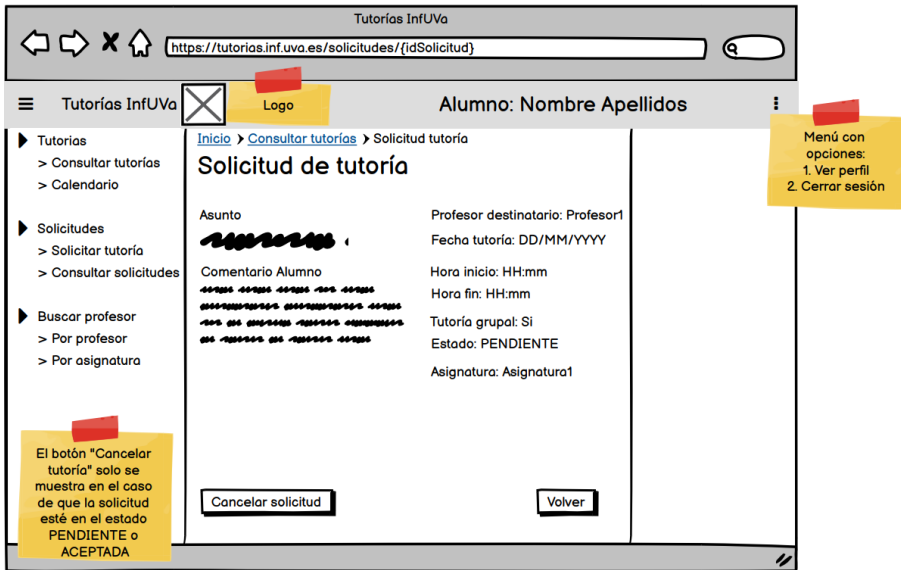


Figura 5.11: Vista de Solicitud de tutoría: en estado pendiente - Alumno

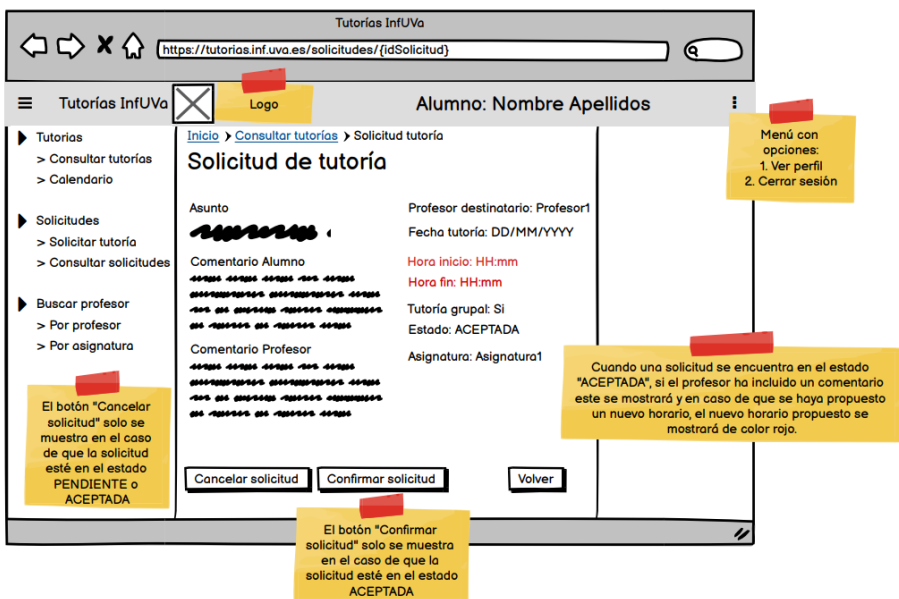


Figura 5.12: Vista de Solicitud de tutoría: en estado aceptada - Alumno

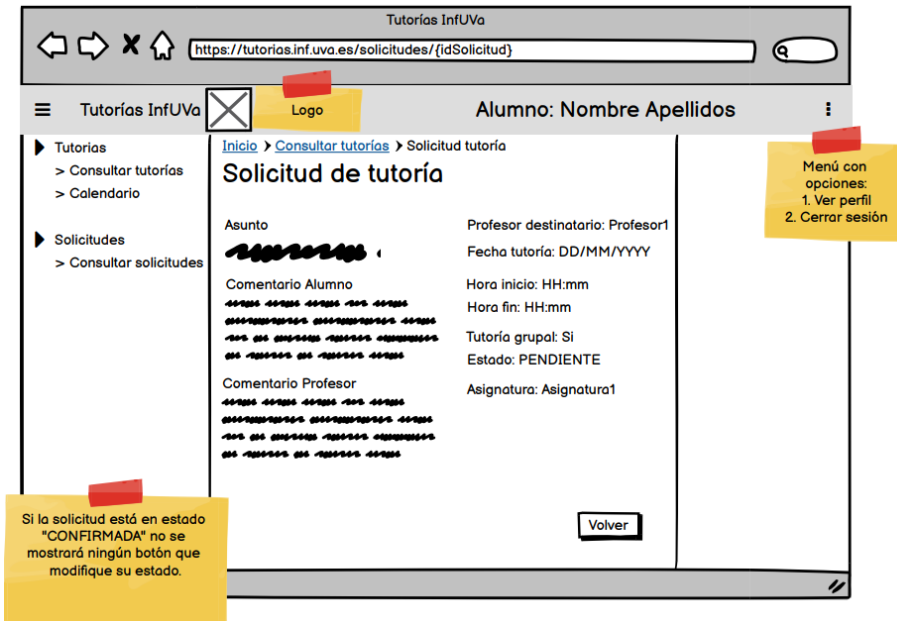


Figura 5.13: Vista de Solicitud de tutoría: en estado confirmada - Alumno

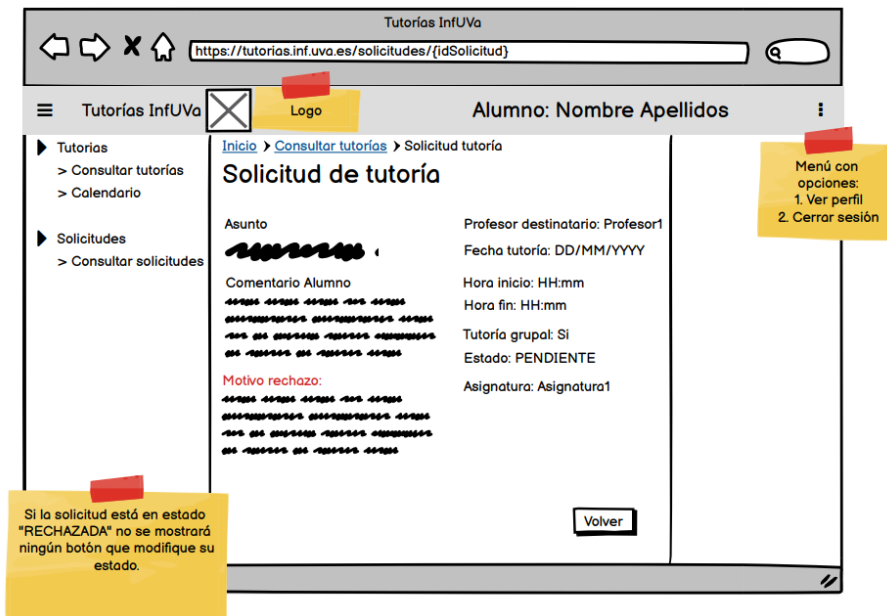


Figura 5.14: Vista de Solicitud de tutoría: en estado rechazada - Alumno

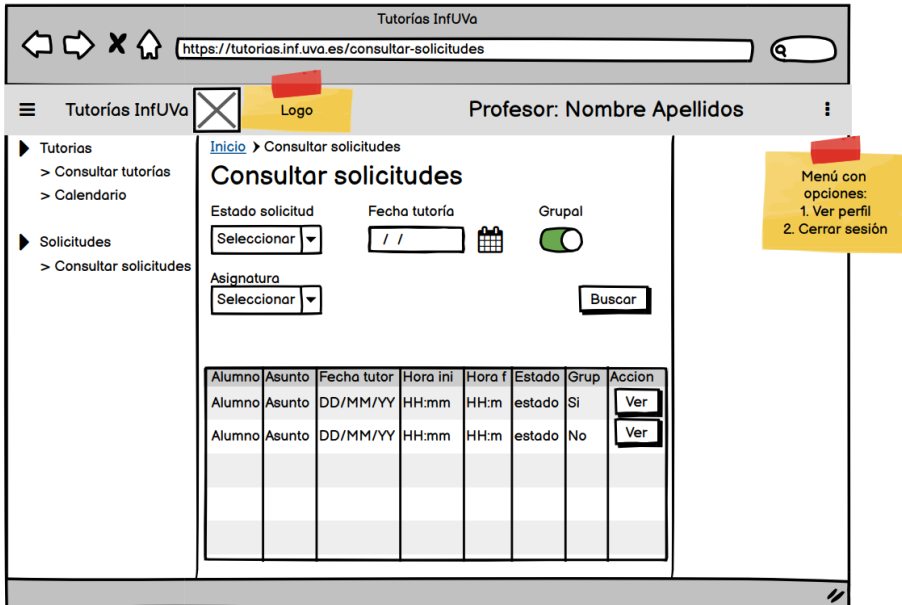


Figura 5.15: Vista de Consultar solicitudes - Profesor

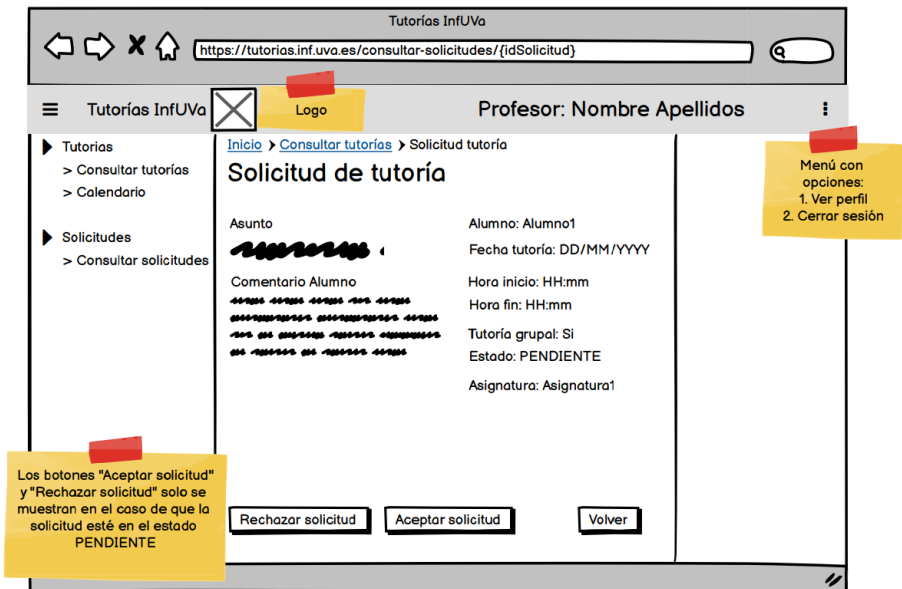


Figura 5.16: Vista de Solicitud de tutoría: en estado pendiente - Profesor

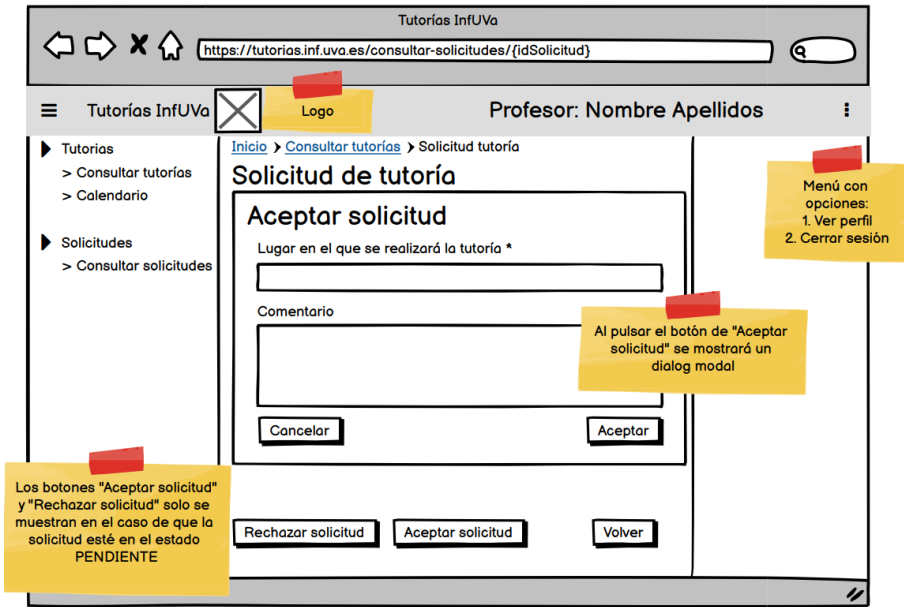


Figura 5.17: Vista de Solicitud de tutoría: aceptar solicitud - Profesor

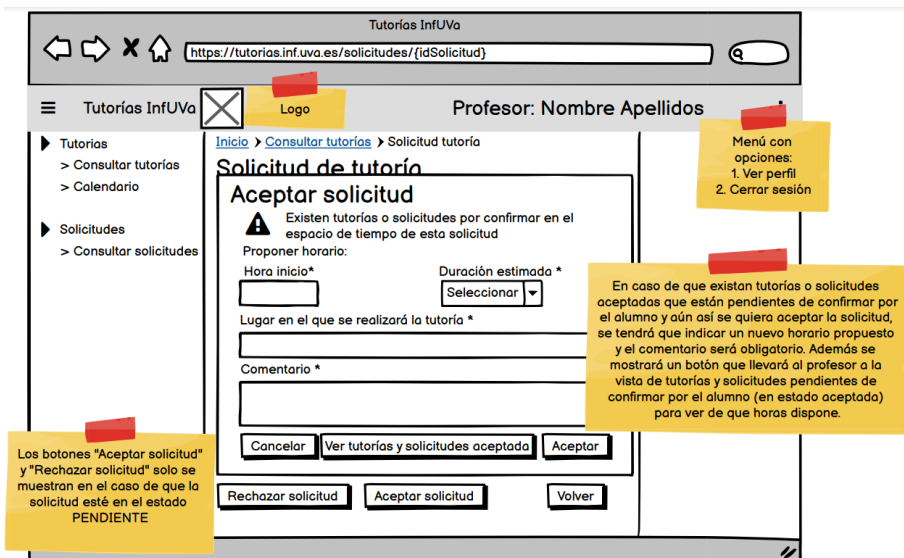


Figura 5.18: Vista de Solicitud de tutoría: aceptar solicitud con propuesta de nuevo horario - Profesor

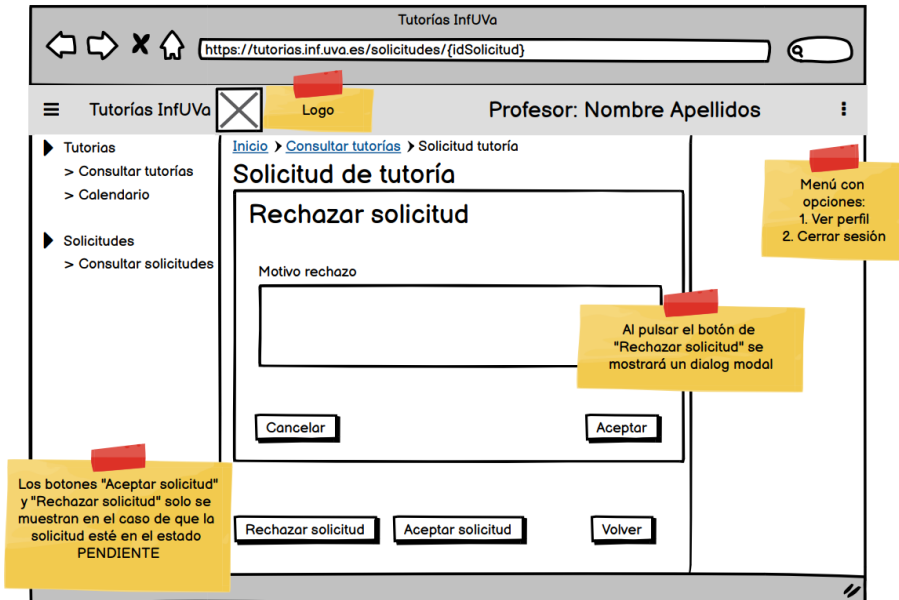


Figura 5.19: Vista de Solicitud de tutoría: rechazar solicitud - Profesor

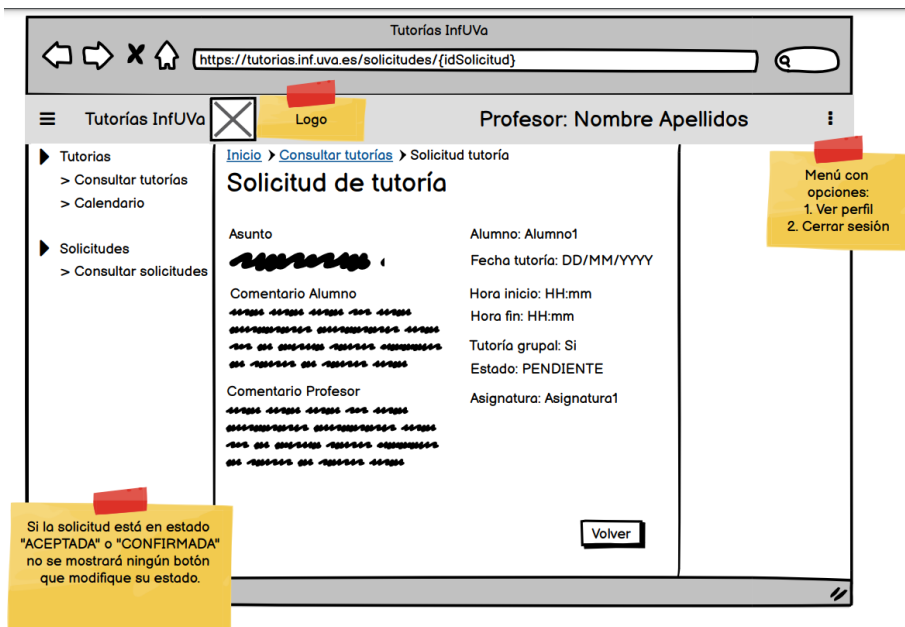


Figura 5.20: Vista de Solicitud de tutoría: en estado aceptada o confirmada - Profesor

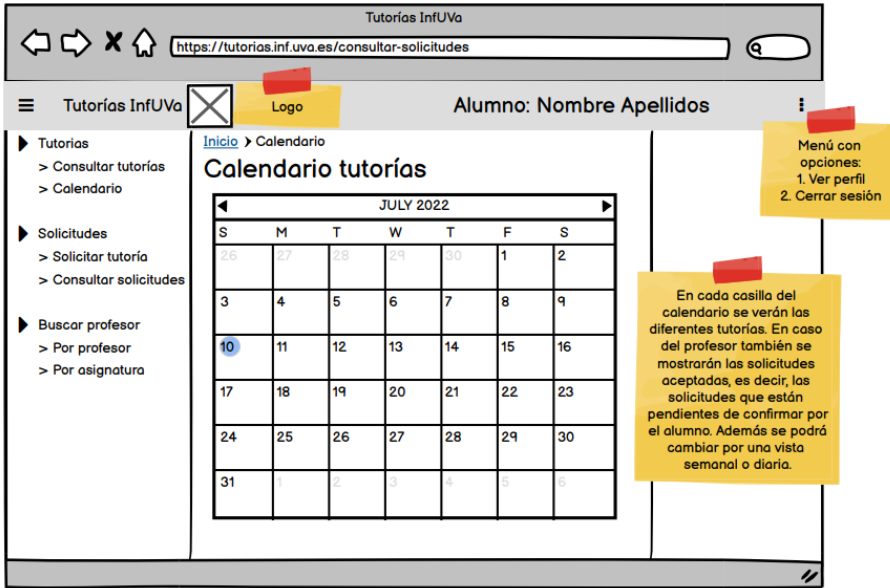


Figura 5.21: Vista de Calendario tutorías - Usuario

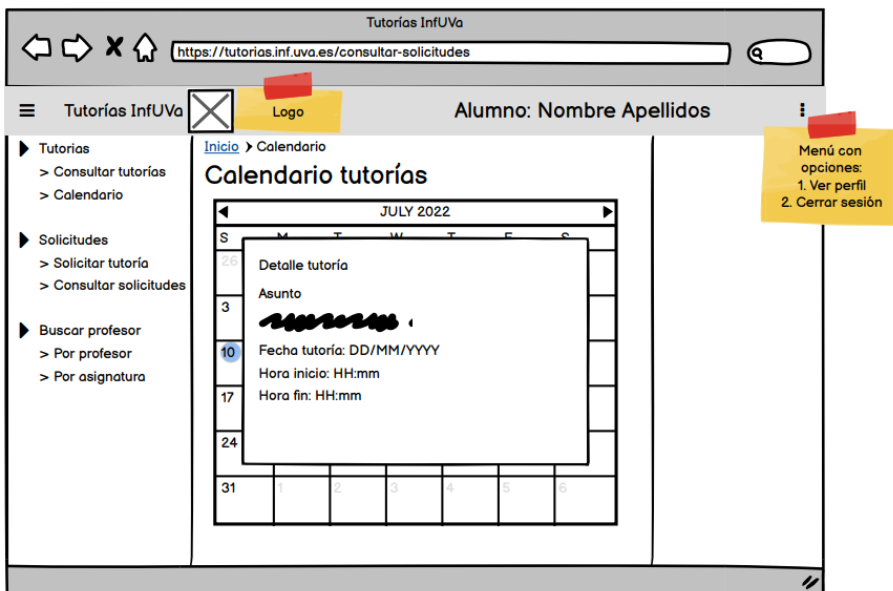


Figura 5.22: Vista de Calendario tutorías: ver detalle tutoría - Usuario

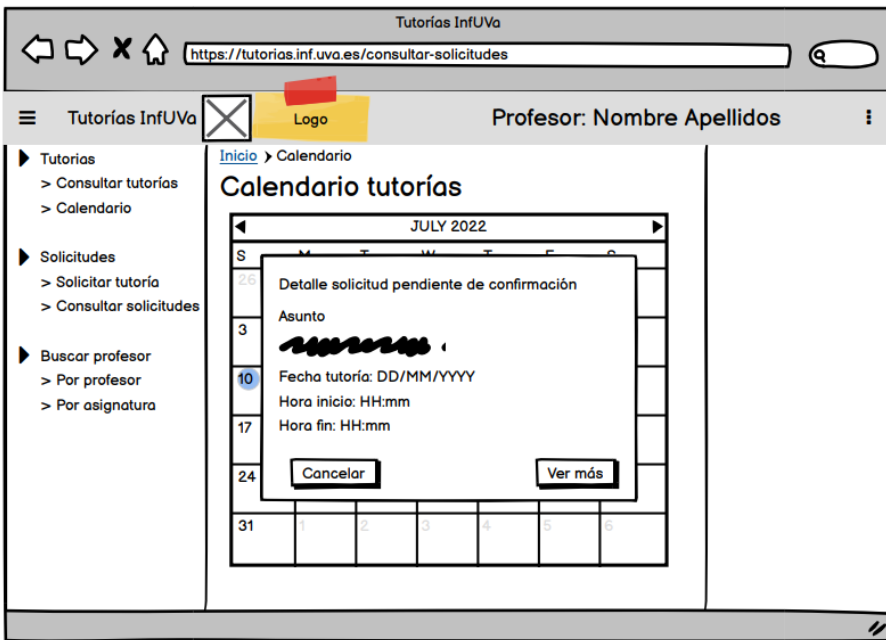


Figura 5.23: Vista de Calendario tutorías: ver detalle solicitud aceptada (pendiente de confirmar) - Profesor

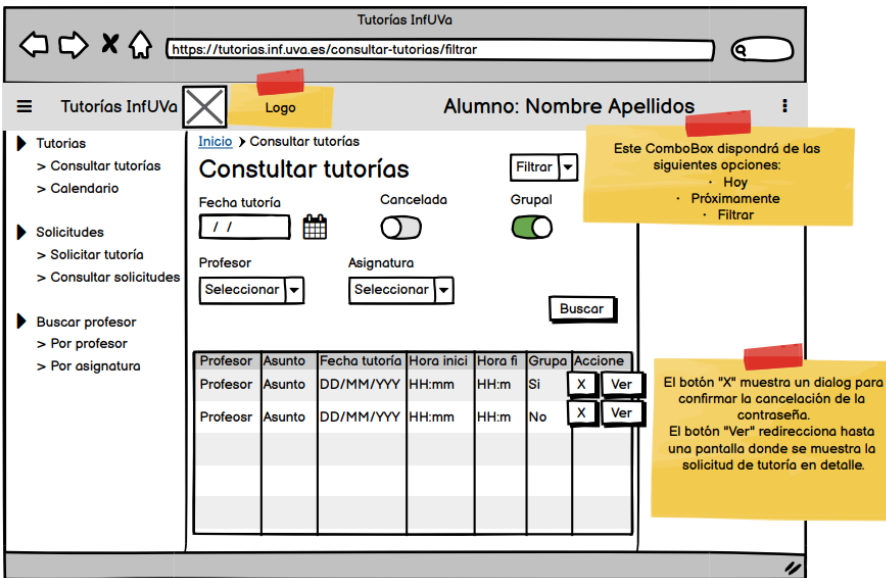


Figura 5.24: Vista de Consultar tutorías: mediante filtros - Alumno

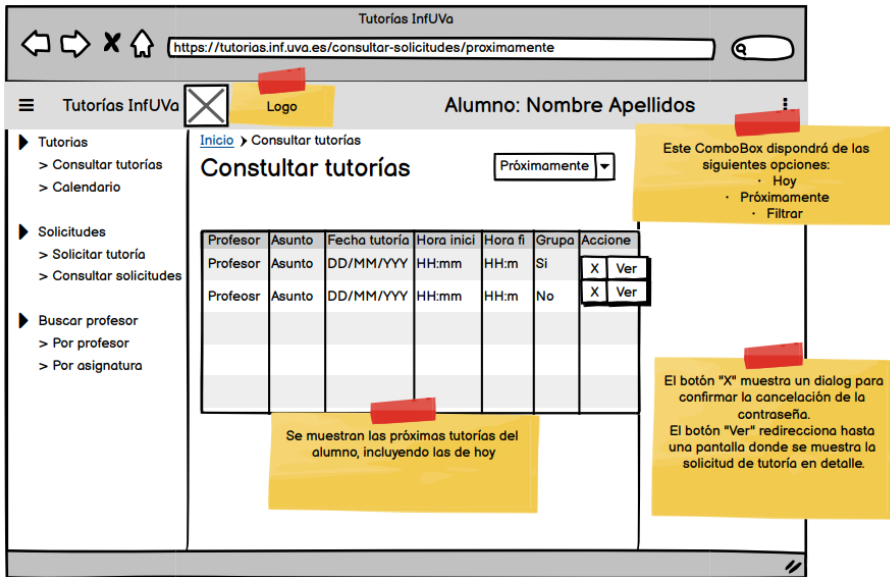


Figura 5.25: Vista de Consultar tutorías: próximamente - Alumno

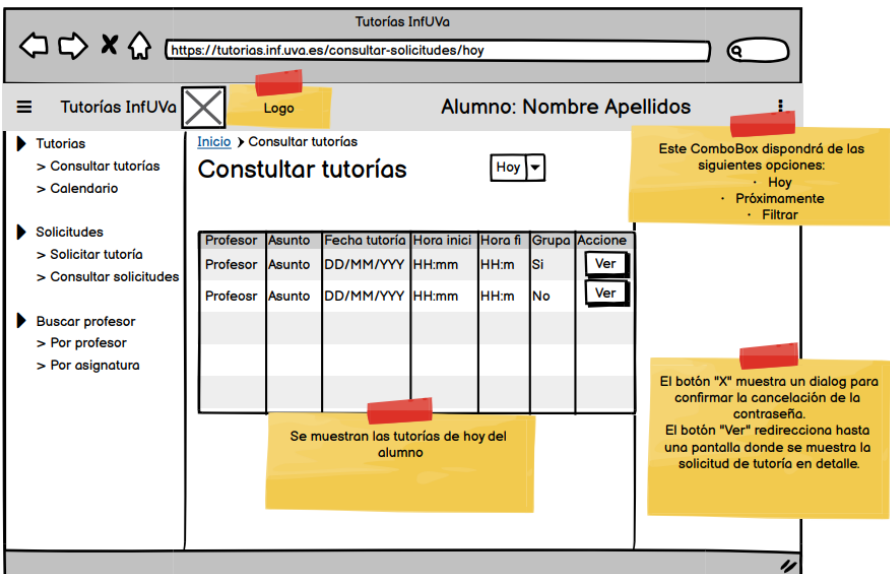


Figura 5.26: Vista de Consultar tutorías: hoy - Alumno

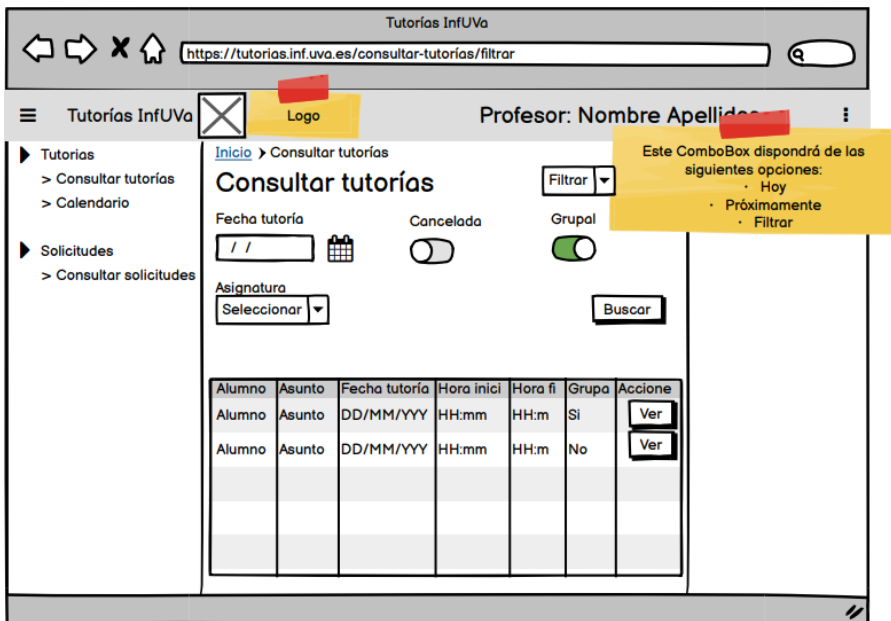


Figura 5.27: Vista de Consultar tutorías: mediante filtros - Profesor

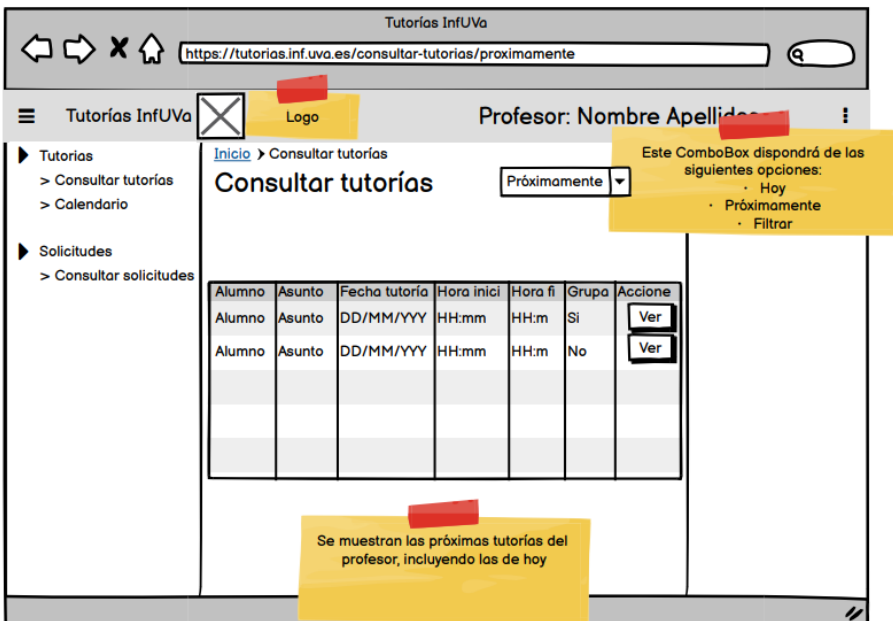


Figura 5.28: Vista de Consultar tutorías: próximamente - Profesor

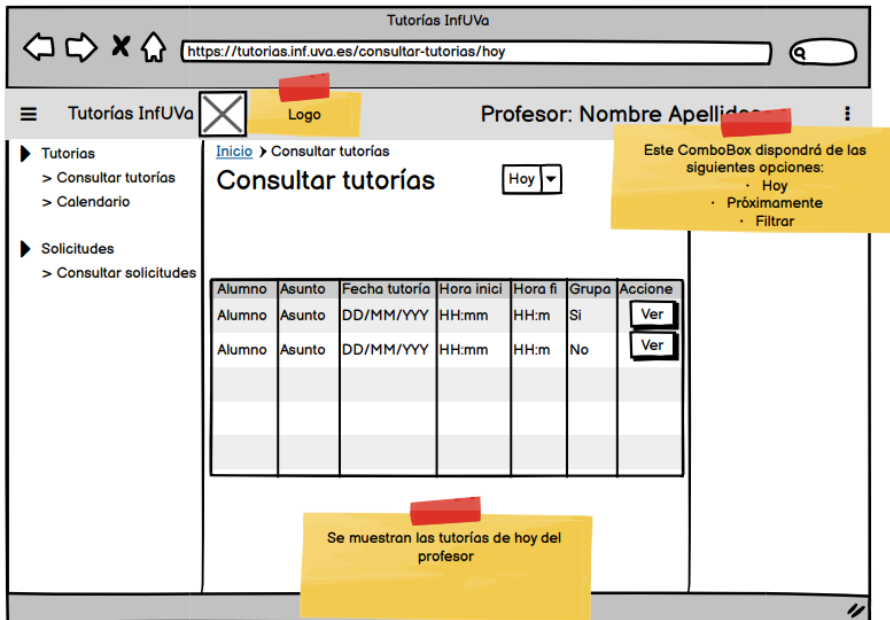


Figura 5.29: Vista de Consultar tutorías: hoy - Profesor

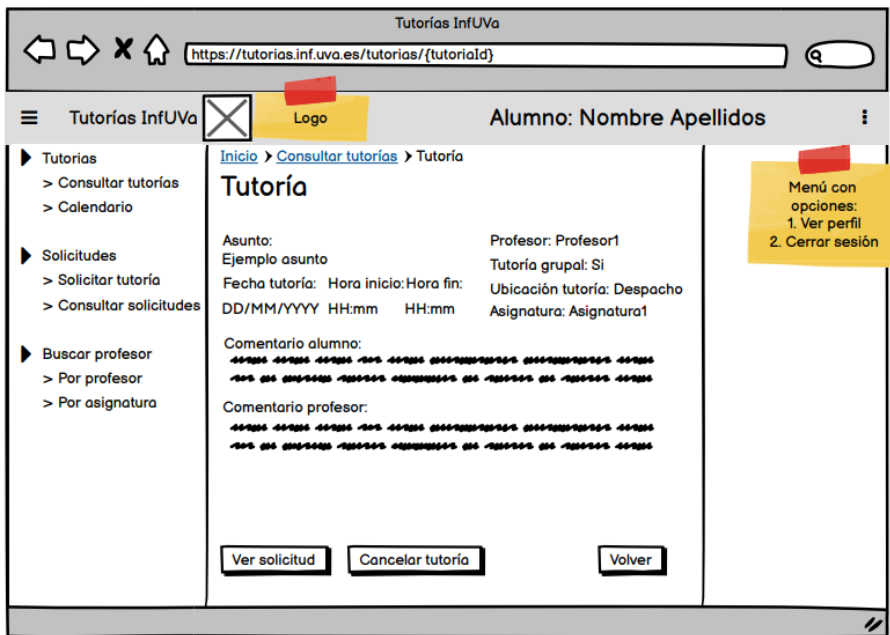


Figura 5.30: Vista de Tutoría - Alumno

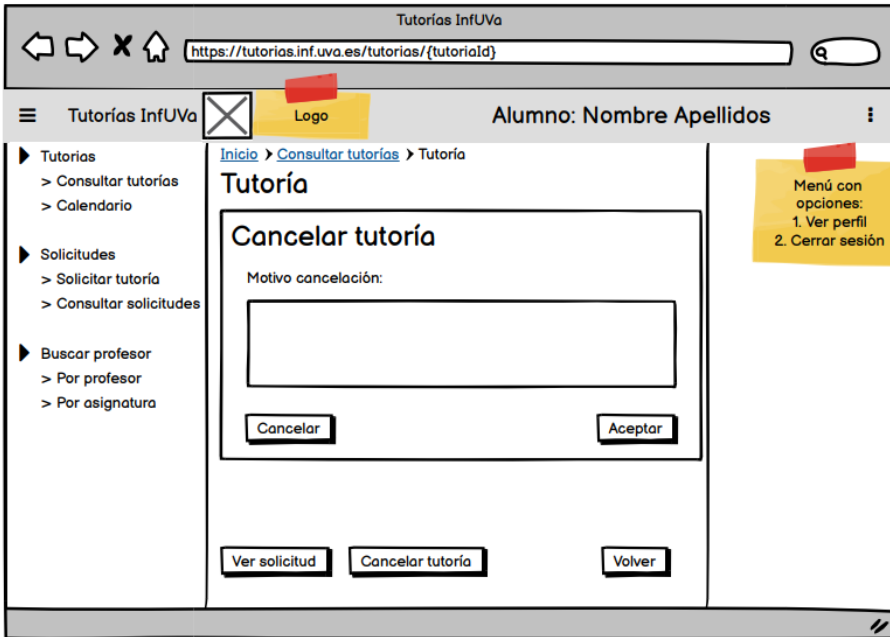


Figura 5.31: Vista de Tutoría: cancelar tutoría - Usuario

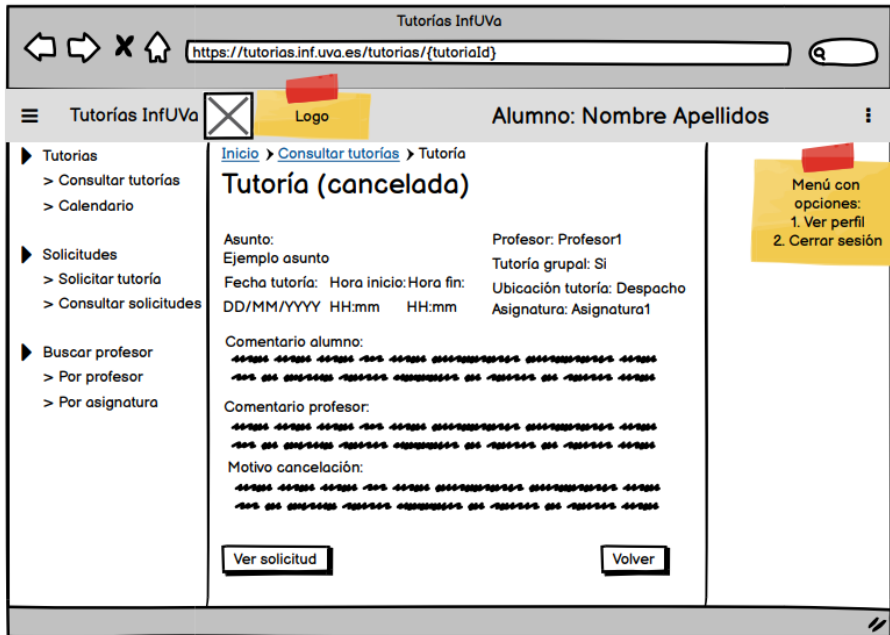


Figura 5.32: Vista de Tutoría (cancelada) - Alumno

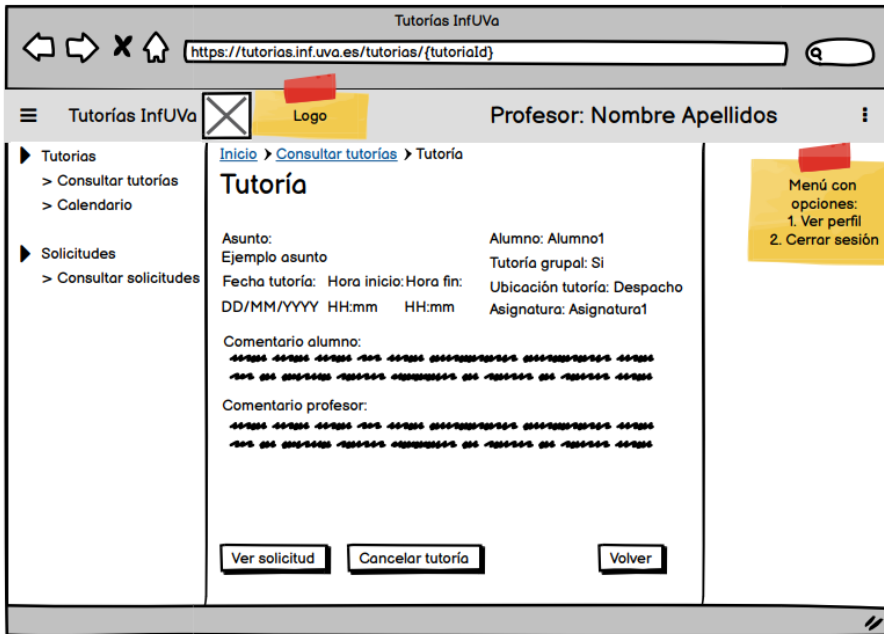


Figura 5.33: Vista de Tutoría - Profesor

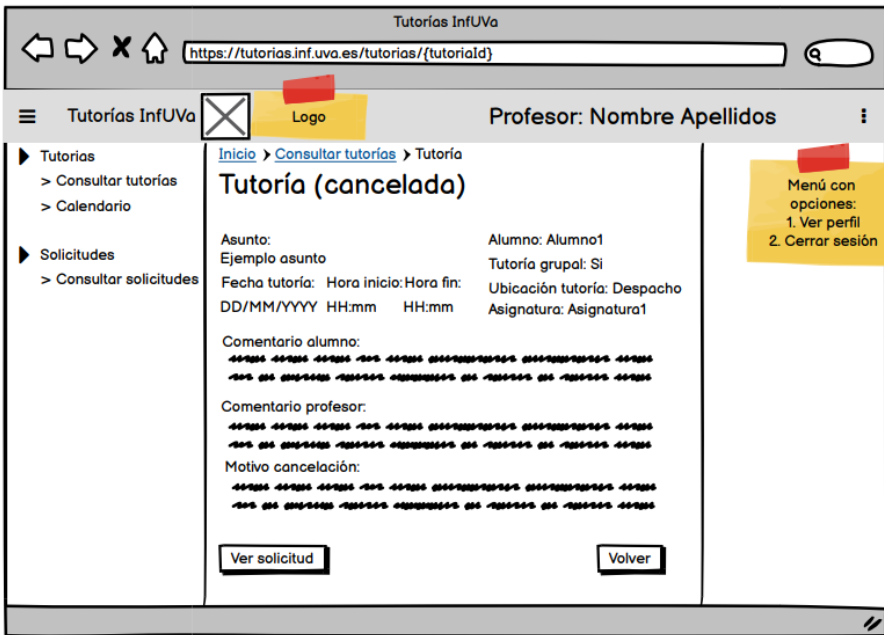


Figura 5.34: Vista de Tutoría (cancelada) - Profesor

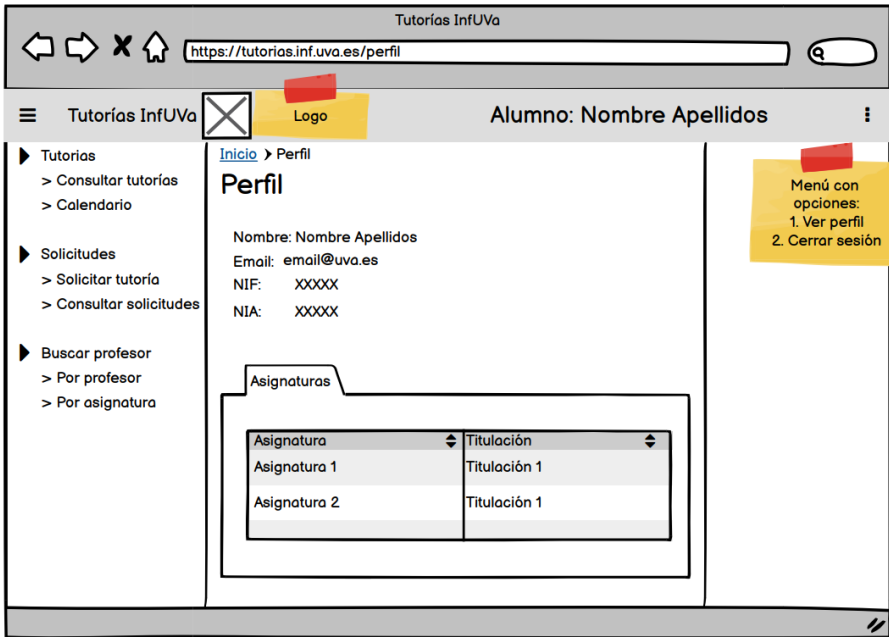


Figura 5.35: Vista de Perfil - Alumno

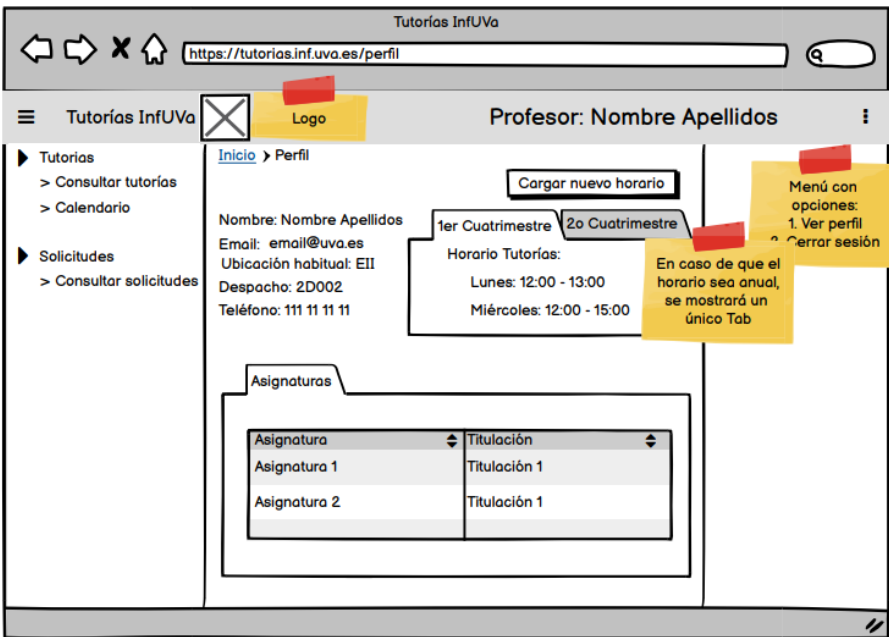


Figura 5.36: Vista de Perfil - Profesor

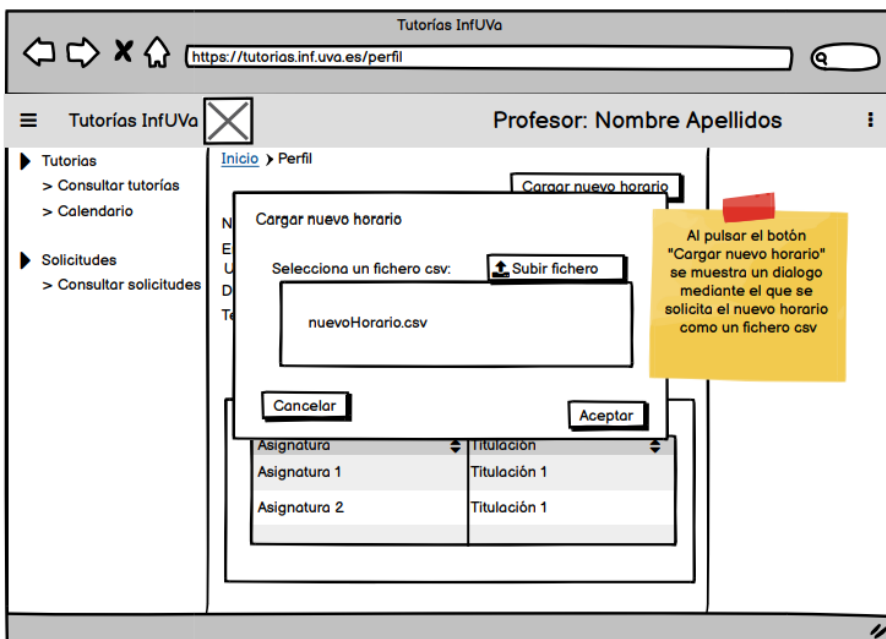


Figura 5.37: Vista de Perfil: cargar nuevo horario - Profesor

5.2. Diseño arquitectónico

5.2.1. Arquitectura en el Cliente (MVVM)

Model View ViewModel [61] es un patrón arquitectónico en el que se desacopla lo máximo posible la vista de la lógica de negocio, y la vista es capaz de gestionar las acciones del usuario. MVVM consta de los siguientes elementos:

- **Modelo (Model)**. Representa la lógica de negocio.
- **Vista (View)**. Representa la información que se muestra al usuario.
- **Modelo de vista (ViewModel)**. Actúa como intermediario entre la vista y el modelo, dicha intermediación se produce mediante data binding [4].



Figura 5.38: Funcionamiento del patrón arquitectónico MVVM. Tomado de [10]

En Angular, que es el framework con el que se va a desarrollar el frontend del proyecto, cada elemento se corresponde de la siguiente manera: la vista se corresponde con los archivos html, el modelo se corresponde con los objetos de dominio y otros elementos del framework como los servicios, y el modelo de vista con las clases TypeScript de los componentes.

En cuanto a la arquitectura general de la aplicación, ésta se divide en 4 paquetes principales:

- **components:** en este paquete se encuentran todos los componentes de la aplicación y se corresponde con el *Modelo de Vista* y la *Vista* del patrón arquitectónico MVVM, ya que los componentes contienen las clases TypeScript y las plantillas HTML. En la Sección 5.5 se muestra en detalle el desarrollo basado en componentes de la aplicación.
- **models:** en este paquete se encuentran las clases del dominio de la aplicación. Se corresponde con el *Modelo* de la patrón arquitectónico MVVM. En este paquete se encuentra el subpaquete **enums**, donde se alojan las enumeraciones que se han desarrollado.
- **services:** en este paquete se encuentran los servicios, los cuales también son parte del *Modelo* de la arquitectura MVVM. En estos servicios es donde se consumirá la *API REST* (ver Sección 5.4) mediante peticiones HTTP.

- **utils:** en este paquete se encuentran las clases de utilidades recurrentes en la aplicación, como constantes, entre otros.

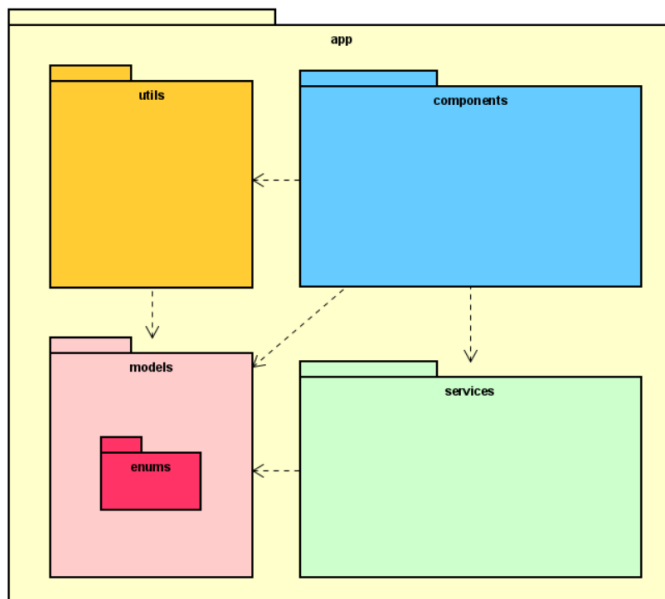


Figura 5.39: Arquitectura general del frontend

5.2.2. Arquitectura en el Servidor

En la parte del servidor, para el desarrollo de la API REST, se ha elegido como patrón arquitectónico una **arquitectura cerrada** compuesta de 3 capas. Una arquitectura cerrada [23] se caracteriza por que cada capa solamente utiliza los servicios de su capa y de la capa inmediatamente inferior, minimizando, así, las dependencias entre capas y reduciendo el impacto de los cambios. A continuación, se detalla cada capa:

- **rest:** en esta capa se encuentran los controladores REST, las excepciones que se pueden dar relacionadas con el protocolo HTTP y los advices, lugar en el que se encuentran las clases con la anotación `@ControllerAdvice` [46], que se encargarán de manejar las excepciones que se indiquen y facilitar al desarrollador la gestión de las respuestas HTTP en caso de que ocurra una excepción, siendo estos controladores los que se encarguen de responder a cada petición con el código de error HTTP que se le haya indicado.
- **business:** en esta capa se encuentra la lógica de negocio de la aplicación. Además, las clases del modelo de dominio son utilizadas como DTO. En esta capa también se encuentra el paquete `scheduled`, donde se encuentran las clases encargadas de tareas automatizadas.

- **persistence:** en esta capa se encuentran las entidades JPA y los repositorios mediante los que se realizarán las diferentes operaciones en la Base de Datos.

En la Figura 5.40 se puede observar, mediante un diagrama de clases a nivel de paquetes, la arquitectura general de la API del backend de la aplicación.

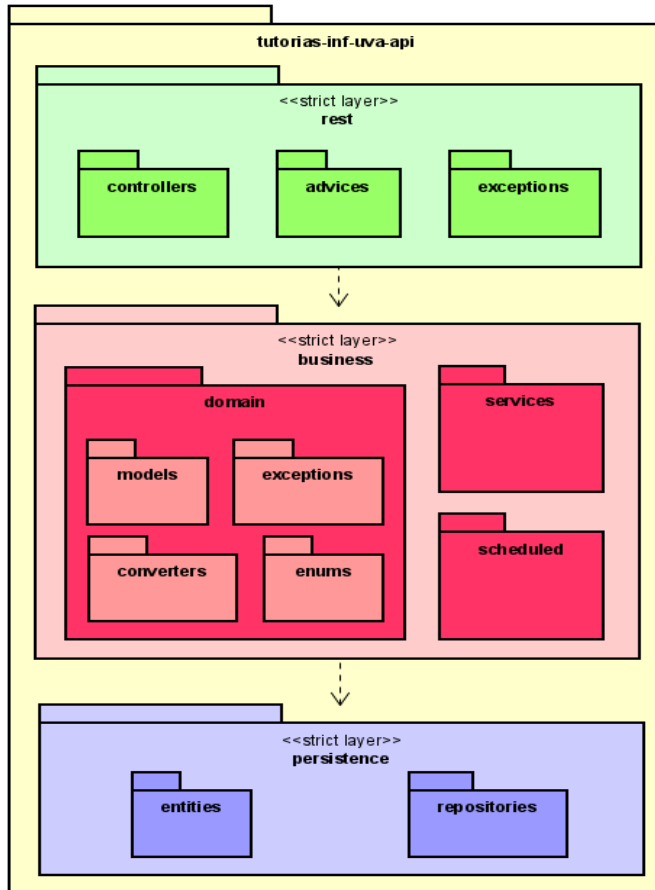


Figura 5.40: Arquitectura general de la API

5.2.3. Arquitectura en el Bot de Telegram

Para la arquitectura del Bot de Telegram, se sigue un esquema muy parecido al seguido en la arquitectura de la API (Ver Sección 5.2.2). Sin embargo, en esta ocasión nos encontramos ante una **arquitectura abierta** [23], pues se dispone de una capa relajada pero que se empleará únicamente para temas de configuración. A continuación, se detallan las cuatro capas:

- **handlers:** en esta capa se encuentran los controladores que se encargarán de gestionar al bot de Telegram. Para el desarrollo de este bot en particular solo ha sido necesario un único controlador, que se encarga de las actualizaciones recibidas y a partir de ahí se realizan las operaciones pertinentes.
- **business:** al igual que en la arquitectura de la API, aquí se encuentra todo lo referente a la lógica de negocio. La mayor diferencia que se puede encontrar tiene que ver con el subpaquete *webclient*, que se encargará de consumir los servicios ofrecidos por la API mediante el protocolo HTTP.
- **persistence:** en esta capa, al igual que en la arquitectura de la API, se encuentran las entidades JPA y los repositorios, a partir de los que se realizarán las operaciones con la Base de Datos.

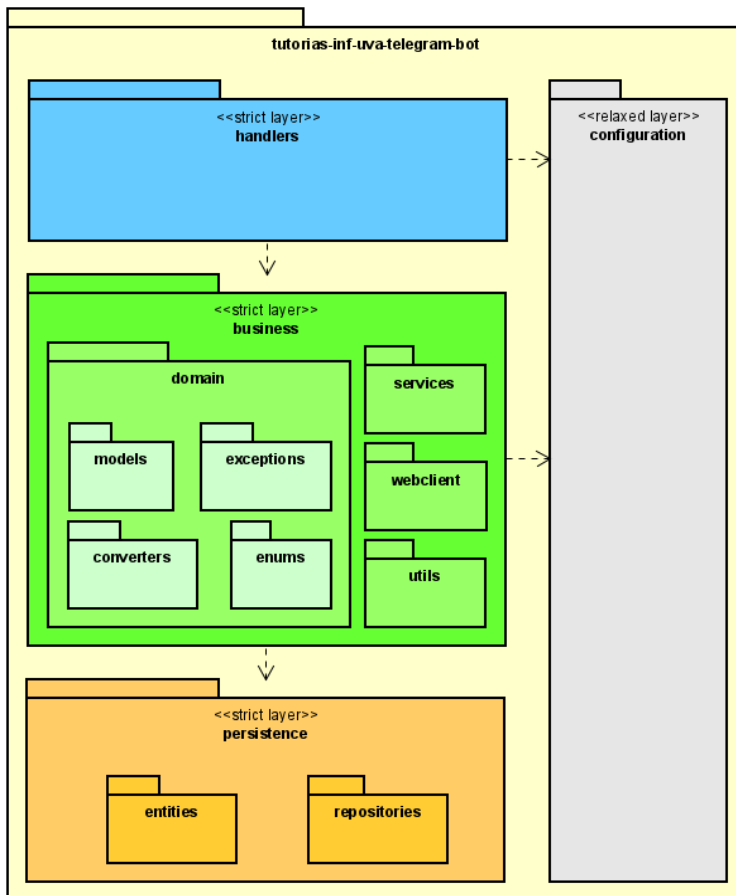


Figura 5.41: Arquitectura general del bot de Telegram

5.3. Patrones de diseño empleados

En esta sección se detallan los patrones de diseño destacados que el alumno ha empleado en el diseño del proyecto, por lo que no se tienen en cuenta los patrones utilizados por los frameworks empleados para el desarrollo del proyecto.

5.3.1. Patrón DTO

El patrón *DTO (Data Transfer Object)* [63] tiene como objetivo reunir en un objeto plano una serie de atributos para ser recuperados, por ejemplo, de un servidor, en una sola invocación, evitando, así, más de una operación costosa.

En este proyecto, tanto en el bot de Telegram como en la API, se utilizan como DTO las clases del modelo de dominio, en las que se mapean los datos necesarios de las entidades de la capa de persistencia (ver Figuras 5.40 y 5.41) a partir de las clases del paquete converters.

5.3.2. Patrón DAO

El patrón *DAO (Data Access Object)* [24] [62] permite abstraer y encapsular el acceso a una fuente de datos. Utilizar este patrón, permite que la tecnología subyacente de acceso a la fuente de datos puede ser cambiada sin tener que modificar la aplicación en su totalidad.

En este proyecto se utiliza el patrón DAO para separar la lógica de negocio del acceso a la base de datos. Los diferentes DAO se encuentran en el paquete repositories. En la Figuras 5.42 y 5.43 se muestran los diagramas de diseño detallado de las interfaces correspondientes a los DAOs de la API y el bot de Telegram, respectivamente.

5.3.3. Patrón Comando

El patrón *Comando* [24] [57] permite encapsular peticiones como un objeto, permitiendo solicitar peticiones a un objeto sin conocer el contenido de dicha operación, ni su receptor. En la Figura 5.44 se muestra la estructura que sigue este patrón.

En este proyecto se ha empleado este patrón para el diseño del bot de Telegram, con la intención de que las invocaciones de las distintas acciones del bot sigan un formato uniforme y que permita la extensión de nuevos comandos de manera más sencilla. En las Figuras 5.45 y 5.46 se muestran los diagramas de clases de las relaciones entre clases y el diseño detallado, respectivamente, del patrón comando en el diseño del bot de Telegram.

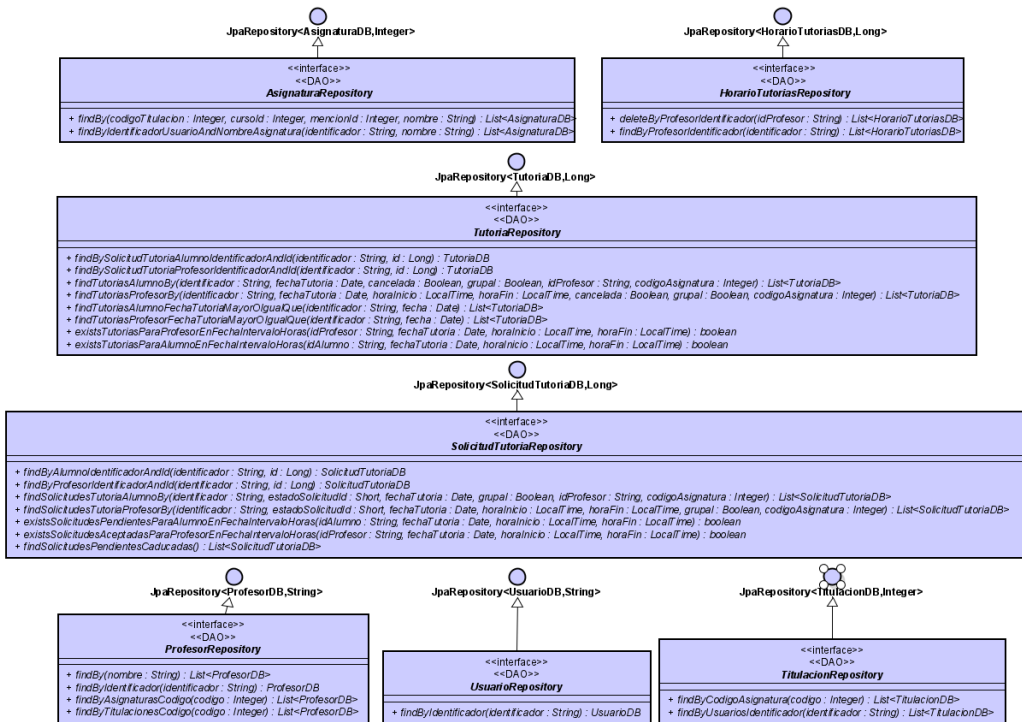


Figura 5.42: Diseño detallado: DAOs API REST

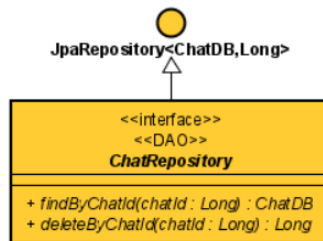


Figura 5.43: Diseño detallado: DAOs del bot de Telegram

5.3.4. Patrón Thread pool

El patrón de diseño *Thread pool* [64] consiste en mantener un conjunto de hilos esperando la ejecución de una pequeña tarea de manera concurrente (ver Figura 5.47). Con este patrón se evita el coste que supone la creación y destrucción de hilos bajo demanda, mejorando el rendimiento y la estabilidad del sistema.

En este proyecto se aplica este patrón en el bot de Telegram, haciendo que cada tarea

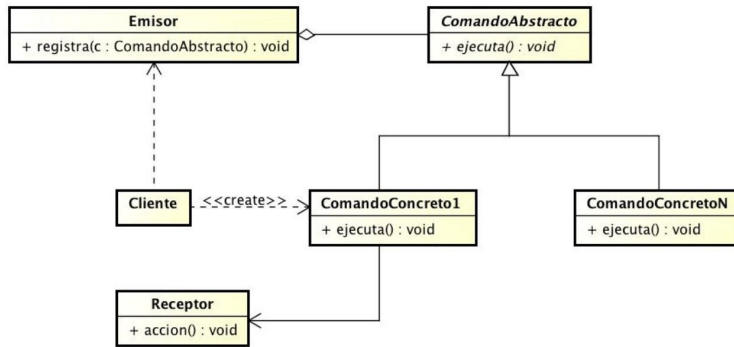


Figura 5.44: Patrón Comando: Estructura

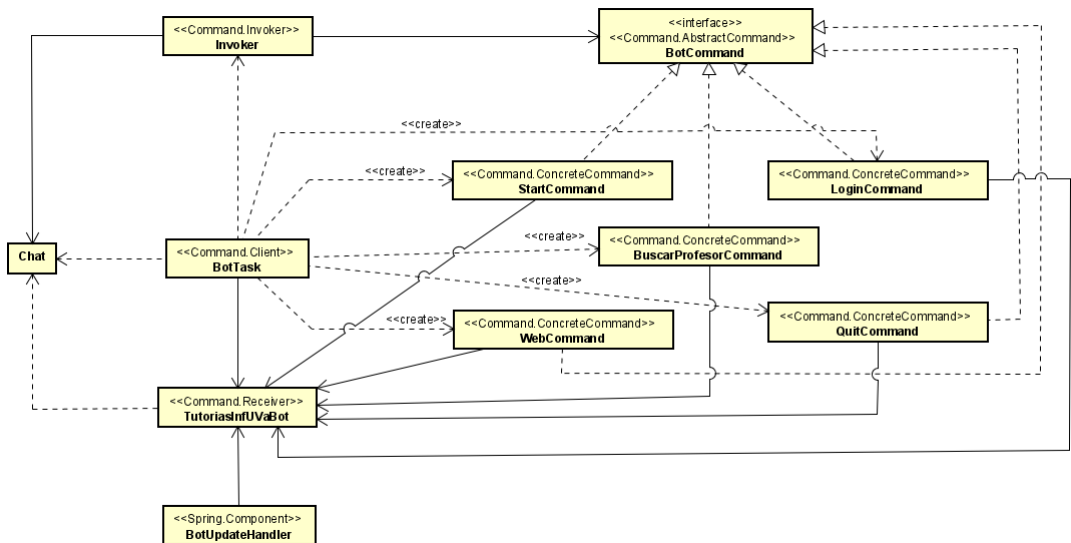


Figura 5.45: Patrón Comando en el bot de Telegram: Diseño de las relaciones entre clases

que realice el bot se ejecute en un hilo que no esté ocupado.

5.4. Diseño de la API REST

En las Figuras 5.48 y 5.49 se muestra el diseño de la API REST en forma de diagrama de recursos y relaciones. Para facilitar la comprensión del diagrama, éste se ha dividido en dos partes diferenciadas: una dedicada a la parte de solicitudes y tutorías, y otra dedicada al resto de recursos, es decir, los recursos de la institución académica (usuarios, titulaciones, asignaturas, cursos, menciones, etc.).

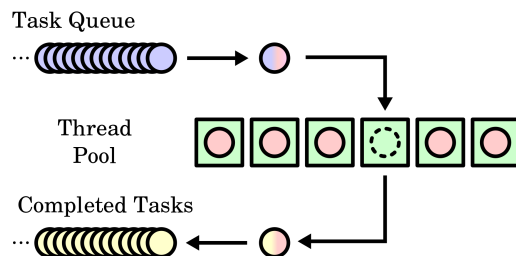


Figura 5.47: Ejemplo de un Thread pool. Tomado de [64]

5.5. Diseño basado en componentes en el frontend de la aplicación

Como se indica en la Sección 3.4.5 del Capítulo 3, dedicado a las diferentes tecnologías utilizadas en el proyecto, una de las características esenciales del framework de desarrollo frontend Angular, es el uso de componentes. Un componente es un elemento reutilizable, compuesto de una lógica, plantilla y estilo (ver Figura 5.50), y que a su vez puede utilizar otros componentes.

Esto permite que cada componente reúna una funcionalidad específica e individual, reduciendo el acoplamiento y favoreciendo la cohesión y el proceso de pruebas.

5.5.1. Componentes creados en el desarrollo del proyecto

A continuación, se enumeran y describen los diferentes componentes angular diseñados para el frontend del proyecto, a partir de los prototipos elaborados y que pueden verse en detalle en la Sección 5.1.

- **App:** es el componente principal y contiene el resto de componentes, formándose, así, la aplicación completa.
- **IniciarSesion:** página de inicio de sesión para los usuarios de la escuela, alumnos o profesores.
- **Header:** cabecera de aplicación web una vez se ha iniciado sesión. En ella se muestra el rol del usuario y su nombre completo. Además, se encuentran las opciones para cerrar sesión y acceder al perfil.
- **Sidebar:** menú lateral izquierdo con las opciones de navegación según el tipo de usuario.
- **RightSidebar:** barra lateral derecha que en caso del alumno muestra un *código QR* que da acceso al bot de Telegram.

5.5. DISEÑO BASADO EN COMPONENTES EN EL FRONTEND DE LA APLICACIÓN

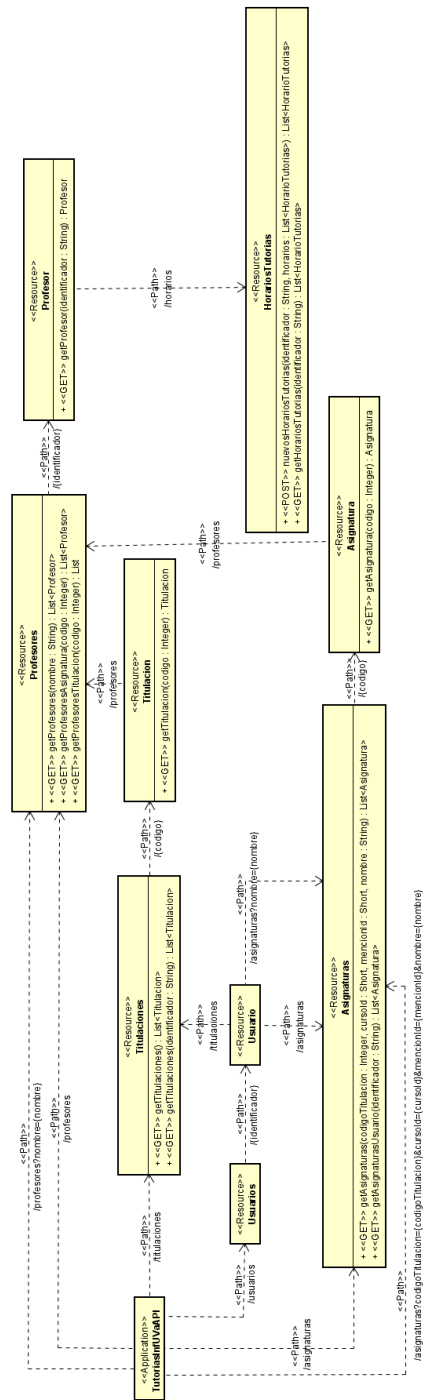


Figura 5.48: Diseño API REST parte 1: Solicitudes y Tutorías

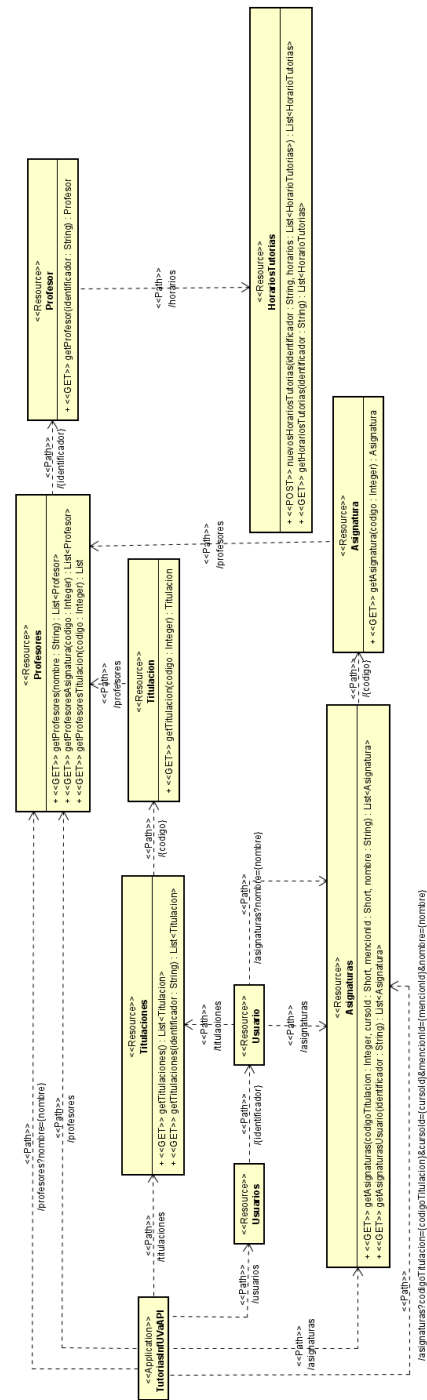


Figura 5.49: Diseño API REST parte 2: resto de recursos

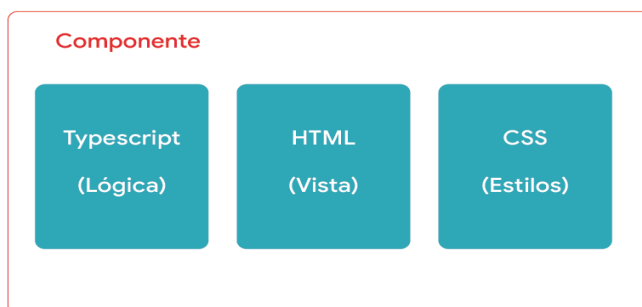


Figura 5.50: Componente en Angular. Tomado de [7]

- **Inicio:** pantalla de inicio. Utiliza los componentes *SolicitudesInicio* y *TutoriasInicio*.
- **SolicitudesInicio:** carrusel donde se muestran las solicitudes pendientes en el caso del profesor y las solicitudes pendientes de confirmación (en estado *aceptada*) en el caso del alumno.
- **TutoriasInicio:** carrusel donde se muestran las próximas tutorías programadas para el usuario.
- **BuscarProfesor:** componente con la función de buscar un profesor. Solo accesible para un alumno. Utiliza los componentes *BusquedaPorProfesor* y *BusquedaPorAsignatura*.
- **BusquedaPorProfesor:** componente con la función de buscar un profesor mediante su nombre.
- **BusquedaPorAsignatura:** componente con la función de buscar un profesor a partir de las asignaturas que imparte.
- **Perfil:** componente que muestra la información del usuario que ha iniciado sesión. Utiliza los componentes *DatosUsuario*, *Asignaturas* y *HorarioTutorias* (en el caso de un profesor).
- **DatosUsuario:** muestra los datos del usuario pasado como input. Utiliza el componente *DatosAlumno* en el caso de que el usuario pasado como input sea un alumno o *DatosProfesor* si es un profesor.
- **DatosAlumno:** muestra los datos específicos de un alumno.
- **DatosProfesor:** muestra los datos específicos de un profesor.
- **Asignaturas:** muestra las asignaturas de un usuario.
- **HorariosTutorias:** muestra los horarios de tutorías de un profesor. Utiliza el componente *CargarHorarioTutorias* en el caso de un profesor.
- **CargarHorarioTutorias:** permite cargar nuevos horarios de tutorías a partir de un fichero CSV. Accesible solo para un profesor.

- **InformacionProfesor:** muestra la información de un profesor. Utiliza los componentes *TutoriasProgramadas*, *Asignaturas*, *DatosUsuario* y *HorarioTutorias*.
- **TutoriasProgramadas:** muestra las tutorías programadas de un profesor.
- **SolicitarTutoria:** permite solicitar tutorías a un alumno. Utiliza el componente *HorarioTutorias*.
- **ConsultarSolicitudesTutoria:** permite filtrar y ver un listado de las solicitudes de un usuario. Utiliza los componentes *FiltrosSolicitudTutoria* y *ListadoSolicitudesTutoria*.
- **FiltrosSolicitudTutoria:** permite seleccionar los filtros mediante los que buscar solicitudes de tutoría.
- **ListadoSolicitudesTutoria:** muestra en forma de lista las solicitudes de tutoría de un usuario.
- **SolicitudTutoria:** muestra el detalle de una solicitud de tutoría y sus acciones posibles. Utiliza los componentes *AceptarSolicitudTutoria* y *RechazarSolicitudTutoria*, únicamente accesibles por un profesor.
- **AceptarSolicitudTutoria:** permite a un profesor aceptar una solicitud de tutoría.
- **RechazarSolicitudTutoria:** permite a un profesor rechazar una solicitud de tutoría.
- **ConsultarTutorias:** muestra las tutorías de un usuario. Utiliza los componentes *TutoriasHoy*, *ProximasTutorias* y *FiltrarTutorias*.
- **Tutoria:** muestra el detalle de una Tutoría y sus acciones. Utiliza el componente *CancelarTutoria*.
- **CancelarTutoria:** permite cancelar una tutoría programada.
- **TutoriasHoy:** muestra las tutorías del día actual para un usuario. Utiliza el componente *ListadoTutorias*.
- **ProximasTutorias:** muestra las próximas tutorías planificadas para un usuario. Utiliza el componente *ListadoTutorias*.
- **FiltrarTutorias:** permite filtrar tutorías y mostrarlas en un listado. Utiliza los componentes *FiltrosTutoria* y *ListadoTutorias*.
- **FiltrosTutoria:** permite seleccionar los filtros mediante los que buscar tutorías.
- **ListadoTutoria:** muestra en forma de lista las tutorías de un usuario.
- **CalendarioTutorias:** muestra mediante un calendario las tutorías de un usuario. Utiliza los componentes *DetalleSolicitudPendienteConfirmar* y *DetalleTutoria*.
- **DetalleSolicitudPendienteConfirmar:** muestra el detalle de una solicitud pendiente de confirmar seleccionada en el calendario. Solo accesible para un profesor.
- **DetalleTutoria:** muestra el detalle de una tutoría seleccionada en el calendario.

En la Figura 5.51 se muestra en forma de diagrama de componentes las dependencias entre estos.

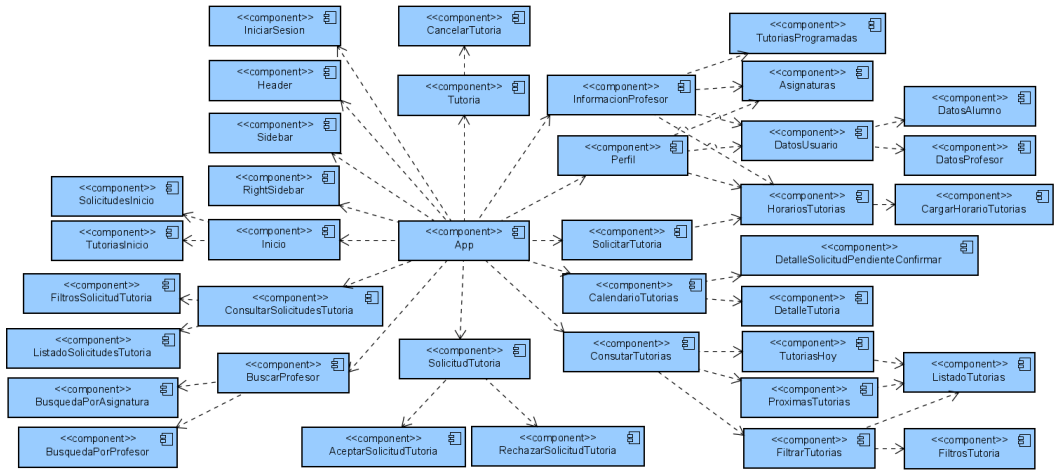


Figura 5.51: Diagrama de Componentes

5.6. Diseño del bot de Telegram

Para el diseño del bot de Telegram se tiene como principal objetivo que sea sencillo de usar. Para ello, se diseñan 5 comandos:

- ***/start***. Todo bot de Telegram necesita de este comando para comenzar la interacción con el mismo. Es por ello, que al ejecutarlo, este comando dará la bienvenida.
- ***/login***. Con este comando se iniciará sesión, lo que permitirá el acceso al resto de funcionalidad del bot, restringiendo, así, la misma, a los usuarios de Telegram que no pertenezcan a la Escuela de Ingeniería Informática.
- ***/buscar_profesor***. Con este comando se inicia el proceso de búsqueda de un profesor para conocer su información. Se diseñan 3 tipos de búsqueda: una búsqueda general, que permitirá buscar entre todas las asignaturas de las titulación en la que esté matriculado un alumno; una búsqueda a partir de las asignaturas matriculadas, restringiendo la búsqueda del profesor a, únicamente, las asignaturas en las que esté matriculado el alumno; y una búsqueda por nombre de asignatura, lo que permitirá buscar una asignatura a partir de una cadena de texto y, a partir de los resultados, seleccionar al profesor del que se quiere conocer su información.
- ***/web***. Este comando dará acceso a la web de Tutorías InfUva.
- ***/quit***. Con este comando se terminará la sesión iniciada, haciendo que no se puedan utilizar el resto de funcionalidades hasta que se vuelva a iniciar sesión.

Para el desarrollo de este bot se empleará la librería de Java *TelegramBots* [9].

5.7. Diseño de la comunicación entre objetos

En esta sección se muestran los diagramas de secuencia en diseño, correspondientes a la historia de usuario **HU05: Como alumno quiero hacer una solicitud de tutoría a un profesor para facilitar la tarea de solicitar una tutoría**. De esta manera, se pretende tener un ejemplo de la comunicación entre objetos de una historia representativa que sirva como caso extrapolable al resto de comunicaciones entre objetos del proyecto, ya que, por falta de tiempo y que se trata de un trabajo de ámbito académico, la realización de diagramas de secuencia para todas las historias de usuario queda fuera del alcance de éste.

Además, para simplificar el proceso, algunos subdiagramas son omitidos. Sin embargo, en los *Interaction Use* cuyo subdiagrama no es representado se intenta ser lo más preciso posible en su descripción para dejar claro las actividades que realizan. En la Figura 5.52 se puede observar la secuencia principal de la historia de usuario **HU05**.

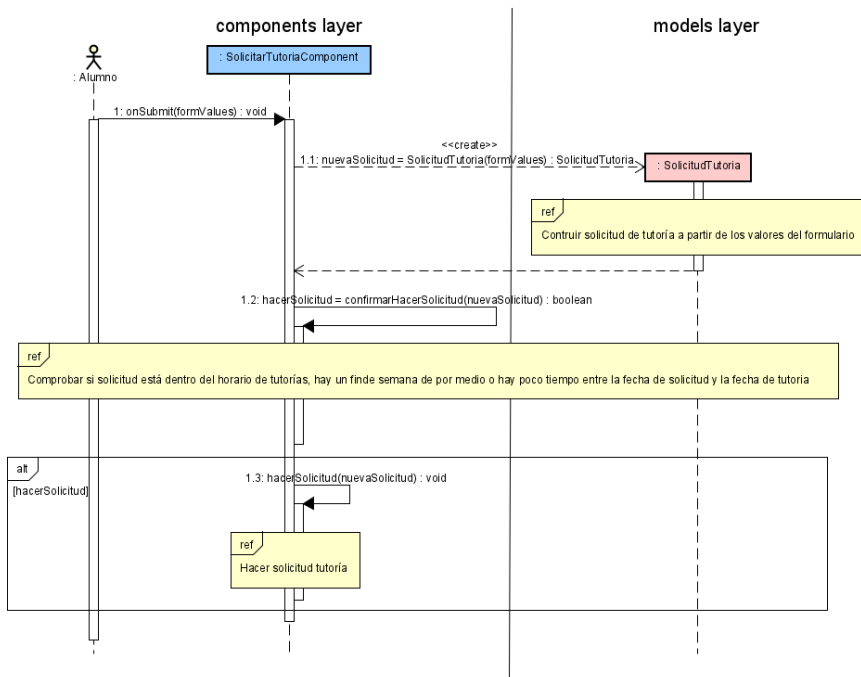


Figura 5.52: Secuencia principal - HU05: Solicitar tutoría

5.8. Diseño del modelo de datos

Para la realización del modelo relacional de la aplicación de tutorías, se ha partido del modelo de dominio de la aplicación (ver Figura 4.2). En cada uno de los sprints se ha diseñado únicamente la parte necesaria para cada historia de usuario, teniendo como resultado final el

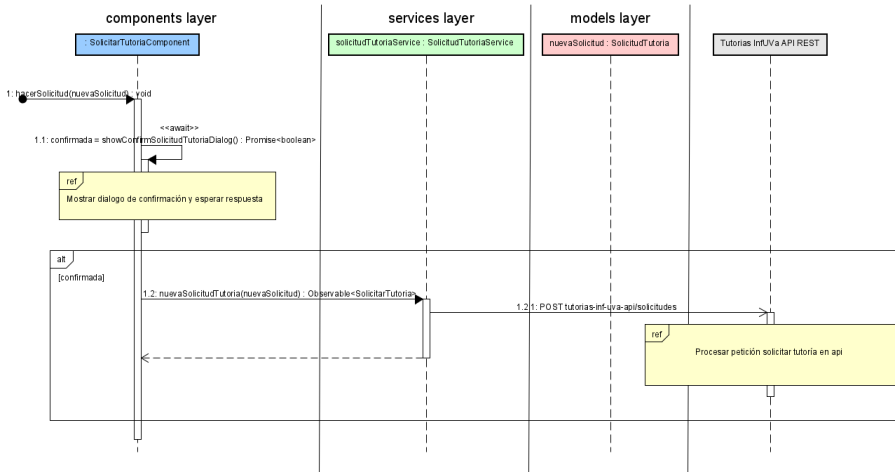


Figura 5.53: Subdiagrama *Interaction Use* “Hacer solicitud tutoría” - HU05: Solicitar tutoría

modelo relacional de la Figura 5.57. Hay que destacar que, para trasladar la herencia de las clases *Alumno* y *Profesor* con la clase *Usuario* al modelo relacional, se ha escogido la opción de emplear una tabla para cada una de las clases, haciendo que no haya posibles campos nulos.

En cuanto al bot de Telegram, se decide prescindir del modelo de datos, pues únicamente consiste en una única entidad “Chat”, que es creada al interactuar con el bot y eliminada al terminar la sesión con el mismo.

5.9. Despliegue de la aplicación web y del bot de Telegram

A pesar de que como primera opción se pretendía desplegar el proyecto en Heroku [26], finalmente, debido a diferentes problemas (explicados en la Sección 7.1.9 del Capítulo 7, dedicado al seguimiento del proyecto), se ha decidido hacerlo sobre una máquina virtual ofrecida por la Escuela.

En la Figura 5.58 se puede observar el diagrama de despliegue diseñado tanto para la aplicación web como para el bot de Telegram. Como se puede observar en la figura, se ha decidido que las aplicaciones desarrolladas estén contenidas en contenedores docker (ver Sección 3.4.14), lo que facilita el despliegue y evita problemas con dependencias que se pudieran necesitar en la máquina virtual.

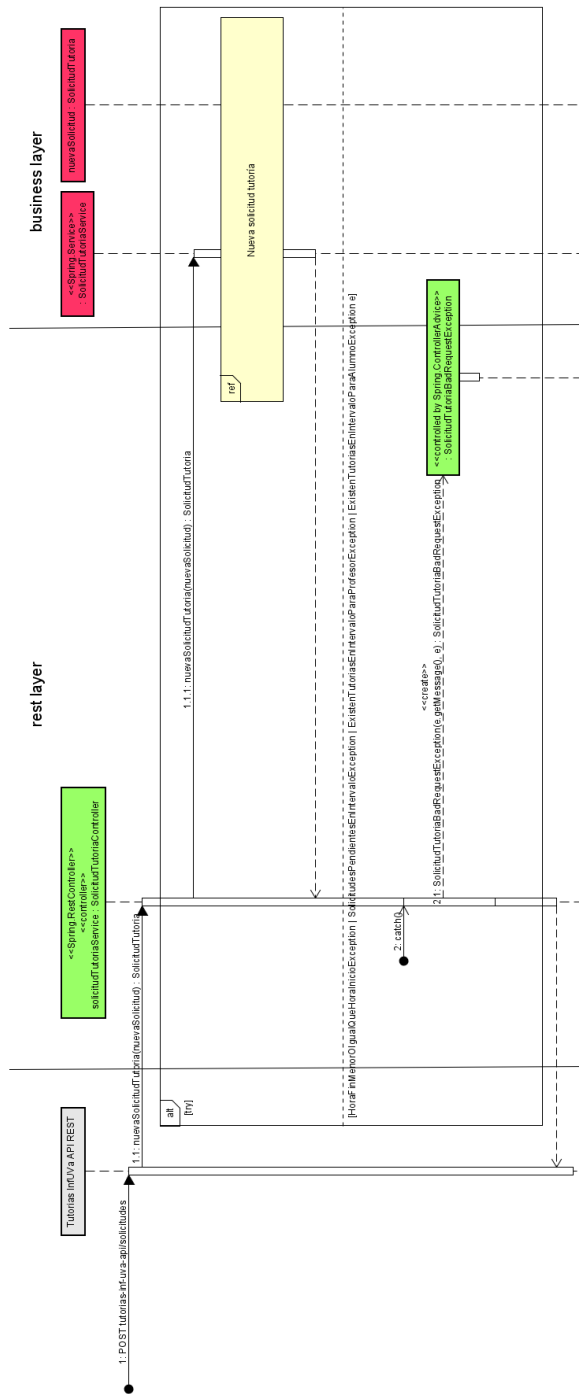


Figura 5.54: Subdiagrama *Interaction Use* “Procesar petición solicitar tutoría en api” - HU05: Solicitar tutoría

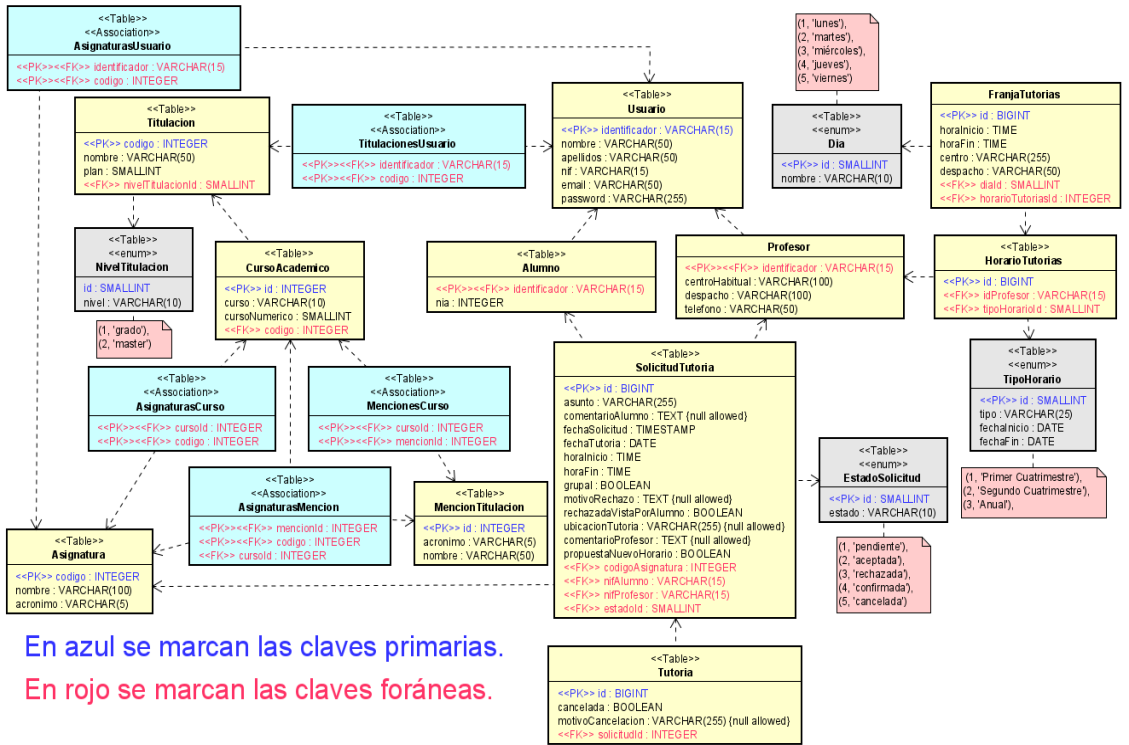


Figura 5.57: Modelo de Datos: modelo relacional

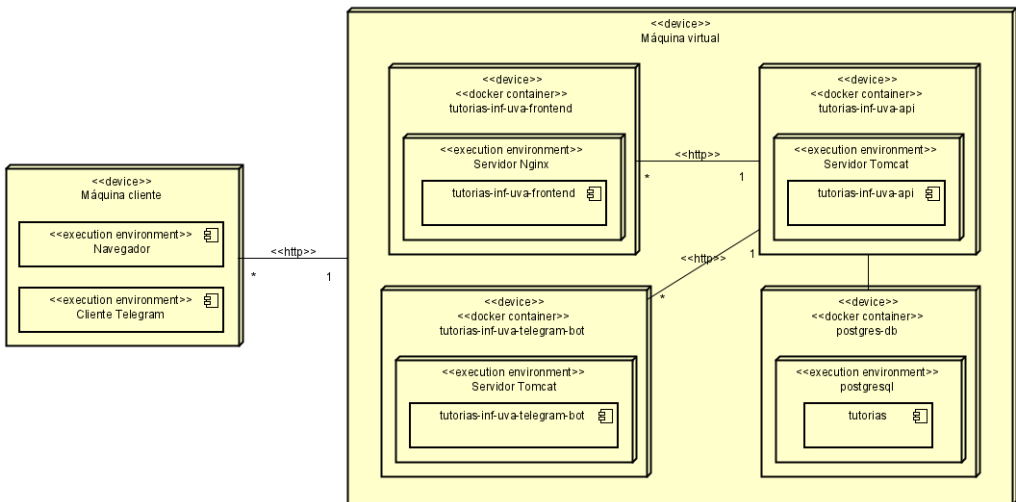


Figura 5.58: Diagrama de despliegue

Capítulo 6

Implementación y pruebas

6.1. Implementación

6.1.1. Organización del código

En esta sección se muestra la organización del código y se explican las distintas partes del mismo. Por brevedad, algunos ficheros de configuración serán omitidos, pues no han sido de demasiada relevancia al tratarse de partes de test automatizados o de configuraciones de los propios frameworks, en las que el alumno no ha participado en su elaboración o modificación.

Estructura del repositorio en GitLab

En la Figura 6.1 se muestra la estructura seguida en el repositorio GitLab (ver Apéndice B). A continuación, se explica en detalle cada una de las carpetas y ficheros:

- En la carpeta ***documentacion*** se encuentra el *Astah* del proyecto. Este fichero contiene toda la documentación referente al modelado de Análisis y Diseño de la aplicación, conteniendo diagramas, que por brevedad no se han mostrado en este presente documento. El `.gitignore` de este nivel sirve para ignorar el fichero `.lock` que genera Astah cuando el proyecto está abierto.
- Las siguientes carpetas, ***tutorias-inf-uva-api***, ***tutorias-inf-uva-frontend*** y ***tutorias-inf-uva-telegram-bot*** contienen las aplicaciones elaboradas y serán detalladas en sus respectivas secciones.
- El fichero `.env` contiene las variables de entorno que se utilizan en el fichero `docker-compose.yml`, para así evitar que estas estén “*hard-codeadas*”, es decir, que estén incrustadas directamente en el mismo fichero.

- El fichero *docker-compose.yml* tiene la función de “dockerizar” el proyecto. Se explica con más detalle en la Sección 6.1.2.
- Aunque no se incluye en la figura, también se encuentra un directorio, *horarios_pruebas*, con los ficheros CSV utilizado para las pruebas de la carga del nuevo horario por parte de un profesor (ver Sección 6.2).

```
tfg-richardalbanfernandez/  
├─ documentacion/  
│   └─ .gitignore  
│   └─ tutoriasInfUva.asta  
├─ tutorias-inf-uva-api/  
├─ tutorias-inf-uva-frontend/  
├─ tutorias-inf-uva-telegram-bot/  
├─ .env  
└─ docker-compose.yml
```

Figura 6.1: Estructura del repositorio en GitLab

Estructura del código en el frontend

A continuación, se muestra la estructura del proyecto frontend, el cual sigue una estructura en mayor o menor medida dictada por el framework Angular, pero con la flexibilidad de poder organizar la parte del código como deseemos sin necesidad de tener que modificar ficheros de configuración. En la Figura 6.2 se muestra la estructura seguida, y que es detallada a continuación:

- La carpeta *src* contiene el código fuente de la aplicación dentro de la subcarpeta *app*, las imágenes y la configuración de idioma de los textos en los componentes PrimeNG y el calendario de Angular [19] se encuentran dentro de la subcarpeta *assets*. En la subcarpeta *environments* se encuentran las variables de entorno utilizadas en el proyecto, que pueden ser distintas según como se decida construir la aplicación, desarrollo o producción, además de otros posibles entornos que podemos crear. En este proyecto se emplea *environment.prod.ts* para desplegar el proyecto en la máquina virtual y *environment.ts* durante el desarrollo y despliegue local. Por último, en la carpeta *src* se encuentran, también, los ficheros principales de la aplicación: *index.html*, *main.ts* y *styles.scss*.
- Dentro de la carpeta *app* encontramos la organización ideada por el alumno durante las fases de diseño (ver Capítulo 5). En la carpeta *components* se encuentran todos los componentes desarrollados para la aplicación (ver Sección 5.5). En la carpeta *guards* [3] se encuentran las secciones de código que prevendrán el acceso no autorizado a las diferentes partes de la aplicación, como puede ser el acceso a la funcionalidad de la misma sin haber iniciado sesión previamente. En la carpeta *models* se encuentra las clases del modelo de dominio de la aplicación y, en la subcarpeta *enums* se encuentran los tipos de datos que representan enumeraciones, como los estados de una solicitud o

los días de la semana. Dentro de la carpeta *services* se encuentran los servicios encargados, principalmente, de realizar las distintas peticiones a la API REST. También, se encargan de otras tareas típicas de la aplicación, como el servicio de eventos, el cual adapta las tutorías como un evento para ser mostradas en el calendario. En la carpeta *utils* se encuentran las clases de utilidades de uso recurrente en la aplicación. Cabe destacar el fichero *app-routing.module.ts*, encargado del sistema de rutas de la aplicación [3]. El resto de ficheros de la carpeta *app* se corresponde con el componente principal de la aplicación, el *AppComponent*.

- De vuelta en el fichero raíz, nos encontramos con algunos ficheros de configuración [6]: el de Angular (*angular.json*), el fichero de configuración para el servidor Nginx (*nginx.conf*), dos ficheros Dockerfile, el fichero de configuración de dependencias (*package.json*) y el fichero de información de los paquetes instalados (*package-lock.json*). También están los ficheros *.gitignore* y *.dockerignore*, con los que ignoraremos los ficheros indicados en ambas herramientas, y el archivo *README.md* con la información por defecto del proyecto Angular.

```
tutorias-inf-uva-frontend/
├── src/
│   ├── app/
│   │   ├── componentes/
│   │   ├── guards/
│   │   ├── models/
│   │   │   └── enums/
│   │   ├── services/
│   │   ├── utils/
│   │   ├── app-routing.module.ts
│   │   ├── app.component.html
│   │   ├── app.component.scss
│   │   ├── app.component.ts
│   │   └── app.module.ts
│   ├── assets/
│   │   ├── i18n/
│   │   │   └── es.json
│   │   └── images/
│   ├── environments/
│   │   ├── environment.prod.ts
│   │   └── environment.ts
│   ├── index.html
│   ├── main.ts
│   └── styles.scss
├── .dockerignore
├── .gitignore
├── angular.json
├── Dockerfile
├── Dockerfile.local
├── nginx.conf
├── package-lock.json
├── package.json
└── README.md
```

Figura 6.2: Estructura del código en el frontend

Estructura del código en el backend (API REST)

Para el desarrollo de la API REST se emplea Spring Boot [47] con el gestor Apache Maven [42], por lo que el proyecto sigue la estructura característica de un proyecto Maven [41], teniendo el código fuente dentro de “*src/main/java*” y los recursos en “*src/main/resources*”. En la Figura 6.3 se muestra la estructura, que es detallada a continuación:

- Bajo la estructura de directorios “*es/uva/inf/tutorias*”, creada a partir del *groupId* del POM, se encuentra el código de la aplicación, siguiendo la arquitectura por capas diseñada (ver Sección 5.2.2). Aquí, se encuentra “*TutoriasInfUVaApiApplication.java*”, que contiene la clase principal de esta aplicación.
- En la carpeta ***rest***, capa de nivel 1, se encuentran los controladores REST de los distintos endpoints bajo la subcarpeta ***controllers***. En la carpeta ***exceptions*** se encuentran las excepciones que se pueden producir en esta capa debido a códigos de error referentes al protocolo HTTP, que son gestionados e introducidos en la respuesta HTTP mediante los controladores ***advice***, que se encuentran en la carpeta ***advices***.
- Bajo la carpeta ***business***, capa de nivel 2, encontramos la lógica de negocio de la aplicación, gestionada mediante los servicios de la carpeta ***services*** y modelada a partir de las clases del modelo de dominio, enumeraciones y excepciones, que se encuentran en las carpeta ***models***, ***enums*** y ***exceptions***, respectivamente, bajo la carpeta ***domain***. En esta carpeta, también se encuentran, dentro de ***converters***, los convertidores, encargados de mapear los diferentes campos de las entidades (de la capa de persistencia) con las clases del modelo de dominio.
- En la carpeta ***persistence***, capa de nivel 3, se encuentran las entidades JPA y las clases DAO bajo ***entities*** y ***repositories***, respectivamente.
- En la carpeta ***resources*** encontramos los diferentes ficheros de configuración. En primer lugar, tenemos los *application.properties*, que además, se han dividido en dos a mayores: uno que es usado durante el desarrollo de manera local (seguido de “*-local*”) y otro, que es usado para el despliegue (seguido de “*-prod*”). En esta carpeta de recursos también encontramos los scripts de configuración de la base de datos, teniendo uno para la creación de tablas y otro para poblarlas.
- Por último, volviendo a la raíz, encontramos el fichero Dockerfile, cuya función se explica en la Sección 6.1.2, el fichero .gitignore y el xml del POM, que es la unidad fundamental de trabajo en un proyecto Maven [40], cuya información y configuración permitirán la construcción del proyecto.

Estructura del código en el bot de Telegram

La estructura del código en el bot de Telegram, también realizada mediante Spring Boot y Maven, es muy similar a la de la API, por lo que únicamente se comentarán las diferencias:



Figura 6.3: Estructura del código de la API REST

- En la capa de negocio, carpeta **business**, se encuentra el directorio **utils**, con las clases de utilidades recurrentes, y el directorio **webclient**, donde se encuentra la clase que consumirá los servicios de la API REST.
- En la capa de configuración, carpeta **configuration**, se encuentran las diferentes clases de configuración del proyecto: *BotConfiguration* para la configuración del bot, *RestClientConfiguration* con la información de la API a consumir, y *TutoriasInfUVaWebProperties*, que contiene la información de la web creada para acceder a ella mediante el bot.
- Por último, en la capa de los manejadores, carpeta **handlers** se encuentran los controladores del bot. En este caso, solo hay uno, *BotUpdateHandler*, que se encarga de manejar las actualizaciones recibidas (nuevos mensajes, comandos o callbacks).

6.1.2. Dockerización del proyecto

Dockerizar una aplicación es como se conoce al hecho de empaquetar una aplicación software para ser distribuida y ejecutada mediante *Docker*. Para dockerizar una aplicación necesitamos un archivo conocido como Dockerfile [13], que consiste en una serie de instrucciones para tener, como resultado final, una imagen que puede ser ejecutada. En las Figuras

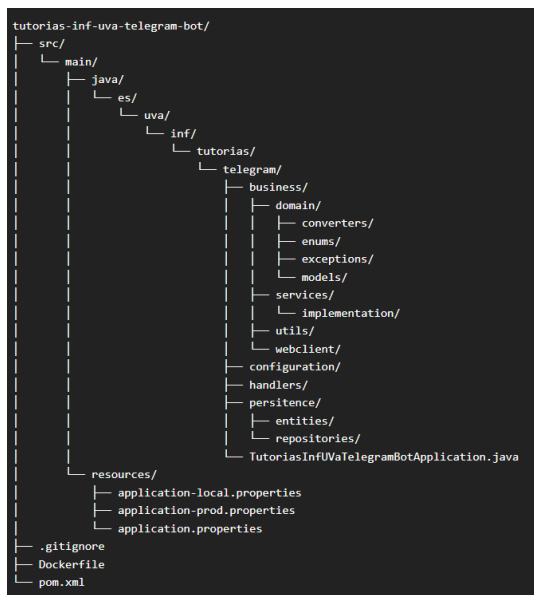


Figura 6.4: Estructura del código en el bot de Telegram

6.5, 6.6 y 6.7 se muestran los ficheros Dockerfile del frontend, API REST y bot desarrollados, respectivamente.

```

### STAGE 1: Build ###
FROM node:alpine AS build
WORKDIR /usr/src/app
COPY package.json ./
RUN npm install
COPY . .
RUN npm run build --prod

### STAGE 2: Run ###
FROM nginx:alpine
COPY nginx.conf /etc/nginx/conf.d/default.conf
COPY --from=build /usr/src/app/dist/tutorias-inf-uva-frontend /usr/share/nginx/html
EXPOSE 80

```

Figura 6.5: Dockerfile del frontend

Puesto que este proyecto, consta de 3 aplicaciones, necesitamos utilizar la herramienta *Compose* de Docker [15] para correr los contenedores mediante un fichero YAML en el que configuramos los servicios. Por brevedad, no se muestra el contenido del fichero *docker-compose.yml* en el presente documento, sin embargo, estará accesible en el repositorio del proyecto (ver Apéndice B). En el fichero *docker-compose.yml* se definen los servicios, cada uno de ellos será un contenedor docker, y su respectiva configuración. A continuación, se describe brevemente el contenido del fichero, para que pueda ser comprendido:

```

### STAGE 1: Build ###
FROM maven:3.8.6-openjdk-11-slim AS build
RUN mkdir -p /workspace
WORKDIR /workspace
COPY pom.xml /workspace
COPY src /workspace/src
RUN mvn -f pom.xml clean package -Pprod -DskipTests

### STAGE 2: Run ###
FROM openjdk:11-slim
COPY --from=build /workspace/target/*.jar app.jar
EXPOSE 8080
ENTRYPOINT ["java", "-jar", "app.jar"]
    
```

Figura 6.6: Dockerfile de la API REST

```

### STAGE 1: Build ###
FROM maven:3.8.6-openjdk-11 AS build
RUN mkdir -p /workspace
WORKDIR /workspace
COPY pom.xml /workspace
COPY src /workspace/src
RUN mvn -f pom.xml clean package -Pprod -DskipTests

### STAGE 2: Run ###
FROM openjdk:11-slim
COPY --from=build /workspace/target/*.jar app.jar
EXPOSE 8080
ENTRYPOINT ["java", "-jar", "app.jar"]
    
```

Figura 6.7: Dockerfile del bot de Telegram

- Se definen 5 servicios: db, api, frontend, frontend-local (para desplegarlo de manera local) y bot.
- En cada servicio se define el nombre que tendrá la imagen y el contenedor, los puertos (el de la máquina local y el del contenedor), y la ubicación del fichero Dockerfile mediante el que se construirá la imagen. Además se indican los perfiles a los que pertenece cada servicio, lo que determinará que servicios serán contruidos como contenedores.
- Entre el servicio db y api se establece una dependencia, de manera que db sea contruido antes que api.

6.1.3. Principales dificultades y retos

A lo largo de la planificación y ejecución del proyecto, han surgido diferentes problemas o ciertas tareas que han supuesto un reto. Todo ello se comenta en el Capítulo 7, dedicado al seguimiento, pero en esta sección, se reúnen aquellas dificultades y retos que han destacado por encima del resto:

- **Bot de Telegram.** Sin lugar a dudas, el desarrollo del bot de Telegram ha sido el mayor reto del proyecto. Y no lo ha sido porque hacer un bot en Telegram sea complicado, si no porque el bot que se ha tenido que desarrollar necesita conocer tanto el estado de una petición como poder manejar varias tareas de manera concurrente, pues se tiene que atender a varios clientes al mismo tiempo desde un único servidor. Tras intentar varias soluciones para el correcto manejo de threads, finalmente se tomo como opción el uso del patrón Thread Pool (ver Sección 5.3.4 del Capítulo 5). De esta manera, se ha podido gestionar con eficiencia el uso de hilos.
- **Cambio en el modelo de dominio inicial.** Durante el sprint 0, se creó un modelo de dominio partiendo de un análisis erróneo. En un primer momento, se había ideado la aplicación como de uso exclusivo de la Escuela de Ingeniería Informática, haciendo que el modelo de dominio inicial se enfocara a las titulaciones que ésta impartía. Sin embargo, a la hora de la implementación, el alumno se dio cuenta de que esta primera aproximación era demasiado poco flexible a que se introdujeran posibles cambios en un futuro, por ejemplo, para otras titulaciones que no fueran las impartidas por la

Escuela, y, además, no se había modelado de manera adecuada la posibilidad de que un curso académico pudiera tener menciones. En la Sección 4.3 del Capítulo 4, dedicado al análisis, y en la Sección 7.1.4 del Capítulo 7, dedicado al seguimiento, se comenta este problema.

- **Despliegue del proyecto.** En un principio se planeaba desplegar el proyecto en *Heroku* [26], sin embargo, puesto que las 3 partes del proyecto (frontend, api y bot) se encuentran en un mismo repositorio de GitLab, la herramienta no detectaba las tecnologías utilizadas provocando diferentes errores, con los que no se podía lidiar por falta de tiempo, ya que ni siquiera se conoce si existe alguna posible solución. *Heroku* también dispone de la posibilidad de desplegar proyectos mediante Docker, por lo que se pasó a “dockerizar” el proyecto (ver Sección 6.1.2), para lo que se tuvieron que obtener conocimientos básicos de esta herramienta, pues nunca se había utilizado. Realizada la “dockerización”, para lo que se utiliza el fichero `docker-compose.yml`, se trató de desplegar en *Heroku*. Sin embargo, esta requería de un tipo de fichero YAML [67], con una estructura distinta al `docker-compose.yml`, por lo que, finalmente, se descartó la idea de desplegar el proyecto empleando esta herramienta y se utilizó una máquina virtual ofrecida por la Escuela. En el Capítulo 7, dedicado al seguimiento, se comenta este problema.

6.1.4. Licencia

Se ha dotado a este proyecto de la licencia de software libre *BSD de 3 cláusulas* [29]. Esto significa que la redistribución y uso del código fuente y binario, con o sin modificaciones, está permitido siempre que se cumplan las 3 cláusulas que dictamina esta licencia.

BSD 3-Clause License

Copyright (c) 2022, Albán Fernández, Richard
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

6.2. Plan de pruebas y evaluación

En esta sección se detallan los casos de prueba manuales realizados para probar el correcto funcionamiento de la web creada y el bot de Telegram.

6.2.1. Casos de prueba

Los siguientes casos de prueba han sido utilizados para determinar si una historia de usuario responde ante diversas situaciones de manera adecuada. En la siguiente sección, Sección 6.2.2, se encuentran los resultados de las pruebas y en caso de que alguna de ellas no haya sido superada, se añade qué error se ha encontrado y la solución que se ha tomado para solventarlo.

HU03 - Buscar profesor

CP-01	Buscar profesor por nombre vacío
Descripción	Buscar un profesor
Actor	Alumno
Escenario	Vista “Búsqueda por profesor”
Entrada	Se pulsa el botón buscar sin introducir ningún nombre.
Resultado esperado	Se muestran todos los profesores.

Tabla 6.1: CP-01: Buscar profesor por nombre vacío

CP-02	Buscar profesor por cadena de texto coincidente con nombre o apellidos
Descripción	Buscar un profesor
Actor	Alumno
Escenario	Vista “Búsqueda por profesor”
Entrada	Se introduce una cadena de texto con un texto que pueda tener varias coincidencias. Se pulsa en el botón buscar.
Resultado esperado	Se muestran todos los profesores cuyos nombres o apellidos contienen la cadena de texto introducida.

Tabla 6.2: CP-02: Buscar profesor por cadena de texto coincidente con nombre o apellidos

CP-03	Buscar profesor por nombre y apellidos exactos
Descripción	Buscar un profesor
Actor	Alumno
Escenario	Vista “Búsqueda por profesor”
Entrada	Se introduce una cadena de texto con el nombre y apellidos de un profesor de manera exacta. Se pulsa en el botón buscar.
Resultado esperado	Se muestra únicamente el profesor con ese nombre y apellidos, o más de uno en caso de que exista la coincidencia de un profesor con el mismo nombre y apellidos.

Tabla 6.3: CP-03: Buscar profesor por nombre y apellidos exactos

CP-04	Buscar profesor por cadena de texto no coincidente
Descripción	Buscar un profesor
Actor	Alumno
Escenario	Vista “Búsqueda por profesor”
Entrada	Se introduce una cadena de texto que no coincide con ningún profesor. Se pulsa en el botón buscar.
Resultado esperado	Se muestra el texto “No se encontraron resultados”.

Tabla 6.4: CP-04: Buscar profesor por cadena de texto no coincidente

CP-05	Buscar profesor por asignatura a partir de la titulación
Descripción	Buscar un profesor a partir de las asignaturas que imparte en una titulación.
Actor	Alumno
Escenario	Vista “Búsqueda por asignatura”
Entrada	Se selecciona la titulación. Se pulsa en el botón buscar.
Resultado esperado	Se muestran las asignaturas de la titulación seleccionada y por cada asignatura los profesores que la imparten. En caso de que una asignatura no tenga docencia se mostrará el mensaje “Asignatura sin docencia”.

Tabla 6.5: CP-05: Buscar profesor por asignatura a partir de la titulación

CP-06	Buscar profesor por asignatura a partir de la titulación y cadena de texto en el nombre de la asignatura
Descripción	Buscar un profesor a partir de las asignaturas que imparte en una titulación y que coincida con una cadena de texto como el nombre de la asignatura.
Actor	Alumno
Escenario	Vista “Búsqueda por asignatura”
Entrada	Se selecciona la titulación. Se introduce una cadena de texto en “Nombre asignatura”. Se pulsa en el botón buscar.
Resultado esperado	Se muestran las asignaturas de la titulación seleccionada que contengan la cadena de texto introducida y por cada asignatura los profesores que la imparten. En caso de que una asignatura no tenga docencia se mostrará el mensaje “Asignatura sin docencia”.

Tabla 6.6: CP-06: Buscar profesor por asignatura a partir de la titulación y cadena de texto en el nombre de la asignatura

CP-07	Buscar profesor por asignatura a partir de la titulación y curso académico
Descripción	Buscar un profesor a partir de las asignaturas que imparte en una titulación y curso académico.
Actor	Alumno
Escenario	Vista “Búsqueda por asignatura”
Entrada	Se selecciona la titulación. Se selecciona el curso académico. Se pulsa en el botón buscar.
Resultado esperado	Se muestran las asignaturas de la titulación y curso seleccionado y por cada asignatura los profesores que la imparten. En caso de que una asignatura no tenga docencia se mostrará el mensaje “Asignatura sin docencia”.

Tabla 6.7: CP-07: Buscar profesor por asignatura a partir de la titulación y curso académico

CP-08	Buscar profesor por asignatura a partir de la titulación, curso académico y cadena de texto en el nombre de la asignatura
Descripción	Buscar un profesor a partir de las asignaturas que imparte en una titulación, curso y que coincida con una cadena de texto como el nombre de la asignatura.
Actor	Alumno
Escenario	Vista “Búsqueda por asignatura”
Entrada	Se selecciona la titulación. Se selecciona el curso académico. Se introduce una cadena de texto en “Nombre asignatura”. Se pulsa en el botón buscar.
Resultado esperado	Se muestran las asignaturas de la titulación y curso seleccionados que contengan la cadena de texto introducida y por cada asignatura los profesores que la imparten. En caso de que una asignatura no tenga docencia se mostrará el mensaje “Asignatura sin docencia”.

Tabla 6.8: CP-08: Buscar profesor por asignatura a partir de la titulación, curso académico y cadena de texto en el nombre de la asignatura

CP-09	Buscar profesor por asignatura a partir de la titulación, curso académico y mención
Descripción	Buscar un profesor a partir de las asignaturas que imparte en una titulación, curso y mención indicados.
Actor	Alumno
Escenario	Vista “Búsqueda por asignatura”
Entrada	Se selecciona la titulación. Se selecciona el curso académico. Si el curso académico tiene distintas menciones, se selecciona una de ellas. Se pulsa en el botón buscar.
Resultado esperado	Se muestran las asignaturas de la titulación, curso y mención seleccionados y por cada asignatura los profesores que la imparten. En caso de que una asignatura no tenga docencia se mostrará el mensaje “Asignatura sin docencia”.

Tabla 6.9: CP-09: Buscar profesor por asignatura a partir de la titulación, curso académico y mención

CP-10	Buscar profesor por asignatura a partir de la titulación, curso académico, mención y cadena de texto en el nombre de la asignatura
Descripción	Buscar un profesor a partir de las asignaturas que imparte en una titulación, curso, mención y que coincida con una cadena de texto como el nombre de la asignatura.
Actor	Alumno
Escenario	Vista “Búsqueda por asignatura”
Entrada	Se selecciona la titulación. Se selecciona el curso académico. Si el curso académico tiene distintas menciones, se selecciona una de ellas. Se introduce una cadena de texto en “Nombre asignatura”. Se pulsa en el botón buscar.
Resultado esperado	Se muestran las asignaturas de la titulación, curso y mención seleccionados que contengan la cadena de texto introducida y por cada asignatura los profesores que la imparten. En caso de que una asignatura no tenga docencia se mostrará el mensaje “Asignatura sin docencia”.

Tabla 6.10: CP-10: Buscar profesor por asignatura a partir de la titulación, curso académico, mención y cadena de texto en el nombre de la asignatura

CP-11	Buscar profesor por asignatura a partir de una cadena de texto en el nombre de la asignatura no coincidente
Descripción	Buscar un profesor a partir de una cadena de texto que actúa como nombre de la asignatura, que no tenga ninguna asignatura coincidente.
Actor	Alumno
Escenario	Vista “Búsqueda por asignatura”
Entrada	Se selecciona la titulación, pues es obligatorio. Se introduce una cadena de texto en “Nombre asignatura”. Se pulsa en el botón buscar.
Resultado esperado	Se muestra el mensaje “No se encontraron resultados”.

Tabla 6.11: CP-11: Buscar profesor por asignatura a partir de una cadena de texto en el nombre de la asignatura no coincidente

HU04 - Ver la información detallada de un profesor

CP-12	Se muestra la información del profesor cuyo horario de tutorías está dividido por cuatrimestres
Descripción	Ver la información del profesor con horario de tutorías distinto según cuatrimestre
Actor	Alumno
Escenario	Vista “Información profesor”
Entrada	Se selecciona el tab del primer cuatrimestre o segundo cuatrimestre.
Resultado esperado	Se muestran los horarios de tutorías del cuatrimestre escogido.

Tabla 6.12: CP-12: Se muestra la información del profesor cuyo horario de tutorías está dividido por cuatrimestres

CP-13	Se muestra la información del profesor cuyo horario de tutorías es únicamente anual
Descripción	Ver la información del profesor con mismo horario de tutorías en todo el año
Actor	Alumno
Escenario	Vista “Información profesor”
Entrada	No es necesario seleccionar ningún tab pues solo se muestra el tab “Anual”.
Resultado esperado	Se muestran los horarios de tutorías de forma anual.

Tabla 6.13: CP-13: Buscar profesor por asignatura a partir de una cadena de texto en el nombre de la asignatura no coincidente

HU05 - Solicitar tutoría

CP-14	Solicitar tutoría dentro de horario de tutorías del profesor
Descripción	Se solicita una tutoría rellenando todos los campos.
Actor	Alumno
Escenario	Vista "Solicitar tutoría"
Entrada	<p>Se rellena el campo profesor.</p> <p>Se selecciona la asignatura a la que hará referencia la tutoría.</p> <p>Se marca el checkbox de "Tutoría grupal"</p> <p>Se introduce la fecha de la tutoría.</p> <p>Se introduce la hora de inicio de la tutoría.</p> <p>Se selecciona una duración estimada de 15 minutos.</p> <p>Se introduce un asunto.</p> <p>Se introduce un comentario.</p> <p>Se pulsa en el botón "Solicitar tutoría".</p> <p>Se muestra un diálogo modal con el siguiente mensaje: "¿Quieres solicitar una tutoría para el día DD/MM/YYYY de HH:mm a HH:mm?".</p> <p>Se pulsa el botón "Si".</p>
Resultado esperado	<p>Se redirige a la pantalla "Solicitud tutoría".</p> <p>Se muestra, en la parte superior, un mensaje de éxito con el siguiente contenido: "La solicitud de tutoría se ha realizado con éxito"</p>

Tabla 6.14: CP-14: Solicitar tutoría dentro de horario de tutorías del profesor

CP-15	Solicitar tutoría fuera del horario de tutorías del profesor
Descripción	Se solicita una tutoría rellenando todos los campos.
Actor	Alumno
Escenario	Vista “Solicitar tutoría”
Entrada	<p>Se rellena el campo profesor.</p> <p>Se selecciona la asignatura a la que hará referencia la tutoría.</p> <p>Se marca el checkbox de “Tutoría grupal”</p> <p>Se introduce la fecha de la tutoría.</p> <p>Se introduce la hora de inicio de la tutoría.</p> <p>Se selecciona una duración estimada de 15 minutos.</p> <p>Se introduce un asunto.</p> <p>Se introduce un comentario.</p> <p>Se pulsa en el botón “Solicitar tutoría”.</p> <p>Se muestra un diálogo modal con el siguiente mensaje: “La tutoría a solicitar está fuera del horario de tutorías del profesor, ¿quieres realizar la solicitud de todas maneras?”.</p> <p>Se pulsa el botón “Si”.</p> <p>Se muestra un diálogo modal con el siguiente mensaje: “¿Quieres solicitar una tutoría para el día DD/MM/YYYY de HH:mm a HH:mm?”.</p> <p>Se pulsa el botón “Si”.</p>
Resultado esperado	<p>Se redirige a la pantalla “Solicitud tutoría”.</p> <p>Se muestra, en la parte superior, un mensaje de éxito con el siguiente contenido: “La solicitud de tutoría se ha realizado con éxito”</p>

Tabla 6.15: CP-15: Solicitar tutoría fuera del horario de tutorías del profesor

CP-16	Solicitar tutoría para un intervalo de tiempo en el que ya hay tutorías establecidas para un profesor
Descripción	Se solicita una tutoría en un intervalo de tiempo cuya hora de inicio y fin se solapa con tutorías ya establecidas para un profesor. Se considera que una solicitud de tutoría se solapa con un conjunto de tutorías ya establecidas cuando para una cierta fecha se cumple: $\forall t \exists st \neg [(t.horaInicio < st.horaInicio \wedge t.horaFin \leq st.horaInicio) \vee (t.horaInicio \geq st.horaFin \wedge t.horaFin > st.horaFin)]$, siendo t una tutoría y st una solicitud de tutoría.
Actor	Alumno
Escenario	Vista “Solicitar tutoría”
Entrada	Se rellena el campo profesor. Se selecciona la asignatura a la que hará referencia la tutoría. Se marca el checkbox de “Tutoría grupal” Se introduce la fecha de la tutoría. Se introduce la hora de inicio de la tutoría. Se selecciona una duración estimada. Se introduce un asunto. Se introduce un comentario. Se pulsa en el botón “Solicitar tutoría”. Se muestra un diálogo modal con el siguiente mensaje: “¿Quieres solicitar una tutoría para el día DD/MM/YYYY de HH:mm a HH:mm?”. Se pulsa el botón “Si”.
Resultado esperado	Se muestra el mensaje de error: “Existen tutorías en ese intervalo para el profesor”.

Tabla 6.16: CP-16: Solicitar tutoría para un intervalo de tiempo en el que ya hay tutorías establecidas para un profesor

CP-17	Solicitar tutoría para un intervalo de tiempo en el que ya hay tutorías establecidas para el alumno solicitante
Descripción	Se solicita una tutoría en un intervalo de tiempo cuya hora de inicio y fin se solapa con tutorías ya establecidas para el propio alumno. Se considera que una solicitud de tutoría se solapa con un conjunto de tutorías ya establecidas cuando para una cierta fecha se cumple: $\forall t \exists st \neg [(t.horaInicio < st.horaInicio \wedge t.horaFin \leq st.horaInicio) \vee (t.horaInicio \geq st.horaFin \wedge t.horaFin > st.horaFin)]$, siendo t una tutoría y st una solicitud de tutoría.
Actor	Alumno
Escenario	Vista “Solicitar tutoría”
Entrada	Se rellena el campo profesor. Se selecciona la asignatura a la que hará referencia la tutoría. Se marca el checkbox de “Tutoría grupal” Se introduce la fecha de la tutoría. Se introduce la hora de inicio de la tutoría. Se selecciona una duración estimada. Se introduce un asunto. Se introduce un comentario. Se pulsa en el botón “Solicitar tutoría”. Se muestra un diálogo modal con el siguiente mensaje: “¿Quieres solicitar una tutoría para el día DD/MM/YYYY de HH:mm a HH:mm?”. Se pulsa el botón “Si”.
Resultado esperado	Se muestra el mensaje de error: “Ya tienes tutorías en ese intervalo”

Tabla 6.17: CP-17: Solicitar tutoría para un intervalo de tiempo en el que ya hay tutorías establecidas para el alumno solicitante

CP-18	Solicitar tutoría con algún campo obligatorio sin rellenar
Descripción	Se intenta solicitar una tutoría dejando un campo obligatorio sin rellenar.
Actor	Alumno
Escenario	Vista “Solicitar tutoría”
Entrada	Se rellena el campo profesor. Se introduce la fecha de la tutoría. Se introduce la hora de inicio de la tutoría. No se selecciona una duración estimada. Se introduce un asunto. Se pulsa en el botón “Solicitar tutoría”.
Resultado esperado	El botón “Solicitar tutoría” está deshabilitado por lo que no se puede solicitar la tutoría.

Tabla 6.18: CP-18: Solicitar tutoría con algún campo obligatorio sin rellenar

CP-19	Ver horario de tutorías del profesor al que se le quiere solicitar la tutoría
Descripción	Al introducir el profesor deseado para la solicitud, se quiere recordar el horario de tutorías del mismo.
Actor	Alumno
Escenario	Vista “Solicitar tutoría”
Entrada	Se introduce el campo profesor. Se pulsa en el botón “Ver horario profesor”.
Resultado esperado	Se muestra un diálogo con el horario de tutorías del profesor.

Tabla 6.19: CP-19: Ver horario de tutorías del profesor al que se le quiere solicitar la tutoría

CP-20	Ver horario de tutorías del profesor al que se le quiere solicitar la tutoría
Descripción	Se intenta ver el horario de tutorías sin introducir ningún profesor.
Actor	Alumno
Escenario	Vista “Solicitar tutoría”
Entrada	Se pulsa en el botón “Ver horario profesor”.
Resultado esperado	El botón “Ver horario de tutorías” está deshabilitado.

Tabla 6.20: CP-20: Ver horario de tutorías del profesor al que se le quiere solicitar la tutoría

CP-21	Solicitar tutoría sin seleccionar asignatura hace que el campo “Comentario” sea obligatorio
Descripción	Se intenta solicitar una tutoría sin seleccionar asignatura, haciendo que el campo comentario sea obligatorio.
Actor	Alumno
Escenario	Vista “Solicitar tutoría”
Entrada	Se rellena el campo profesor. Se introduce la fecha de la tutoría. Se introduce la hora de inicio de la tutoría. Se selecciona una duración estimada. Se introduce un asunto. Se pulsa en el botón “Ver horario profesor”.
Resultado esperado	El botón “Solicitar tutoría” está deshabilitado por lo que no se puede solicitar la tutoría.

Tabla 6.21: CP-21: Solicitar tutoría sin seleccionar asignatura hace que el campo “Comentario” sea obligatorio

CP-22	Solicitar tutoría en un espacio de tiempo menor a un día
Descripción	Se solicita una tutoría con un tiempo entre la hora de solicitud y la hora de tutoría menor a 1 día, por lo que al realizar la solicitud se muestra un mensaje indicando que la solicitud puede no ser atendida a tiempo.
Actor	Alumno
Escenario	Vista “Solicitar tutoría”
Entrada	<p>Se rellena el campo profesor.</p> <p>Se selecciona la asignatura a la que hará referencia la tutoría.</p> <p>Se introduce como fecha de tutoría el día siguiente al actual.</p> <p>Se introduce la hora de inicio de la tutoría.</p> <p>Se selecciona una duración estimada.</p> <p>Se introduce un asunto.</p> <p>Se introduce un comentario.</p> <p>Se pulsa en el botón “Solicitar tutoría”.</p> <p>Se muestra un diálogo modal con el siguiente mensaje: “La solicitud puede no ser atendida a tiempo, ¿quieres continuar?”</p> <p>Se pulsa el botón “Si”.</p> <p>Se muestra un diálogo modal con el siguiente mensaje: “¿Quieres solicitar una tutoría para el día DD/MM/YYYY de HH:mm a HH:mm?”.</p> <p>Se pulsa el botón “Si”.</p>
Resultado esperado	<p>Se redirige a la pantalla “Solicitud tutoría”.</p> <p>Se muestra, en la parte superior, un mensaje de éxito con el siguiente contenido: “La solicitud de tutoría se ha realizado con éxito”</p>

Tabla 6.22: CP-22: Solicitar tutoría en un espacio de tiempo menor a un día

CP-23	Solicitar tutoría un viernes, sábado o domingo para el lunes siguiente
Descripción	Se solicita una tutoría un viernes, sábado y domingo para el lunes siguiente, por lo que al realizar la solicitud se muestra un mensaje indicando que la solicitud puede no ser atendida a tiempo.
Actor	Alumno
Escenario	Vista “Solicitar tutoría”
Entrada	<p>Se rellena el campo profesor.</p> <p>Se selecciona la asignatura a la que hará referencia la tutoría.</p> <p>Se introduce la fecha de la tutoría para el lunes siguiente.</p> <p>Se introduce la hora de inicio de la tutoría.</p> <p>Se selecciona una duración estimada.</p> <p>Se introduce un asunto.</p> <p>Se introduce un comentario.</p> <p>Se pulsa en el botón “Solicitar tutoría”.</p> <p>Se muestra un diálogo modal con el siguiente mensaje: “La solicitud puede no ser atendida a tiempo, ¿quieres continuar?”</p> <p>Se pulsa el botón “Si”.</p> <p>Se muestra un diálogo modal con el siguiente mensaje: “¿Quieres solicitar una tutoría para el día DD/MM/YYYY de HH:mm a HH:mm?”.</p> <p>Se pulsa el botón “Si”.</p>
Resultado esperado	<p>Se redirige a la pantalla “Solicitud tutoría”.</p> <p>Se muestra, en la parte superior, un mensaje de éxito con el siguiente contenido: “La solicitud de tutoría se ha realizado con éxito”</p>

Tabla 6.23: CP-23: Solicitar tutoría un viernes, sábado o domingo para el lunes siguiente

CP-24	Solicitar tutoría en una fecha e intervalo de horas que se solape con otras solicitudes pendientes que hayamos realizado
Descripción	Se solicita una tutoría en una fecha e intervalo de horas que se solape con otras solicitudes pendientes que hayamos realizado, por lo que se muestra un mensaje de error indicando el problema.
Actor	Alumno
Escenario	Vista “Solicitar tutoría”
Entrada	Se rellena el campo profesor. Se selecciona la asignatura a la que hará referencia la tutoría. Se marca el checkbox de “Tutoría grupal” Se introduce la fecha de la tutoría. Se introduce la hora de inicio de la tutoría. Se selecciona una duración estimada. Se introduce un asunto. Se introduce un comentario. Se pulsa en el botón “Solicitar tutoría”. Se muestra un diálogo modal con el siguiente mensaje: “¿Quieres solicitar una tutoría para el día DD/MM/YYYY de HH:mm a HH:mm?”. Se pulsa el botón “Si”.
Resultado esperado	Se muestra el mensaje de error: “Tienes solicitudes pendientes que se solapan con la nueva solicitud”.

Tabla 6.24: CP-24: Solicitar tutoría en una fecha e intervalo de horas que se solape con otras solicitudes pendientes que hayamos realizado

HU06 - Cancelar solicitud de tutoría

CP-25	Cancelar solicitud en estado pendiente o aceptada desde el listado de solicitudes
Descripción	Se cancela una solicitud pendiente o aceptada desde la lista de tutorías del alumno.
Actor	Alumno
Escenario	Vista “Consultar solicitudes”
Entrada	Se selecciona el estado “Pendiente” o “Aceptada”. Se pulsa en el botón “Buscar”. Se pulsa en el botón con el icono de una papelera de una solicitud pendiente. Se muestra un diálogo modal de confirmación con el siguiente mensaje: “¿Estás seguro de que quieres cancelar la solicitud de tutoría?” Se pulsa en el botón “Sí”.
Resultado esperado	Se muestra el mensaje de éxito: “La solicitud de tutoría se ha cancelado”. Se muestra el cambio en el elemento de la lista.

Tabla 6.25: CP-25: Cancelar solicitud en estado pendiente o aceptada desde el listado de solicitudes

CP-26	Cancelar solicitud en estado pendiente o aceptada desde la vista en detalle de la solicitud
Descripción	Se cancela una solicitud pendiente o aceptada desde la opción de cancelar solicitud de la vista detallada de la solicitud de tutoría.
Actor	Alumno
Escenario	Vista “Consultar solicitudes” y Vista “Solicitud tutoría”
Entrada	Se selecciona el estado “Pendiente” o “Aceptada”. Se pulsa en el botón “Buscar”. Se pulsa en el botón con un icono de un ojo y un tooltip que dice “Ver solicitud” de una solicitud pendiente o aceptada. Se muestra la vista “Solicitud tutoría” Se pulsa en el botón “Cancelar solicitud”. Se muestra un diálogo modal de confirmación con el siguiente mensaje: “¿Estás seguro de que quieres cancelar la solicitud de tutoría?” Se pulsa en el botón “Sí”.
Resultado esperado	Se muestra el mensaje de éxito: “La solicitud de tutoría se ha cancelado”. Se muestra la solicitud en estado “CANCELADA”.

Tabla 6.26: CP-26: Cancelar solicitud en estado pendiente o aceptada desde la vista en detalle de la solicitud

HU01 - Iniciar sesión

CP-27	Iniciar sesión como alumno
Descripción	Se inicia sesión en el sistema como un alumno.
Actor	Alumno
Escenario	Vista “Iniciar sesión”
Entrada	Se introduce el usuario de la escuela. Se introduce la contraseña. Se pulsa en el botón “Iniciar sesión”.
Resultado esperado	Se muestra la pantalla de Inicio, en la que se aprecia en el sidebar izquierdo que las funcionalidades son distintas a las de un profesor.

Tabla 6.27: CP-27: Iniciar sesión como alumno

CP-28	Iniciar sesión como profesor
Descripción	Se inicia sesión en el sistema como un profesor.
Actor	Profesor
Escenario	Vista “Iniciar sesión”
Entrada	Se introduce el usuario de la escuela. Se introduce la contraseña. Se pulsa en el botón “Iniciar sesión”.
Resultado esperado	Se muestra la pantalla de Inicio, en la que se aprecian en el sidebar izquierdo que las funcionalidades son distintas a las de un alumno.

Tabla 6.28: CP-28: Iniciar sesión como profesor

CP-29	Iniciar sesión con usuario no existente
Descripción	Se intenta iniciar sesión en el sistema como un usuario que no existe.
Actor	Usuario
Escenario	Vista “Iniciar sesión”
Entrada	Se introduce el usuario que no existe. Se introduce la contraseña. Se pulsa en el botón “Iniciar sesión”.
Resultado esperado	Se muestra el mensaje de error: “Usuario o contraseña incorrectos”

Tabla 6.29: CP-29: Iniciar sesión con usuario no existente

CP-30	Iniciar sesión con contraseña incorrecta
Descripción	Se intenta iniciar sesión en el sistema con una contraseña incorrecta.
Actor	Usuario
Escenario	Vista “Iniciar sesión”
Entrada	Se introduce el usuario de la escuela. Se introduce la contraseña incorrecta. Se pulsa en el botón “Iniciar sesión”.
Resultado esperado	Se muestra el mensaje de error: “Usuario o contraseña incorrectos”

Tabla 6.30: CP-30: Iniciar sesión con contraseña incorrecta

CP-31	Iniciar sesión sin introducir credenciales
Descripción	Se intenta iniciar sesión en el sistema sin introducir las credenciales.
Actor	Usuario
Escenario	Vista “Iniciar sesión”
Entrada	Se pulsa en el botón “Iniciar sesión”.
Resultado esperado	El botón “Iniciar sesión” está deshabilitado.

Tabla 6.31: CP-31: Iniciar sesión sin introducir credenciales

CP-32	Acceder a funcionalidades sin haber iniciado sesión
Descripción	Se intenta acceder a las distintas funcionalidades de la aplicación sin haber iniciado sesión previamente.
Actor	Usuario
Escenario	Navegador
Entrada	Se introduce en el navegador una ruta para acceder a la funcionalidad de la aplicación sin haber iniciado sesión.
Resultado esperado	Se redirige a la pantalla de inicio de sesión.

Tabla 6.32: CP-32: Acceder a funcionalidades sin haber iniciado sesión

HU02 - Cerrar sesión

CP-33	Cerrar sesión
Descripción	Se termina la sesión y se sale del sistema.
Actor	Usuario
Escenario	Cualquier vista de la aplicación a excepción de la vista de inicio de sesión.
Entrada	Se pasa el puntero del ratón sobre las opciones. Se pulsa la opción del dropdown “Cerrar sesión” Se muestra un diálogo de confirmación con el siguiente mensaje: “Vas a terminar la sesión, ¿quieres continuar?”. Se pulsa la opción “Si”.
Resultado esperado	Se muestra la pantalla de inicio de sesión.

Tabla 6.33: CP-33: Cerrar sesión

HU07 - Aceptar solicitud de tutoría

CP-34	Aceptar solicitud pendiente
Descripción	Se acepta una solicitud de tutoría en estado pendiente.
Actor	Profesor
Escenario	Vista “Solicitud tutoría”
Entrada	Se pulsa el botón “Aceptar solicitud”. Se muestra un diálogo modal. Se introduce el lugar donde tendrá lugar la tutoría. Se introduce un comentario opcional. Se pulsa el botón “Aceptar”.
Resultado esperado	Se muestra el mensaje de éxito: “Solicitud aceptada con éxito”. Se muestra el cambio en el estado de la solicitud.

Tabla 6.34: CP-34: Aceptar solicitud pendiente

CP-35	Aceptar solicitud pendiente para horario en el que el profesor ya tiene tutorías o solicitudes pendientes de confirmación
Descripción	Se intenta aceptar una solicitud de tutoría en un espacio de tiempo en el que el profesor ya tiene tutorías o solicitudes por confirmar.
Actor	Profesor
Escenario	Vista “Solicitud tutoría”
Entrada	Se pulsa el botón “Aceptar solicitud”. Se muestra un diálogo modal. Se muestra un mensaje de advertencia con el texto: “Tienes tutorías o solicitudes por confirmar en el espacio de tiempo de esta solicitud”. Se propone un nuevo horario, por lo que se introduce una nueva hora de inicio y duración estimada. Se introduce el lugar donde tendrá lugar la tutoría. Se introduce un comentario que pasa a ser obligatorio. Se pulsa el botón “Aceptar”.
Resultado esperado	Se muestra el mensaje de éxito: “Solicitud aceptada con éxito”. Se muestra el cambio en el estado de la solicitud.

Tabla 6.35: CP-35: Aceptar solicitud pendiente para horario en el que el profesor ya tiene tutorías o solicitudes pendientes de confirmación

CP-36	Aceptar solicitud pendiente proponiendo horario en el que existen solicitudes por confirmar o tutorías
Descripción	Se intenta aceptar una solicitud de tutoría en un espacio de tiempo en el que el profesor ya tiene tutorías o solicitudes por confirmar.
Actor	Profesor
Escenario	Vista “Solicitud tutoría”
Entrada	<p>Se pulsa el botón “Aceptar solicitud”.</p> <p>Se muestra un diálogo modal.</p> <p>Se muestra un mensaje de advertencia con el texto: “Tienes tutorías o solicitudes por confirmar en el espacio de tiempo de esta solicitud”.</p> <p>Se propone un nuevo horario, por lo que se introduce una nueva hora de inicio y duración estimada, pero en este caso con datos en los que ya existan tutorías o solicitudes por confirmar.</p> <p>Se introduce el lugar donde tendrá lugar la tutoría.</p> <p>Se introduce un comentario que pasa a ser obligatorio.</p> <p>Se pulsa el botón “Aceptar”.</p>
Resultado esperado	Se muestra el mensaje de error: “Existen tutorías en ese espacio de tiempo” o “Existen solicitudes por confirmar en ese espacio de tiempo”.

Tabla 6.36: CP-36: Aceptar solicitud pendiente proponiendo horario en el que existen solicitudes por confirmar o tutorías

HU08 - Rechazar solicitud de tutoría

CP-37	Rechazar solicitud pendiente
Descripción	Se rechaza una solicitud de tutoría en estado pendiente.
Actor	Profesor
Escenario	Vista “Solicitud tutoría”
Entrada	Se pulsa el botón “Rechazar solicitud”. Se muestra un diálogo modal. Se introduce el motivo de rechazo de la tutoría. Se pulsa el botón “Aceptar”.
Resultado esperado	Se muestra el mensaje de éxito: “Solicitud rechazada con éxito” Se muestra el cambio en el estado de la solicitud.

Tabla 6.37: CP-37: Rechazar solicitud pendiente

CP-38	Rechazar solicitud pendiente sin indicar motivo de rechazo
Descripción	Se intenta rechazar una solicitud de tutoría en estado pendiente sin introducir ningún motivo de rechazo.
Actor	Profesor
Escenario	Vista “Solicitud tutoría”
Entrada	Se pulsa el botón “Rechazar solicitud”. Se muestra un diálogo modal. Se pulsa el botón “Aceptar”.
Resultado esperado	El botón de “Aceptar” está deshabilitado por lo que no se puede rechazar la solicitud.

Tabla 6.38: CP-38: Rechazar solicitud pendiente sin indicar motivo de rechazo

CP-39	Rechazar solicitud pendiente indicando un motivo de rechazo con menos de 10 caracteres
Descripción	Se intenta rechazar una solicitud de tutoría en estado pendiente introduciendo un motivo con menos de 10 caracteres.
Actor	Profesor
Escenario	Vista “Solicitud tutoría”
Entrada	Se pulsa el botón “Rechazar solicitud”. Se muestra un diálogo modal. Se introduce un motivo de rechazo de 9 caracteres. Se pulsa el botón “Aceptar”.
Resultado esperado	El botón de “Aceptar” está deshabilitado por lo que no se puede rechazar la solicitud.

Tabla 6.39: CP-39: Rechazar solicitud pendiente indicando un motivo de rechazo con menos de 10 caracteres

HU19 - Confirmar solicitud de tutoría

CP-40	Confirmar solicitud de tutoría
Descripción	Se confirma una solicitud de tutoría en el estado aceptada.
Actor	Alumno
Escenario	Vista “Solicitud tutoría”
Entrada	Se pulsa el botón “Confirmar solicitud”. Se muestra un diálogo modal de confirmación con el mensaje: “Vas a confirmar tu asistencia a la tutoría, ¿quieres continuar?”. Se pulsa el botón “Si”.
Resultado esperado	Se muestra el mensaje de éxito: “La solicitud se ha confirmado” Se muestra el cambio en el estado de la solicitud.

Tabla 6.40: CP-40: Confirmar solicitud de tutoría

HU09 - Consultar solicitudes mediante filtros

CP-41	Consultar solicitudes mediante filtros (Profesor)
Descripción	Se consultan las solicitudes de tutoría mediante diversos filtros (estado de la solicitud, fecha de la tutoría, si es grupal o no, o por asignatura impartida por el profesor).
Actor	Profesor
Escenario	Vista “Consultar solicitudes”
Entrada	Se selecciona el estado de la solicitud. Se introduce una fecha de tutoría. Se indica si es grupal. Se selecciona una asignatura. Se pulsa el botón buscar.
Resultado esperado	Se muestran los registros encontrados mediante los filtros o, en caso de no encontrar resultados, un mensaje con el siguiente texto: “No se encontraron resultados”.

Tabla 6.41: CP-41: Consultar solicitudes mediante filtros (Profesor)

CP-42	Consultar solicitudes mediante filtros (Alumno)
Descripción	Se consultan las solicitudes de tutoría mediante diversos filtros (estado de la solicitud, fecha de la tutoría, si es grupal o no, el profesor destinatario, o por asignatura).
Actor	Alumno
Escenario	Vista “Consultar solicitudes”
Entrada	Se selecciona el estado de la solicitud. Se introduce una fecha de tutoría. Se indica si es grupal. Se introduce el profesor. Se selecciona una asignatura. Se pulsa el botón buscar.
Resultado esperado	Se muestran los registros encontrados mediante los filtros o, en caso de no encontrar resultados, un mensaje con el siguiente texto: “No se encontraron resultados”.

Tabla 6.42: CP-42: Consultar solicitudes mediante filtros (Alumno)

HU13 - Consultar calendario de tutorías

CP-43	Ver detalle de tutoría
Descripción	Se consulta el detalle de una tutoría mediante el calendario de tutorías.
Actor	Usuario
Escenario	Vista “Calendario de tutorías”
Entrada	Se pulsa en una evento con punto de color azul, el cual simboliza que es una tutoría.
Resultado esperado	Se muestra el detalle de la tutoría.

Tabla 6.43: CP-43: Ver detalle de tutoría

CP-44	Ver detalle de solicitud tutoría aceptada
Descripción	Se consulta el detalle de una solicitud de tutoría que está pendiente de ser confirmada, mediante el calendario de tutorías.
Actor	Profesor
Escenario	Vista “Calendario de tutorías”
Entrada	Se pulsa en una evento con punto de color verde, el cual simboliza que es una solicitud en estado aceptada.
Resultado esperado	Se muestra el detalle de la solicitud de tutoría en estado aceptada

Tabla 6.44: CP-44: Ver detalle de solicitud tutoría aceptada

CP-45	Ver más información de una tutoría mediante el calendario de tutorías
Descripción	Se consulta el detalle de una tutoría y se pulsa en el botón “Ver tutoría” para conocer más detalladamente la tutoría.
Actor	Usuario
Escenario	Vista “Calendario de tutorías”
Entrada	Se pulsa en una evento con punto de color azul, el cual simboliza que es una tutoría. Se muestra el detalle de la tutoría. Se pulsa en el botón “Ver más” .
Resultado esperado	Se navega hasta la vista “Tutoría” de la tutoría.

Tabla 6.45: CP-45: Ver más información de una tutoría mediante el calendario de tutorías

CP-46	Ver más información de una solicitud pendiente de confirmación mediante el calendario de tutorías
Descripción	Se consulta el detalle de una solicitud de tutoría que está pendiente de ser confirmada y se pulsa en el botón “Ver más” para conocer más detalladamente la solicitud.
Actor	Profesor
Escenario	Vista “Calendario de tutorías”
Entrada	Se pulsa en una evento con punto de color verde, el cual simboliza que es una solicitud en estado aceptada. Se muestra el detalle de la solicitud de tutoría en estado aceptada. Se pulsa en el botón “Ver más” .
Resultado esperado	Se navega hasta la vista “Solicitud tutoría” de la tutoría en estado aceptada.

Tabla 6.46: CP-46: Ver más información de una solicitud pendiente de confirmación mediante el calendario de tutorías

HU14 - Consultar listado de tutorías

CP-47	Consultar próximas tutorías
Descripción	Se consultan las próximas tutorías que tendrán lugar.
Actor	Usuario
Escenario	Vista “Consultar tutorías”
Entrada	Se selecciona en el ComboBox la opción “Próximamente”.
Resultado esperado	Se muestran las tutorías que tendrán lugar próximamente. No se incluyen las tutorías canceladas ni las tutorías cuya fecha de finalización ya ha pasado.

Tabla 6.47: CP-47: Consultar próximas tutorías

CP-48	Consultar tutorías que tienen lugar en el día de hoy
Descripción	Se consultan las tutorías que tendrán lugar en el día de hoy.
Actor	Usuario
Escenario	Vista “Consultar tutorías”
Entrada	Se selecciona en el ComboBox la opción “Hoy”.
Resultado esperado	Se muestran las tutorías que tendrán lugar hoy. No se incluyen las tutorías canceladas.

Tabla 6.48: CP-48: Consultar tutorías que tienen lugar en el día de hoy

CP-49	Consultar tutorías mediante filtros (Alumno)
Descripción	Se consultan las tutorías mediante filtros.
Actor	Alumno
Escenario	Vista “Consultar tutorías”
Entrada	Se selecciona en el ComboBox la opción “Filtrar”. Se introduce o no la fecha de tutoría. Se indica si está cancelada la tutoría. Se indica si la tutoría es grupal. Se selecciona o no al profesor de la tutoría. Se selecciona o no la asignatura de la tutoría. Se pulsa en el botón “Buscar”.
Resultado esperado	Se muestran las tutorías que coinciden con los filtros seleccionados.

Tabla 6.49: CP-49: Consultar tutorías mediante filtros (Alumno)

CP-50	Consultar tutorías mediante filtros (Profesor)
Descripción	Se consultan las tutorías mediante filtros.
Actor	Profesor
Escenario	Vista “Consultar tutorías”
Entrada	Se selecciona en el ComboBox la opción “Filtrar”. Se introduce o no la fecha de tutoría. Se indica si está cancelada la tutoría. Se indica si la tutoría es grupal. Se selecciona o no la asignatura de la tutoría. Se pulsa en el botón “Buscar”.
Resultado esperado	Se muestran las tutorías que coinciden con los filtros seleccionados.

Tabla 6.50: CP-50: Consultar tutorías mediante filtros (Profesor)

CP-51	Consultar tutoría
Descripción	Se consulta el detalle de una tutoría.
Actor	Usuario
Escenario	Vista “Consultar tutorías”
Entrada	De alguna de las tutorías listadas se pulsa el botón con el tooltip “Ver tutoría”
Resultado esperado	Se muestra la tutoría detalladamente. Si ésta no está cancelada se muestra un botón para realizar la acción.

Tabla 6.51: CP-51: Consultar tutoría

HU15 - Cancelar tutoría

CP-52	Cancelar tutoría
Descripción	Se cancela una tutoría.
Actor	Usuario
Escenario	Vista "Tutoría"
Entrada	Se pulsa el botón "Cancelar solicitud". Se muestra un diálogo modal en el que se tiene que introducir un motivo de cancelación de al menos 10 caracteres. Se introduce el motivo de cancelación. Se pulsa el botón "Aceptar"
Resultado esperado	Se muestra el mensaje de éxito: "Tutoría cancelada con éxito". Se muestran los cambios en la pantalla: En el título se muestra el texto "(cancelada)" y desaparece el botón "Cancelar tutoría".

Tabla 6.52: CP-52: Cancelar tutoría

HU10 - Consultar perfil

CP-53	Consultar perfil
Descripción	Se consulta el perfil del usuario que ha iniciado sesión.
Actor	Usuario
Escenario	Vista "Perfil"
Entrada	Ninguna.
Resultado esperado	Se muestra el perfil del usuario, donde se muestran sus datos y asignaturas en las que está matriculado. En el caso del profesor, se muestra su horario de tutorías.

Tabla 6.53: CP-53: Consultar perfil

HU21 - Consultar solicitudes pendientes desde la pantalla de inicio

CP-54	Consultar solicitudes pendientes
Descripción	Se consultan las solicitudes pendientes desde la pantalla de inicio.
Actor	Profesor
Escenario	Vista “Inicio”
Entrada	Ninguna.
Resultado esperado	Se muestran las solicitudes pendientes en forma de carrusel. En caso de no haber solicitudes pendientes no se mostrará contenido en el carrusel.

Tabla 6.54: CP-54: Consultar solicitudes pendientes

HU22 - Consultar solicitudes aceptadas desde la pantalla de inicio

CP-55	Consultar solicitudes aceptadas
Descripción	Se consultan las solicitudes aceptadas desde la pantalla de inicio.
Actor	Alumno
Escenario	Vista “Inicio”
Entrada	Ninguna.
Resultado esperado	Se muestran las solicitudes aceptadas en forma de carrusel. En caso de no haber solicitudes aceptadas no se mostrará contenido en el carrusel.

Tabla 6.55: CP-55: Consultar solicitudes pendientes

HU24 - Consultar las próximas tutorías desde la pantalla de inicio

CP-56	Consultar próximas tutorías
Descripción	Se consultan las próximas tutorías desde la pantalla de inicio.
Actor	Usuario
Escenario	Vista “Inicio”
Entrada	Ninguna.
Resultado esperado	Se muestran las próximas tutorías en forma de carrusel. En caso de no haber tutorías próximamente no se mostrará contenido en el carrusel.

Tabla 6.56: CP-56: Consultar próximas tutorías

HU23 - Rechazar automáticamente las solicitudes pendientes que estén caducadas

CP-57	Rechazar automáticamente las solicitudes pendientes caducadas
Descripción	Se rechazan de manera automática las solicitudes pendientes cuya fecha de tutoría y hora de inicio ya hayan pasado.
Actor	Tarea automatizada
Escenario	Fichero application.properties.
Entrada	En el application.properties se ha establecido una propiedad en la que se indica cuándo se realizará automáticamente el rechazo de estas solicitudes. Para esta prueba se establece un horario cercano y se crean unas solicitudes pendientes con una fecha de tutoría que ya haya pasado, así como la hora de inicio y fin, en la Base de Datos. Se espera hasta el horario establecido.
Resultado esperado	Las solicitudes tutorías caducadas pasan a estar en el estado "RECHAZADA" y con el motivo de cancelación: "Solicitud rechazada automáticamente. Motivo: Solicitud caducada".

Tabla 6.57: CP-57: Rechazar automáticamente las solicitudes pendientes caducadas

HU12 - Cargar horario mediante fichero de extensión csv

CP-58	Cargar un fichero que cumple con el formato adecuado
Descripción	<p>Se carga un fichero de extensión csv cuyo formato respeta el siguiente formato:</p> <ul style="list-style-type: none"> ■ Cabecera con 6 campos: dia, horaInicio, horaFin, centro, despacho, tipoHorario. ■ Columna 'dia': nombre en castellano del día en el que se desea establecer el horario. No importa si la cadena lleva tildes ni si está en mayúsculas o minúsculas o una combinación de ambas. Solo se permiten días de lunes a viernes. Ejemplos válidos: 'Miercoles', 'Miércoles', 'miercoLes'. ■ Columna 'horaInicio': cadena en formato 'HH:mm'. Representa la hora de inicio de la franja de tutorías. ■ Columna 'horaFin': cadena en formato 'HH:mm'. Representa la hora de fin de la franja de tutorías. ■ Columna 'centro': cadena de caracteres que representa el centro académico en el que el profesor estará presente durante la franja del horario. ■ Columna 'despacho': cadena de caracteres que representa el despacho del profesor en el que se encontrará en la franja de tutorías. ■ Columna 'tipoHorario': número que representa el tipo de horario. Puede tomar el valor '1' para indicar que es una franja para el primer cuatrimestre, el valor '2' para el segundo cuatrimestre, y el valor '3' si es anual. En caso de que se quiera cargar un horario anual solo puede incluir este tipo de horario, por el contrario, si se trata de un horario que se distingue según el cuatrimestre, no podrá incluir el tipo de horario anual.
Actor	Profesor
Escenario	Vista "Perfil"
Entrada	<p>Se pulsa el botón "Cargar nuevo horario".</p> <p>Se abre un diálogo modal con el mensaje: "Seleccione un fichero en formato csv:"</p> <p>Se pulsa el botón seleccionar y se selecciona el fichero csv con el nuevo horario acorde al formato establecido. También se puede arrastrar el fichero y soltar sobre la caja de contenido.</p> <p>Se muestra el fichero que se quiere subir.</p> <p>Se pulsa el botón "Subir"</p>
Resultado esperado	Se muestra el mensaje de éxito: "Horario cargado con éxito".

Tabla 6.58: CP-58: Cargar un fichero que cumple con el formato adecuado

CP-59	Cargar un fichero csv sin contenido
Descripción	Se carga un fichero de extensión csv sin contenido.
Actor	Profesor
Escenario	Vista “Perfil”
Entrada	Se pulsa el botón “Cargar nuevo horario”. Se abre un diálogo modal con el mensaje: “Seleccione un fichero en formato csv:” Se pulsa el botón seleccionar y se selecciona el fichero csv sin contenido. También se puede arrastrar el fichero y soltar sobre la caja de contenido. Se muestra el fichero que se quiere subir. Se pulsa el botón “Subir”.
Resultado esperado	Se muestra el mensaje de error: “El fichero no cumple con el formato”.

Tabla 6.59: CP-59: Cargar un fichero csv sin contenido

CP-60	Cargar un fichero csv con una cabecera incorrecta
Descripción	Se carga un fichero de extensión csv que no cumple con la cabecera establecida (insertar referencia).
Actor	Profesor
Escenario	Vista “Perfil”
Entrada	Se pulsa el botón “Cargar nuevo horario”. Se abre un diálogo modal con el mensaje: “Seleccione un fichero en formato csv:” Se pulsa el botón seleccionar y se selecciona el fichero csv con la cabecera incorrecta. También se puede arrastrar el fichero y soltar sobre la caja de contenido. Se muestra el fichero que se quiere subir. Se pulsa el botón “Subir”.
Resultado esperado	Se muestra el mensaje de error: “El fichero no cumple con el formato”.

Tabla 6.60: CP-60: Cargar un fichero csv con una cabecera incorrecta

CP-61	Cargar un fichero csv con un tipo de horario anual y cuatrimestral
Descripción	Se carga un fichero de extensión csv que no cumple con la condición de que cuando existe un elemento con horario de tipo anual, no puede haber elementos con horario de tipos cuatrimestrales.
Actor	Profesor
Escenario	Vista "Perfil"
Entrada	Se pulsa el botón "Cargar nuevo horario". Se abre un diálogo modal con el mensaje: "Seleccione un fichero en formato csv:" Se pulsa el botón seleccionar y se selecciona el fichero csv. También se puede arrastrar el fichero y soltar sobre la caja de contenido. Se muestra el fichero que se quiere subir. Se pulsa el botón "Subir".
Resultado esperado	Se muestra el mensaje de error: "El fichero no cumple con el formato".

Tabla 6.61: CP-61: Cargar un fichero csv con un tipo de horario anual y cuatrimestral

CP-62	Cargar un fichero csv con alguna hora de inicio mayor que la hora de fin
Descripción	Se carga un fichero de extensión csv que contiene algún elemento en el que la hora de fin es inferior a la hora de inicio.
Actor	Profesor
Escenario	Vista "Perfil"
Entrada	Se pulsa el botón "Cargar nuevo horario". Se abre un diálogo modal con el mensaje: "Seleccione un fichero en formato csv:" Se pulsa el botón seleccionar y se selecciona el fichero csv. También se puede arrastrar el fichero y soltar sobre la caja de contenido. Se muestra el fichero que se quiere subir. Se pulsa el botón "Subir".
Resultado esperado	Se muestra el mensaje de error: "Los nuevos horarios de tutorías no son válidos".

Tabla 6.62: CP-62: Cargar un fichero csv con alguna hora de inicio mayor que la hora de fin

CP-63	Cargar un fichero csv con alguna hora de inicio mayor que la hora de fin
Descripción	Se carga un fichero de extensión csv que contiene algún elemento en el que la hora de fin es inferior a la hora de inicio.
Actor	Profesor
Escenario	Vista “Perfil”
Entrada	Se pulsa el botón “Cargar nuevo horario”. Se abre un diálogo modal con el mensaje: “Seleccione un fichero en formato csv:” Se pulsa el botón seleccionar y se selecciona el fichero csv. También se puede arrastrar el fichero y soltar sobre la caja de contenido. Se muestra el fichero que se quiere subir. Se pulsa el botón “Subir”.
Resultado esperado	Se muestra el mensaje de error: “Los nuevos horarios de tutorías no son válidos”.

Tabla 6.63: CP-63: Cargar un fichero csv con alguna hora de inicio mayor que la hora de fin

CP-64	Cargar un fichero csv con cadena que no representa un día en la columna ‘dia’
Descripción	Se carga un fichero de extensión csv que contiene algún elemento en el que la columna ‘día’ contiene una cadena que no representa un día de la semana de lunes a viernes.
Actor	Profesor
Escenario	Vista “Perfil”
Entrada	Se pulsa el botón “Cargar nuevo horario”. Se abre un diálogo modal con el mensaje: “Seleccione un fichero en formato csv:” Se pulsa el botón seleccionar y se selecciona el fichero csv. También se puede arrastrar el fichero y soltar sobre la caja de contenido. Se muestra el fichero que se quiere subir. Se pulsa el botón “Subir”.
Resultado esperado	Se muestra el mensaje de error: “El fichero no cumple con el formato”.

Tabla 6.64: CP-64: Cargar un fichero csv con cadena que no representa un día en la columna ‘dia’

HU16 - Acceder a la información de un profesor mediante el bot de Telegram

CP-65	Consultar información de un profesor sin identificarse
Descripción	Se quiere consultar la información de un profesor sin haberse identificado previamente.
Actor	Alumno
Escenario	Bot de Telegram "TutoriasInfUVaBot"
Entrada	Se introduce el comando <code>"/buscar_profesor"</code> .
Resultado esperado	Se muestra el mensaje: "Para utilizar las funciones del bot es necesario que inicie sesión con el comando <code>/login</code> ".

Tabla 6.65: CP-65: Consultar información de un profesor sin identificarse

CP-66	Iniciar sesión en el bot de Telegram con credenciales válidas
Descripción	Se inicia sesión en el bot de Telegram con una credenciales válidas.
Actor	Alumno
Escenario	Bot de Telegram "TutoriasInfUVaBot"
Entrada	Se introduce el comando <code>"/login"</code> . Se muestra el mensaje "Introduce tu nombre de usuario de la Escuela:". Se introduce el nombre de usuario de la Escuela. Se muestra el mensaje "Introduce tu contraseña:". Se introduce la contraseña.
Resultado esperado	Se muestra el mensaje: "Sesión iniciada: ¡Bienvenid@, nombreAlumno apellidosAlumno!".

Tabla 6.66: CP-66: Iniciar sesión en el bot de Telegram con credenciales válidas

CP-67	Acceder a la información de un profesor mediante un tipo de búsqueda general
Descripción	Se busca un profesor a través del bot de Telegram, realizando una búsqueda de tipo general.
Actor	Alumno
Escenario	Bot de Telegram “TutoriasInfUVaBot”
Entrada	<p>Se introduce el comando “/buscar_profesor”.</p> <p>Se muestra el mensaje “¿De qué manera quieres empezar a buscar al profesor?” y las opciones: “Búsqueda general”, “A partir de mis asignaturas” y “Por nombre de asignatura”.</p> <p>Se selecciona la opción “Búsqueda general”.</p> <p>Se muestra el mensaje “Selecciona una titulación” y las opciones que se corresponden con las titulaciones del alumno.</p> <p>Se selecciona una titulación.</p> <p>Se muestra el mensaje “Selecciona un curso:” y como opciones se muestran los cursos de la titulación.</p> <p>Se selecciona un curso.</p> <p>En este paso existen 2 posibilidades:</p> <ol style="list-style-type: none"> 1. Si el curso tiene menciones asociadas. 2. Si el curso no tiene menciones asociadas, el caso 1 no se realiza y se salta hasta el 2. <p>Caso 1:</p> <p>Se muestra el mensaje “Selecciona una mención:” y como opciones se muestran las menciones del curso académico.</p> <p>Se selecciona una mención.</p> <p>Caso 2:</p> <p>Se muestra el mensaje “Selecciona una asignatura” y como opciones se muestran las asignaturas del curso académico (y mención en caso de haber tenido que seleccionar una).</p> <p>Se selecciona una asignatura.</p> <p>Se muestra el mensaje “Selecciona un profesor” y como opciones se muestran los profesores que imparten la asignatura escogida.</p> <p>Se selecciona un profesor.</p>
Resultado esperado	Se muestra el mensaje con la siguiente información del profesor: nombre completo, email, teléfono, centro habitual y despacho.

Tabla 6.67: CP-67: Acceder a la información de un profesor mediante un tipo de búsqueda general

CP-68	Acceder a la información de un profesor buscando a partir de mis asignaturas
Descripción	Se busca un profesor a través del bot de Telegram, realizando una búsqueda a partir de las asignaturas del alumno que ha iniciado sesión.
Actor	Alumno
Escenario	Bot de Telegram "TutoriasInfUVaBot"
Entrada	Se introduce el comando "/buscar_profesor". Se muestra el mensaje "¿De qué manera quieres empezar a buscar al profesor?" y las opciones: "Búsqueda general", "A partir de mis asignaturas" y "Por nombre de asignatura". Se selecciona la opción "A partir de mis asignaturas". Se muestra el mensaje "Selecciona una asignatura" y como opciones se muestran las asignaturas del alumno. Se selecciona una asignatura. Se muestra el mensaje "Selecciona un profesor" y como opciones se muestran los profesores que imparten la asignatura escogida. Se selecciona un profesor.
Resultado esperado	Se muestra el mensaje con la siguiente información del profesor: nombre completo, email, teléfono, centro habitual y despacho.

Tabla 6.68: CP-68: Acceder a la información de un profesor buscando a partir de mis asignaturas

CP-69	Acceder a la información de un profesor buscando a partir del nombre de una asignatura
Descripción	Se busca un profesor a través del bot de Telegram, realizando una búsqueda a partir del nombre de una asignatura.
Actor	Alumno
Escenario	Bot de Telegram "TutoriasInfUVaBot"
Entrada	Se introduce el comando "/buscar_profesor". Se muestra el mensaje "¿De qué manera quieres empezar a buscar al profesor?" y las opciones: "Búsqueda general", "A partir de mis asignaturas" y "Por nombre de asignatura". Se selecciona la opción "Por nombre asignatura". Se muestra el mensaje: "Introduce el nombre de la asignatura:" Se introduce una cadena de caracteres. Se muestra el mensaje "Selecciona una asignatura" y como opciones se muestran las asignaturas que contienen la cadena introducida. Se selecciona una asignatura. Se muestra el mensaje "Selecciona un profesor" y como opciones se muestran los profesores que imparten la asignatura escogida. Se selecciona un profesor.
Resultado esperado	Se muestra el mensaje con la siguiente información del profesor: nombre completo, email, teléfono, centro habitual y despacho.

Tabla 6.69: CP-69: Acceder a la información de un profesor buscando a partir del nombre de una asignatura

CP-70	Acceder a la información de un profesor buscando a partir del nombre de una asignatura
Descripción	Se busca un profesor a través del bot de Telegram mediante el tipo de búsqueda por nombre, y se introduce una cadena texto que no coincide con ninguna asignatura.
Actor	Alumno
Escenario	Bot de Telegram "TutoriasInfUVaBot"
Entrada	Se sigue el proceso correspondiente al tipo de búsqueda seleccionado. Se selecciona una asignatura que no tiene docencia.
Resultado esperado	Se muestra el mensaje: "Asignatura sin docencia"

Tabla 6.70: CP-70: Acceder a la información de un profesor buscando a partir del nombre de una asignatura

CP-71	Buscar asignatura mediante el tipo de búsqueda por nombre de asignatura e introducir una cadena de texto que no coincida con ninguna asignatura
Descripción	Se busca un profesor a través del bot de Telegram mediante cualquier tipo de búsqueda y se selecciona una asignatura que no tiene docencia.
Actor	Alumno
Escenario	Bot de Telegram "TutoriasInfUVaBot"
Entrada	Se introduce el comando "/buscar_profesor". Se muestra el mensaje "¿De qué manera quieres empezar a buscar al profesor?" y las opciones: "Búsqueda general", "A partir de mis asignaturas" y "Por nombre de asignatura". Se selecciona la opción "Por nombre asignatura". Se muestra el mensaje: "Introduce el nombre de la asignatura." Se introduce una cadena de caracteres que no coincida con el nombre de alguna asignatura.
Resultado esperado	Se muestra el mensaje: "No se encontraron resultados"

Tabla 6.71: CP-71: Buscar asignatura mediante el tipo de búsqueda por nombre de asignatura e introducir una cadena de texto que no coincida con ninguna asignatura

CP-72	Terminar la sesión con el bot
Descripción	Se quiere terminar la sesión iniciada con el bot.
Actor	Alumno
Escenario	Bot de Telegram "TutoriasInfUVaBot"
Entrada	Se introduce el comando "/quit".
Resultado esperado	Se muestra el mensaje: "Sesión terminada. ¡Hasta la próxima, nombreAlumno apellidosAlumno!"

Tabla 6.72: CP-72: Terminar la sesión con el bot

CP-73	Intentar buscar un profesor cuando la API de TutoriasInfUVa no está disponible
Descripción	Se intenta buscar un profesor cuando la API de TutoriasInfUVa no se encuentra disponible.
Actor	Alumno
Escenario	Bot de Telegram “TutoriasInfUVaBot”
Entrada	Se introduce el comando “/buscar_profesor”. Se muestra el mensaje “¿De qué manera quieres empezar a buscar al profesor?” y las opciones: “Búsqueda general”, “A partir de mis asignaturas” y “Por nombre de asignatura”. Se selecciona una opción. En caso de seleccionar “Por nombre de asignatura”, se introduce el nombre.
Resultado esperado	Se muestra el mensaje: “Proceso terminado con error. Póngase en contacto con los técnicos”.

Tabla 6.73: CP-73: Intentar buscar un profesor cuando la API de TutoriasInfUVa no está disponible

HU17 - Acceder a la aplicación web TutoriasInfUVa mediante el bot de Telegram

CP-74	Acceder a la web mediante el bot de Telegram sin haberse identificado previamente
Descripción	Se quiere acceder a la web de TutoriasInfUVa mediante el bot de Telegram, TutoriasInfUVaBot, sin haberse identificado previamente.
Actor	Alumno
Escenario	Bot de Telegram “TutoriasInfUVaBot”
Entrada	Se introduce el comando “/web”
Resultado esperado	Se muestra el mensaje: “Para utilizar las funciones del bot es necesario que inicie sesión con el comando /login”.

Tabla 6.74: CP-74: Acceder a la web mediante el bot de Telegram sin haberse identificado previamente

CP-75	Acceder a la web mediante el bot de Telegram habiendo iniciado sesión previamente
Descripción	Se quiere acceder a la web de TutoriasInfUVa mediante el bot de Telegram, TutoriasInfUVaBot, habiéndose identificado previamente.
Actor	Alumno
Escenario	Bot de Telegram “TutoriasInfUVaBot”
Entrada	Se introduce el comando “/web” Se muestra un botón con el texto “Abrir TutoriasInfUVa”. Se pulsa el botón “Abrir TutoriasInfUVa”. Se muestra un diálogo con el mensaje “¿Abrir este enlace?” y un enlace a la web. Se pulsa el botón “Abrir”.
Resultado esperado	Se navega hasta la web TutoriasInfUVa.

Tabla 6.75: CP-75: Acceder a la web mediante el bot de Telegram habiendo iniciado sesión previamente

6.2.2. Resultados de las pruebas

A continuación, se muestran los resultados de los casos de prueba y, en caso de error, se indica como se ha solucionado el problema.

Caso de prueba	Resultado	Solución
CP-01	OK	-
CP-02	Fallo: cuando se escriben espacios al principio o al final, no se encuentran los mismos resultados que se deberían encontrar al no poner dichos espacios.	Se ha añadido el código necesario para que no se de dicho error. Dicho código se basa en eliminar los espacios en blanco en el string resultante del formulario.
CP-03	OK	-
CP-04	OK	-
CP-05	OK	-
CP-06	OK	-
CP-07	OK	-
CP-08	OK	-
CP-09	Fallo: está programado para un caso específico (el Grado en Ingeniería Informática), pero para otros casos puede variar y tener otras menciones u otros cursos y no se está teniendo en cuenta.	Para resolver este problema se ha tenido que modificar tanto el diseño de la base de datos como el de modelo de dominio, tal y como se indica en la Sección 7.1.4, del capítulo dedicado al seguimiento.

Tabla 6.76 continuación de la página anterior

Caso de prueba	Resultado	Solución
CP-10	Fallo: Al igual que en el CP-09, funciona, pero únicamente para el caso de ejemplo que se está probando.	Misma solución que en el CP-09.
CP-11	Fallo: No se muestra el mensaje “No se encontraron resultados”	En caso de que no se encuentren resultados, el accordion no se mostrará y en su lugar se mostrará el mensaje en cuestión. Para ello se crea un template, donde se encuentra el mensaje, al que solo se accede si los elementos del accordion son igual a 0.
CP-12	OK	-
CP-13	OK	-
CP-14	OK	-
CP-15	OK	-
CP-16	OK	-
CP-17	Error: se permiten solicitar tutorías en un intervalo de tiempo en el que el usuario ya tiene otras tutorías.	Para resolver este problema, se ha comprobado si el alumno tiene tutorías en el intervalo de tiempo en el que ha solicitado la nueva tutoría.
CP-18	OK	-
CP-19	OK	-
CP-20	OK	-
CP-21	OK	-
CP-22	OK	-
CP-23	OK	-
CP-24	OK	-
CP-25	OK	-
CP-26	OK	-
CP-27	OK	-
CP-28	OK	-
CP-29	OK	-
CP-30	OK	-
CP-31	OK	-
CP-32	OK	-
CP-33	OK	-
CP-34	OK	-
CP-35	OK	-
CP-36	OK	-
CP-37	OK	-
CP-38	OK	-

Tabla 6.76 continuación de la página anterior

Caso de prueba	Resultado	Solución
CP-39	OK	-
CP-40	OK	-
CP-41	OK	-
CP-42	OK	-
CP-43	OK	-
CP-44	OK	-
CP-45	OK	-
CP-46	OK	-
CP-47	Fallo: Se muestran las tutorías canceladas.	Se ha añadido en la consulta sql que las tutorías no estén canceladas.
CP-48	Fallo: Se muestran las tutorías canceladas.	Se ha añadido en la consulta sql que las tutorías no estén canceladas.
CP-49	OK	-
CP-50	OK	-
CP-51	OK	-
CP-52	OK	-
CP-53	OK	-
CP-54	OK	-
CP-55	OK	-
CP-56	OK	-
CP-57	OK	-
CP-58	OK	-
CP-59	OK	-
CP-60	OK	-
CP-61	OK	-
CP-62	Fallo: El horario se carga con éxito y no se muestra el mensaje de error.	Se ha comprobado que en cada franja la hora de inicio sea menor que la hora de fin.
CP-63	OK	-
CP-64	OK	-
CP-65	OK	-
CP-66	OK	-
CP-67	OK	-
CP-68	OK	-
CP-69	OK	-
CP-70	OK	-
CP-71	OK	-
CP-72	OK	-

Tabla 6.76 continuación de la página anterior

Caso de prueba	Resultado	Solución
CP-73	Error: el bot no responde con nada.	El problema se debía a que no se controlaba la excepción WebClientRequestException producida al tratar de realizar peticiones al servidor cuando este no está disponible.
CP-74	OK	-
CP-75	OK	-

Tabla 6.76: Resultados de los casos de prueba y sus respectivas soluciones

Capítulo 7

Seguimiento del proyecto

7.1. Seguimiento de los sprints realizados

7.1.1. Sprint 0 (08/02/2022 - 23/02/2022)

Se emplea el sprint 0 para la realización de las tareas de planificación y documentación inicial, además de la descarga del software necesario para el trabajo. También, se realizan las labores de análisis y diseño inicial de la aplicación. Para este sprint se han empleado dos semanas en las que se ha trabajado un total de **20 horas**.

En la Tabla 7.1 se puede ver el resumen de las tareas realizadas durante el sprint 0.

7.1.2. Sprint 1 (23/02/2022 - 09/03/2022)

Tras las tareas de documentación inicial del sprint 0, en este sprint comienza la ejecución del proyecto. Durante la reunión de Sprint Planning se añaden al Sprint Backlog las tareas referentes a las historias de usuario **HU03** y **HU04**, que se corresponden a la búsqueda de un profesor según diferentes filtros y la visualización de la información detallada de un profesor por parte de un alumno, respectivamente.

El tiempo dedicado a este sprint es de un total de **40 horas y 22 minutos** de las 50 estimadas. El que se haya trabajado menos tiempo del estimado se debe, principalmente, a la ocurrencia del riesgo **R06**, “**Ausencia por consulta médica u otra actividad**”, puesto que durante la primera semana del sprint el alumno tuvo que asistir al médico y otras actividades en varias ocasiones, es por ello que durante la primera semana únicamente se pudieron trabajar 10 horas, quedando así muy lejos de las 25 horas semanales estimadas. Debido a este riesgo se aplicó durante la siguiente semana de sprint el plan de contingencia por el que se trabajó durante fines de semana. Sin embargo, durante la segunda semana de

Tarea	Tiempo empleado	Estado
Documentación de objetivos	1 hora	Completada
Definición del Product Backlog inicial	2 horas	Completada
Documentación de la planificación inicial y calendarización	3 horas	Completada
Documentación del plan de riesgos	3 horas	Completada
Documentación de la estimación de costes	1 hora y 30 minutos	Completada
Documentación del marco de trabajo	2 horas y 30 minutos	Completada
Descarga del software necesario	1 hora y 30 minutos	Completada
Análisis: descripción de historias de usuario iniciales y modelo de dominio inicial	3 horas y 30 minutos	Completada
Diseño: arquitectura inicial	2 horas	Completada

Tabla 7.1: Tareas del sprint 0

sprint se produjeron otros riesgos, en concreto, el riesgo **R01**, “**Falta de formación del equipo de desarrollo**”, debido a que la poca experiencia con la configuración del backend de la aplicación provocó que se comenzara tarde con el desarrollo y que se trabajase más horas de las estimadas en la historia de usuario *HU03*, provocando, además, que no se pudiera comenzar con la otra historia de usuario programada para el sprint.

Por estas razones, las tareas de diseño, implementación y pruebas de la *HU03* y todas las tareas de la *HU04* se pasan al siguiente sprint. Además, durante la reunión de Sprint Review, la tutora indica al alumno que ordene las historias de usuario por prioridad para facilitar la selección de historias de usuario durante la planificación del sprint.

En la Tabla 7.2 se puede ver el resumen de las tareas realizadas durante el sprint 1.

7.1.3. Sprint 2 (09/03/2022 - 23/03/2022)

Durante el Sprint Planning de este nuevo sprint se decide, que a parte de las tareas inacabadas del sprint anterior, se va a añadir al Sprint Backlog una tarea pequeña que no suponga una gran parte del tiempo pero que permita avanzar en el proyecto. Es por ello, que la historia de usuario **HU05** da lugar a una historia de usuario más pequeña cuyo identificador se decide que va a ser el **HU18**, para respetar el orden correlativo de las historias de usuario iniciales, y cuya descripción es la siguiente: **HU18 - Como equipo de desarrollo quiero idear el formulario de solicitud de tutoría sin llegar a desarrollarlo**. A esta historia de usuario se le valora con 1 punto de historia.

Además, se ha procedido a realizar la ordenación de historias de usuario por prioridad para facilitar su selección en el sprint planning. El resultado de esta ordenación se puede ver en el *Product Backlog final* (ver Sección 2.7).

Historia de usuario	Tiempo estimado	Tiempo empleado	Tareas	Estado
HU03	25 horas	39 horas	Análisis	Completado
			Diseño	Iniciada
			Desarrollo	Iniciada
			Pruebas	No iniciado
HU04	25 horas	0 horas	Análisis	No iniciado
			Diseño	No iniciado
			Desarrollo	No iniciado
			Pruebas	No iniciado
-	-	1 hora y 22 minutos	Documentación de seguimiento del sprint 0 y del Sprint Planning del sprint 1	Completado

Tabla 7.2: Tareas del sprint 1

Para este Sprint, finalmente, se dedican un total de **35 horas y 15 minutos** de las 50 planificadas. Pese a que durante la primera semana de sprint si que se consiguen realizar las 25 horas semanales, ha sido durante la segunda semana cuando el alumno, debido a problemas personales y tener que acudir a citas médicas durante 3 días, no ha podido llevar a cabo un trabajo fluido, lo que ha hecho que el objetivo de las 50 horas de sprint no se lleve a cabo. Por lo tanto, se procede a la ejecución de los planes de contingencia de los riesgos **R03**, **R05** y **R06**, que son los que se han identificado durante este sprint (ver riesgos detalladamente en la Sección 2.4.2). Además, se toma la decisión de planificar menos tareas para el próximo sprint y, también, el alumno se plantea la posibilidad de reducir en futuros sprints las horas de trabajo diario si observa que durante el siguiente sprint no se consigue llevar a cabo el total de horas diarias inicial, para así establecer una planificación más realista a las condiciones dadas.

Pese a estos inconvenientes, el alumno ha manifestado que se siente más ágil a la hora de llevar a cabo ciertas tareas, en especial las dedicadas al frontend, que le ocupaban una gran cantidad de tiempo al no estar tan acostumbrado, pero que ha logrado conseguir resultados lo suficientemente vistosos en un tiempo menor.

En la Tabla 7.3 se puede ver el resumen de las tareas realizadas durante el sprint 2.

7.1.4. Sprint 3 (23/03/2022 - 06/04/2022)

En el Sprint Planning del tercer sprint, se añaden al Sprint Backlog las historias de usuario **HU05** y **HU06**, a parte de las tareas no terminadas durante el anterior sprint. Por el momento no se deciden tomar alternativas en cuanto al tiempo de trabajo diario.

Durante este sprint el alumno se ha encontrado con varias dificultades, tanto a nivel personal como de trabajo en el proyecto, provocando que el trabajo no pudiera ser todo

Historia de usuario	Tiempo estimado	Tiempo empleado	Tareas	Estado
HU03	20 horas	22 horas y 30 minutos	Diseño	Completado
			Desarrollo	Completado
			Pruebas	No iniciado
HU04	25 horas	10 horas y 19 minutos	Análisis	Completado
			Diseño	Iniciado
			Desarrollo	Iniciado
			Pruebas	No iniciado
HU18	5 horas	0 horas	Análisis	No iniciado
			Diseño	No iniciado
-	-	2 horas y 26 minutos	Documentación de seguimiento del Sprint Review y Retrospective del sprint 1 y del Sprint Planning del sprint 2	Completado

Tabla 7.3: Tareas del sprint 2

lo constante como se hubiera deseado, además de que el trabajo se ha visto entorpecido debido a la identificación de un error a la hora del análisis del modelo de dominio que ha repercutido en el diseño. Dicho error no pudo ser resuelto hasta que el alumno solicitó ayuda a su tutora durante la reunión de *Sprint Review*, donde se identificó la causa y se solucionó, provocando variaciones en el modelo de dominio, ya previamente modificado por el alumno. El problema consistía en que, tal y como se había realizado el modelo de dominio inicial (ver Sección 4.2), los cursos académicos se habían establecido como una enumeración, sin estar relacionados con una o varias posibles menciones de una titulación. Esto provocó que el diseño y, por lo tanto, la implementación no se realizaran de la manera correcta, pues, en la *HU03* al realizar la búsqueda por asignatura, el filtrado de una asignatura por mención y/o curso no se realizaba de manera adecuada. En el modelo de dominio final (ver Sección 4.3) se soluciona este problema, creando una clase para los cursos académicos y relacionando las asignaturas con sus posibles menciones, dando lugar, también, a una clase asociativa, *AsignaturasMencion*, que permitirá identificar a las asignaturas específicas de una mención.

Por lo tanto, en este sprint se han manifestado los riesgos **R03** y **R05**, además de un nuevo riesgo, que consiste en el análisis inicial erróneo y sus consecuencias, cuyo detalle se puede ver en la Tabla 7.4.

Las horas dedicadas durante este sprint fueron de un total de **35 horas**, de manera que no se llega a cumplir con las 50 horas de trabajo establecidas. Como resultado del trabajo irregular en este sprint, las tareas no terminadas se pasan al siguiente.

En la Tabla 7.5 se puede ver el resumen de las tareas realizadas durante el sprint 3.

ID del riesgo	R09
Título	Análisis inicial incorrecto
Categoría del riesgo	Estructura
Descripción del riesgo	Durante el proyecto existe la posibilidad de que el análisis inicial no estuviese elaborado de la manera más precisa, lo que puede ocasionar problemas en el diseño y desarrollo de la aplicación.
Probabilidad del riesgo	Bajo
Impacto del riesgo	Alto
Plan de mitigación	Realizar una tarea de investigación exhaustiva y la elaboración precisa de los requisitos necesarios de la aplicación a elaborar.
Plan de contingencia	<ul style="list-style-type: none"> ■ Depurar tanto el análisis como el resto de fases afectadas. ■ Retrasar las tareas afectadas. ■ Dedicar más tiempo para evitar que la planificación se vea demasiado afectada.

Tabla 7.4: Riesgo R09: Análisis inicial incorrecto

7.1.5. Sprint 4 (06/04/2022 - 27/04/2022)

Durante la reunión del Sprint Planning, se determina que lo mejor para favorecer el estado estado del proyecto es terminar las tareas pendientes y no añadir nuevas. Además, se indica al alumno que si es capaz de terminar todas las tareas dedique el resto del tiempo a la elaboración de la memoria del trabajo. También se toma la decisión de mantener las 5 horas de trabajo diario y que durante las vacaciones de Semana Santa, en las que en un principio el alumno planeaba tomar como descanso, ahora éste mismo ha decidido que trabajará alguno de los días para tratar de adelantar trabajo y solventar el estado frágil en el que se encuentra el proyecto, debido a la irregularidad en el trabajo a causa de factores externos.

En la reunión de Sprint Review, el alumno indica que las tareas que se han pasado del anterior sprint al actual se han llevado a cabo con éxito, concluyendo así con las historias de usuario **HU03** y **HU04**. En cuanto a las nuevas tareas del sprint, únicamente se han podido completar todas las correspondientes a la **HU18**. En lo referente a la historia de usuario **HU05**, pese a haber dedicado más horas de las estimadas, no se ha logrado terminar con las tareas de diseño, desarrollo y pruebas. El principal problema para no haber concluido esta historia de usuario en su totalidad ha consistido en la dificultad por parte del alumno de hacer frente a algunos errores inesperados en la base de datos: las secuencias de generación de los id, por ejemplo, la de las nuevas solicitudes, provocan errores ya que en la sentencia de creación se establecieron sus nombres con comillas dobles, de manera que JPA no encontraba dichas secuencias. Para solucionar este problema se volvieron a crear las secuencias sin emplear comillas dobles y también se reemplazó el tipo de dato de los id a los que hacen referencia estas secuencias por el tipo de dato SQL Bigint. Se hizo lo mismo con los atributos de

Historia de usuario	Tiempo estimado	Tiempo empleado	Tareas	Estado
HU03	2 horas	15 horas	Pruebas	Iniciado
			Depuración Análisis	Iniciado
			Depuración Diseño	Iniciado
			Depuración Desarrollo	Iniciado
HU04	15 horas	18 horas y 6 minutos	Diseño	Completado
			Desarrollo	Iniciado
			Pruebas	No iniciado
HU18	5 horas	0 horas	Análisis	No iniciado
			Diseño	No iniciado
HU05	20 horas	0 horas	Análisis	No iniciado
			Diseño	No iniciado
			Desarrollo	No iniciado
			Pruebas	No iniciado
HU06	10 horas	0 horas	Análisis	No iniciado
			Diseño	No iniciado
			Desarrollo	No iniciado
			Pruebas	No iniciado
-	-	1 hora y 54 minutos	Documentación de seguimiento del Sprint Review y Retrospective del sprint 2 y del Sprint Planning del sprint 3	Completado

Tabla 7.5: Tareas del sprint 3

las clases en Java referentes a las tablas con generación de índices, en las que el tipo de dato Integer se sustituyó por Long. Otro problema con el que se lidió en esta historia de usuario se encontraba a la hora de validar los formularios de solicitud de tutoría debido a las diferentes posibilidades que se pueden dar. Es por ello que esta historia de usuario (**HU05**) está inacabada y que la historia de usuario **HU06** no ha podido dar comienzo.

Además, al comienzo del sprint, durante la elaboración de las pruebas de la historia de usuario **HU04**, un problema en el backend con las relaciones bidireccionales entre distintas entidades hizo que se produjeran algunos cambios en la arquitectura de la aplicación para llevar a cabo la introducción de DTOs. El problema se debía a que cuando una entidad JPA pasa de la capa de negocio hasta la capa rest, se recupera el objeto completo, es decir, con todas sus relaciones, ignorando el hecho de que las hayamos marcado explícitamente como relaciones con carga perezosa (*fetch=FetchType.LAZY*), y, además, si la relación es bidireccional se produce un desbordamiento en la pila y los datos enviados al cliente son erróneos. Para resolver este problema, las entidades JPA, en lugar de representar las clases

de dominio se pasan al paquete de entidades de capa de persistencia. Además, la capa de dominio deja de ser transversal, pasando a estar dentro de la capa de negocio, donde se crean las clases de dominio, que siguen una estructura idéntica a las entidades, pero sin las anotaciones. Por último, se introducen en la capa de negocio, unos convertidores, que se encargarán de mapear los campos de las entidades a las clases del modelo de dominio y viceversa. Por lo tanto, las clases del modelo de dominio, pasan ahora a actuar como DTO.

Durante el sprint, vuelve a producirse el riesgo **R03**, además de que el alumno no logra realizar las 50 horas estimadas del sprint, haciendo un total de **45 horas y 1 minuto**. Es por ello que el alumno trata de identificar el porqué de estos retrasos, además de estar preocupado por no cumplir con la planificación, lo que le lleva a pensar en cómo afrontar los siguientes 2 sprints puesto que es consciente de que no ha logrado gestionarse de la mejor manera para cumplir con los tiempos, que en la fecha actual, estos empiezan a apretar a pesar de contar con los 2 sprints de refuerzo.

En la Tabla 7.6 se puede ver el resumen de las tareas realizadas durante el sprint 4.

7.1.6. Sprint 5 (27/04/2022 - 11/05/2022)

En el Sprint Planning del sprint 5, el penúltimo sprint a falta de conocer si se utilizarán los sprints de refuerzo, se determina que las tareas inacabadas del sprint anterior se pasan a éste y además se añaden las **HU01** y **HU02** al Sprint Backlog. Además, la tutora hace énfasis en dedicar todo el tiempo posible a la documentación de la memoria del TFG en caso de terminar con las historias de usuario.

En la Sprint Review, el alumno muestra sus avances. En cuanto a las historias de usuario **HU05** y **HU06** la tutora indica algunos cambios a incluir:

- Se deben añadir mensajes de error cuando el usuario solicite una tutoría en un intervalo donde ya había solicitado una tutoría.
- Se deben añadir mensajes de advertencia de que la solicitud puede no ser atendida, cuando se mandan solicitudes de un día para otro y esta se ha enviado en un horario tardío, y lo mismo cuando la solicitud se ha mandado un viernes en horario tardío o durante el fin de semana con la intención de ser realizada el lunes.
- En caso de que una solicitud no haga referencia a una asignatura, el campo de comentario debe ser obligatorio.
- El icono y color del botón para cancelar solicitudes deben ser más representativos. Se ha propuesto un icono de una papelería y un color rojo.

En cuanto a la historia de usuario **HU01** se han realizado sus respectivas tareas de análisis y diseño, y la **HU02** no se ha comenzado. El total de horas dedicadas durante el sprint ha sido de **40 horas y 9 minutos**, por lo que no se ha llegado a las 50 horas planificadas pese a que estas horas han tenido una mayor productividad respecto al resto de sprints. Como retrospectiva se ha incidido en que la productividad general del proyecto no es la esperada,

7.1. SEGUIMIENTO DE LOS SPRINTS REALIZADOS

Historia de usuario	Tiempo estimado	Tiempo empleado	Tareas	Estado
HU03	5 horas	7 horas y 50 minutos	Pruebas	Completado
			Depuración Análisis	Completado
			Depuración Diseño	Completado
			Depuración Desarrollo	Completado
HU04	5 horas	3 horas y 51 minutos	Desarrollo	Completado
			Pruebas	Completado
HU18	5 horas	1 hora y 21 minutos	Diseño	Completado
HU05	25 horas	26 horas y 25 minutos	Análisis	Completado
			Diseño	Iniciado
			Desarrollo	Iniciado
			Pruebas	No iniciado
HU06	10 horas	0 horas	Análisis	No iniciado
			Diseño	No iniciado
			Desarrollo	No iniciado
			Pruebas	No iniciado
-	-	51 minutos	Documentación de seguimiento del Sprint Review y Retrospective del sprint 3 y del Sprint Planning del sprint 4	Completado

Tabla 7.6: Tareas del sprint 4

pues a falta de un sprint el proyecto se encuentra aún lejos de cumplir los objetivos, pero que no se tiene que tener un enfoque negativo pues aún se dispone del tiempo necesario para concluir con el proyecto teniendo en cuenta los sprints de apoyo planificados. Se pone énfasis en que el tiempo restante se debe aprovechar al máximo para poder llevar tanto la aplicación como la documentación lo mejor posible, pero que esto no sirva para planificar de más en los siguientes sprints.

En la Tabla 7.7 se puede ver el resumen de las tareas realizadas durante el sprint 5.

7.1.7. Sprint 6 (11/05/2022 - 25/05/2022)

En la planificación del que sería el último sprint, el sprint 6, se añaden al Sprint Backlog las tareas inacabadas del sprint anterior y las tareas referentes a las historias de usuario

Historia de usuario	Tiempo estimado	Tiempo empleado	Tareas	Estado
HU05	20 horas	21 horas y 51 minutos	Diseño	Completado
			Desarrollo	Iniciado
			Pruebas	Iniciado
HU06	10 horas	15 horas y 26 minutos	Análisis	Completado
			Diseño	Completado
			Desarrollo	Iniciado
			Pruebas	Iniciado
HU01	10 horas	1 hora y 22 minutos	Análisis	Iniciado
			Diseño	Iniciado
			Desarrollo	No iniciado
			Pruebas	No iniciado
HU02	5 horas	0 horas	Análisis	No iniciado
			Diseño	No iniciado
			Desarrollo	No iniciado
			Pruebas	No iniciado
-	-	1 hora y 30 minutos	Documentación de seguimiento del Sprint Review y Retrospective del sprint 4 y del Sprint Planning del sprint 5	Completado

Tabla 7.7: Tareas del sprint 5

HU07 y **HU08**, que se corresponden a la aceptación y rechazo de tutorías, respectivamente, por parte del profesor. Se decide incluir estas 2 historias de usuario a mayores de lo inacabado en el sprint anterior, pues se trata de historias muy parecidas a lo ya realizado en dicho sprint con la historia de usuario HU06, que consistía en la cancelación de tutorías por parte del alumno. Además, se vuelve a hacer énfasis en centrar parte del tiempo en la realización de tareas de documentación de la memoria para evitar que esta se haga cuesta arriba en los tramos finales del proyecto.

El sprint 6 se ha llevado a cabo con bastante éxito. A falta de un problema identificado durante la demostración que hizo el alumno en la reunión de Sprint Review, el resto de tareas consiguieron llevarse a cabo con éxito. El problema identificado en la Sprint Review se trataba de que no se mostraban los mensajes de advertencia cuando el alumno realizaba una solicitud para el día siguiente o cuando hay un fin de semana de por medio, es por ello que las tareas de desarrollo y pruebas no se dan por finalizadas y se tendrán que pasar al sprint siguiente.

La historia de usuario **HU07** ha requerido más tiempo del estimado, esto se debe a que durante la fase de análisis en la que alumno realizó un diagrama de actividades del proceso

de negocio “Solicitar tutoría”, identificó que tal y como se estaba desarrollando la aplicación no se contemplaba el caso de que un profesor podría proponer un nuevo horario al alumno en caso de tratar de aceptar una solicitud de tutoría en un espacio de tiempo en el que el profesor no tuviese espacio para atender la solicitud. Debido a esto, se tuvieron que modificar diferentes diagramas y elementos de la web para añadir la posibilidad de que el horario de la solicitud de tutoría podría quedar modificado, por lo que se da lugar a una nueva historia de usuario, la **HU19 - Confirmar solicitud de tutoría**. Además de este problema, otro que alargó el desarrollo de esta historia de usuario fue un problema en la parte backend en la que el driver de PostgreSQL no podía determinar el tipo de dato de algunos parámetros. Tras varias horas investigando en internet se dio con la solución [35], que consistía en castear al tipo adecuado los parámetros en la propia consulta sql afectada.

En cuanto a la tarea dedicada a la documentación de la memoria, se comenzó a trasladar lo escrito en el borrador a la versión final en L^AT_EX[43]. Aunque al comienzo, el alumno se encontró con problemas, sobre todo, con las tablas, tras unas horas de dedicación logró acostumbrarse a este editor de texto y trasladar gran parte de la documentación a su versión final, gracias, en gran medida, al uso de la buena documentación que posee Overleaf [36].

El tiempo total dedicado en este sprint fue de **59 horas y 23 minutos**, superando con creces lo planificado por sprint. Además, se ha logrado concluir con todas las tareas a excepción del problema identificado durante la demo de la Sprint Review. Este sprint ha servido como motivación de que aún existe la posibilidad de que el alumno pueda terminar el proyecto únicamente empleando los sprints de refuerzo y que es el camino a seguir para concluir el trabajo con la mayor calidad posible.

En la Tabla 7.8 se puede ver el resumen de las tareas realizadas durante el sprint 6.

7.1.8. Sprint de refuerzo 1 (25/05/2022 - 08/06/2022)

Con este sprint se da comienzo a la recta final del proyecto, quedando únicamente los sprints de refuerzo, siendo el sprint actual el primero de ellos. Durante la reunión de Sprint Planning se decide que la historia de usuario **HU11**, cuyo propósito es establecer un horario de tutorías por parte del profesor, se **elimina** y la **HU12**, en la que se indicaba que la información del horario de un profesor se recogerá automáticamente desde la información de la UVa para no duplicar esfuerzos, cambia para que el horario se cargue a partir de un fichero CSV. Además, se añade la **HU20 - Cómo miembro de la administración de la escuela quiero que la población (titulaciones, asignaturas, usuarios y su información) de la aplicación se carguen mediante un fichero**, a la que se valora con una puntuación de historia de 3. Para el Sprint Backlog se determina que, a parte de las tareas no concluidas de la historia de usuario **HU05**, se incluirán en el sprint las historias de usuario **HU19**, **HU09**, **HU13** y **HU14**. Además, se incluye una tarea de documentación de la memoria.

Durante el sprint se sigue un ritmo de trabajo que permite que la tareas se desarrollen en un tiempo menor al planificado, a excepción de las tareas correspondientes a la historia de usuario **HU14**, donde casi se dobla el tiempo planificado. Esto se debe a que durante el diseño de esta tarea se decidió plantear una solución que permitiera no repetir código mediante herencia, debido a que a la hora de realizar las peticiones necesarias en el servicio

Historia de usuario	Tiempo estimado	Tiempo empleado	Tareas	Estado
HU05	5 horas	5 horas y 44 minutos	Desarrollo	Iniciado
			Pruebas	Iniciado
HU06	1 hora y 30 minutos	1 hora	Desarrollo	Completado
			Pruebas	Completado
HU01	8 horas y 30 minutos	13 horas y 30 minutos	Análisis	Completado
			Diseño	Completado
			Desarrollo	Completado
			Pruebas	Completado
HU02	5 horas	1 hora	Análisis	Completado
			Diseño	Completado
			Desarrollo	Completado
			Pruebas	Completado
HU07	10 horas	25 horas y 51 minutos	Análisis	Completado
			Diseño	Completado
			Desarrollo	Completado
			Pruebas	Completado
HU08	10 horas	1 hora y 38 minutos	Análisis	Completado
			Diseño	Completado
			Desarrollo	Completado
			Pruebas	Completado
-	10 horas	8 horas y 30 minutos	Documentación memoria	Completado
-	-	2 horas y 10 minutos	Documentación de seguimiento del Sprint Review y Retrospective del sprint 5 y del Sprint Planning del sprint 6	Completado

Tabla 7.8: Tareas del sprint 6

del frontend TutoriaService, se duplica mucho código, pues se tienen métodos distintos para las acciones del profesor o del alumno, por ejemplo, existen métodos como “getProximasTutoriasAlumno” y “getProximasTutoriasProfesor”. Primero se decidió plantear un solución en la que crear una interfaz ITutoriaService con los métodos necesarios, y después 2 clases concretas, TutoriaAlumnoService y TutoriaProfesorService. Sin embargo, trabajar con herencia en los servicios de angular es un tanto extraño, ya que en cada clase donde sean inyectados estos servicios concretos, hay que indicar cual se de ellos se utilizará, por lo que finalmente

se descarto esta solución y se planteó una nueva solución en la que el servicio TutoriaService, en vez de tener métodos específicos para cada tipo de usuario, tendría métodos en los que como parámetro se pasaría un Usuario y en la lógica de los distintos métodos dependiendo de la clase de la que sea instancia el usuario se realizaría una petición u otra. Esta solución se aplica también al servicio SolicitudTutoriaService. Es por esta razón que el desarrollo de esta historia de usuario se alargó más de lo debido.

En la reunión semanal con la tutora, el alumno indicó que había avanzado lo suficiente como para incluir nuevas historias de usuario en el sprint, por lo que se añadieron al Sprint Backlog las historias de usuario **HU15** y **HU10**, correspondientes a la cancelación de tutorías y a la consulta del perfil por parte del usuario. Ambas historias de usuario se desarrollan sin problemas, en especial la historia de usuario HU10, en la que tan solo se tuvo que reutilizar componentes ya creados a los que se le agregó pequeñas modificaciones.

También, en la reunión de Sprint Weekly la tutora indico al alumno que en la pantalla de inicio sería interesante que el profesor pudiera ver qué solicitudes tienes pendientes y que tutorías tiene próximamente, es por ello que, debido al buen ritmo de trabajo, se añadieron al Sprint Backlog tres nuevas historias de usuario: **HU21 - Como profesor quiero que las solicitudes pendientes se muestren en la pantalla de inicio para acceder a ellas con mayor facilidad**, **HU22 - Como alumno quiero que las solicitudes en estado aceptada se muestren en la pantalla de inicio para acceder a ellas más fácilmente** y **HU24 - Como usuario quiero que las próximas tutorías se muestren en la pantalla de inicio para facilitar su acceso**, con una puntuación de historia de 1 en cada una de ellas. Además, el alumno también vio que podría ser útil que las solicitudes caducadas en estado pendiente se rechacen automáticamente, para evitar que el profesor tenga que rechazarlas una a una, por lo que también se añade la **HU23 - Como profesor quiero que todas las solicitudes pendientes que no hayan sido atendidas a tiempo se rechacen automáticamente para automatizar la gestión**, con una puntuación de historia de 1.

Estas nuevas historias de usuario se desarrollaron sin problemas, pudiendo terminarlas todas en un tiempo total de **54 horas y 21 minutos**.

En la Tabla 7.9 se puede ver el resumen de las tareas realizadas durante el sprint de refuerzo 1.

7.1.9. Sprint de refuerzo 2 (08/06/2022 - 29/06/2022)

Entrados ya en el último sprint planificado, el sprint de refuerzo 2, queda por terminar con las últimas historias de usuario del Product Backlog. En la reunión de Sprint Planning se decide añadir, entonces, al Sprint Backlog todas las historias restantes, es decir, las historias de usuario **HU16**, **HU17**, correspondientes al bot de Telegram, **HU12** y **HU20**. Además, la tutora le comunica al alumno que no sería posible realizar la reunión semanal, y que durante este Sprint final se centre en terminar la funcionalidad restante y dejar la documentación para el tiempo que quede desde el final del sprint hasta la entrega del trabajo.

Durante el sprint se mantiene el ritmo de trabajo del sprint anterior, sin embargo, en esta

Historia de usuario	Tiempo estimado	Tiempo empleado	Tareas	Estado
HU05	1 hora	18 minutos	Desarrollo	Completado
			Pruebas	Completado
HU19	5 horas	5 horas y 37 minutos	Análisis	Completado
			Diseño	Completado
			Desarrollo	Completado
			Pruebas	Completado
HU09	15 horas	5 horas y 52 minutos	Análisis	Completado
			Diseño	Completado
			Desarrollo	Completado
			Pruebas	Completado
HU13	20 horas	5 horas y 21 minutos	Análisis	Completado
			Diseño	Completado
			Desarrollo	Completado
			Pruebas	Completado
HU14	10 horas	19 horas y 47 minutos	Análisis	Completado
			Diseño	Completado
			Desarrollo	Completado
			Pruebas	Completado
HU15	10 horas	5 horas y 46 minutos	Análisis	Completado
			Diseño	Completado
			Desarrollo	Completado
			Pruebas	Completado
HU10	10 horas	1 hora y 52 minutos	Análisis	Completado
			Diseño	Completado
			Desarrollo	Completado
			Pruebas	Completado
HU21	5 horas	1 hora y 41 minutos	Análisis	Completado
			Diseño	Completado
			Desarrollo	Completado
			Pruebas	Completado
HU22	5 horas	30 minutos	Análisis	Completado
			Diseño	Completado
			Desarrollo	Completado
			Pruebas	Completado
HU24	5 horas	2 horas y 10 minutos	Análisis	Completado
			Diseño	Completado
			Desarrollo	Completado
			Pruebas	Completado
HU23	10 horas	2 horas	Análisis	Completado
			Diseño	Completado
			Desarrollo	Completado
			Pruebas	Completado
-	-	55 minutos	Documentación memoria	Completado
-	-	2 horas y 32 minutos	Documentación de seguimiento del Sprint Review y Retrospective del sprint 6 y del Sprint Planning del sprint de refuerzo 1	Completado

Tabla 7.9: Tareas del sprint de refuerzo 1

ocasión, al tener que afrontar tareas con tecnologías con las que nunca se había trabajado, que son las correspondientes al bot de Telegram, provocó que a pesar de la cantidad de horas empleadas estas no produjeran un trabajo tan rápido. Durante la fase de análisis de la historia de usuario **HU16**, se barajaron las posibles opciones en las que desarrollar en bot de Telegram. A estas alturas de proyecto, emplear tecnologías no conocidas no era una opción pues no se disponía del tiempo necesario, por lo que se investigó la posibilidad de hacerlo con node.js o con java. En un principio, la primera opción resultaba más atractiva para el alumno, pero finalmente fue la segunda opción la que se eligió, ya que existía la posibilidad de desarrollar el bot con Spring Boot.

Durante el desarrollo del bot de Telegram se encontraron algunas dificultades. La principal, es que se trataba de una aplicación que necesita conocer el estado para que las operaciones se lleven a cabo correctamente, teniendo que programar con hilos las diferentes tareas que ejecute el bot. Para ello se acabó utilizando un Thread Pool que ejecuta una tarea cuando un hilo está libre (ver más en Sección 5.3.4). Además, como el bot de Telegram funciona mediante comandos, se decidió implementar en este bot el patrón de diseño **“Patrón Comando”**. A partir de ahí, aunque se consumió mucho tiempo, la realización de la historia de usuario **HU17** era prácticamente inmediata, ya que solo había que añadir un comando para acceder a la web de tutorías.

En cuanto, a la historia de usuario **HU12**, no se encontraron dificultades. Debido a la falta de tiempo, el alumno solicitó a la tutora cancelar el desarrollo de la **HU20** y dejarlo como funcionalidad futura.

Puesto que la tutora no pudo reunirse el día del Sprint Review final, planificado inicialmente para el día 22 de junio de 2022 (ver Sección 2.3), se decidió ampliar la duración de este último sprint de refuerzo una semana más, teniendo dicha reunión el 29 de junio. En esa semana, se dedicó el tiempo a arreglar detalles y a volver a ejecutar todas las pruebas manuales diseñadas (ver Sección 6.2) para comprobar que la versión final de la aplicación web y el bot de Telegram funcionan adecuadamente. Además, se intentó desplegar la aplicación en Heroku [26], sin embargo, puesto a que el proyecto se encuentra en un único repositorio de GitLab, Heroku no era capaz de detectar el proyecto adecuadamente. Como alternativa, se trató desplegar el proyecto en Heroku “dockerizándolo” previamente. Debido a la nula experiencia con Docker por parte del alumno se decidió crear una historia de usuario con una tarea de aprendizaje. Dicha historia de usuario es la siguiente **HU25 - Como equipo de desarrollo quiero “dockerizar” las aplicaciones para poder ejecutarlas en diferentes entornos sin preocuparnos por las dependencias**, con una puntuación de 6.

El tiempo total dedicado en este sprint es de un total de **120 horas y 16 minutos** de las 75 horas planificadas. La gran cantidad de horas de más empleadas en este sprint se han debido a las dificultades encontradas con el bot de Telegram y al tratar de desplegar en Heroku. En la reunión de Sprint Review se comentaron los problemas a la hora de desplegar en dicha plataforma, a lo que la tutora ofreció como alternativa desplegar en una máquina virtual de la Escuela.

En la Tabla 7.10 se puede ver el resumen de las tareas realizadas durante el sprint de refuerzo 2.

Historia de usuario	Tiempo estimado	Tiempo empleado	Tareas	Estado
HU16	30 horas	73 horas y 6 minutos	Análisis	Completado
			Diseño	Completado
			Desarrollo	Completado
			Pruebas	Completado
HU17	30 horas	54 minutos	Análisis	Completado
			Diseño	Completado
			Desarrollo	Completado
			Pruebas	Completado
HU12	15 horas	24 horas y 53 minutos	Análisis	Completado
			Diseño	Completado
			Desarrollo	Completado
			Pruebas	Completado
HU20	15 horas	0 horas y 21 minutos	Análisis	Bloqueada
			Diseño	Bloqueada
			Desarrollo	Bloqueada
			Pruebas	Bloqueada
HU25	10 horas	6 horas y 50 minutos	Aprendizaje	Completado
			Diseño	Completado
			Desarrollo	Completado
			Pruebas	Completado
-	-	6 horas y 42 minutos	Ejecución de todas las pruebas manuales	Completado
-	-	6 horas y 26 minutos	Despliegue en Heroku	No completado
-	-	1 hora y 11 minutos	Documentación memoria	Completado
-	-	1 hora y 25 minutos	Documentación de seguimiento del Sprint Review y Retrospective del sprint de refuerzo 1 y del Sprint Planning del sprint de refuerzo 2	Completado

Tabla 7.10: Tareas del sprint de refuerzo 2

7.1.10. Momentos finales del proyecto (29/06/2022 - 14/07/2022)

Durante las aproximadamente 2 semanas de margen entre la finalización de los sprints y la entrega del Trabajo de Fin de Grado, se realizaron algunas tareas que no fue posible acabarlas durante los sprints de refuerzo, como el despliegue del proyecto y la elaboración de la documentación del TFG, además de una pequeña modificación en el diseño de la API REST que simplificó el diagrama y el código desarrollado. Esta modificación consiste en que para las peticiones *GET* de los recursos *SolicitudesTutoria* y *Tutorias*, en lugar de tener operaciones separadas para el alumno y profesor, pasarían a ser operaciones para el usuario en general, reduciendo así la duplicación de código y obteniendo los beneficios de la herencia.

En cuanto al despliegue de las diferentes aplicaciones, se lleva a cabo en una máquina virtual de la Escuela, que más haya de pequeñas dificultades en la elaboración del *docker-compose.yml* debido a la nula experiencia previa, no se tuvieron mayores dificultades, a excepción de un problema con el espacio en disco de la máquina virtual, el cual era demasiado poco para realizar el despliegue. Sin embargo, se comunicó rápidamente a la tutora y se pudo resolver otorgando más espacio de disco a la máquina virtual.

Por último, lo que más tiempo ha llevado en esos instantes finales del proyecto ha sido la elaboración del documento del Trabajo de Fin de Grado, puesto que aún quedaban varias secciones sin documentar debido a que durante la ejecución de los sprints el tiempo dedicado a la documentación se empleó, sobre todo, a las secciones de seguimiento, realización de diagramas, documentación de las pruebas manuales y la planificación inicial realizada en el sprint 0. Además, se ha empleado mucho tiempo en ajustar los diagramas y diferentes tablas en el diagrama para que sean legibles, y otras pequeñas tareas del estilo.

En este tramo final se ha empleado un total de **69 horas y 10 minutos**. En la Tabla 7.11 se muestra la tabla con el resumen de las tareas realizadas.

Tarea	Tiempo empleado	Estado
Modificación en el diseño de la API REST	3 horas y 19 minutos	Completada
Modificación en el código referente a los cambios en el diseño de la API REST	51 minutos	Completada
Despliegue en la máquina virtual	6 horas y 23 minutos	Completada
Documentación final del documento del TFG	58 horas y 37 minutos	Completada

Tabla 7.11: Tareas de los momentos finales del proyecto

7.2. Resumen del desarrollo del proyecto (08/02/2022 - 14/07/2022)

Tras finalizar el proyecto, es interesante comparar la planificación con como se ha desarrollado el proyecto realmente. Para ello, en esta sección veremos el tiempo que se ha empleado en el proyecto realmente y se comparará con la planificación inicial. Además, se verá como cambiarían los costes estimados durante la planificación teniendo en cuenta la duración real del proyecto.

7.2.1. Tiempo estimado y empleado

Durante la planificación inicial 2.3 se estimó, partiendo como base de las horas de las que consta el Trabajo de Fin de Grado [53], que para el desarrollo del proyecto serían necesarias un total de 300 horas (sin contar el periodo de planificación). Puesto a que esta planificación fue ideada como caso ideal en el que no se manifestara ningún riesgo, se planificaron 2 sprints extra para actuar de colchón en caso de que las consecuencias de los riesgos retrasaran el correcto desarrollo del proyecto. De esta manera, se estableció como fecha final del proyecto el día 22 de junio, planificándose una duración máxima de 400 horas para ejecución del proyecto, y un total de 420 horas para el proyecto contando el período de planificación (*sprint 0*).

Sin embargo, el proyecto ha finalizado el día 14 de julio, casi un mes más tarde de lo planificado como máximo, empleándose un total de 498 horas y 57 minutos para la ejecución del proyecto. Por lo tanto, si sumamos el tiempo de planificación inicial se obtiene un tiempo total de dedicación al proyecto de **518 horas y 57 minutos**, superando con creces el tiempo requerido para la elaboración del TFG.

Podemos concluir que la planificación inicial no ha sido del todo precisa pese a haber planificado 2 sprints a mayores. Viendo el seguimiento documentado, podemos ver que fue durante los primeros sprints cuando se produjeron más riesgos, retrasándose gran parte del trabajo a los siguientes sprints. Además, también podemos observar que es en los 3 sprints finales cuando se comienza a cumplir con los tiempos estimados para cada sprint, por lo tanto, quizá, una planificación con una duración por sprint menor y una fecha de finalización de proyecto algo más amplia hubiera favorecido el desarrollo del proyecto. Sin embargo, la incertidumbre es la razón por la que se ha empleado Scrum como marco de trabajo, por lo que no hay que ver la finalización tardía del proyecto como algo puramente negativo. Gracias a la utilización de este marco de trabajo se ha podido adaptar los sprints finales de manera que se pudiera remontar el trabajo retrasado.

7.2.2. Costes finales

Puesto que el proyecto finalmente ha durado casi 5 meses y un total de 519 horas, aproximadamente, se volverán a calcular los costes, tanto para el caso simulado como para el real, para ver como el retraso en la finalización del proyecto ha afectado a éstos.

Coste simulado final

Teniendo en cuenta que los conceptos son los mismos, únicamente debemos calcular los costes con los nuevos datos: 6 meses y 519 horas. En la Tabla 7.12 se muestra cada coste individual y el total.

Concepto	Precio	Cantidad	Total
Hora de trabajo de desarrollador de software	21'23€/hora	519 horas	11018'37€
Equipo (Lenovo ThinkBook 15 G2 ITL)	13'72€/mes	5 meses	68'60€
Licencia Rocket.Chat	0€/mes	5 meses	0€
Licencia Balsamiq Cloud	7'95€/mes	5 meses	39'75€
Licencia Astah Professional	8'82€/mes	5 meses	44'10€
Total			11170'82€

Tabla 7.12: Coste simulado final

La estimación del coste simulado con la normalización del 20%, para hacer frente a posibles sobrecostes, era de un total de 8298'67€, por lo que no cubriría con el coste simulado final, que asciende a **un total de 11170'82€**.

Coste real final

Para el coste real final también tenemos el mismo concepto que en la estimación del coste real, por lo que hacemos el cálculo con los 5 meses de duración del proyecto.

En la Tabla 7.13 se muestra el resumen del cálculo del coste total final para el caso real, teniendo como coste real final **un total de 218'65€**, que si lo comparamos con el coste real estimado, que era de 174'92€, podemos ver que para este caso tampoco cubriríamos el coste real del proyecto.

Concepto	Precio	Cantidad	Total
Equipo (Asus ROG Strix G15 G513QR-HF010T)	43'73€/mes	5 meses	218'65€
Total			218'65€

Tabla 7.13: Coste real final

Capítulo 8

Conclusiones y trabajo futuro

8.1. Conclusiones

Una vez finalizado el proyecto, se puede concluir que la realización de tareas de gestión, desarrollo y documentación para un solo miembro de un equipo de desarrollo no es una tarea sencilla, y que el lema de *“La unión hace la fuerza”* en el mundo del desarrollo de software es totalmente cierto. Sin embargo, el llevar a cabo este reto para poner a prueba todos los conocimientos adquiridos durante el grado, tiene como resultado una sensación muy gratificante, puesto que a pesar de las dificultades y la enorme cantidad de trabajo, se ha logrado cumplir con todos los objetivos, tanto del proyecto como personales.

De hecho, a nivel personal, se han superado las expectativas, puesto que se ha conseguido elaborar una web con un diseño y funcionalidad atractivos, y un bot cuya funcionalidad de búsqueda de información de los profesores se considera muy útil para el uso de los alumnos. Además, del aprendizaje inesperado de la herramienta Docker, que en un principio no se planificaba utilizar, y que ha despertado en el alumno un interés por el mundo DevOps que desconocía.

8.2. Líneas de trabajo futuras

Como trabajo futuro, se plantean las siguientes propuestas, que de no ser por el tiempo limitado, se habrían querido implementar durante el proyecto:

- **Web completamente responsive para adaptarse a dispositivos móviles.** Puesto que los dispositivos móviles, concretamente los *smartphones*, están muy extendidos en la actualidad, se considera que la funcionalidad de la aplicación web desarrollada sería muy apropiada para ellos.

- **Test automatizados y CI/CD.** Durante el desarrollo del proyecto, se ha echado de menos el uso de test automatizados y CI/CD, teniendo que realizar todas las pruebas manuales, elaboradas durante la fase de pruebas de cada historia de usuario, al final del desarrollo del proyecto para verificar que todo funcionaba adecuadamente, y, además, teniendo que desplegar de forma manual de manera constante, provocando que se empleara demasiado tiempo en tareas que podrían estar automatizadas.
- **Seguridad.** Si algo se puede mejorar en este proyecto, es la seguridad. Debido al poco conocimiento en la implementación de seguridad en el desarrollo de aplicaciones web, ni siquiera se planteó como una opción, ya que su aprendizaje hubiera consumido, posiblemente, la mayor parte del proyecto.
- **Implementación de un rol de administración.** Debido al carácter académico de la web, el rol de un administrador que se encargara del alta y baja de alumnos y profesores, se considera que hubiera sido una opción más que acertada.

Bibliografía

- [1] 40deFiebre. ¿qué es el diseño responsive? <https://www.40defiebre.com/que-es/disenio-responsive>. Accessed: 2022-07-06.
- [2] Amazon. About 1-click ordering. <https://www.amazon.es/gp/help/customer/display.html?nodeId=202031590>. Accessed: 2022-07-04.
- [3] Angular. Common routing tasks. <https://angular.io/guide/router>. Accessed: 2022-07-12.
- [4] Angular. Understanding binding. <https://angular.io/guide/binding-overview>. Accessed: 2022-07-09.
- [5] Angular. What is angular? <https://angular.io/guide/what-is-angular>. Accessed: 2022-07-06.
- [6] Angular. Workspace and project file structure. <https://angular.io/guide/file-structure>. Accessed: 2022-07-12.
- [7] Aprendiendo Angular. Componentes. <https://ngchallenges.gitbook.io/project/componentes>. Accessed: 2022-07-10.
- [8] Astah. The power of software modeling. <https://astah.net/>. Accessed: 2022-07-05.
- [9] Ruben Bermudez. Telegram bot java library. <https://github.com/rubenlagus/TelegramBots>. Accessed: 2022-07-06.
- [10] bravent. Xamarin.forms y mvvm. <https://www.bravent.net/xamarin-forms-y-mvvm/>. Accessed: 2022-02-22.
- [11] Balsamiq Cloud. The perfect tool for designing and reviewing user interfaces online. <https://balsamiq.com/wireframes/cloud/>. Accessed: 2022-07-05.
- [12] Visual Studio Code. Code editing. redefined. <https://code.visualstudio.com/>. Accessed: 2022-07-06.
- [13] Docker. Dockerfile reference. <https://docs.docker.com/engine/reference/builder/>. Accessed: 2022-07-06.
- [14] Docker. The industry-leading container runtime. <https://www.docker.com/products/container-runtime/>. Accessed: 2022-07-06.

- [15] Docker. Overview of docker compose. <https://docs.docker.com/compose/>. Accessed: 2022-07-13.
- [16] Docker. What is a container? <https://www.docker.com/resources/what-container/>. Accessed: 2022-07-06.
- [17] File Format Docs. Documentation. <https://docs.fileformat.com/web/scss/>. Accessed: 2022-07-06.
- [18] Eclipse Foundation. Desktop ides. <https://www.eclipse.org/ide/>. Accessed: 2022-07-06.
- [19] FullCalendar. Angular component. <https://fullcalendar.io/docs/angular>. Accessed: 2022-03-29.
- [20] Git. –everything-is-local. <https://git-scm.com/>. Accessed: 2022-07-05.
- [21] GitLab. Issue boards. https://docs.gitlab.com/ee/user/project/issue_board.html. Accessed: 2022-07-06.
- [22] GitLab. The one devops platform. <https://about.gitlab.com/>. Accessed: 2022-07-06.
- [23] Yania Crespo González-Carvajal. Apuntes del tema 3 de la asignatura diseño de software. Escuela de Ingeniería Informática de la Universidad de Valladolid. Accessed: 2022-07-10.
- [24] Yania Crespo González-Carvajal. Apuntes del tema 5 de la asignatura diseño de software. Escuela de Ingeniería Informática de la Universidad de Valladolid. Accessed: 2022-07-10.
- [25] H2. H2 database engine. <https://www.h2database.com/html/main.html>. Accessed: 2022-07-10.
- [26] Heroku. Cloud application platform — heroku. <https://www.heroku.com/home>. Accessed: 2022-07-10.
- [27] Bob Hughes and Mike Cotterell. *Software Project Management*. McGraw-Hill Education, 5 edition, 2009.
- [28] Indeed. Ofertas de trabajo, bolsa de trabajo — buscar empleo en indeed españa. <https://es.indeed.com/?r=us>. Accessed: 2022-02-15.
- [29] Open Source Initiative. The 3-clause bsd license. <https://opensource.org/licenses/BSD-3-Clause>. Accessed: 2022-07-14.
- [30] Kinsta. ¿qué es nginx y cómo funciona? <https://kinsta.com/es/base-de-conocimiento/que-es-nginx/>. Accessed: 2022-07-06.
- [31] Manuel Trigas Gallego. Metodología scrum. <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>. Accessed: 2022-02-17.
- [32] MAX REHKOPF. Historias de usuario con ejemplos y plantilla. <https://www.atlassian.com/es/agile/project-management/user-stories>. Accessed: 2022-07-05.
- [33] Node.js. Acerca de node.js. <https://nodejs.org/es/about/>. Accessed: 2022-07-06.

- [34] NPM. About npm. <https://docs.npmjs.com/about-npm>. Accessed: 2022-07-06.
- [35] STACK OVERFLOW. Postgres sql ‘could not determine data type of parameter’ by hibernate. <https://stackoverflow.com/questions/56089400/postgres-sql-could-not-determine-data-type-of-parameter-by-hibernate>. Accessed: 2022-05-21.
- [36] Overleaf. Documentation. <https://www.overleaf.com/learn>. Accessed: 2022-07-13.
- [37] Overleaf. The easy to use, online, collaborative latex editor. <https://www.overleaf.com/>. Accessed: 2022-07-05.
- [38] PostgreSQL. Postgresql: The world’s most advanced open source relational database. <https://www.postgresql.org/>. Accessed: 2022-07-06.
- [39] PrimeFaces. Next-gen angular ui. <https://www.primefaces.org/primeng/>. Accessed: 2022-07-06.
- [40] Apache Maven Project. Introduction to the pom. <https://maven.apache.org/guides/introduction/introduction-to-the-pom.html>. Accessed: 2022-07-12.
- [41] Apache Maven Project. Introduction to the standard directory layout. <https://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html>. Accessed: 2022-07-12.
- [42] Apache Maven Project. Welcome to apache maven. <https://maven.apache.org/>. Accessed: 2022-07-06.
- [43] The LaTeX Project. Latex – a document preparation system. <https://www.latex-project.org/>. Accessed: 2022-07-05.
- [44] Rocket.Chat. La plataforma de comunicaciones en la que puedes confiar plenamente. <https://es.rocket.chat/get-started>. Accessed: 2022-07-05.
- [45] Scrum. What is scrum? <https://www.scrum.org/resources/what-is-scrum>. Accessed: 2022-02-17.
- [46] Spring. Exception handling in spring mvc. <https://spring.io/blog/2013/11/01/exception-handling-in-spring-mvc>. Accessed: 2022-07-10.
- [47] Spring. Spring boot. <https://spring.io/projects/spring-boot>. Accessed: 2022-07-06.
- [48] Spring. Why spring? <https://spring.io/why-spring>. Accessed: 2022-07-06.
- [49] Telegram. Bot code examples. <https://core.telegram.org/bots/samples>. Accessed: 2022-07-06.
- [50] Telegram. Bots: An introduction for developers. <https://core.telegram.org/bots>. Accessed: 2022-07-06.
- [51] Telegram. Preguntas frecuentes. <https://telegram.org/faq>. Accessed: 2022-07-06.

- [52] TypeScript. Typescript is javascript with syntax for types. <https://www.typescriptlang.org/>. Accessed: 2022-07-06.
- [53] Universidad de Valladolid. Proyecto docente del trabajo de fin de grado 2021-2022 (Mención Ingeniería de Software). https://albergueweb1.uva.es/guia_docente/uploads/2021/545/46976/1/Documento.pdf. Accessed: 2022-02-18.
- [54] Angular University. Angular single page applications (spa): What are the benefits? <https://blog.angular-university.io/why-a-single-page-application-what-are-the-benefits-what-is-a-spa/>. Accessed: 2022-07-10.
- [55] W3Schools. Css tutorial. <https://www.w3schools.com/css/>. Accessed: 2022-07-06.
- [56] Webex. Una aplicación para todo y todos. <https://www.webex.com/es/index.html>. Accessed: 2022-07-05.
- [57] Wikipedia. Command (patrón de diseño). [https://es.wikipedia.org/wiki/Command_\(patr%C3%B3n_de_dise%C3%B1o\)](https://es.wikipedia.org/wiki/Command_(patr%C3%B3n_de_dise%C3%B1o)). Accessed: 2022-07-11.
- [58] Wikipedia. Devops. <https://es.wikipedia.org/wiki/DevOps>. Accessed: 2022-07-06.
- [59] Wikipedia. Entorno de desarrollo integrado. https://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado. Accessed: 2022-07-06.
- [60] Wikipedia. Lazy loading. https://es.wikipedia.org/wiki/Lazy_loading. Accessed: 2022-07-06.
- [61] Wikipedia. Model–view–viewmodel. <https://en.wikipedia.org/wiki/Model>. Accessed: 2022-02-22.
- [62] Wikipedia. Objeto de acceso a datos. https://es.wikipedia.org/wiki/Objeto_de_acceso_a_datos. Accessed: 2022-07-11.
- [63] Wikipedia. Objeto de transferencia de datos. https://es.wikipedia.org/wiki/Objeto_de_transferencia_de_datos. Accessed: 2022-07-11.
- [64] Wikipedia. Thread pool. https://en.wikipedia.org/wiki/Thread_pool. Accessed: 2022-07-11.
- [65] Wikipedia. Tomcat. <https://es.wikipedia.org/wiki/Tomcat>. Accessed: 2022-07-06.
- [66] Wikipedia. Visual studio code. https://es.wikipedia.org/wiki/Visual_Studio_Code. Accessed: 2022-07-06.
- [67] YAML. The official yaml web site. <https://yaml.org/>. Accessed: 2022-07-13.

Apéndice A

Manuales

A.1. Manual de despliegue local

Para desplegar la aplicación de manera local se debe tener instalado *Docker* y, de manera opcional, se deberá instalar *Git* para clonar el repositorio. De forma alternativa, se puede descargar el proyecto desde el propio repositorio (ver Apéndice B).

Una vez se tenga el software necesario, lo primero es clonar el repositorio, para ello, se debe ejecutar la siguiente instrucción desde una terminal de línea de comandos:

```
$ git clone https://gitlab.inf.uva.es/ricalba/tfg-  
richardalbanfernandez.git
```

A continuación, se debe ejecutar el siguiente comando y esperar a que termine:

```
$ docker compose —profile local up -d
```

¡Listo! El proyecto estará desplegado localmente. Para acceder a la web de manera local utiliza la siguiente url: <http://localhost:4200>. Para acceder al bot de Telegram, utiliza el enlace del Apéndice B. Para aprender a utilizarlos, sigue el manual de usuario de la siguiente sección (ver Sección A.2).

A continuación, se muestra la instrucción necesaria para detener el despliegue local:

```
$ docker compose —profile local down —rmi all
```

Con este comando se detendrán los contenedores y se eliminarán tanto los contenedores como las imágenes creadas.

A.2. Manual de usuario

Para la utilización de las diferentes aplicaciones, se han creado unos usuarios de prueba para poder acceder a todas las funcionalidades. Los usuarios son los siguientes:

■ Alumnos:

- *Alumno 1:*
 - **identificador:** alumno1
 - **contraseña:** password
- *Alumno 2:*
 - **identificador:** alumno2
 - **contraseña:** password
- *Alumno 3:*
 - **identificador:** alumno3
 - **contraseña:** password

■ Profesores:

- *Profesor 1:*
 - **identificador:** profesor1
 - **contraseña:** password
- *Profesor 2:*
 - **identificador:** profesor2
 - **contraseña:** password
- *Profesor 3:*
 - **identificador:** profesor3
 - **contraseña:** password

A.2.1. Manual de usuario de la aplicación web

Para facilitar la comprensión del manual, este se va a dividir según el tipo de usuario: Alumno, Profesor y Usuario no identificado.

Usuario no identificado

La primera vez que accedemos a la web, lo hacemos como un usuario no identificado, de manera que no tendremos acceso a la funcionalidad de la aplicación hasta que iniciemos sesión con nuestras credenciales de la institución académica, en este caso la Escuela de Ingeniería Informática de la Universidad de Valladolid. En la pantalla de *login* introducimos nuestras credenciales, como se muestra en la Figura A.1. Tras ello, si las credenciales son correctas, seremos llevados hasta la página de *Inicio*.

Alumno

En la pantalla de *Inicio* del *Alumno*, como se observa en la Figura A.2, hay bastante información, por lo que iremos por partes. En la parte superior, se encuentra el *header*, donde se muestra nuestro rol de alumno junto a nuestro nombre completo. Mediante las opciones del header, podremos acceder a nuestro perfil, donde se muestra nuestra información académica (ver Figura A.3), y cerrar sesión.

En el centro de la pantalla de *Inicio* podemos ver 2 carruseles: en el primero se muestran las próximas tutorías que tenemos programadas, mientras que en el segundo se muestran las

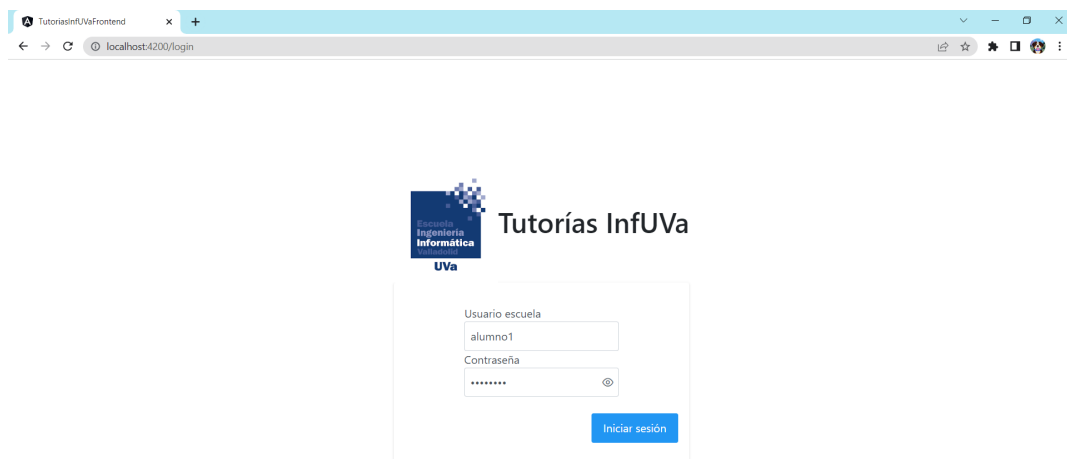


Figura A.1: Pantalla de Inicio de sesión de la web Tutorías InfUVA

solicitudes que un profesor ha aceptado y están pendientes de nuestra confirmación. Podemos acceder al detalle de cada elemento pulsando sobre el botón con el icono de un ojo.

En el menú lateral izquierdo, se encuentran las diferentes secciones y funcionalidades de la web. Empezaremos a buscar un profesor al que solicitaremos una tutoría. Para ello, pulsamos sobre la opción del menú **Búsqueda por profesor**, navegando, así, hasta la pantalla **Buscar profesor**, donde nos encontraremos en el tab *Profesor* del tabMenu “Búsqueda por:”. Para realizar la búsqueda, introducimos una cadena de texto en el buscador, por ejemplo, buscaremos algún profesor con el nombre “Malenia” (ver Figura A.4). A continuación, realizamos la búsqueda de un profesor, pero en este caso, trataremos de buscarlo a partir de las asignaturas que imparte. Para ello, pulsamos en la opción del tabMenu *Asignatura* o pulsamos en el menú lateral la opción **Búsqueda por asignatura**. En este tipo de búsqueda (ver Figura A.5), seleccionaremos una titulación y, de manera opcional, seleccionaremos un curso académico, una mención (en caso de haberla) y una cadena con el nombre de la asignatura. Por último, realizamos la búsqueda pulsando el botón “Buscar”. Las asignaturas que coincidan con los filtros indicados se mostrarán en forma de *TreeTable* [39], por lo que si pulsamos alguno de los nodos, se desplegará un listado con los docentes encargados de impartir la materia elegida.

Antes de solicitar una tutoría, veremos la información de un profesor. Para ello, pulsamos sobre el botón “Ver perfil” de alguno de los profesores listados, lo que nos llevará hasta su información (ver Figura A.6). En esta pantalla de **Información profesor** podemos ver la información de un profesor: como su despacho, centro habitual, etc.; los horarios de tutorías del profesor; las asignaturas que imparte y las tutorías que tiene programadas.

Para solicitar una tutoría podemos hacerlo desde el botón de “Solicitar tutoría” en los

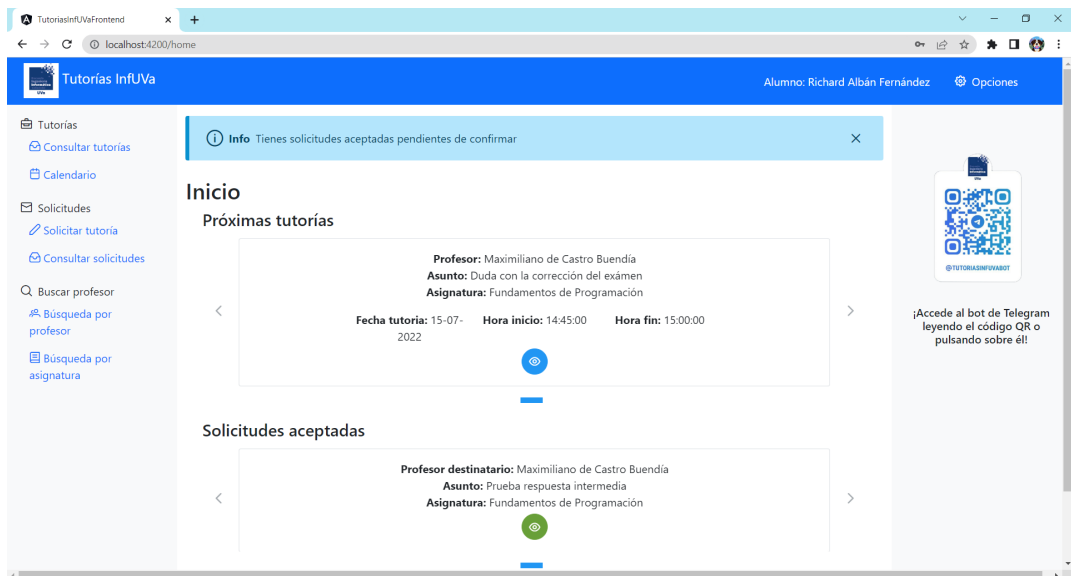


Figura A.2: Pantalla de Inicio de la web Tutorías InfUVa - Alumno

resultados de búsqueda de un profesor (ver Figuras A.4 y A.5), de forma que el profesor quedaría seleccionado, o podemos hacerlo desde la opción del menú **Solicitar tutoría**. Por cualquiera de los métodos, navegamos hasta la pantalla **Solicitar tutoría**, en donde tendremos que rellenar, como mínimo, los campos obligatorios (ver Figura A.7). Hay que destacar, que en caso de no seleccionar una asignatura, el campo comentario es obligatorio. En esta pantalla, podremos consultar el horario de tutorías del profesor seleccionado al pulsar sobre el botón “Ver horario tutorías”, como se ve en la Figura A.8.

Una vez solicitada la tutoría, se nos muestra en la pantalla **Solicitud tutoría** todos los detalles de la nueva solicitud (ver Figura A.9). En esta pantalla es donde se visualizará el ciclo de vida de la solicitud de tutoría, desde que está pendiente, hasta que pasa a estar confirmada. Cuando una solicitud está pendiente o aceptada, el alumno podrá cancelarla. Enviada la solicitud, debemos esperar a que el profesor la gestione, pudiendo ser rechazada o aceptada. Si el profesor acepta nuestra solicitud pasará al estado aceptada y nosotros deberemos decidir si confirmamos la asistencia a la tutoría, en cuyo caso la solicitud pasa a estado confirmada (ver Figura A.10) y genera una tutoría para el horario establecido

Para consultar las solicitudes que hemos realizado, tenemos que seleccionar en el menú lateral la opción **Consultar solicitudes**, lo que nos llevará a una pantalla con ese mismo nombre. Aquí podremos filtrar y consultar todas las solicitudes que hayamos realizado (ver Figura A.11). El filtrado de solicitudes se podrá realizar sobre los siguientes campos: estado de la solicitud, fecha de la tutoría, si es grupal, el profesor destinatario y la asignatura a la que hace referencia.

En cuanto a las tutorías, podemos consultarlas mediante un calendario o en forma de lista. Para la primera opción, seleccionamos en el menú **Calendario**, lo que nos llevará hasta la

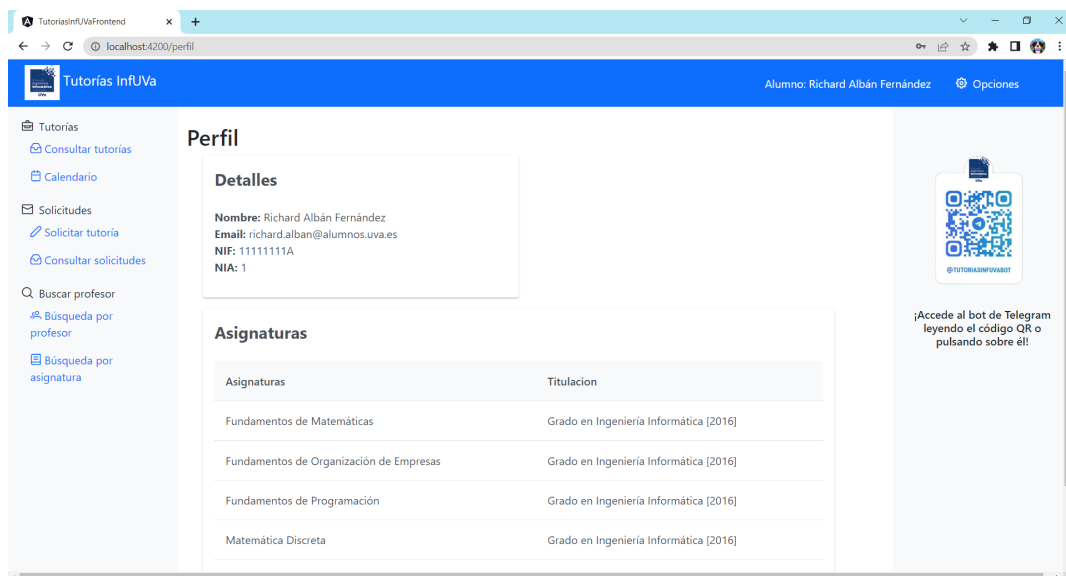


Figura A.3: Pantalla de Perfil de la web Tutorías InfUVA - Alumno

pantalla *Calendario tutorías*, donde a partir de un calendario podremos ver las tutorías que tenemos programadas. Si pulsamos sobre una de ellas se mostrará su detalle, pudiendo ver más información si se presiona el botón “Ver más” (ver Figura A.12).

Por otro lado, podemos consultar las tutorías seleccionando en el menú *Consultar tutorías*, navegando, así, hasta el listado de tutorías. En este listado, podremos seleccionar si queremos ver las tutorías que tienen lugar en el día de hoy, las que tienen lugar próximamente o filtrarlas según diversos parámetros (ver Figura A.13). El filtrado de tutorías se podrá realizar sobre los siguientes campos: fecha de la tutoría, si se ha cancelado, si es grupal, el profesor destinatario y la asignatura a la que hace referencia. Para ver una tutoría pulsamos sobre el icono del ojo, navegando así hasta la pantalla *Tutoría* (ver Figura A.14). En esta pantalla se mostrará el detalle de la tutoría, pudiendo ver la solicitud que ha originado esta tutoría mediante el botón “Ver solicitud”, o cancelarla, incluyendo un motivo de cancelación, mediante el botón “Cancelar tutoría”.

Por último, podremos acceder, mediante cualquiera de las pantallas, al bot de Telegram. Para ello, podemos leer el código QR con un dispositivo móvil o, simplemente, pulsar sobre él. Para aprender a utilizar el bot de Telegram, en la Sección A.2.2 se muestra cómo hacerlo.

Profesor

En la pantalla de *Inicio* del *Profesor*, como se observa en la Figura A.15, hay bastante información, por lo que iremos por partes. En la parte superior, se encuentra el *header*, donde se muestra nuestro rol de profesor junto a nuestro nombre completo, además de unas opciones en las que profundizaremos más adelante. En el centro de la pantalla podemos ver

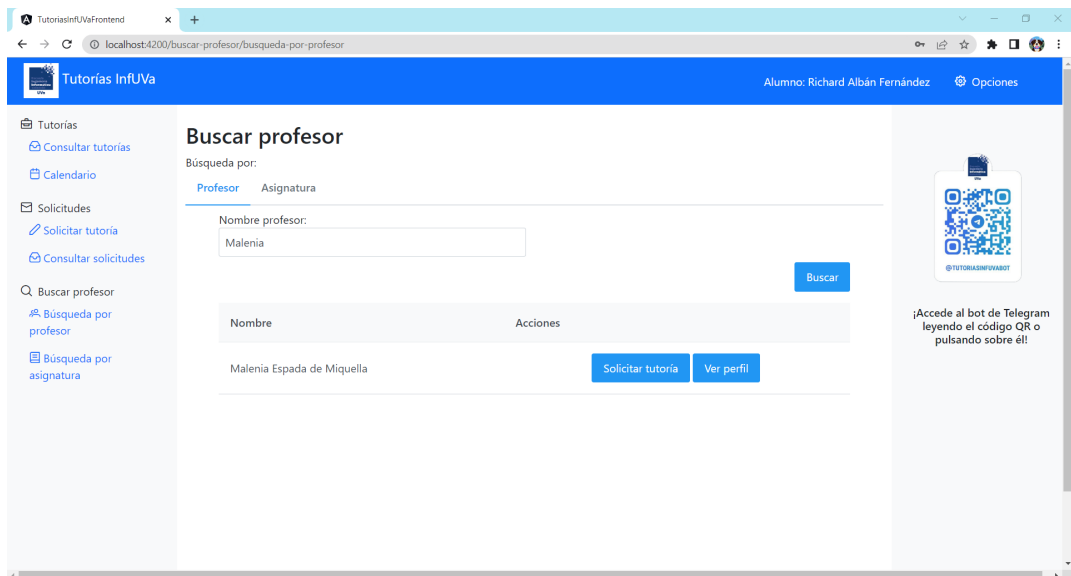


Figura A.4: Pantalla de Buscar profesor: búsqueda por profesor, de la web Tutorías InfUva - Alumno

2 carruseles: en el primero se muestran las próximas tutorías que tenemos programadas, mientras que en el segundo se muestran las solicitudes que tenemos pendientes. Podemos acceder al detalle de cada elemento pulsando sobre el botón con el icono de un ojo.

Para consultar las solicitudes según diversos filtros y en forma de listado, debemos pulsar en la opción del menú lateral **Consultar solicitudes**. En esta pantalla (ver Figura A.16), podremos filtrar las solicitudes según diversos parámetros: su estado, la fecha de la tutoría, si es grupal y por asignatura. Lo siguiente que debemos hacer es pulsar el botón “Buscar” y seleccionar uno de los resultados pulsando en el botón con el icono de un ojo. De esta manera accedemos al detalle de una solicitud. En esta pantalla, **Solicitud tutoría**, a parte de ver el detalle de la solicitud, podremos llevar a cabo distintas acciones en caso de que la solicitud esté en estado pendiente: rechazarla, indicando el motivo del rechazo, o aceptarla, con la posibilidad de poder indicar un nuevo horario para la tutoría si ya hay tutorías en el horario de la solicitud (ver Figura A.17). Las solicitudes caducadas, es decir, las que su fecha de tutoría ya ha pasado, se rechazarán de manera automática.

Para ver las tutorías que tenemos programadas, podremos hacerlo de dos formas: en una vista de calendario (mediante la opción del menú **Calendario**) o a partir de un listado (mediante la opción **Consultar tutorías**). En la pantalla **Calendario tutorías** se muestra el calendario donde veremos las tutorías, así como las solicitudes pendientes de confirmación por parte del alumno. Al pulsar sobre una de ellas, se mostrará en un dialogo modal el detalle de la solicitud aceptada (ver Figura A.18) o la tutoría, pudiendo ver el detalle completo si pulsamos en el botón “Ver más”. En cuanto a la pantalla **Consultar tutorías** podremos seleccionar como queremos que se vea el listado (ver Figura A.19): por las próximas tutorías programadas, por las tutorías programadas para el día actual o por las tutorías obtenidas

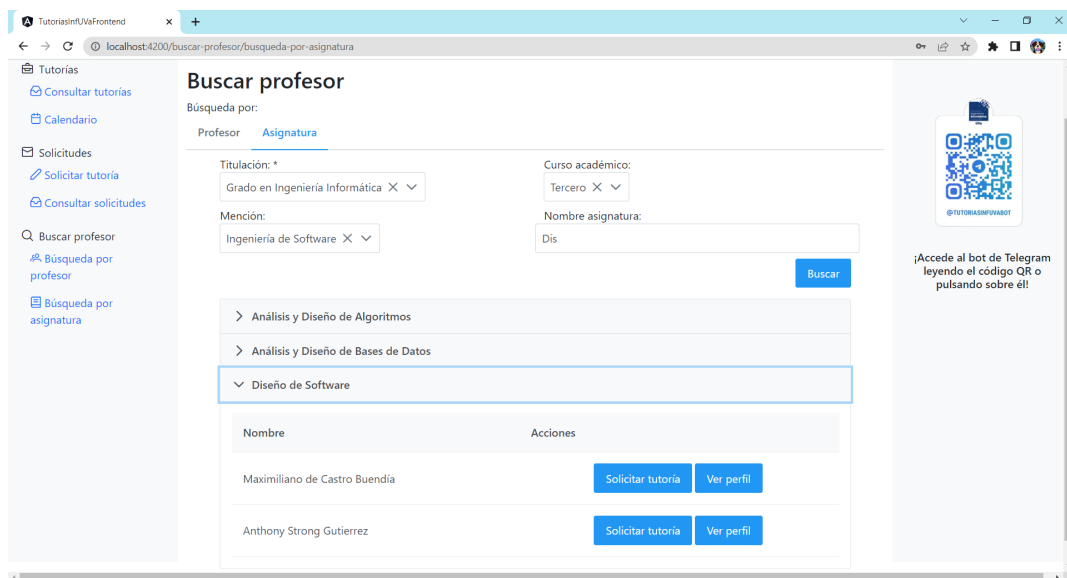


Figura A.5: Pantalla de Buscar profesor: búsqueda por asignatura, de la web Tutorías InfUva - Alumno

mediante filtros. En cuanto a los filtros, estos son: fecha de la tutoría, si está cancelada, si es grupal o por alguna de las asignaturas del profesor. Pulsando sobre el icono del ojo, con el tooltip “Ver tutoría”, de alguna de las tutorías listadas, accederemos a su detalle. En esta pantalla, **Tutoría** (ver Figura A.20), a parte de mostrarse el detalle de la tutoría, se puede ver la solicitud que la ha originado mediante el botón “Ver solicitud”, o cancelarla, incluyendo un motivo de cancelación, mediante el botón “Cancelar tutoría”.

Mediante el header visible a través de todas las pantallas una vez se ha iniciado sesión en la aplicación web, podemos acceder a las opciones: cerrar sesión o ver perfil. En la pantalla de **Perfil** (ver Figura A.21), se muestra la información académica del profesor: su datos de contacto, horario de tutorías y asignaturas que imparte. Por último, el profesor puede cargar nuevos horarios de tutorías pulsando en el botón “Cargar nuevo horario”, de manera que se abrirá un dialogo modal (ver Figura A.22) en el que subir un fichero CSV con los nuevos horarios. Este fichero CSV debe tener el siguiente formato:

- Cabecera con 6 campos: día, horaInicio, horaFin, centro, despacho, tipoHorario.
- Columna ‘día’: nombre en castellano del día en el que se desea establecer el horario. No importa si la cadena lleva tildes ni si está en mayúsculas o minúsculas o una combinación de ambas. Solo se permiten días de lunes a viernes. Ejemplos válidos: ‘Miercoles’, ‘Miércoles’, ‘miercoLes’.
- Columna ‘horaInicio’: cadena en formato ‘HH:mm’. Representa la hora de inicio de la franja de tutorías.

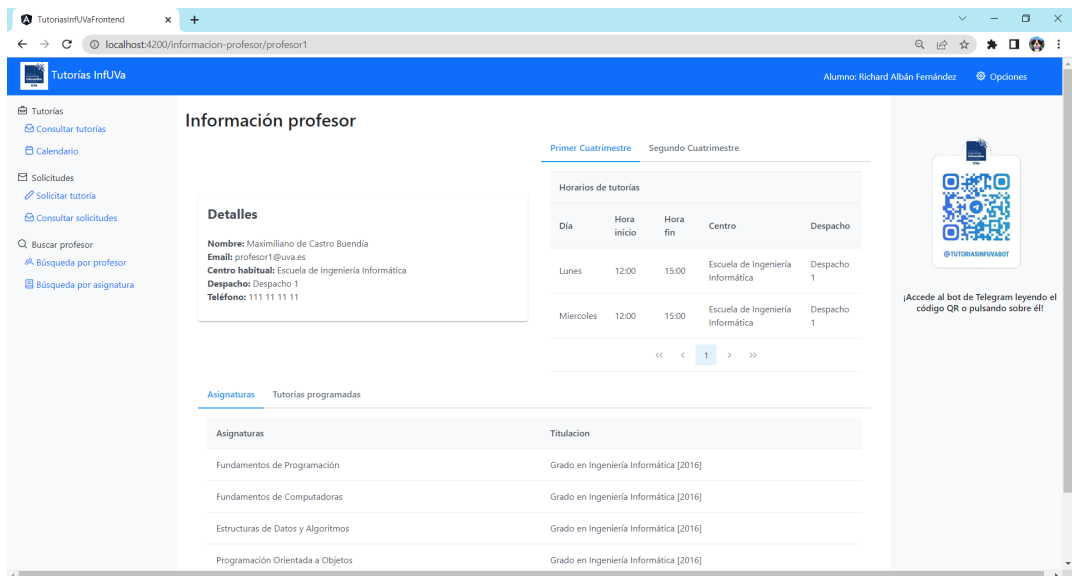


Figura A.6: Pantalla de Información profesor, de la web Tutorías InfUVa - Alumno

- Columna ‘horaFin’: cadena en formato ‘HH:mm’. Representa la hora de fin de la franja de tutorías.
- Columna ‘centro’: cadena de caracteres que representa el centro académico en el que el profesor estará presente durante la franja del horario.
- Columna ‘despacho’: cadena de caracteres que representa el despacho del profesor en el que se encontrará en la franja de tutorías.
- Columna ‘tipoHorario’: número que representa el tipo de horario. Puede tomar el valor ‘1’ para indicar que es una franja para el primer cuatrimestre, el valor ‘2’ para el segundo cuatrimestre, y el valor ‘3’ si es anual. En caso de que se quiera cargar un horario anual solo puede incluir este tipo de horario, por el contrario, si se trata de un horario que se distingue según el cuatrimestre, no podrá incluir el tipo de horario anual.

A.2.2. Manual de usuario del bot de Telegram

Una vez accedemos al bot de Telegram, se nos muestra la pantalla de inicio, donde se describe lo que puede hacer el bot (ver Figura A.23). En este punto tendremos que ejecutar el comando `/start` de forma obligatoria para comenzar la interacción con el bot, pero en esta primera ocasión de una manera especial, ya que es la aplicación de Telegram la que mediante el botón “Iniciar” ejecutará nuestro comando `/start`. Como se muestra en la Figura A.24, este comando nos dará la bienvenida.

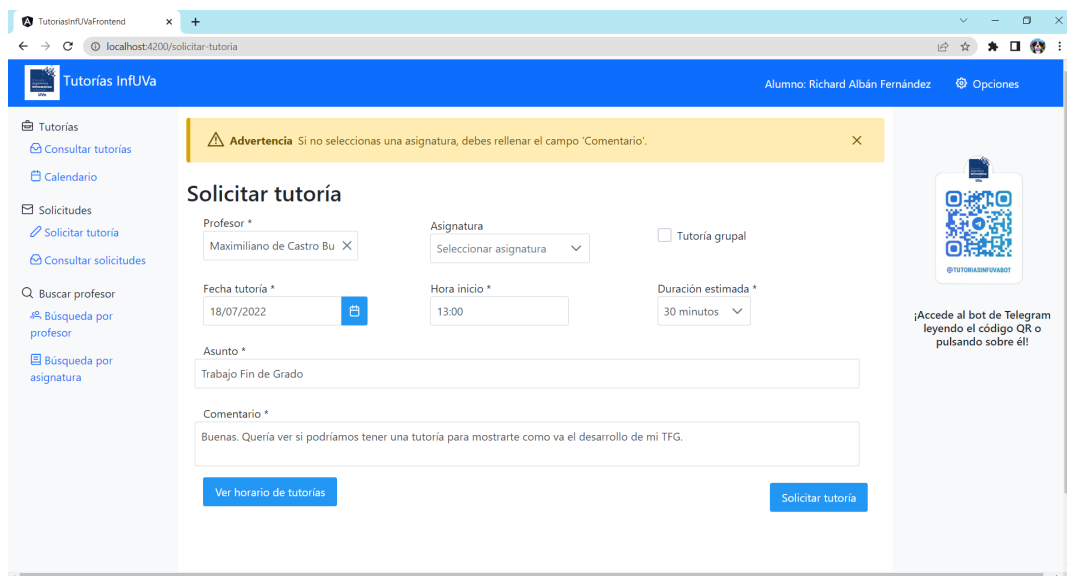


Figura A.7: Pantalla de Solicitar tutoría de la web Tutorías InfUVA - Alumno

Presionando el botón “Menú” podremos ver los distintos comandos del bot (ver Figura A.25), junto con su descripción, y podremos ejecutarlos sin necesidad de escribirlos en el chat, únicamente presionando sobre el comando que deseemos ejecutar.

Para poder acceder a la funcionalidad del bot, debemos iniciar sesión con nuestras credenciales de la Escuela, para lo que utilizamos el comando */login*. Como se ve en la Figura A.26, el bot nos responderá y nos solicitará el usuario y, posteriormente, la contraseña. Tras esto, si las credenciales son correctas, nos indicará que hemos iniciado sesión y nos dará la bienvenida utilizando nuestro nombre completo.

A continuación, vamos a buscar un profesor, que es la característica más interesante de este bot. Para ello, utilizamos el comando */buscar_profesor*. Como se muestra en la Figura A.27, el bot nos responderá y pedirá que seleccionemos uno de los tres tipos de búsqueda:

- **Búsqueda general.** Mediante este tipo de búsqueda el bot nos irá preguntando y mostrando opciones que debemos seleccionar.
 1. En primer lugar nos preguntará la titulación (ver Figura A.28).
 2. A continuación, el curso académico (ver Figura A.29).
 3. En caso de que el curso académico esté asociado a alguna mención, nos pedirá que seleccionemos una de ellas (ver Figura A.30).
 4. Lo siguiente que nos pedirá será la asignatura (ver Figura A.31).
 5. Por último, nos mostrará los profesores que imparten la asignatura (ver Figura A.32), donde seleccionaremos el profesor que estábamos buscando, mostrándonos su información de contacto (ver Figura A.33).

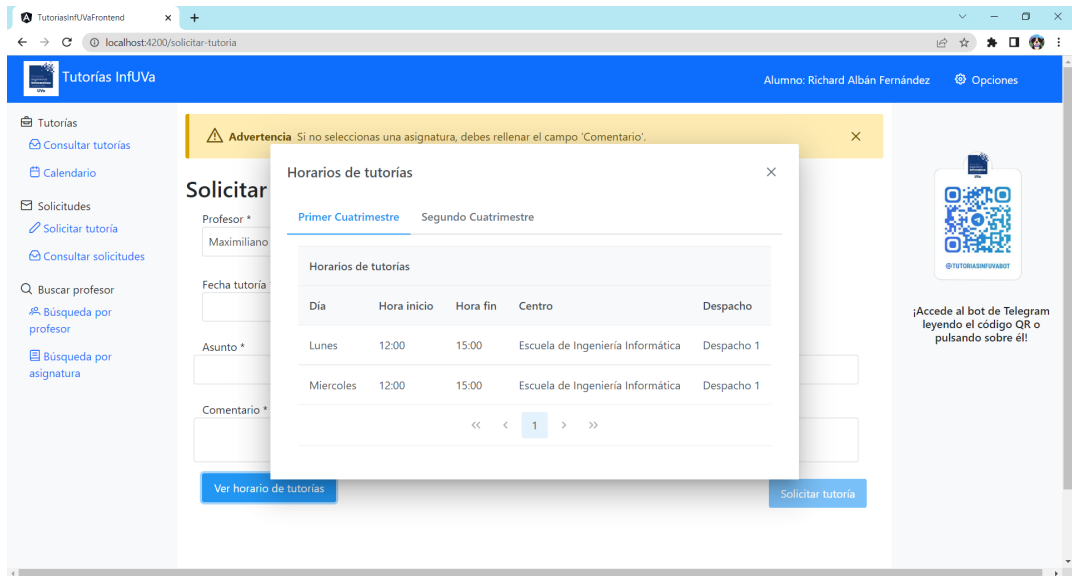


Figura A.8: Pantalla de Solicitar tutoría: horarios, de la web Tutorías InfUVA - Alumno

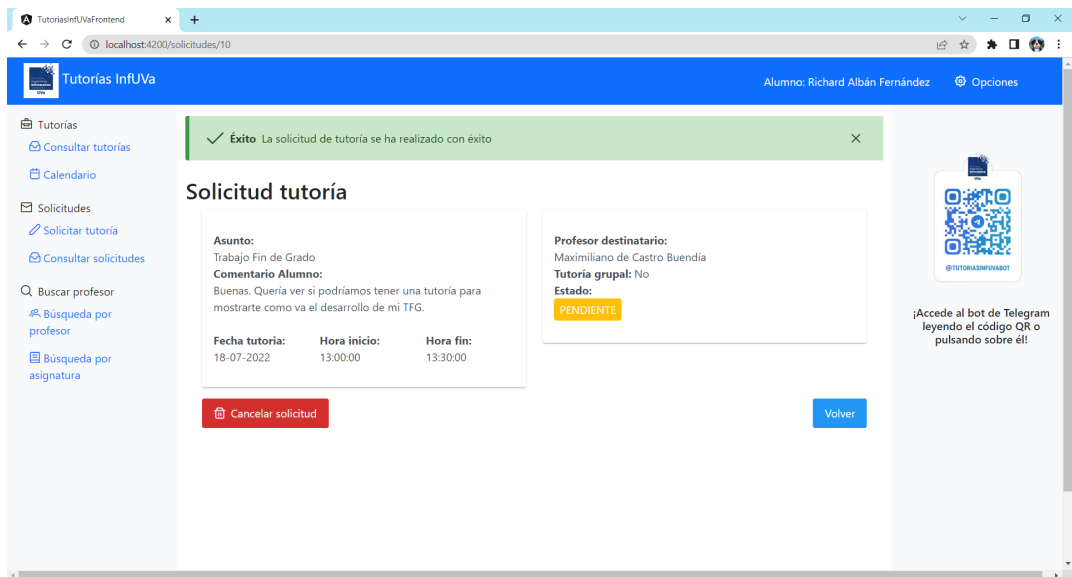


Figura A.9: Pantalla de Solicitud tutoría: pendiente, de la web Tutorías InfUVA - Alumno

- **A partir de mis asignaturas.** En este tipo de búsqueda, el bot restringe el espacio de búsqueda a las asignaturas en las que el alumno, que ha iniciado sesión, está matriculado, y a partir de esas asignaturas sigue el mismo proceso de búsqueda que la *Búsqueda general*.

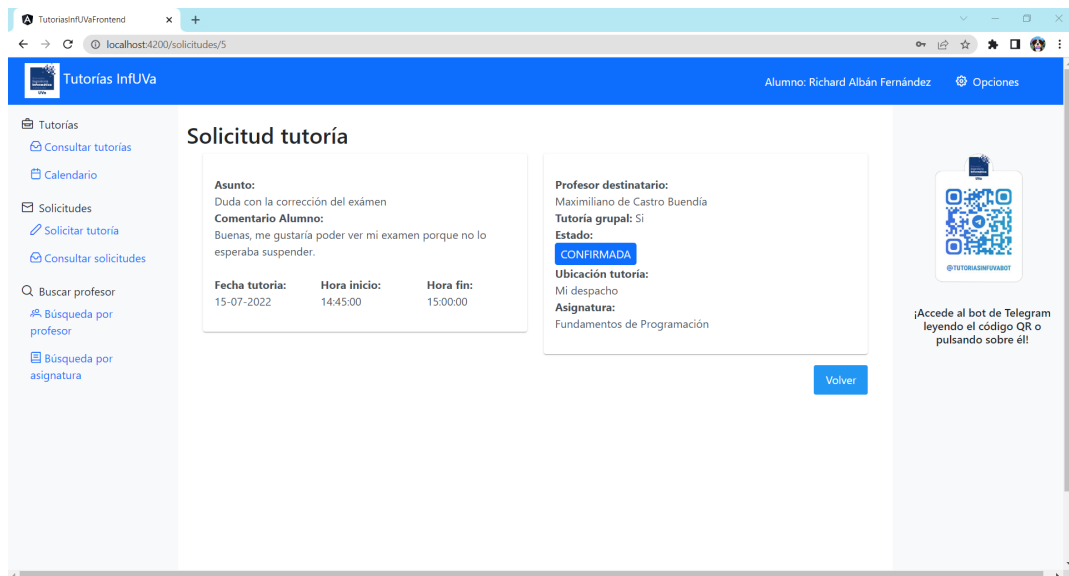


Figura A.10: Pantalla de Solicitud tutoría: confirmada, de la web Tutorías InfUVA - Alumno

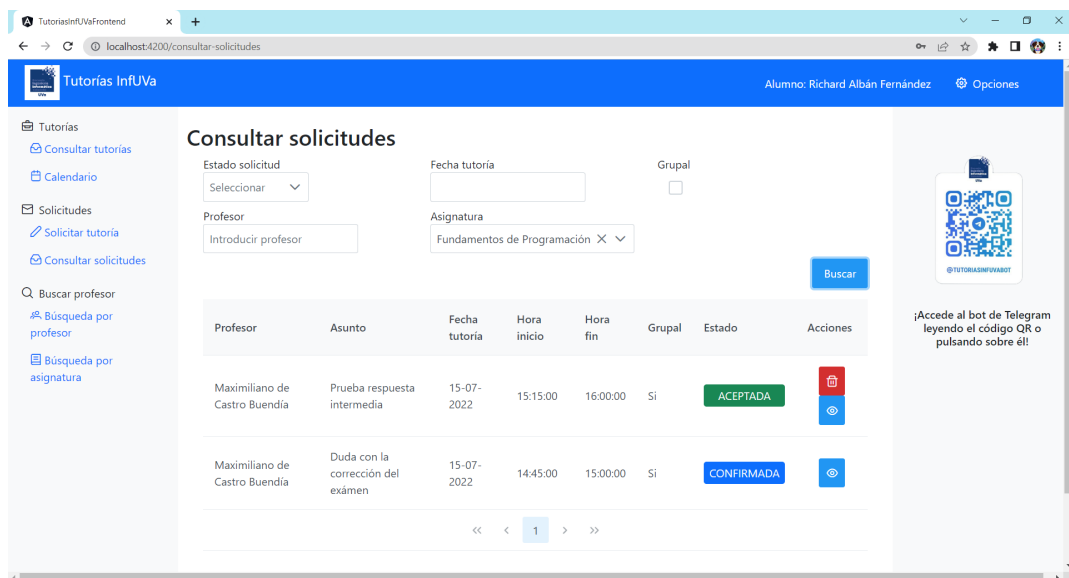


Figura A.11: Pantalla de Consultar solicitudes de la web Tutorías InfUVA - Alumno

- Por nombre asignatura.** En la búsqueda por nombre de asignatura, el bot solicita al alumno que introduzca el nombre de una asignatura. El nombre introducido puede ser parcial, por lo que el bot busca todas las asignaturas que contengan el texto introducido (ver Figura A.28). A partir de ahí (de las asignaturas coincidentes), sigue el mismo proceso que la *Búsqueda general*.

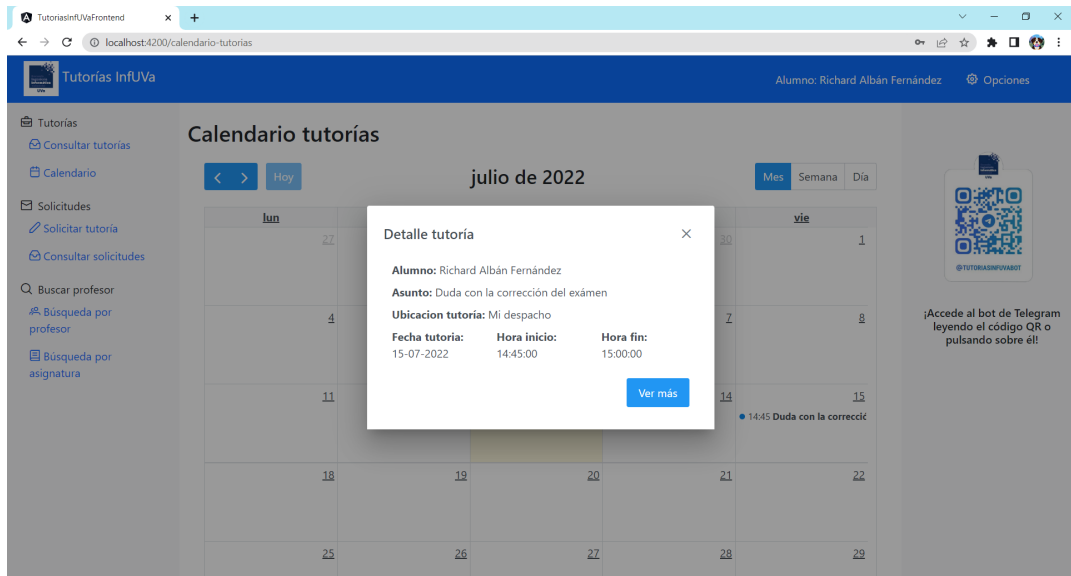


Figura A.12: Pantalla de Calendario tutorías: detalle tutoría, de la web Tutorías InfUVa - Alumno

Para facilitar el acceso a la web de Tutorías InfUVa, el bot de Telegram cuenta con el comando `/web`, proporcionando un acceso directo hasta dicha web (ver Figura A.35). Y, por último, para terminar con la sesión con el bot, éste cuenta con el comando `/quit`, que, además, nos mostrará un mensaje de despedida (ver Figura A.36).

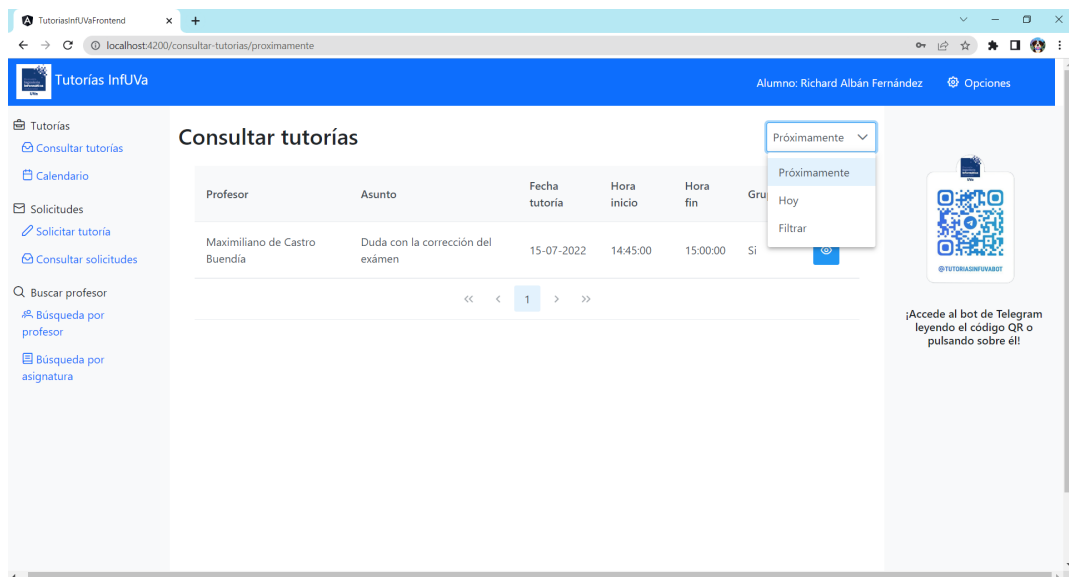


Figura A.13: Pantalla de Consultar tutorías de la web Tutorías InfUva - Alumno

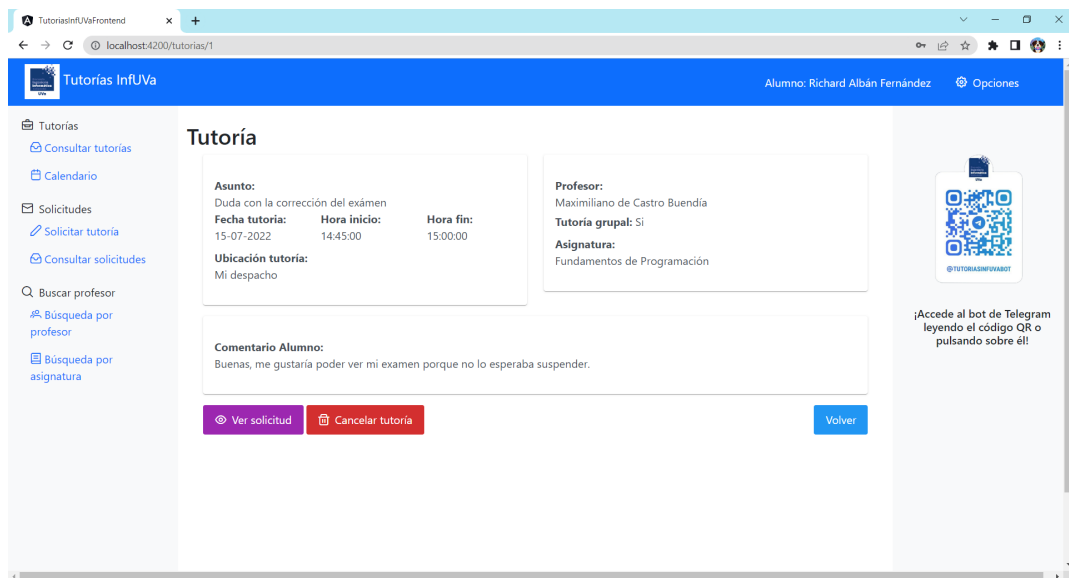


Figura A.14: Pantalla de Tutoría de la web Tutorías InfUva - Alumno

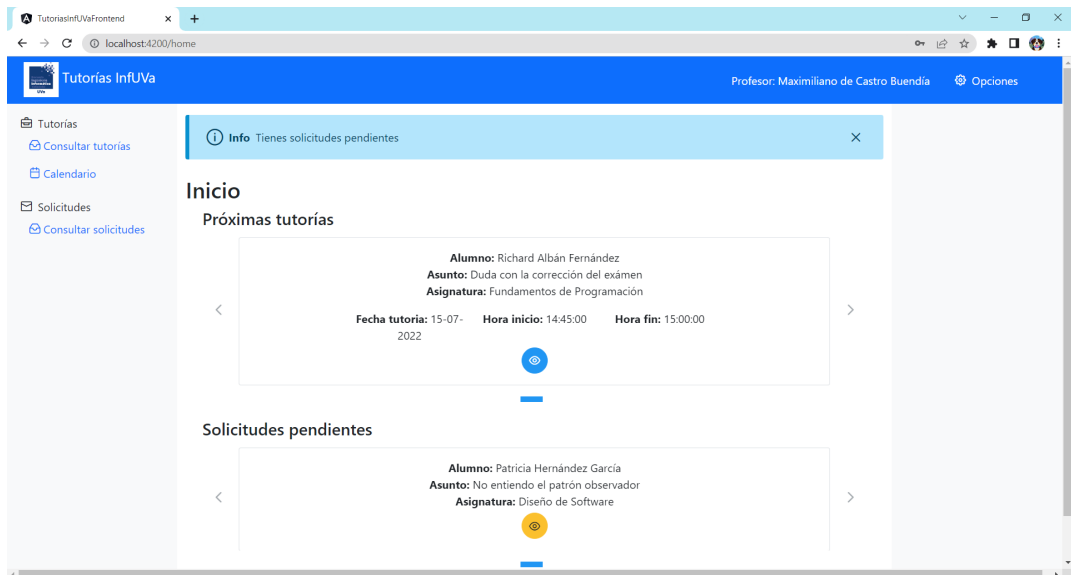


Figura A.15: Pantalla de Inicio de la web Tutorías InfUva - Profesor

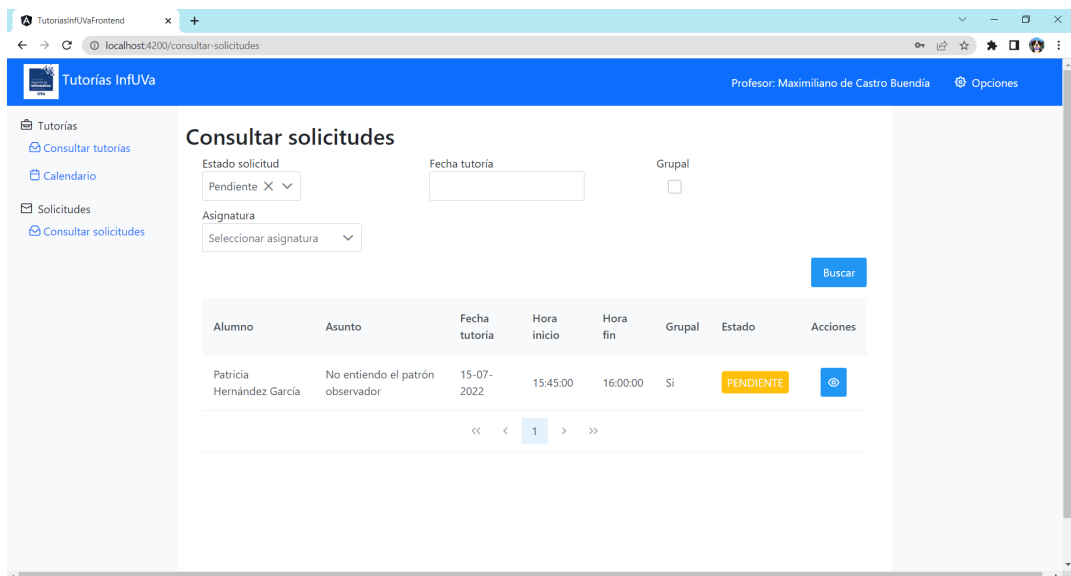


Figura A.16: Pantalla de Consultar solicitudes de la web Tutorías InfUva - Profesor

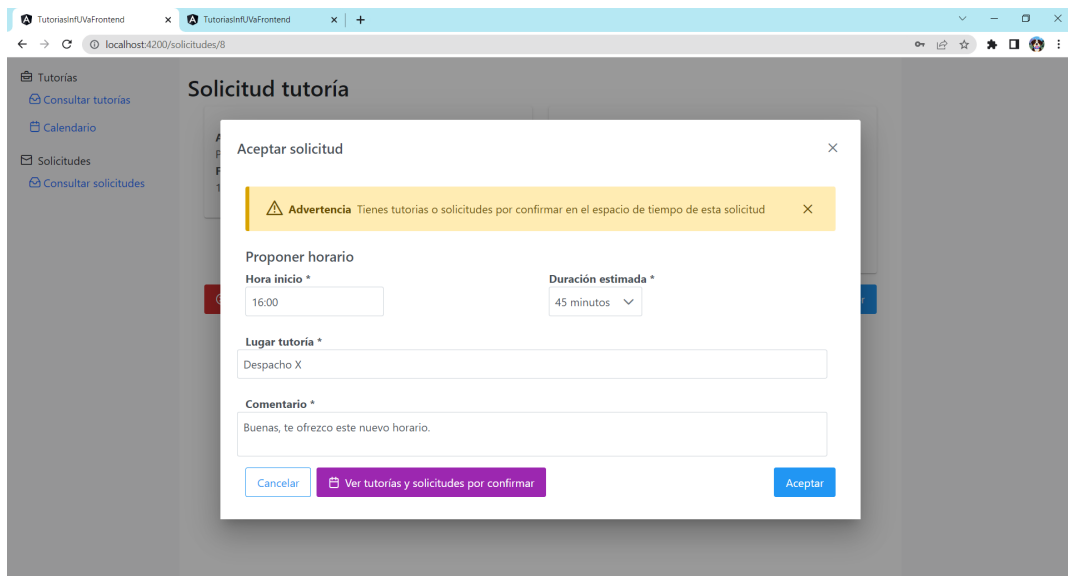


Figura A.17: Pantalla de Solicitud tutoría: aceptar solicitud, de la web Tutorías InfUva - Profesor

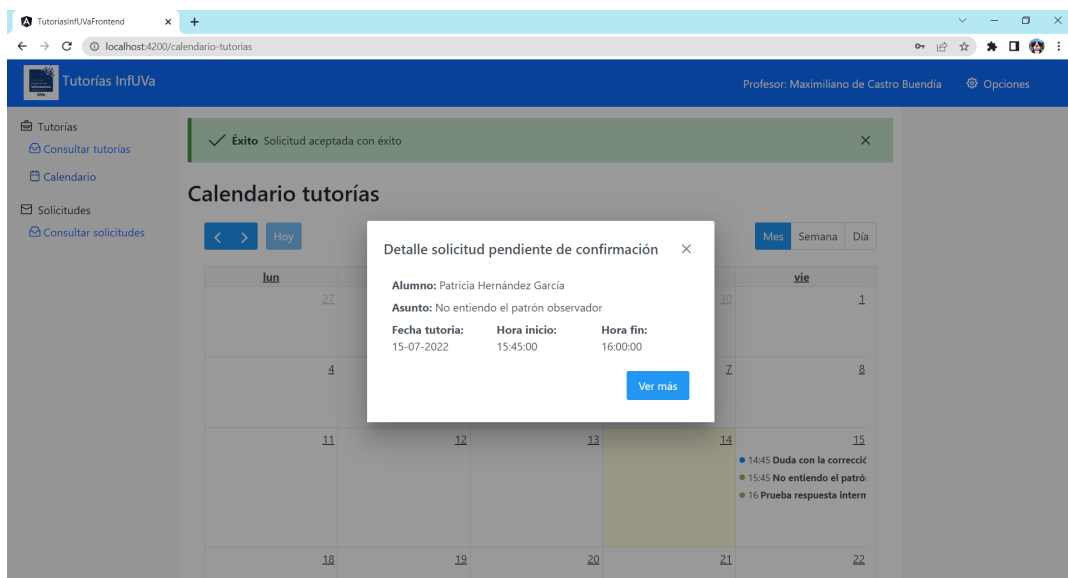


Figura A.18: Pantalla de Calendario tutorías: detalle solicitud pendiente de confirmación, de la web Tutorías InfUva - Profesor

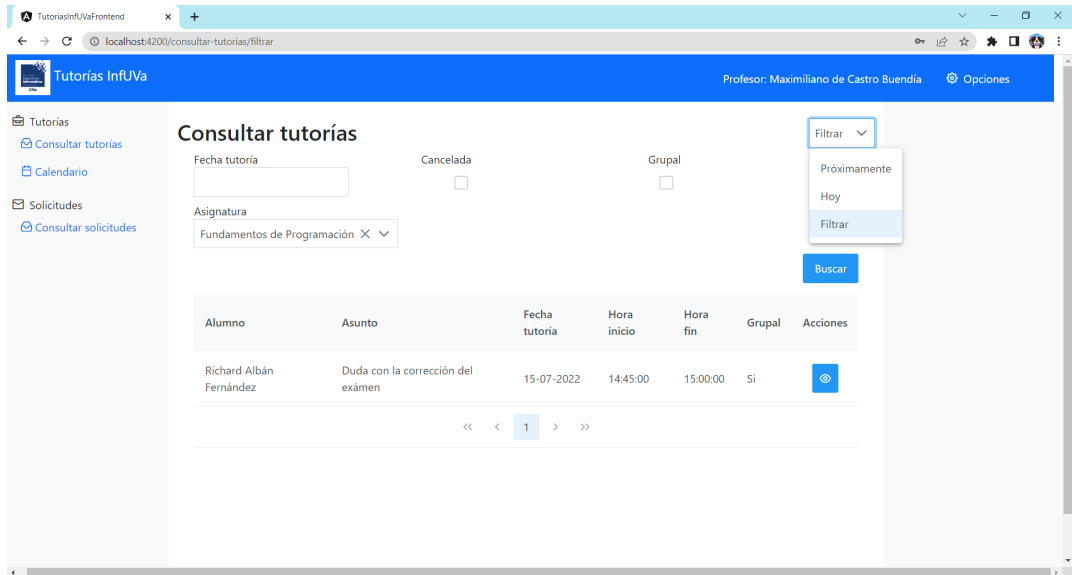


Figura A.19: Pantalla de Consultar tutorías de la web Tutorías InfUVA - Profesor

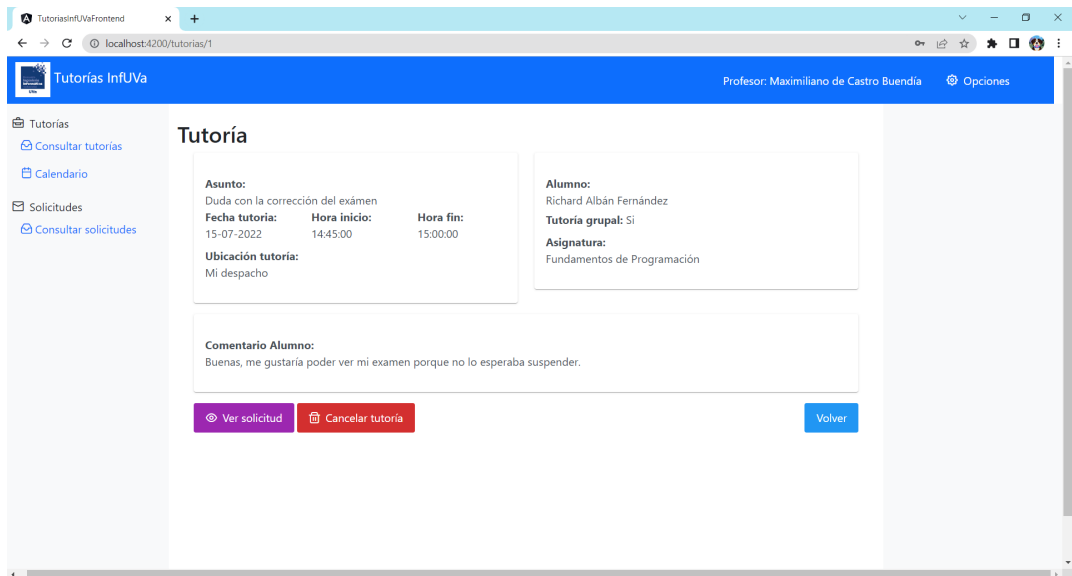


Figura A.20: Pantalla de Tutoría de la web Tutorías InfUVA - Profesor

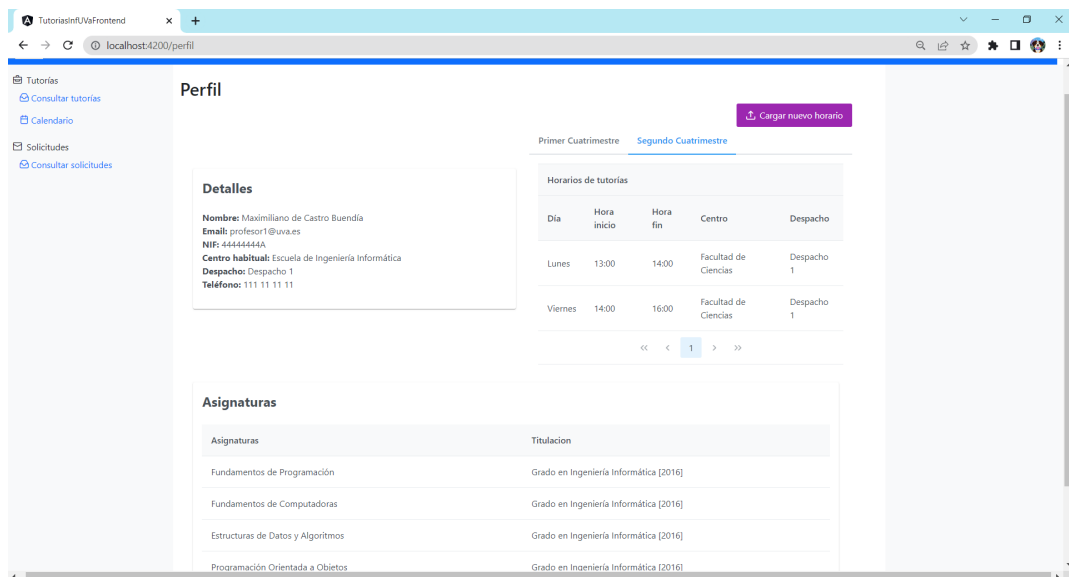


Figura A.21: Pantalla de Perfil de la web Tutorías InfUVA - Profesor

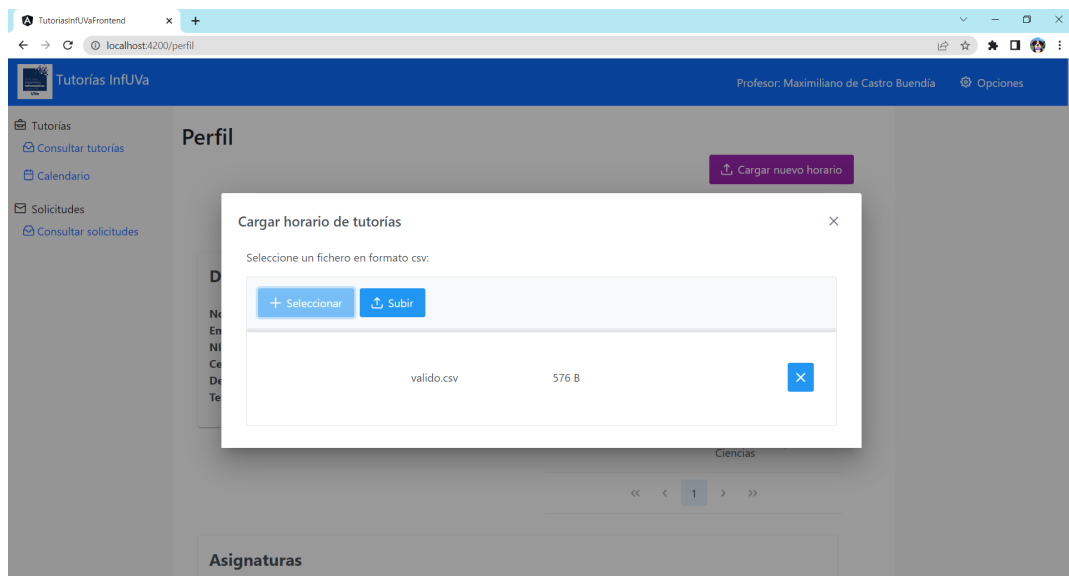


Figura A.22: Pantalla de Perfil: cargar horario tutorías, de la web Tutorías InfUVA - Profesor

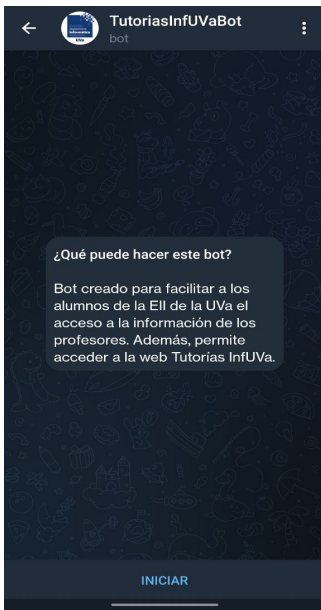


Figura A.23: Inicio

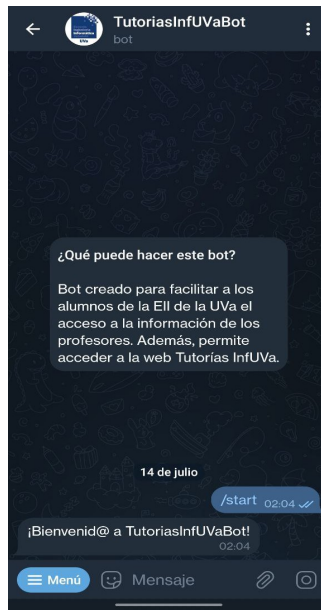


Figura A.24: Iniciar bot



Figura A.25: Comandos

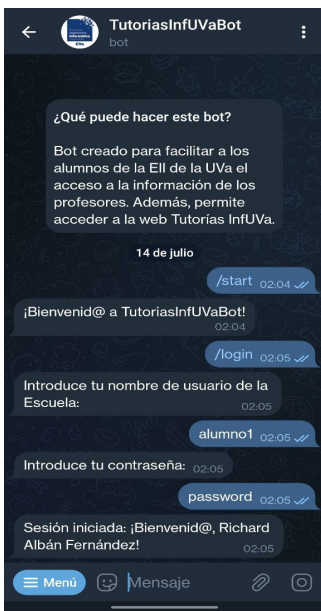


Figura A.26: Iniciar sesión en bot

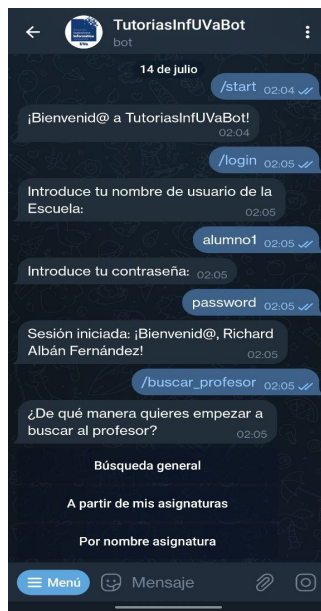


Figura A.27: Buscar profesor - tipo búsqueda

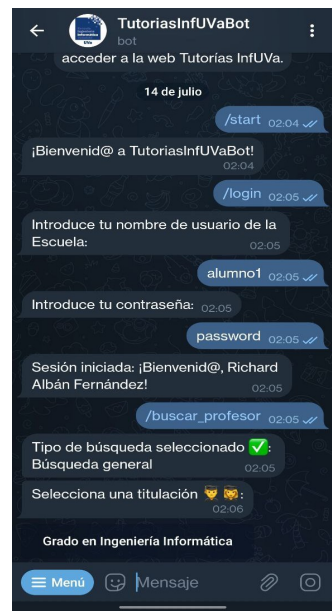


Figura A.28: Buscar profesor - titulaciones

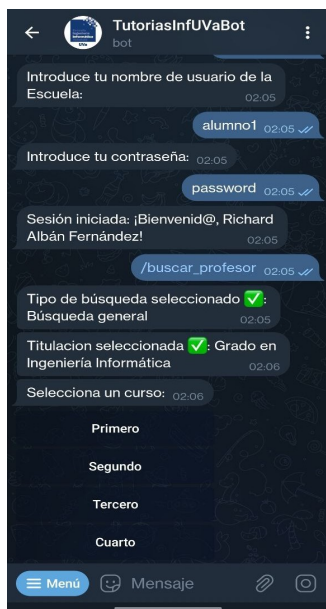


Figura A.29: Buscar profesor - cursos

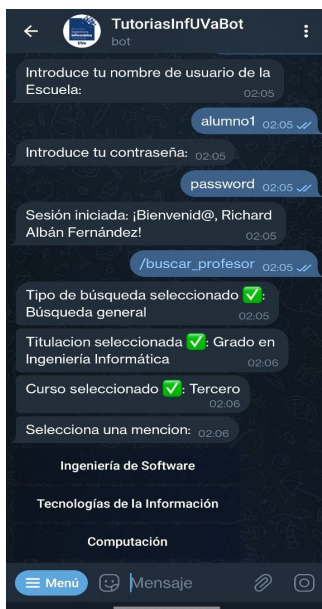


Figura A.30: Buscar profesor - menciones

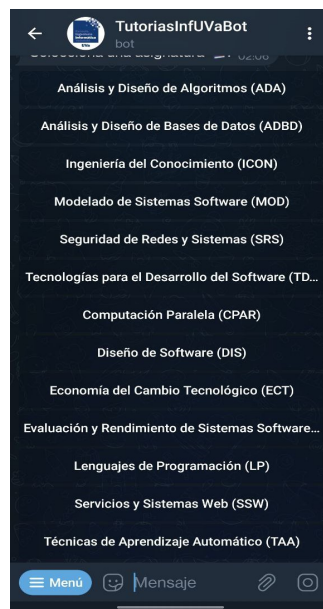


Figura A.31: Buscar profesor - asignaturas

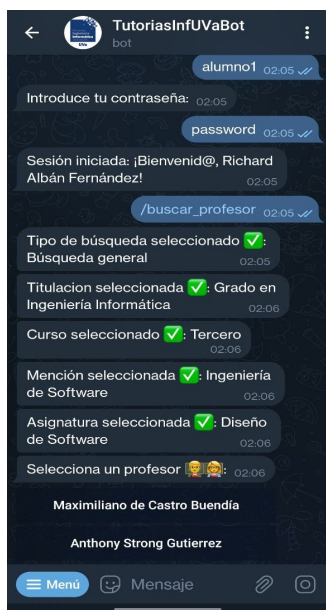


Figura A.32: Buscar profesor - profesores

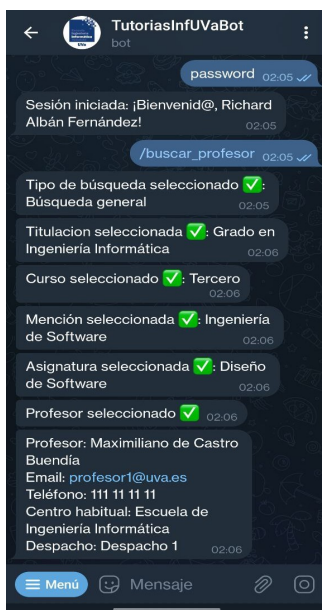


Figura A.33: Buscar profesor - detalle profesor



Figura A.34: Buscar profesor - por nombre

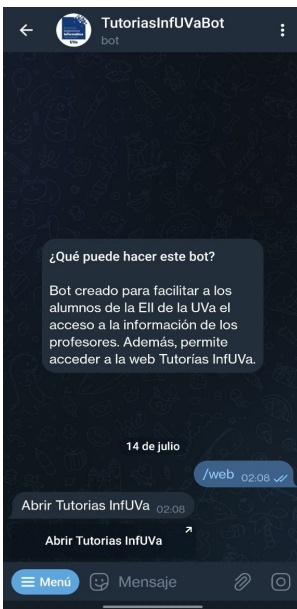


Figura A.35: Acceso a la web

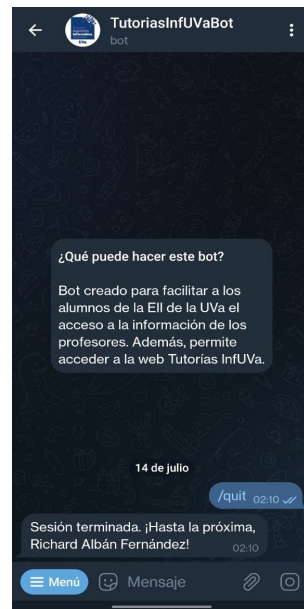


Figura A.36: Terminar sesión

Apéndice B

Resumen de enlaces adicionales

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

- Repositorio GitLab con el código y la documentación en Astah: <https://gitlab.inf.uva.es/ricalba/tfg-richardalbanfernandez>.
- Repositorio en GitHub utilizado a modo de backup: <https://github.com/Richard-11/TFG-RichardAlbanFernandez>
- Aplicación web Tutorías InfUVa desplegada: <http://virtual.lab.inf.uva.es:20094/>
- Bot de Telegram: <https://t.me/TutoriasInfUVaBot>