



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Mención Ingeniería de Software

Aplicación web basada en ReactJS enfocada en la
visualización y creación de horarios universitarios

Autor:

D. Francisco Cura Martín

Tutor Académico:

D. Joaquín Nicolás Adiego Rodríguez

Agradecimientos

En primer lugar, quiero agradecer a mi familia y amigos cercanos, quienes han resultado una irremplazable fuente de apoyo e inspiración, en los buenos y malos momentos, a lo largo de todos estos años.

A su vez, agradezco a mis compañeros de carrera, los cuales han hecho mis días mucho más agradables y me han enseñado a seguir adelante con ánimos de superación.

También quiero agradecer a mi tutor universitario, quien ha actuado como un gran compañero, sin dejar de lado su labor de supervisión, guiado y consejo propias de alguien con suma experiencia en este ámbito.

Resumen

Este proyecto consiste en el desarrollo de una aplicación web cuyo objetivo es la visualización por parte de los estudiantes y la creación por parte del profesorado, de horarios universitarios.

La principal motivación que supone el desarrollo de este proyecto de Trabajo Fin de Grado es el deseo de simplificar el proceso de gestión de horarios por parte del estudiantado y la creación de estos por parte del profesorado a través del Subdirector de Organización Docente.

Por ello, en el presente documento se constata la planificación del desarrollo, adaptado a la propia naturaleza de la situación que conlleva.

Por último, posteriormente se presenta dicho proceso en el desarrollo de la aplicación, haciendo uso del lenguaje JavaScript, junto con su Framework React.

Abstract

This project consists in the development of a web application whose objective is the visualizing of university timetables by students and the creation of them by the teaching staff.

The main motivation behind the development of this Final Degree Project is the desire to simplify the timetable management process for students and its creation by teaching staff through the Assistant Director of Teaching Organisation.

For this reason, this document shows the planning of the development, adapted to the nature of the situation involved.

Finally, this process is then presented in the development of the application, making use of the JavaScript language, together with its React Framework.

Índice general

Agradecimientos	2
Resumen	3
Abstract	4
Índice general	5
Introducción	16
1. Introducción.....	16
1.1. Contexto.....	16
1.2. Motivación	17
1.3. Objetivos	17
1.4. Objetivos personales	18
1.5. Usuarios objetivo	19
1.6. Estructuración de la memoria	19
 Planificación del proyecto	 21
2. Planificación del proyecto	21
2.1. Introducción	21
2.2. Glosario.....	21
2.3. Artefactos del proyecto.....	22
2.4. Restricciones del proyecto.....	23
2.5. Metodología empleada	23
2.5.1. Scrum.....	23
2.5.2. Organización en equipo	23
2.5.3. Elementos destacados	24
2.5.3.1. Eventos	24
2.5.3.2. Artefactos.....	25
2.6. Planificación	26
2.6.1. Adaptación de Scrum al marco del proyecto	26
2.6.2. Planificación inicial de sprints	27

2.6.3.	Análisis de riesgos	27
2.7.	Seguimiento del proyecto	34
2.7.1.	Incidencias relacionadas con riesgos	34
Análisis.....		36
3.	Introducción.....	36
3.1.	Análisis de requisitos.....	36
3.1.1.	Requisitos funcionales	36
3.1.2.	Requisitos de información	38
3.1.3.	Requisitos no funcionales.....	39
3.1.4.	Reglas de negocio	40
3.2.	Casos de uso	41
3.2.1.	Actores principales.....	41
3.2.2.	Diagrama de casos de uso	42
3.2.3.	Descripción de los casos de uso	43
3.3.	Modelo de dominio.....	53
3.3.1.	Subdirector.....	54
3.3.2.	Alumno.....	54
3.3.3.	Asignatura	54
3.3.4.	Evento	56
3.3.5.	Horario	56
3.3.6.	Date	57
3.4.	Diagramas de secuencia.....	58
Diseño.....		66
4.	Patrón de diseño.....	66
4.1.	Modelo – Vista – Controlador.....	66
4.1.1.	Modelo – Vista – Controlador en el proyecto	68
4.2.	Diseño web.....	68

4.2.1.	Diseño separado en páginas	68
4.2.2.	Componentes del proyecto	69
4.3.	Diagrama de componentes	75
4.4.	Prototipo de interfaces	75

Implementación.....81

5.1.	Herramienta de desarrollo	81
5.2.	Tecnologías empleadas	81
5.2.1.	React.....	82
5.2.2.	Node.js.....	83
5.2.3.	npm.....	83
5.2.4.	Metodología ágil	85
5.2.5.	Jira	86
5.2.6.	Unified Modeling Language	86
5.2.7.	Astah.....	87
5.2.8.	Sistema de gestión de versiones.....	87
5.3.	Implementación de funcionalidad	87
5.3.1.	Carga inicial y enrutamiento de la aplicación web.....	88
5.3.2.	Lectura de ficheros y obtención de eventos de calendario.....	89

Pruebas..... 91

6.1.	Pruebas realizadas sobre el cliente	91
6.1.1.	Test de navegabilidad	91
6.1.2.	Test de visualización genérica de horario troncal general	93
6.1.3.	Test de visualización genérica de horario troncal general con sesiones de laboratorio	94
6.1.4.	Test de visualización genérica de horario troncal específico	95
6.1.5.	Test de visualización genérica de horario troncal específico con sesiones de laboratorio	96

6.1.6.	Test de visualización genérica de horario de mención	97
6.1.7.	Test de visualización genérica de horario de mención con sesiones de laboratorio	98
6.1.8.	Test de visualización personalizada de horario	99
6.1.9.	Test de visualización personalizada de horario con sesiones de laboratorio	100
6.1.10.	Test de inicio de sesión en sección de administrador	101
6.1.11.	Test de carga de fichero en sección de administrador	101
6.1.12.	Test de visualización de horario en sección de administrador	102
6.1.13.	Test de manipulación de sesión de asignatura con Drag & Drop.....	102
6.1.14.	Test de manipulación de sesión de asignatura con Resize	103
6.1.15.	Test de guardado de fichero en sección de administrador	103
6.1.16.	Test de exportación de horario a formato PDF	104
6.1.17.	Test de modificación de sesión de asignatura en sección de administrador	104
6.1.18.	Test de borrado de sesión de asignatura en sección de administrador	105
6.1.19.	Test de borrado de incidencias de sesión de asignatura en sección de administrador	105
6.1.20.	Test de creación de sesión de asignatura en sección de administrador	106
6.1.21.	Test de limpieza de formulario de creación de sesión de asignatura en sección de administrador	107
6.1.22.	Test de cierre de sesión en sección de administrador	108

Conclusiones y trabajo futuro.....109

7.	Conclusiones y enfoques de trabajo a futuro	109
7.1.	Conclusiones.....	109
7.2.	Enfoques de trabajo futuro.....	110

Bibliografía.....	111
Guía de instalación.....	114
A.1. Instalación.....	114
A.1.1. Instalación de NVM	115
A.1.2. Ejecución del proyecto	115
Manual de usuario.....	117
B.1. Sección de inicio	117
B.2 Sección para visualización genérica	119
B.3 Sección para visualización personalizada	122
B.4 Pop – ups de asignaturas	124
B.5 Información adicional de asignaturas	125
B.6 Inicio de sesión	126
B.7 Cierre de sesión	128
B.8 Modificación de horario y asignaturas.....	129
B.9 Creación de asignaturas	131
B.10 Modificación de asignatura	132
B.11 Guardado, impresión en formato PDF y apertura de fichero	133

Índice de figuras

Figura 1: Diagrama de Casos de Uso - Sistema	42
Figura 2: Flujo del Caso de Uso - Iniciar sesión	43
Figura 3: Flujo del Caso de Uso - Seleccionar filtros	44
Figura 4: Flujo del Caso de Uso - Visualizar horario	45
Figura 5: Flujo del Caso de Uso - Crear sesión de asignatura.....	46
Figura 6: Flujo del Caso de Uso - Modificar sesión de asignatura	47
Figura 7: Flujo del Caso de Uso - Borrar sesión de asignatura.....	48
Figura 8: Flujo del Caso de Uso - Guardar fichero	49
Figura 9: Flujo del Caso de Uso - Cargar fichero	50
Figura 10: Flujo del Caso de Uso - Cerrar sesión	51
Figura 11: Flujo del Caso de Uso - Exportar a PDF	52
Figura 12: Modelo de dominio - Sistema	53
Figura 13: Diagrama de secuencia - Iniciar sesión	58
Figura 14: Diagrama de secuencia - Seleccionar filtros.....	59
Figura 15: Diagrama de secuencia - Visualizar horario	59
Figura 16: Diagrama de secuencia - Crear sesión de asignatura	60

Figura 17: Diagrama de secuencia - Modificar sesión de asignatura	61
Figura 18: Diagrama de secuencia - Borrar sesión de asignatura	61
Figura 19: Diagrama de secuencia - Guardar fichero	62
Figura 20: Diagrama de secuencia - Cargar fichero.....	63
Figura 21: Diagrama de secuencia - Cerrar sesión	64
Figura 22: Diagrama de secuencia - Exportar a PDF.....	65
Figura 23: Diagrama MVC	67
Figura 24: Diagrama de Componentes	75
Figura 25: Prototipo de interfaz - Sección introducción.....	76
Figura 26: Prototipo de interfaz - Visualización genérica	77
Figura 27: Prototipo de interfaz - Visualización personalizada	78
Figura 28: Prototipo de interfaz - Inicio de sesión	78
Figura 29: Prototipo de interfaz - Modificación de horario	79
Figura 30: Prototipo de interfaz - Creación de asignaturas	80
Figura 31: Fragmento de código - Lectura de fichero y creación de eventos.....	90
Figura 32: Comandos de verificación de versiones	115
Figura 33: comando de instalación de Node Package Manager	116
Figura 34: comando de ejecución de proyecto	116

Figura 35: Retroalimentación de ejecución	116
Figura 36: Diseño - Sección introducción	118
Figura 37: Diseño - Visualización genérica - estándar	119
Figura 38: Diseño - Visualización genérica - criterios	120
Figura 39: Diseño - Visualización genérica - laboratorios	121
Figura 40: Diseño - Visualización personalizada - estándar	122
Figura 41: Diseño - Visualización personalizada - criterios, asignaturas	123
Figura 42: Diseño - Visualización genérica - pop-ups.....	124
Figura 43: Diseño - Visualización genérica - información adicional.....	125
Figura 44: Diseño - Inicio de sesión	126
Figura 45: Diseño - Inicio de sesión - credenciales erróneas	127
Figura 46: Diseño - Cierre de sesión	128
Figura 47: Diseño - Modificación de asignatura - Drag & Drop.....	129
Figura 48: Diseño - Modificación de asignatura - Resize.....	130
Figura 49: Diseño - Creación de asignatura.....	131
Figura 50: Diseño - Modificación de asignatura - formulario.....	132
Figura 51: Diseño - Guardado, Impresión PDF y apertura de fichero.....	133

Índice de tablas

Tabla 1: Conceptos.....	22
Tabla 2: Roles de Scrum.....	24
Tabla 3: Planificación inicial de sprints.....	27
Tabla 4: Riesgos - Probabilidad - Impacto	27
Tabla 5: Riesgo 0 - Falta de disponibilidad	28
Tabla 6: Riesgo 1 - Falta de experiencia de planificación.....	28
Tabla 7: Riesgo 2 - Falta de formación en tecnologías	29
Tabla 8: Riesgo 3 - Falta de retroalimentación.....	29
Tabla 9: Riesgo 4 - Falta de medios	30
Tabla 10: Riesgo 5 - Nuevas especificaciones	30
Tabla 11: Riesgo 6 - Baja calidad de código.....	31
Tabla 12: Riesgo 7 - Inexperiencia en riesgos.....	31
Tabla 13: Riesgo 8 - Planificación de riesgos insuficiente	32
Tabla 14: Riesgo 9 - Diseño de interfaz incorrecto	32
Tabla 15: Riesgo 10 - Dependencia de componentes externos	33
Tabla 16: Riesgo 11 - Incumplimiento de la fecha de entrega.....	33

Tabla 17: Seguimiento del proyecto	34
Tabla 18: Requisitos funcionales	37
Tabla 19: Requisitos de información	38
Tabla 20: Requisitos no funcionales	39
Tabla 21: Reglas de negocio	40
Tabla 22: Test de navegabilidad	92
Tabla 23: Test de visualización genérica de horario troncal general.....	93
Tabla 24: Test de visualización genérica de horario troncal general con sesiones de laboratorio	94
Tabla 25: Test de visualización genérica de horario troncal específico	95
Tabla 26: Test de visualización genérica de horario troncal específico con sesiones de laboratorio	96
Tabla 27: Test de visualización genérica de horario de mención	97
Tabla 28: Test de visualización genérica de horario de mención con sesiones de laboratorio	98
Tabla 29: Test de visualización personalizada de horario.....	99
Tabla 30: Test de visualización personalizada de horario con sesiones de laboratorio	100
Tabla 31: Test de inicio de sesión en sección de administrador	101

Tabla 32: Test de carga de fichero	101
Tabla 33: Test de visualización de horario en sección de administrador	102
Tabla 34: Test de manipulación de sesión de asignatura con Drag & Drop	102
Tabla 35: Test de manipulación de sesión de asignatura con Resize.....	103
Tabla 36: Test de guardado de fichero en sección de administrador.....	103
Tabla 37: Test de exportación de horario a formato PDF	104
Tabla 38: Test de modificación de sesión de asignatura en sección de administrador	104
Tabla 39: Test de borrado de sesión de asignatura en sección de administrador	105
Tabla 40: Test de borrado de incidencias de sesión de asignatura en sección de administrador	105
Tabla 41: Test de creación de sesión de asignatura en sección de administrador.....	106
Tabla 42: Test de limpieza de formulario de sesión de asignatura en sección de administrador	107
Tabla 43: Test de cierre de sesión en sección de administrador	108

Capítulo 1

Introducción

1. Introducción

1.1. Contexto

La planificación de la vida universitaria requiere una suma cantidad de aspectos técnicos y personales en la vida de las personas, uno fundamental es la gestión de las responsabilidades que esta nueva etapa conlleva.

A lo largo de esta etapa, los estudiantes se ven sometidos a un proceso de cambio en sus vidas, siendo necesario adaptarla a la rutina de clases universitaria divulgada a través de numerosos documentos, entre ellos los horarios universitarios.

A su vez, el profesorado juega un papel fundamental en esta labor, puesto que, en consenso con el resto de la comunidad universitaria, son los planificadores y creadores de estos documentos.

De cara a afrontar este problema, hasta ahora en la creación de horarios ha desempeñado una labor fundamental una herramienta cuyas funciones a lo largo de los años presentan margen de mejora en términos de flexibilidad, apariencia, utilidad y accesibilidad.

Para abordar tal magnitud de desafío que supone este proceso, se ha llevado a cabo el desarrollo del presente proyecto como Trabajo Fin de Grado de Ingeniería Informática, en la mención de Ingeniería de Software, con la intención de modernizar y optimizar un aspecto tan fundamental como es la gestión de horarios, a través de la utilización de un sistema eficiente que simplifique dicha labor.

1.2. Motivación

La idea principal detrás de este proyecto y la aplicación resultado de este, surge de la necesidad de aportar al estudiantado y profesorado una herramienta que sirva de solución a la gestión y creación de horarios.

En este trabajo se establece el desarrollo de una aplicación web que permita la visualización de horarios personalizados por parte de los estudiantes y la creación de horarios por parte de los profesores, de una manera más dinámica y actualizada que facilite ambas tareas.

Por ello, por parte del alumnado se pretende permitir el filtro y visualización de los horarios existentes en función de su plan de estudios, curso en el que se encuentren, grupos a los que pertenezcan, aulas en las que se desarrollen sus actividades universitarias y, por último, por asignaturas que cursen actualmente.

Por parte del profesorado, la pretende mejorar la eficiencia en la planificación de horarios, de forma que se realice una gestión conjunta de las asignaturas impartidas en diferentes grados y menciones, manteniéndose separadas en horarios distintos en base a criterios como el año de cursado, el cuatrimestre y la mención a la que pertenecen, entre otros aspectos.

De este modo, se posibilita la creación de un sistema confiable y seguro en el que relegar la visualización, creación, actualización y publicación de horarios.

1.3. Objetivos

El objetivo de este proyecto es la creación de un sistema confiable enfocado en una aplicación web que permita la visualización, creación, actualización y consiguiente publicación de horarios universitarios. Por ello, los objetivos principales de dicho proyecto son los siguientes:

- Realización de un análisis previo de las tecnologías existentes más adecuadas con el objetivo de afianzar las bases de la aplicación a través de su uso eficiente.

- Realización de un análisis de funcionalidades que este sistema debe desempeñar con el objetivo de satisfacer la interacción de los usuarios al tratar de visualizar, crear, actualizar o publicar horarios.
- Diferenciado de usuarios del sistema en usuarios comunes y administrador.
- Realización de una batería de pruebas robusta que permita determinar el correcto funcionamiento del sistema en diversas situaciones.
- Diseño de una interfaz de usuario sencilla, intuitiva y fiel a los estándares universitarios, permitiendo la visualización, creación, actualización y publicación de horarios.
- Automatización de procesos que reduzcan la carga de trabajo manual en la visualización, creación, actualización, modificación y publicación de horarios, con el fin de evitar errores y situaciones redundantes.

1.4. Objetivos personales

Los objetivos personales del autor, relacionados con el desarrollo de este proyecto, se basan en:

- Comprensión de las tecnologías vanguardistas en el desarrollo de aplicaciones web.
- Comprensión de la creación de una aplicación web desde su inicio.
- Comprensión del proceso de implementación y despliegue de una aplicación web.
- Fragmentación, distribución y organización del trabajo requerido por el proceso de desarrollo, empleando metodologías ágiles.
- Adquisición de experiencia y fluidez en el entorno del desarrollo web actual, aprendiendo a emplear Frameworks útiles y fundamentales en el desarrollo de aplicaciones relacionadas con este ámbito.
- Adquisición de habilidades de diseño de interfaces intuitivas.

1.5. Usuarios objetivo

La propuesta de este proyecto es objeto de beneficio para diferentes tipos de usuarios:

En primer lugar, la aplicación se encuentra enfocada al uso por su parte de alumnos universitarios, con la finalidad de visualizar sus horarios en base a diferentes criterios. Todo ello con el objetivo de permitir una simplificación del proceso de adaptación a las circunstancias personales que suponen el cursado de diferentes tipos de asignaturas.

En segundo lugar, del uso de la aplicación también puede beneficiarse el profesorado universitario, con la finalidad de emplear un proceso simplificado y optimizado de creación, actualización, modificación y publicación de documentos relacionados con horarios.

Por último, como aspecto adicional el sistema podría adaptarse a futuras necesidades, permitiendo su empleo en diferentes instituciones o su integración en otras plataformas universitarias existentes.

1.6. Estructuración de la memoria

El siguiente documento se haya estructurado en los siguientes capítulos:

- **Capítulo 2 Planificación del proyecto**, en este capítulo se detallan los aspectos relacionados con la planificación, haciéndose hincapié en la metodología y organización del trabajo durante el desarrollo, así como en diversos elementos entre los que destacan los eventos y el análisis de riesgos, los cuales juegan un papel clave en este ámbito.
- **Capítulo 3 Análisis**, en este capítulo se detallan aspectos relacionados con el análisis de requisitos, casos de uso y modelos del sistema.
- **Capítulo 4 Diseño**, en este capítulo se presenta el patrón de diseño a desarrollar en la implementación de la aplicación, así como los componentes empleados y su prototipo de interfaces.

- **Capítulo 5 Implementación**, en este capítulo se presentan aspectos del proceso de desarrollo de la aplicación web, tales como herramientas de desarrollo, tecnologías empleadas en diversos ámbitos y, por último, detalles técnicos de la implementación de su funcionalidad.
- **Capítulo 6 Pruebas**, en este capítulo se presenta la batería de pruebas realizadas sobre la aplicación web, con el objetivo de comprobar su correcto funcionamiento.
- **Capítulo 7 Conclusiones y trabajo a futuro**, en este capítulo se presentan las conclusiones del proyecto desarrollado y enfoques a futuro que podría tomar.

A su vez, posterior al desarrollo de estos capítulos, se proporcionan una serie de apartados adicionales como la bibliografía empleada, una Guía de instalación y un Manual de usuario.

Capítulo 2

Planificación del proyecto

2. Planificación del proyecto

2.1. Introducción

En este capítulo es presentado el plan de desarrollo seguido en la creación de la aplicación web enfocada en la gestión de horarios.

Este proyecto, realizado por el alumno Francisco Cura Martín, ha sido desarrollado utilizando una de las metodologías ágiles más relevante, siendo adaptada a las necesidades propias de la naturaleza de la situación.

2.2. Glosario

En la siguiente tabla se especifican diferentes elementos que aparecerán citados con frecuencia a lo largo de este documento, junto con su significado y papel dentro del proyecto:

Siglas	Especificación de concepto
VS Code	Visual Studio Code, editor de código fuente desarrollado por Microsoft, con soporte para numerosos lenguajes de programación (<i>Visual Studio Code, s.f.</i>).
Astah	Astah professional, herramienta de modelado de software enfocada en la creación de diagramas.
JS	JavaScript, lenguaje de programación ampliamente utilizado en el desarrollo web.
JSON	JavaScript Object Notation, formato de texto sencillo para el intercambio de datos, basado en un subconjunto de la notación de objetos de JavaScript.
React	React, Framework de JavaScript enfocado en la creación de interfaces dinámicas y reutilizables, empleado componentes.
Semantic UI	Framework destinado a diseño de interfaces de manera responsive empleando HTML y CSS.

Tabla 1: Conceptos

2.3. Artefactos del proyecto

En este apartado son especificados los artefactos del proyecto que son considerados como piezas clave en el desarrollo del presente proyecto, siendo por ello los siguientes:

- Plan enfocado en el desarrollo software
- Análisis de requisitos y casos de uso
- Diseño
- Implementación
- Batería e informe de pruebas
- Conclusiones y enfoque a futuro

2.4. Restricciones del proyecto

Queda establecido que la única restricción presente en el proyecto es su finalización, entrega y defensa, dentro del periodo ordinario de Trabajos de Fin de Grado.

2.5. Metodología empleada

2.5.1. Scrum

Scrum es un marco de trabajo perteneciente a las metodologías ágiles, cuyo uso fomenta la gestión y desarrollo de proyectos, creado por Jeff Sutherland y Jeff Schwaber en la década de 1990, contribuyendo posteriormente al Manifiesto Ágil en 2001 y consolidando Scrum como uno de los enfoques más populares en la gestión de proyectos Ágiles.

Esta metodología se centra en ciclos iterativos llamados sprints, donde equipos organizados trabajan con el fin de proporcionar entregas incrementales del producto, adaptando el proceso a cambios y mejorando de forma continuada a través de reuniones diarias, de revisión y retrospectivas.

2.5.2. Organización en equipo

El enfoque fundamental de este marco de trabajo está basado en el trabajo en equipo de 10 o menos integrantes, dividiéndose el trabajo en “milestones” u objetivos que deben poderse completar a través de iteraciones de duración fijada y limitada, estas iteraciones reciben el nombre de sprints.

En la presente tabla se muestran los tres roles más determinantes en un equipo de metodología Scrum, y sus funciones asociadas:

Rol	Funciones y responsabilidades
Scrum Master	Encargado de la monitorización del proceso, eliminando obstáculos, ayudando al equipo a alcanzar su máximo rendimiento; todo ello a través de la cultura de mejora continua y actuando como líder al servicio del resto de integrantes.
Product Owner	Propietario del producto, determina los criterios que debe cumplir el producto, tomando decisiones relacionadas con la funcionalidad y cambios asociados, asegurándose que se trabaje en lo más valioso del producto.
Equipo de desarrollo	Papel fundamental en el desarrollo del producto. Diseñan, desarrollan, prueban e implementan el producto, entregando incrementos funcionales del mismo al final de cada sprint, autoorganizándose durante todo el proceso.

Tabla 2: Roles de Scrum

2.5.3. Elementos destacados

2.5.3.1. Eventos

Durante el proceso de desarrollo en el marco de trabajo proporcionado por esta metodología, se emplean un conjunto de eventos clave que permiten realizar una gestión del desarrollo y que permiten lograr los objetivos establecidos. Estos elementos proporcionan al equipo una serie de pautas que permite el trabajo de manera eficiente y el mantenimiento de una apta comunicación entre las diferentes secciones del proyecto. Dichos eventos son los siguientes:

- **Sprint**, ciclo de trabajo iterativo con una duración entre 1 y 4 semanas en el que se desarrolla una versión funcional, incremental y entregable del producto.
- **Sprint Planning**, reunión inicial durante cada sprint en la que se definen los objetivos a cumplir durante la iteración, y cómo cumplirlos.

- **Sprint Review**, reunión final de cada sprint en la que se analiza el trabajo realizado, presentando la versión incremental del producto a los stakeholders y recopilando la retroalimentación de cara a futuros avances.
- **Sprint Retrospective**, reunión posterior a la revisión del sprint, en la que se reflexiona acerca de las dificultades encontradas, el enfoque tomado, así como los aspectos que pueden mejorarse y que acciones tomar de cara a ello.
- **Daily Scrum**, reunión diaria con una duración no superior a 15 minutos, destinada a sincronizar el trabajo en equipo, verificar el progreso de cara a cumplir con los objetivos del sprint y detectar problemas durante el desarrollo del ciclo iterativo.
- **Product Backlog Refinement**, es el proceso de revisión, actualización y mejora continua del Product Backlog; frecuentemente se realiza al menos 1 vez por cada Sprint, con la participación del Product Owner, Equipo de desarrollo y, si es necesario, el Scrum Master.

2.5.3.2. Artefactos

Durante el proceso de desarrollo, a su vez, se emplean una serie de artefactos con el objetivo de garantizar el trabajo coordinado y la comunicación entre miembros y stakeholders. Estos artefactos, dado que son evolutivos a lo largo del proyecto, deben tratarse con responsabilidad por parte de todos los miembros, enfocándose en su revisión y actualización oportuna. Los artefactos son los siguientes:

- **Historias de usuario**, forma de describir funcionalidades o requerimientos del usuario final.
- **Sprint Backlog**, conjunto de elementos seleccionados del Product Backlog para ser cumplimentados durante el sprint actual. Contiene objetivos del sprint, historias de usuario seleccionadas, tareas desglosadas y el estado de cada una de ellas.

- **Product Backlog**, conjunto de requisitos ordenados del producto ordenados por prioridad, contiene funcionalidades, mejoras y correcciones pendientes. Es responsabilidad del Product Owner y supone la principal fuente de trabajo para el Equipo de desarrollo.
- **Versión incremental del producto**, resultado del proceso de la iteración de desarrollo durante un sprint. Debe estar basado en el Product Backlog, al igual que ser funcional y utilizable, después de haber sido implementado, probado e integrado con el resto de las funcionalidades previas.

2.6. Planificación

2.6.1. Adaptación de Scrum al marco del proyecto

La elección de esta metodología se debe al marco de trabajo que ofrece, permitiendo distribuir la carga de trabajo de forma gradual y con total flexibilidad. A su vez, esta distribución del trabajo permite disponer de una organización y adaptación frente a todo tipo de circunstancias.

Una vez clarificado este aspecto, se procede a especificar el reparto de los roles dentro de la metodología Scrum.

En primer lugar, el alumno actuará como Product Owner y único miembro del Equipo de desarrollo. El rol de Scrum Master e interesado será ejercido por el tutor académico de la universidad.

Los avances referentes a cada sprint se realizarán a través de una reunión presencial de seguimiento en intervalos de tiempo semanales, con el tutor académico de la universidad. En el caso de incompatibilidad por alguna de las partes, se tratará de establecer una nueva fecha de forma temprana. A su vez, se establece la existencia de una comunicación frecuente por correo electrónico, en caso de suma necesidad por alguna de las partes.

2.6.2. Planificación inicial de sprints

Para el desarrollo de cada sprint, ha decidido fijarse una duración de 2 semanas para cada uno de ellos, siendo adecuado para el alumno con el fin de un correcto desarrollo del proyecto.

Sprint	Fecha de comienzo	Fecha de finalización
Sprint 1	08/02/2025	22/02/2025
Sprint 2	22/02/2025	08/03/2025
Sprint 3	08/03/2025	22/03/2025
Sprint 4	22/03/2025	05/04/2025
Sprint 5	05/04/2025	19/04/2025
Sprint 6	19/04/2025	03/05/2025
Sprint 7	03/05/2025	17/05/2025
Sprint 8	17/05/2025	31/05/2025
Sprint 9	31/05/2025	14/06/2025
Sprint 10	14/06/2025	28/06/2025

Tabla 3: Planificación inicial de sprints

2.6.3. Análisis de riesgos

Un proyecto de esta magnitud exige la existencia de un análisis de riesgos que conlleve su realización, puesto que su manifestación supone una alteración del flujo normal en el transcurso del desarrollo. Por ello, a continuación, se detallan los posibles riesgos existentes a lo largo del proceso, junto con su probabilidad de manifestarse, su nivel de impacto, consecuencias asociadas, niveles de riesgo, planes de prevención y planes de contingencia o respuesta para su mitigación. A su vez, se detalla previamente la información necesaria para determinar el nivel de un riesgo en base a los criterios de probabilidad / impacto.

Prob / Imp	Bajo	Medio	Alto
Baja	Bajo	Bajo	Medio
Media	Bajo	Medio	Alto
Alta	Medio	Alto	Alto

Tabla 4: Riesgos - Probabilidad - Impacto

A partir del modelo de gestión de riesgos propuesto por Barry Boehm en el año 1989 y a través del análisis de la naturaleza del proyecto, ha sido posible determinar la susceptibilidad del proyecto a los siguientes riesgos:

R0	Carencia de disponibilidad
Descripción del riesgo	El alumno carece de tiempo a invertir en el desarrollo o sufre una baja temporal que le suponga distanciarse del proyecto
Probabilidad	Baja
Impacto	Alto
Nivel del riesgo	Medio
Consecuencia	Retrasos en la planificación establecida
Plan de mitigación	Reajustar la planificación de cara a paliar esta ausencia de disponibilidad
Plan de contingencia	Volver a planificar o añadir tiempo extra

Tabla 5: Riesgo 0 - Falta de disponibilidad

R1	Falta de experiencia de planificación
Descripción del riesgo	El alumno presenta dificultades que le impiden desarrollar adecuadamente la planificación propuesta en el proyecto
Probabilidad	Baja
Impacto	Alto
Nivel del riesgo	Medio
Consecuencia	Incumplimiento de los plazos fijados
Plan de mitigación	Reajuste de la planificación por parte del tutor
Plan de contingencia	Revisión y modificación del Sprint Backlog

Tabla 6: Riesgo 1 - Falta de experiencia de planificación

R2	Falta de formación en tecnologías
Descripción del riesgo	El alumno carece de formación previa en las tecnologías empleadas en el proyecto, afectando a aspectos como la calidad o el intervalo de entrega del producto.
Probabilidad	Media
Impacto	Medio
Nivel del riesgo	Medio
Consecuencia	Retrasos en el desarrollo, dificultades de implementación
Plan de mitigación	Capacitación previa al inicio del desarrollo, búsqueda de información y de mentoría experta
Plan de contingencia	Delimitar el alcance del proyecto o revisar los conocimientos adquiridos

Tabla 7: Riesgo 2 - Falta de formación en tecnologías

R3	Falta de retroalimentación
Descripción del riesgo	Carencias de comunicación por alguna de las partes del proyecto, falta de retroalimentación
Probabilidad	Baja
Impacto	Alto
Nivel del riesgo	Medio
Consecuencia	Retraso de revisiones de las actualizaciones del desarrollo
Plan de mitigación	Definir reuniones periódicas y canales de comunicación claros
Plan de contingencia	Establecer nuevas fechas de reunión o revisar de forma autónoma el proyecto, como última instancia

Tabla 8: Riesgo 3 - Falta de retroalimentación

R4	Falta de medios
Descripción del riesgo	Carencia de los medios necesarios causada por un problema técnico, que imposibilita la continuidad del proceso de desarrollo
Probabilidad	Baja
Impacto	Alto
Nivel del riesgo	Medio
Consecuencia	Aplazamiento de la planificación
Plan de mitigación	Delegar en recursos extra para situaciones de emergencia
Plan de contingencia	Adaptar el proyecto a nuevas herramientas, solicitar recursos a personas externas

Tabla 9: Riesgo 4 - Falta de medios

R5	Nuevas especificaciones
Descripción del riesgo	Incorporación de nuevas funcionalidades al proyecto en desarrollo
Probabilidad	Media
Impacto	Medio
Nivel del riesgo	Medio
Consecuencia	Modificaciones en la implementación
Plan de mitigación	Establecer especificaciones claras al inicio del proceso de planificación
Plan de contingencia	Adaptar el proyecto a nuevas herramientas, solicitar recursos a personas externas

Tabla 10: Riesgo 5 - Nuevas especificaciones

R6	Baja calidad de código
Descripción del riesgo	Carencias en la codificación de la aplicación, con aspectos ineficientes o fallas causadas por ello
Probabilidad	Baja
Impacto	Alto
Nivel del riesgo	Medio
Consecuencia	Ralentización del proceso de desarrollo y planificación
Plan de mitigación	Revisión de código y funcionalidades
Plan de contingencia	Refactorización de código para adaptarlo a estándares de calidad y manejo de errores

Tabla 11: Riesgo 6 - Baja calidad de código

R7	Inexperiencia en riesgos
Descripción del riesgo	Carencias en la identificación, evaluación o gestión de los riesgos que puedan aparecer en el desarrollo del proyecto
Probabilidad	Baja
Impacto	Medio
Nivel del riesgo	Bajo
Consecuencia	Falta de preparación ante riesgos no valorados
Plan de mitigación	Curso de capacitación en gestión de riesgos previo al inicio de la planificación
Plan de contingencia	Revisión y actualización de la planificación de riesgos, ejecución de un nuevo análisis de riesgos

Tabla 12: Riesgo 7 - Inexperiencia en riesgos

R8	Planificación de riesgos insuficiente
Descripción del riesgo	Planteamiento insuficiente de todos los posibles riesgos que pueden afectar al desarrollo del proyecto
Probabilidad	Media
Impacto	Alto
Nivel del riesgo	Alto
Consecuencia	Aparición de nuevos riesgos imprevistos
Plan de mitigación	Planificación completa y detallada de riesgos
Plan de contingencia	Reanalizar y valorar pausadamente todo los riesgos

Tabla 13: Riesgo 8 - Planificación de riesgos insuficiente

R9	Diseño de interfaz incorrecto
Descripción del riesgo	El diseño de las interfaces no satisface los propósitos adecuadamente o es incoherente
Probabilidad	Bajo
Impacto	Alto
Nivel del riesgo	Medio
Consecuencia	Interfaz desviada de los propósitos originales o poco intuitiva
Plan de mitigación	Establecer un proceso de especificación detallado acerca de la interfaz o diseño de prototipos incrementales sometidos a pruebas de usabilidad por parte de usuarios reales
Plan de contingencia	Reajustar el diseño en base al feedback recibido por parte de usuarios reales

Tabla 14: Riesgo 9 - Diseño de interfaz incorrecto

R10	Dependencia de componentes externos
Descripción del riesgo	Basar el funcionamiento del sistema componentes externos como APIs, librerías o frameworks puede afectar en caso de dejar de estar disponibles
Probabilidad	Bajo
Impacto	Alto
Nivel del riesgo	Medio
Consecuencia	Fallos en la aplicación, retrasos en el desarrollo
Plan de mitigación	Basar el desarrollo en tecnologías populares cuyos creadores mantengan actualizadas
Plan de contingencia	Buscar una tecnología que actúe de alternativa o implementar manualmente la funcionalidad si se trata de aspectos con cierta viabilidad

Tabla 15: Riesgo 10 - Dependencia de componentes externos

R11	Incumplimiento de la fecha de entrega
Descripción del riesgo	El proyecto no finaliza dentro del plazo acordado inicialmente
Probabilidad	Bajo
Impacto	Alto
Nivel del riesgo	Medio
Consecuencia	Retraso en la fecha de entrega, existencia de posibles penalizaciones
Plan de mitigación	Planificar fechas de entrega o revisión del proceso intermedias
Plan de contingencia	Reestructurar la planificación y realizar la entrega en periodo extraordinario

Tabla 16: Riesgo 11 - Incumplimiento de la fecha de entrega

2.7. Seguimiento del proyecto

A continuación, se muestra información acerca del seguimiento del proceso de desarrollo en base a la planificación propuesta inicialmente, respecto al periodo de comienzo y finalización estimados, así como las fechas reales de finalización de cada sprint.

Sprint	Comienzo estimado	Finalización Estimada	Comienzo real	Finalización real	Desvío
Sprint 1	08/02/2025	22/02/2025	10/02/2025	22/02/2025	2 días
Sprint 2	22/02/2025	08/03/2025	24/02/2025	12/03/2025	6 días
Sprint 3	08/03/2025	22/03/2025	10/02/2025	22/03/2025	2 días
Sprint 4	22/03/2025	05/04/2025	23/03/2025	07/04/2025	3 días
Sprint 5	05/04/2025	19/04/2025	06/04/2025	19/04/2025	1 día
Sprint 6	19/04/2025	03/05/2025	21/04/2025	05/05/2025	3 días
Sprint 7	03/05/2025	17/05/2025	08/05/2025	18/05/2025	6 días
Sprint 8	17/05/2025	31/05/2025	19/05/2025	02/06/2025	4 días
Sprint 9	31/05/2025	14/06/2025	02/06/2025	12/06/2025	4 días
Sprint 10	14/06/2025	28/06/2025	14/06/2025	24/06/2025	4 días

Tabla 17: Seguimiento del proyecto

2.7.1. Incidencias relacionadas con riesgos

Durante el proceso de desarrollo se han manifestado una serie de incidencias, las cuales hacen referencia a ciertos riesgos mencionados en el previo análisis de riesgos.

Inicialmente es posible destacar que el riesgo **R1** y sus consecuencias han estado presentes, puesto que la presencia de dificultades, bien externas o internas del proyecto, ha supuesto un desvío en el plazo de comienzo o finalización reales, a lo largo de los diversos sprints.

Asimismo, el riesgo **R2** y sus consecuencias han tenido su mayor impacto en el inicio del desarrollo, puesto que la carencia de conocimientos ha ralentizado la implementación de funcionalidades y aspectos complementarios del trabajo práctico.

Durante la mencionada implementación, los riesgos **R5** y **R6** se han manifestado a través de la incorporación de nuevas funcionalidades y de la calidad del código respectivamente, puesto que, sumado a los anteriores riesgos, han supuesto una ralentización del proceso de desarrollo.

Por último, el riesgo **R9** ha estado presente en varias etapas del desarrollo, siendo necesaria una reevaluación de las interfaces desarrolladas y el pertinente rediseño de ellas.

Capítulo 3

Análisis

3. Introducción

Este capítulo se encuentra dedicado a un análisis detallado del funcionamiento de la aplicación. Dicho análisis se pretende realizar elaborando un estudio de los requisitos del sistema, así como un estudio de los casos de uso en los que se ven involucrados los actores relacionados con el entorno de uso de la aplicación.

3.1. Análisis de requisitos

El análisis de los requisitos se basa en la identificación, documentación y validación de las necesidades y expectativas de usuarios y stakeholders, definiendo el propósito de un sistema de manera efectiva y estableciendo las bases para el diseño, implementación y pruebas.

3.1.1. Requisitos funcionales

En la siguiente tabla se presentan todos los requisitos funcionales que juegan un papel en el proyecto, los cuales describen funcionalidades, comportamientos y características necesarias que el sistema debe ejercer para cumplir con los objetivos del usuario.

ID	Requisito	Descripción
RF01	Crear horario	El sistema debe permitir la creación de horarios por parte del administrador universitario.
RF02	Modificar horario	El sistema debe permitir la modificación de horarios por parte del administrador universitario.
RF03	Publicar horario	El sistema debe permitir la publicación del horario para ser mostrado en los canales de comunicación habituales.
RF04	Visualizar horario	El sistema debe permitir la visualización del horario al usuario.
RF05	Filtrado por grado de estudios	El sistema debe permitir aplicar un filtro para visualizar horarios en función del grado de estudios escogido.
RF06	Filtrado por curso	El sistema debe permitir aplicar un filtro para visualizar horarios en función del curso seleccionado.
RF07	Filtrado por grupo	El sistema debe permitir aplicar un filtro para visualizar horarios en función del grupo seleccionado.
RF08	Filtrado por mención	El sistema debe permitir aplicar un filtro para visualizar horarios en función de la mención escogida.
RF09	Filtrado por asignatura	El sistema debe permitir aplicar un filtro para visualizar horarios en función de las asignaturas seleccionadas.
RF10	Roles de usuario	El sistema debe permitir diferenciar a los usuarios en estudiantes y administradores, garantizando el acceso solo a funcionalidades permitidas.
RF11	Exportación de Horarios	El sistema debe permitir exportar horarios en formatos PDF o imagen.
RF12	Cargado de fichero	El sistema debe permitir cargar el fichero de horarios escogido por el administrador.
RF13	Escritura de fichero	El sistema debe permitir escribir los cambios realizados en un fichero preexistente o la creación de un fichero nuevo con el nombre deseado

Tabla 18: Requisitos funcionales

3.1.2. Requisitos de información

En la siguiente tabla se presentan todos los requisitos de información del proyecto, los cuales determinan los datos que el sistema debe ser capaz de gestionar, procesar y almacenar para cumplir con los objetivos y funcionamiento establecidos.

ID	Requisito	Descripción
RI01	Información de horarios	El sistema deberá almacenar y gestionar datos sobre los horarios, incluyendo asignaturas, grupos, aulas, profesores y franjas horarias.
RI02	Información de usuarios	El sistema deberá gestionar datos de usuario, diferenciando entre usuarios comunes y administrador.
RI03	Información de asignaturas	El sistema deberá almacenar datos acerca de las asignaturas, incluyendo nombre, código, curso, cuatrimestre y profesor asociado, entre otros datos.
RI04	Información de grupos, cursos y aulas	El sistema deberá almacenar la información relacionada con los grupos, cursos y aulas disponibles.
RI05	Información de reglas y restricciones	El sistema deberá almacenar las reglas de planificación, así como restricciones que eviten solapamientos en horarios.

Tabla 19: Requisitos de información

3.1.3. Requisitos no funcionales

En la siguiente tabla se presentan los requisitos no funcionales, referentes a criterios de calidad y restricciones del sistema, como su rendimiento, seguridad, usabilidad y compatibilidad.

ID	Requisito	Descripción
RNF01	Rendimiento	El sistema deberá ser capaz de procesar y mostrar horarios en un intervalo de tiempo menor a 2 segundos, incluso bajo alta demanda.
RNF02	Integración	El sistema deberá poder integrarse dentro de la plataforma universitaria correspondiente.
RNF03	Interfaz intuitiva	El sistema debe ofrecer una interfaz de usuario sencilla e intuitiva para facilitar la consulta y gestión de horarios.
RNF04	Accesibilidad	El sistema deberá emplear recursos que fomenten el uso universal del mismo, independientemente de las circunstancias de cada uno de los usuarios y del navegador web utilizado.
RNF05	Front - end	El sistema deberá basa sus servicios front – end en una aplicación web.
RNF06	Seguridad	El sistema deberá proporcionar mecanismos adecuados en términos de encriptación de contraseñas de usuario.
RNF07	Disponibilidad	El sistema deberá estar disponible al menos el 98% del tiempo durante el horario universitario.
RNF08	Confiabilidad	El sistema deberá poder recuperarse de manera automática en caso de caída o error crítico.

Tabla 20: Requisitos no funcionales

3.1.4. Reglas de negocio

En la siguiente tabla se presentan las reglas de negocio, que establecen normativas que determinan el funcionamiento del sistema.

ID	Requisito	Descripción
RN01	Restricción de modificaciones	Los horarios publicados solo podrán ser modificados en situaciones permitidas en el reglamento universitario.
RN02	Disponibilidad de Horarios	Los horarios deberán estar disponibles en el sistema al menos una semana antes del inicio del periodo académico.
RN03	Autenticación	La autenticación del administrador deberá ser obligatoria si se desea acceder a las funcionalidades institucionales que el sistema proporciona.
RN04	Gestión de horarios	La creación, modificación y actualización de horarios estará destinada exclusivamente al usuario con credenciales relacionadas con el rol de administrador.
RN05	Publicación de horarios	Un horario solo podrá ser publicado por el administrador en caso de contar con el consenso positivo del profesorado involucrado en dicho documento.
RN06	Registro de modificaciones	Cada una de las modificaciones del horario deben ser reflejadas en el sistema.
RN07	Guardado de Versiones	El sistema debe permitir mantener un historial de versiones de los archivos de horarios, permitiendo la recuperación o consulta de versiones anteriores si fuera necesario.

Tabla 21: Reglas de negocio

3.2. Casos de uso

3.2.1. Actores principales

Durante el análisis del sistema y los contextos de su uso, ha sido posible determinar que existen dos actores principales que interactúan con el sistema, siendo el Alumno y el Administrador, los cuales tienen permisos para emplear ciertas funcionalidades que son comunes a ambos y que son englobadas en un actor Usuario.

3.2.2. Diagrama de casos de uso

A continuación, se muestra el diagrama de Casos de Uso. Se trata de una representación de las principales funcionalidades englobadas en el sistema, de las cuales puede hacer uso el usuario en cada una de las interacciones (Astah, s.f).

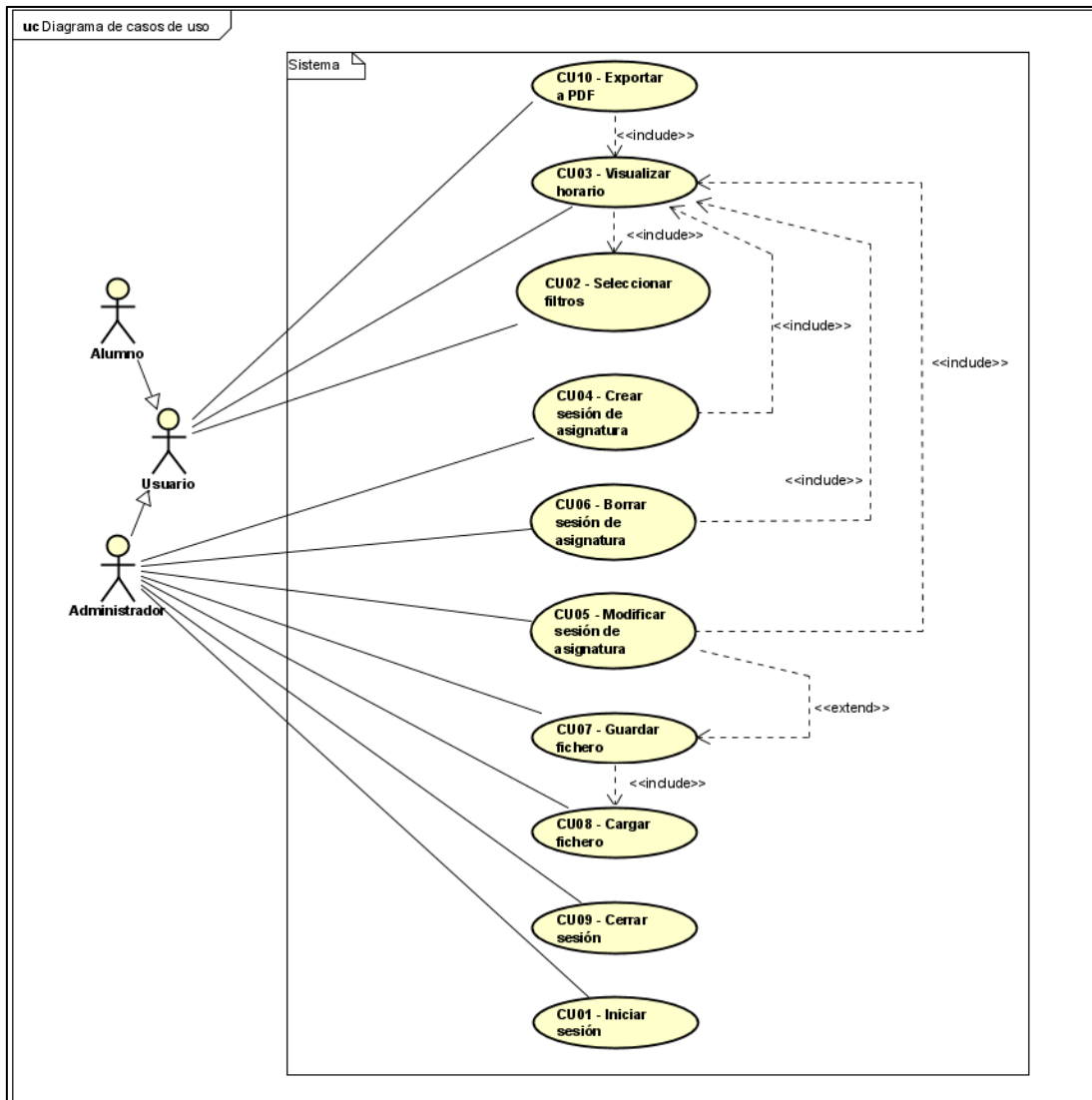


Figura 1: Diagrama de Casos de Uso - Sistema

3.2.3. Descripción de los casos de uso

En esta sección se presenta la secuencia de eventos que tiene lugar en el desarrollo de los casos de uso introducidos previamente, relacionados con la interacción entre el usuario y el sistema con el objetivo de hacer uso de sus funcionalidades. También se muestran situaciones o flujos de eventos alternativos durante la ejecución de cada caso de uso.

ITEM	VALUE
UseCase	CU01 - Iniciar sesión
Summary	Administrador inicia sesión en el sistema.
Actor	Usuario, Administrador
Precondition	Administrador se encuentra previamente registrado en el sistema.
Postcondition	Administrador inicia sus sesión en el sistema.
Base Sequence	1. Administrador selecciona el botón "Iniciar sesión". 2. Sistema muestra al usuario la página de inicio de sesión y solicita sus credenciales. 3. Administrador rellena los campos con sus credenciales de inicio de sesión. 4. Sistema realiza una validación de las credenciales introducidas. 5. Sistema autoriza el acceso al Administrador.
Branch Sequence	4a. Sistema determina que las credenciales introducidas son inválidas, informa al Administrador y solicita la reintroducción de sus datos. El caso de uso vuelve al paso 3.
Exception Sequence	2a, 3a - Administrador cancela la operación, el CU queda sin efecto.
Sub UseCase	
Note	

Figura 2: Flujo del Caso de Uso - Iniciar sesión

ITEM	VALUE
UseCase	CU02 - Seleccionar filtros
Summary	Usuario selecciona filtros en el sistema para visualizar horarios.
Actor	Usuario
Precondition	Usuario se encuentra dentro de una de las secciones de visualización de horarios.
Postcondition	Sistema almacena las opciones de filtrado.
Base Sequence	1. Usuario selecciona opciones de filtrado referentes al plan de estudios. 2. Sistema almacena las opciones seleccionadas.
Branch Sequence	
Exception Sequence	1a - Usuario cancela la operación, el CU queda sin efecto.
Sub UseCase	
Note	

Figura 3: Flujo del Caso de Uso - Seleccionar filtros

ITEM	VALUE
UseCase	CU03 - Visualizar horario
Summary	Usuario consulta horarios en base a opciones de filtrado.
Actor	Usuario
Precondition	Usuario tiene su sesión iniciada en el sistema.
Postcondition	Sistema muestra el horario de acuerdo a los filtros almacenados.
Base Sequence	1. Se ejecuta el CU02 - Seleccionar filtros. 2. Sistema muestra el horario en cuestión.
Branch Sequence	
Exception Sequence	1a - Usuario cancela la operación, el CU queda sin efecto.
Sub UseCase	CU02 - Seleccionar filtros
Note	

Figura 4: Flujo del Caso de Uso - Visualizar horario

ITEM	VALUE
UseCase	CU04 - Crear sesión de asignatura
Summary	Administrador crea la/s asignatura/s que desee.
Actor	Administrador
Precondition	Administrador tiene su sesión iniciada en el sistema.
Postcondition	Sistema actualiza el horario con las asignaturas creadas.
Base Sequence	1. Se ejecuta el CU03 - Visualizar horario. 2. Administrador selecciona la sección "Crear asignaturas". 3. Sistema muestra el formulario de creación de asignaturas. 4. Administrador rellena el formulario y pulsa el botón "crear asignaturas". 5. Sistema almacena el horario actualizado.
Branch Sequence	4a. Administrador pulsa el botón "limpiar formulario". El CU vuelve al paso 3. 5a. Sistema determina una incompatibilidad en la franja horaria donde se trata de situar la asignatura e informa al profesor.
Exception Sequence	2a, 3a, 4b - Administrador cancela la operación, el CU queda sin efecto.
Sub UseCase	CU03 - Visualizar horario
Note	

Figura 5: Flujo del Caso de Uso - Crear sesión de asignatura

ITEM	VALUE
UseCase	CU05 - Modificar sesión de asignatura
Summary	Administrador modifica una sesión de asignatura preexistente.
Actor	Administrador
Precondition	Administrador tiene su sesión iniciada en el sistema.
Postcondition	Sistema almacena el horario con las modificaciones realizadas.
Base Sequence	1. Se ejecuta el CU03 - Visualizar horario. 2. Sistema muestra un menú con distintas secciones. 3. Administrador hace click en sesión de asignatura a modificar. 4. Sistema muestra la asignatura y formulario con su información. 5. Administrador modifica el formulario y pulsa el botón "modificar asignatura". 6. Sistema modifica la asignatura y almacena los cambios realizados.
Branch Sequence	3a. Administrador modifica la sesión de asignatura empleando Drag & Drop, desplazandola a otra sección del horario. 3b. Administrador modifica la sesión de asignatura empleando Resize, aumentando o disminuyendo su duración. 6a. Sistema determina una incompatibilidad tras modificar la sesión de asignatura, informando por pantalla.
Exception Sequence	2a, 3c, 4a, 5a, 6b - Administrador cancela la operación, el CU queda sin efecto.
Sub UseCase	CU03 - Visualizar horario

Figura 6: Flujo del Caso de Uso - Modificar sesión de asignatura

ITEM	VALUE
UseCase	CU06 - Borrar sesión de asignatura
Summary	Administrador borra una sesión de asignatura del horario.
Actor	Administrador
Precondition	Administrador tiene su sesión iniciada en el sistema.
Postcondition	Sistema borra la sesión de asignatura y actualiza el horario con los cambios.
Base Sequence	1. Se ejecuta el CU03 - Visualizar horario. 2. Sistema muestra un menú con distintas secciones. 3. Administrador hace click en sesión de asignatura a borrar. 4. Sistema muestra la sesión de asignatura seleccionada. 5. Administrador pulsa el botón "borrar asignatura". 6. Sistema borra la sesión de asignatura y actualiza el horario con los cambios.
Branch Sequence	
Exception Sequence	2a, 4a, 5a - Administrador cancela la operación, el CU queda sin efecto.
Sub UseCase	CU03 - Visualizar horario
Note	

Figura 7: Flujo del Caso de Uso - Borrar sesión de asignatura

ITEM	VALUE
UseCase	CU07 - Guardar fichero
Summary	Administrador guarda una versión del fichero de horarios con cambios realizados.
Actor	Administrador
Precondition	Administrador tiene su sesión iniciada en el sistema.
Postcondition	Sistema guarda una copia del horario con las modificaciones realizadas.
Base Sequence	1. Administrador pulsa el botón "Guardar horario". 2. Sistema solicita el nombre del horario a guardar. 3. Subdirector proporciona el nombre del documento. 4. Sistema guarda el fichero.
Branch Sequence	3a. Administrador selecciona un horario a sobrescribir. CU continúa en el paso 4.
Exception Sequence	2a, 3b - Administrador, el CU queda sin efecto.
Sub UseCase	CU05 - Modificar sesión de asignatura CU08 - Cargar fichero
Note	

Figura 8: Flujo del Caso de Uso - Guardar fichero

ITEM	VALUE
UseCase	CU08 - Cargar fichero
Summary	Administrador selecciona el fichero de asignaturas a cargar en el sistema.
Actor	Administrador
Precondition	Administrador tiene su sesión iniciada en el sistema.
Postcondition	Sistema carga el fichero seleccionado y realiza su lectura.
Base Sequence	1. Administrador pulsa el botón "cargar fichero". 2. Sistema solicita el fichero a cargar. 3. Administrador selecciona el fichero. 4. Sistema carga el fichero y realiza su lectura.
Branch Sequence	
Exception Sequence	1a, 2a, 3a - Administrador cancela la operación, el CU queda sin efecto.
Sub UseCase	
Note	

Figura 9: Flujo del Caso de Uso - Cargar fichero

ITEM	VALUE
UseCase	CU09 - Cerrar sesión
Summary	Administrador cierra su sesión en el sistema.
Actor	Administrador
Precondition	Administrador tiene su sesión iniciada en el sistema.
Postcondition	Sistema cierra la sesión.
Base Sequence	1. Administrador pulsa el botón "cerrar sesión". 2. Sistema cierra la sesión.
Branch Sequence	
Exception Sequence	1a - Administrador cancela la operación, el CU queda sin efecto.
Sub UseCase	
Note	

Figura 10: Flujo del Caso de Uso - Cerrar sesión

ITEM	VALUE
UseCase	CU10 - Exportar a PDF
Summary	Usuario exporta un horario a formato PDF.
Actor	Usuario
Precondition	
Postcondition	Usuario recibe la descarga del archivo correspondiente.
Base Sequence	1. Se ejecuta el CU03 - Visualizar horario. 2. Usuario selecciona el botón "exportar a PDF". 3. Sistema genera una descarga con el horario escogido.
Branch Sequence	
Exception Sequence	2a - Usuario cancela la operación, el CU queda sin efecto.
Sub UseCase	CU03 - Visualizar horario
Note	

Figura 11: Flujo del Caso de Uso - Exportar a PDF

3.3. Modelo de dominio

A continuación, se muestra el modelo de dominio, donde se reflejan las relaciones entre las principales entidades del sistema. Se trata de una representación de las principales entidades del sistema, junto con sus atributos y relaciones (*Astah, s.f*).

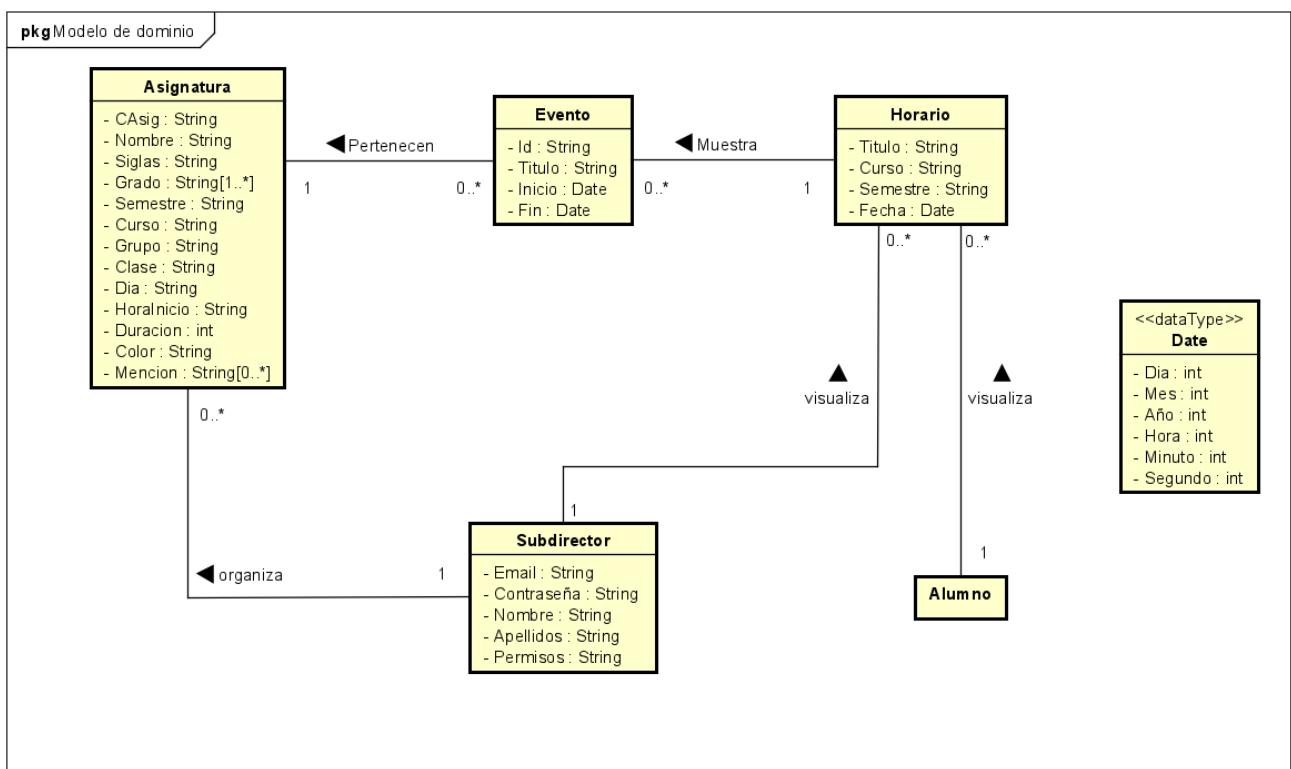


Figura 12: Modelo de dominio - Sistema

A su vez, se describen las entidades representadas en este diagrama, así como cada uno de sus atributos.

3.3.1. Subdirector

Esta clase modela la entidad que se relaciona con la aplicación web de cara a emplear su funcionalidad, como iniciar sesión, visualizar, crear, modificar y publicar horarios.

- Atributos:
 - Email: dirección de correo electrónico asociada al perfil de usuario.
 - Contraseña: clave asociada al perfil de usuario.
 - Nombre: nombre del profesor en particular.
 - Apellidos: apellidos del profesor en particular.
 - Permisos: acciones que dicho usuario puede realizar.

3.3.2. Alumno

Esta clase modela la entidad que corresponde al uso de la aplicación web que realiza el alumnado.

3.3.3. Asignatura

Esta clase modela la entidad que corresponde a cada una de las asignaturas almacenadas en el sistema, las cuales aparecen reflejadas en forma de eventos en cada uno de los horarios.

- Atributos:
 - CAsig: código de la asignatura interno dentro del reglamento universitario.
 - Nombre: nombre completo de la asignatura.
 - Siglas: abreviatura de la asignatura en el sistema.
 - Grado: grado/s universitario/s de la asignatura.
 - Semestre: semestre o cuatrimestre en el que se imparte la asignatura.
 - Curso: número de curso en el que se imparte dicha asignatura dentro del grado de estudios.
 - Clase: aulas físicas en las que se imparte la asignatura.
 - Día: días de la semana en los que imparte la asignatura.
 - HoraInicio: hora de comienzo de la sesión lectiva de la asignatura.
 - Duracion: duración de la sesión lectiva de la asignatura.
 - Color: color de la asignatura dentro del horario lectivo.
 - Mención: menciones a las que pertenece dicha asignatura dentro del grado de estudios.

3.3.4. Evento

Esta clase modela la entidad que corresponde a la visualización de las asignaturas en cada horario, la cual se realiza a través de “eventos” que ocurren en cada uno de ellos.

- Atributos:
 - Id: identificador único para cada evento dentro del sistema.
 - Titulo: título del evento, compuesto por información de la asignatura a la que pertenece.
 - Inicio: fecha de inicio del evento.
 - Fin: fecha de finalización del evento.

3.3.5. Horario

Esta clase modela la entidad correspondiente a cada uno de los horarios almacenados en el sistema y de los cuales es posible hacer uso empleando la aplicación web.

- Atributos:
 - Titulo: parte de la cabecera del horario en la que queda reflejado el grado de estudios, el periodo semestral o cuatrimestral y el año actual.
 - Curso: parte de la cabecera del horario en la que queda reflejado el curso al que pertenece.
 - Semestre: semestre o cuatrimestre al cual pertenece el horario.
 - Fecha: última fecha de modificación del horario.

3.3.6. Date

Esta clase modela la entidad correspondiente al tipo de datos “Date”, presente en cualquier fecha que sea almacenada en el sistema.

- Atributos:
 - Dia: parte de la fecha correspondiente a los días.
 - Mes: parte de la fecha correspondiente al mes.
 - Año: parte de la fecha correspondiente al año.
 - Hora: parte de la fecha correspondiente a las horas.
 - Minuto: parte de la fecha correspondiente a los minutos.
 - Segundo: parte de la fecha correspondiente a los segundos.

3.4. Diagramas de secuencia

A continuación, se muestran diagramas de secuencia, relacionados con las funcionalidades descritas previamente en cada caso de uso. Se trata de una representación de la interacción del usuario con el sistema, paso por paso (Astah, s.f).

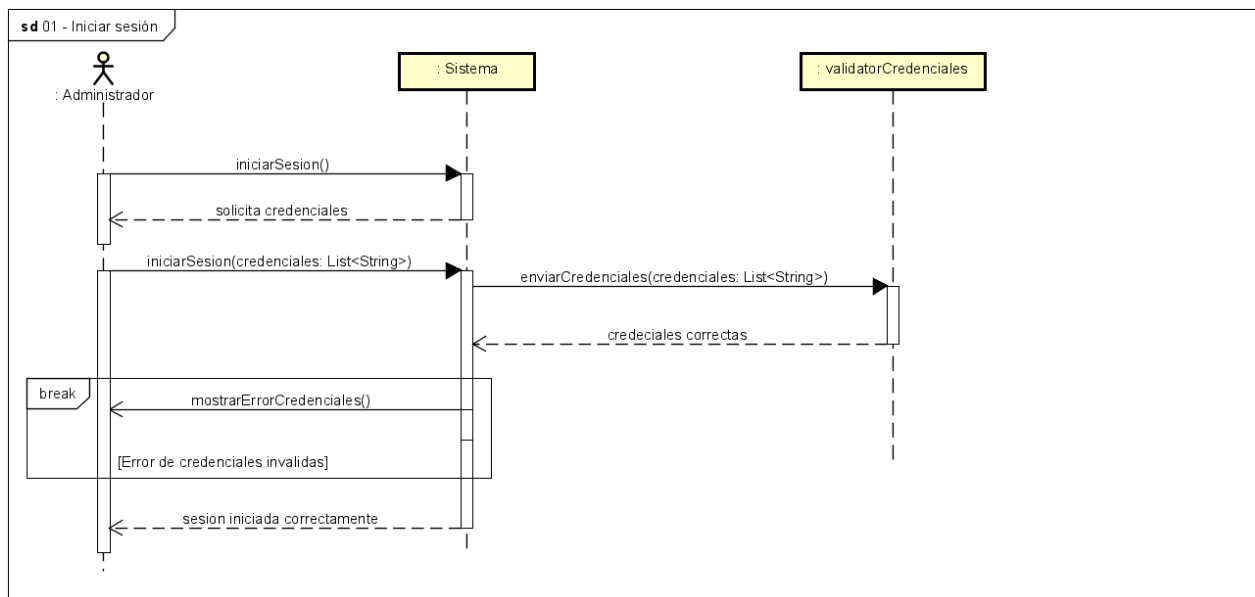


Figura 13: Diagrama de secuencia - Iniciar sesión

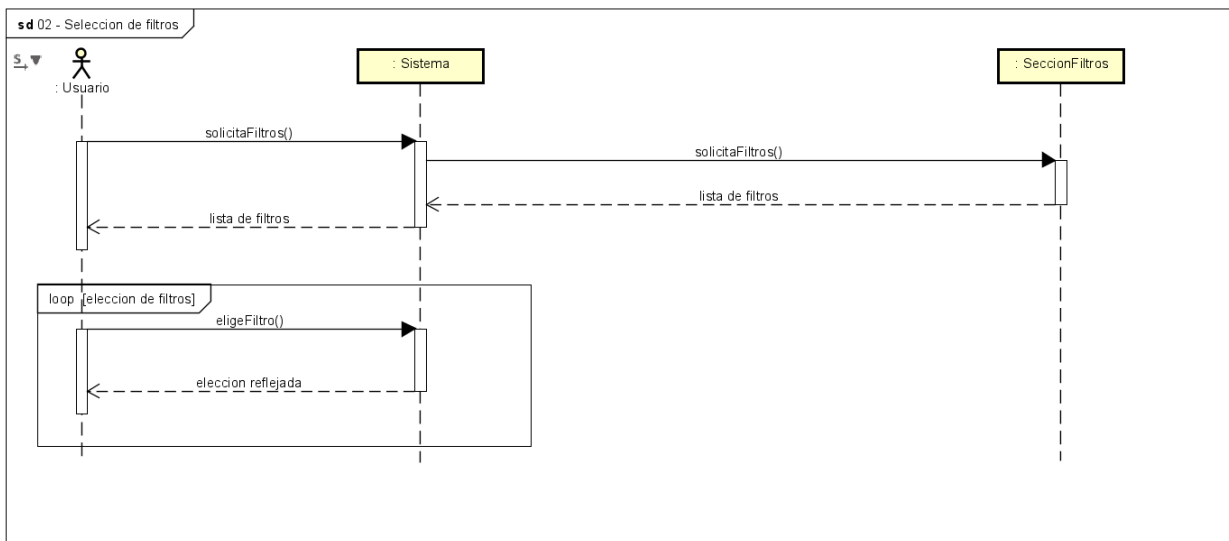


Figura 14: Diagrama de secuencia - Seleccionar filtros

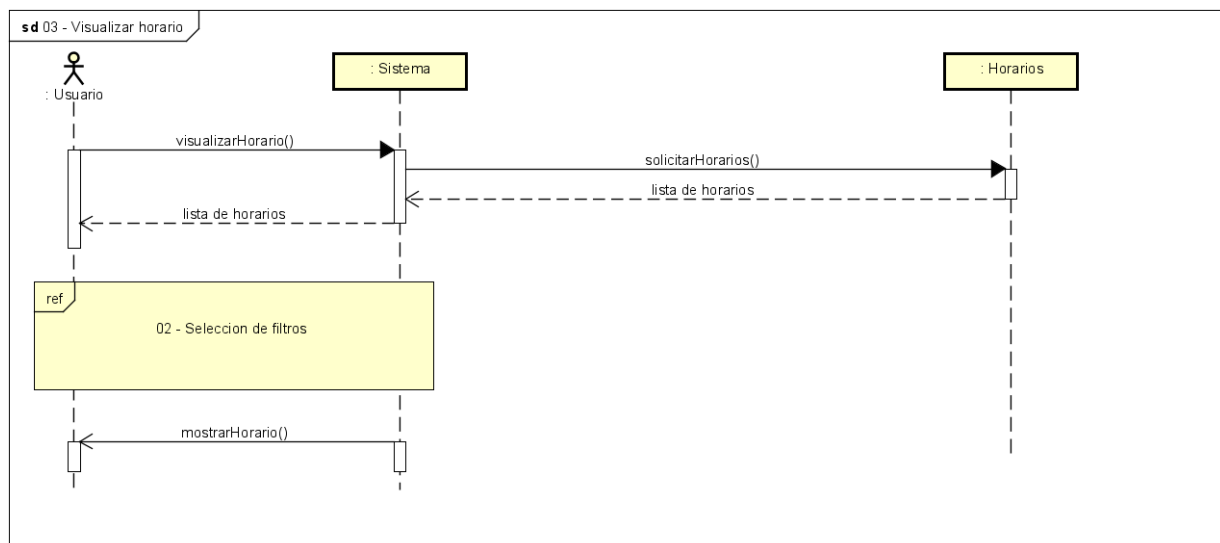


Figura 15: Diagrama de secuencia - Visualizar horario

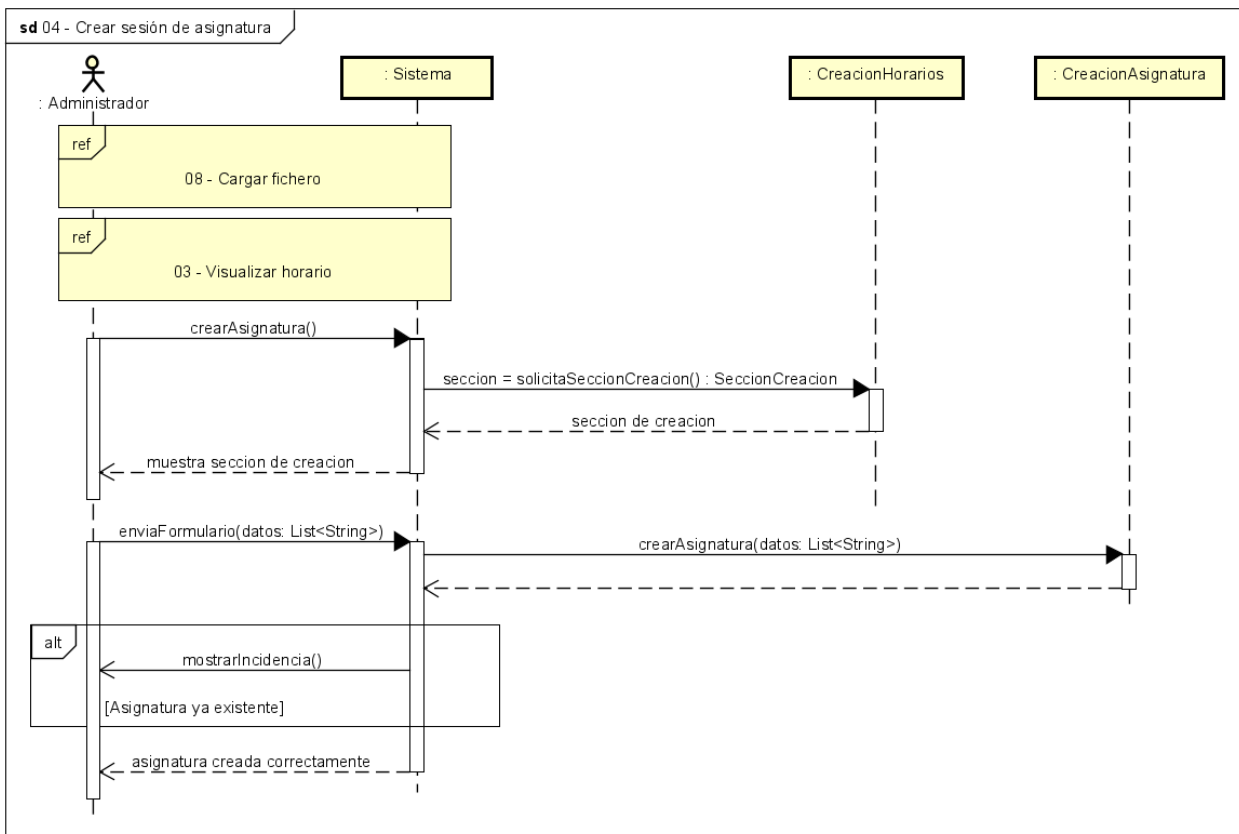


Figura 16: Diagrama de secuencia - Crear sesión de asignatura

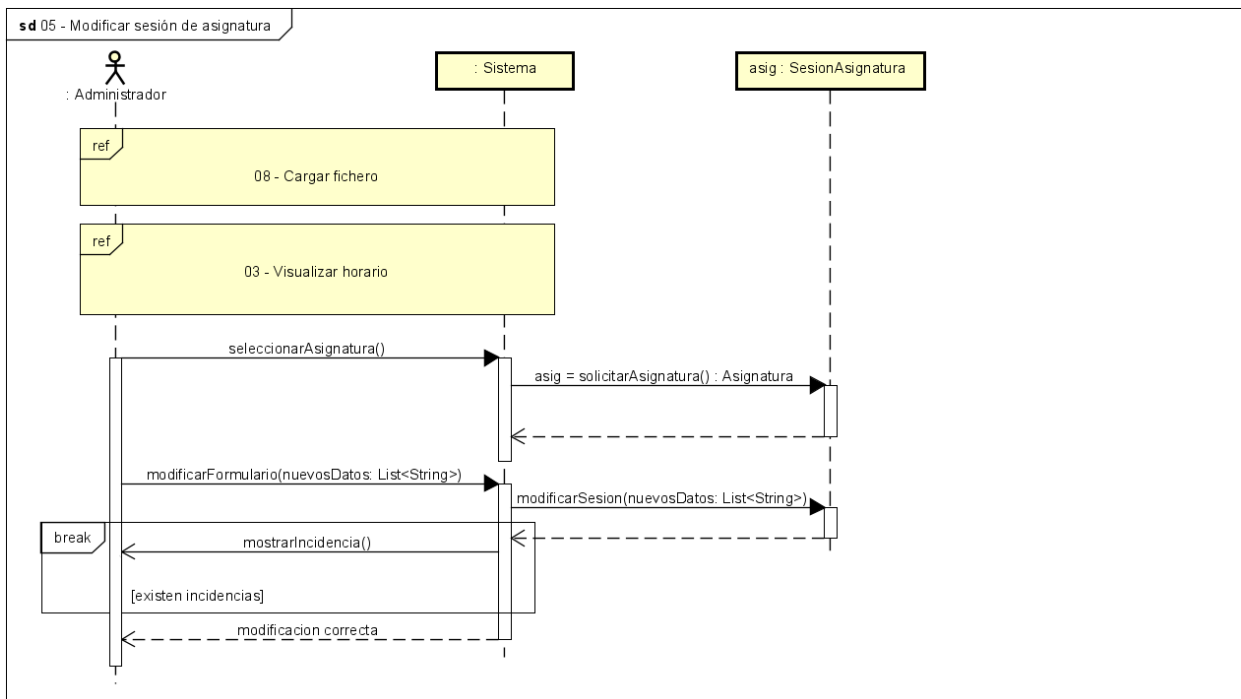


Figura 17: Diagrama de secuencia - Modificar sesión de asignatura

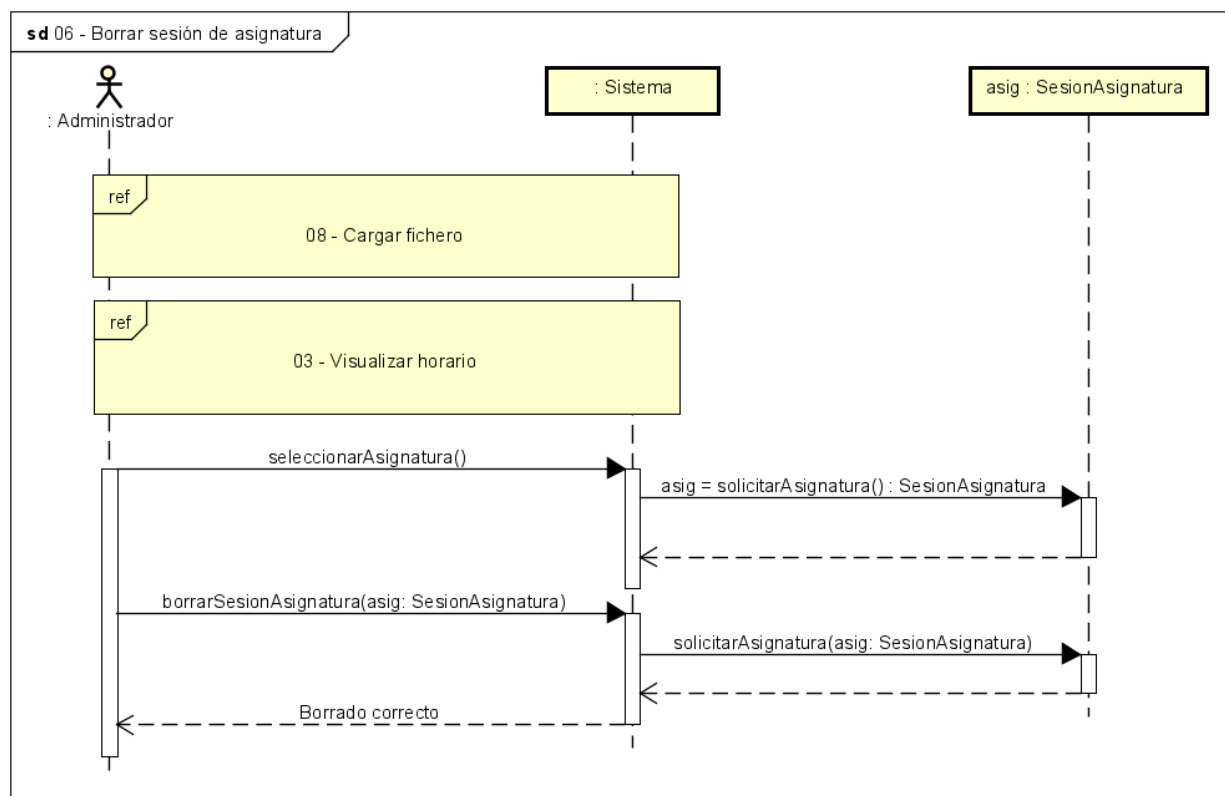


Figura 18: Diagrama de secuencia - Borrar sesión de asignatura

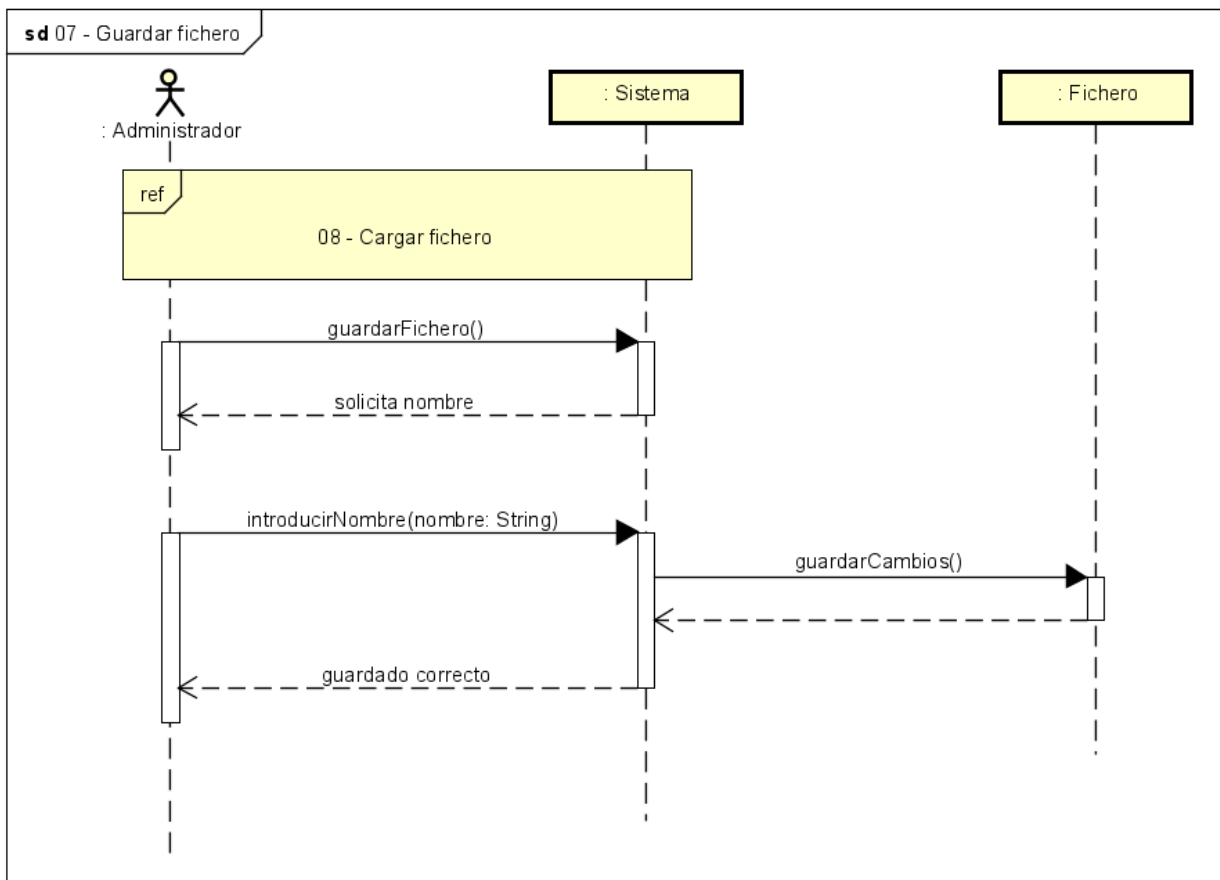


Figura 19: Diagrama de secuencia - Guardar fichero

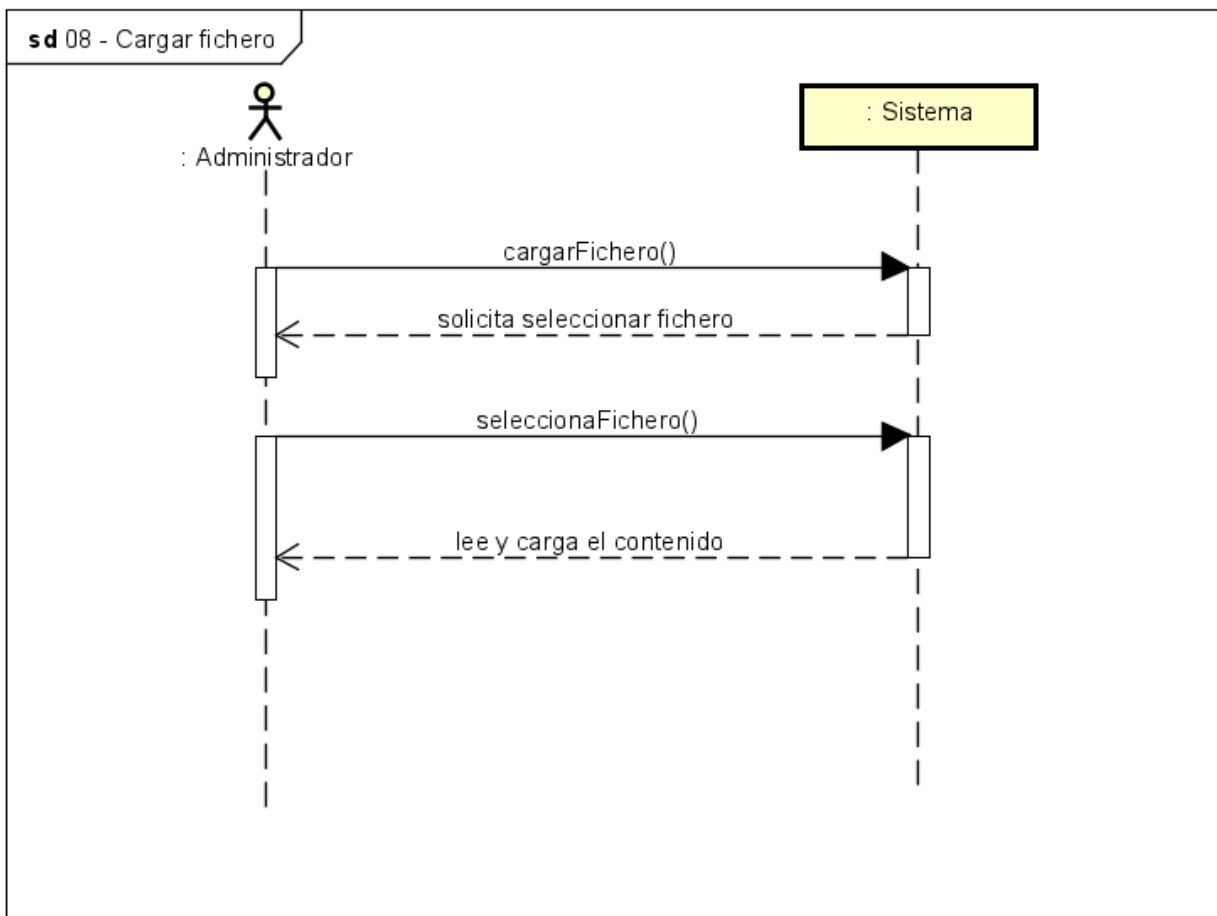


Figura 20: Diagrama de secuencia - Cargar fichero

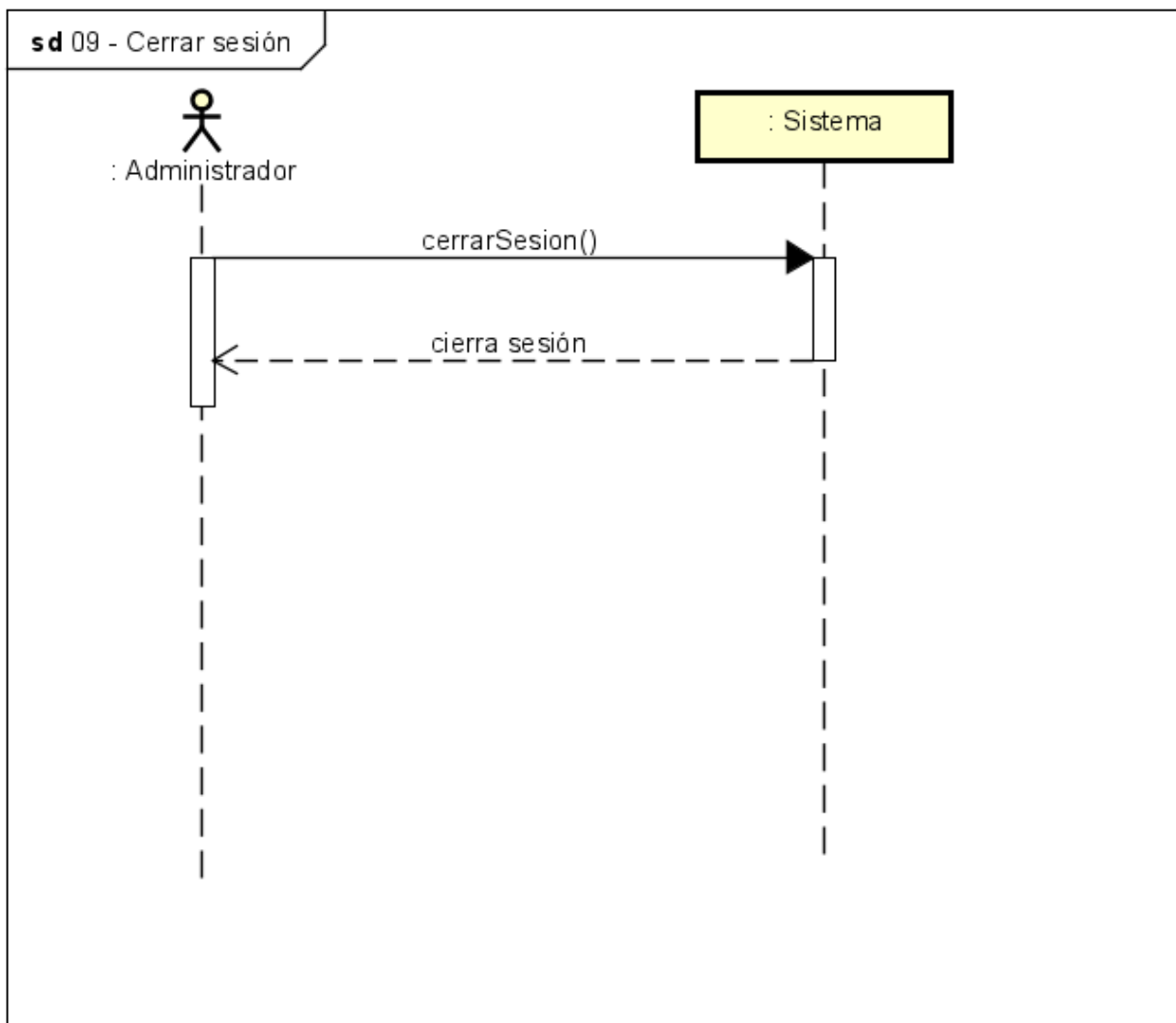


Figura 21: Diagrama de secuencia - Cerrar sesión

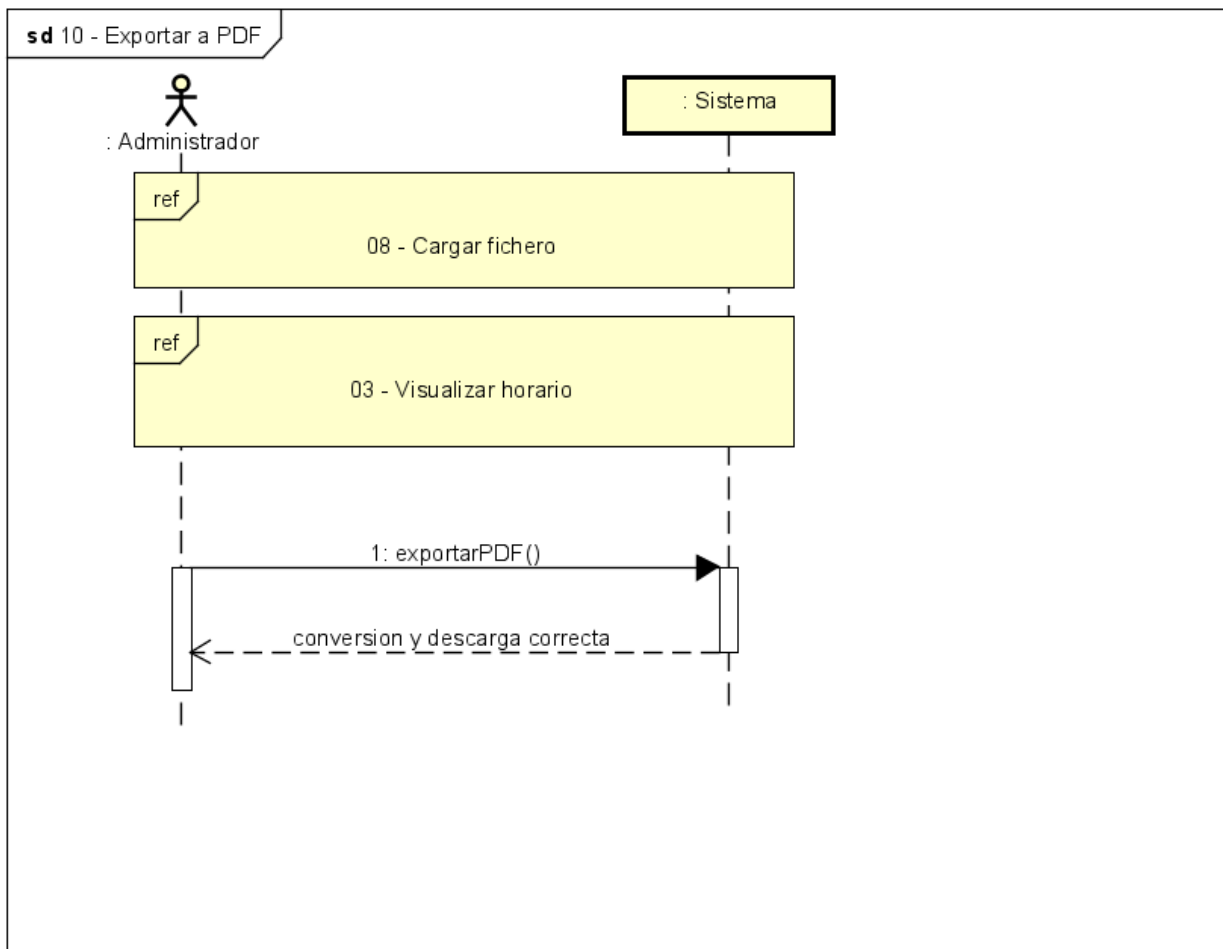


Figura 22: Diagrama de secuencia - Exportar a PDF

Capítulo 4

Diseño

4. Patrón de diseño

Tras quedar establecido el previo análisis, es necesario especificar las decisiones de diseño que han sido tomadas a lo largo del proceso, arquitectura empleada y organización interna del sistema.

4.1. Modelo – Vista – Controlador

El patrón empleado supone un Modelo – Vista – Controlador, siendo un patrón de diseño arquitectónico habitualmente empleado en diversos ámbitos, entre ellos el desarrollo de aplicaciones. Entre los numerosos aspectos que proporciona este patrón, se encuentra la clara separación de la lógica de negocio y la interfaz de usuario (MDN Web Docs, s.f).

De acuerdo con las especificaciones, se divide en tres capas principales:

- Modelo, capa que define los datos que debe contener la aplicación y que representa la estructura y lógica de negocio relacionada con el acceso y manipulación de dichos datos.
- Vista, capa de presentación con la que el usuario interactúa de cara a realizar acciones y observar sus resultados.

- Controlador, capa intermedia que permite la actualización de la vista en relación con los cambios del modelo. Permite la incorporación de lógica adicional, relacionada con la validación y actualización de los datos.

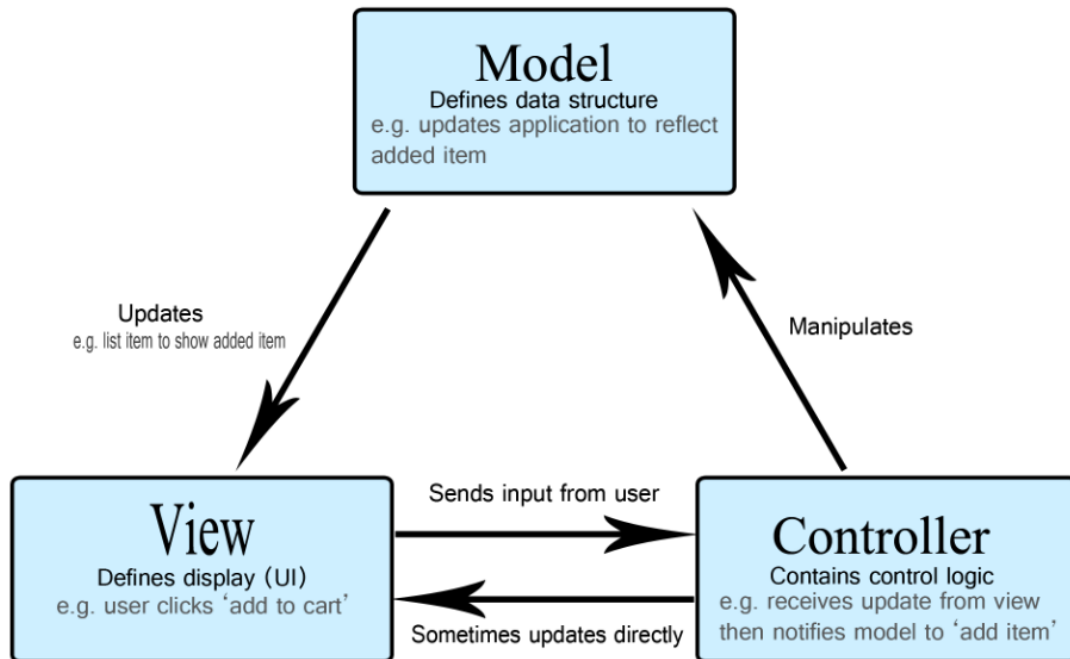


Figura 23: Diagrama MVC

Es posible observar como las tres capas se comunican entre sí de manera unidireccional, respetando la separación de intereses establecida anteriormente. Esta partición y separación de funciones y responsabilidades facilita la realización de cambios en la lógica de negocio e interfaz de usuario sin tener la necesidad alterar el resto de la aplicación.

4.1.1. Modelo – Vista – Controlador en el proyecto

En el contexto de la siguiente aplicación web, este patrón resulta muy útil, puesto que permite la separación de la interfaz de usuario y la lógica de negocio, facilitando la integración y entrelazado de ambas partes. De la siguiente manera, es posible diferenciar y asociar las capas vistas previamente, en el contexto de la aplicación:

- Vista, React juega un papel fundamental en relación con la ejecución de la vista.
- Modelo, incluye los datos almacenados en relación con los horarios que son manejados dentro del sistema.
- Controlador, el back-end en JavaScript interactúa con el modelo de datos y la interfaz de usuario de cara al acceso, manipulación y envío de actualizaciones a la vista.

4.2. Diseño web

4.2.1. Diseño separado en páginas

El planteamiento y estructuración del diseño web se basa en un enfoque basado en la separación en páginas. Dentro de este enfoque, se desarrollan componentes que corresponden a cada una de las páginas que componen la aplicación, siendo diseñados con la intención de incluir todos los aspectos necesarios para la correcta interacción del usuario con cada una de las partes de la app. A su vez, ciertos componentes incluyen otros elementos secundarios, componentes de menor tamaño, los cuales son compartidos entre distintas secciones.

Este enfoque de diseño web proporciona un conjunto de ventajas:

- Separación de código, facilitando su mantenimiento y la realización de cambios en el mismo a medida que nuevas funcionalidades son incorporadas o los requisitos son modificados.

- Reutilización de código, puesto que la “componentización” de la app supone el reutilizado de componentes y código en diversas secciones a lo largo de su estructura.
- Simplificación de las pruebas unitarias, siendo repartida su creación entre los distintos componentes y verificando su correcto funcionamiento de forma independiente al resto del sistema.
- Eficiencia, puesto que, de acuerdo con el funcionamiento de React y su sistema de Virtual DOM (V-DOM), es posible actualizar solo los componentes que sufren cambios de estado, en lugar de volver a construir todo el árbol de componentes, suponiendo un gran salto en la eficiencia y rendimiento del sistema de app web.

4.2.2. Componentes del proyecto

A continuación, se muestra y detalla el conjunto de componente cuya implementación ha sido desarrollada:

- **App:** componente padre que enlaza con el resto de los componentes, incluyendo el enrutamiento entre ellos.
- **AuthProvider:** componente desarrollado, relacionado con la creación del contexto y comprobación de autenticación del subdirector.
- **Routes:** componente importado, relacionado con la creación de rutas dentro del sistema, actuando como elemento padre del componente “Route” y sirviendo de wrapper.
- **Route:** componente importado, relacionado con una ruta específica dedicada a un sección, dentro del contexto del sistema.
- **PrivateRoute:** componente desarrollado, destinado a redirigir al subdirector a la sección de inicio de sesión dedicada a la administración del sistema.

- **IntroSection:** componente desarrollado que muestra una sección introductoria a la app web, dedicado a su uso por el estudiantado para la elección entre ambos tipos de consulta de horarios. Cuenta con los siguientes componentes hijos:
 - **ImageContainer:** componente desarrollado cuya funcionalidad consiste en mostrar una imagen representando el logo de la facultad.
 - **TypeVisualIntroSection:** componente desarrollado y reutilizado para mostrar cada una de las dos opciones de visualización de horarios ofrecidas al usuario.
 - **IntroButton:** componente desarrollado que actúa de botón para enlazar al usuario con su opción de visualización de horarios elegida.
- **Login:** componente que muestra un formulario de inicio de sesión, enfocado al acceso a la gestión de horarios por parte del subdirector de organización docente. Cuenta con los siguientes componentes hijos:
 - **ImageContainer.**
 - **LoginInput:** componente desarrollado y dedicado a la introducción de la contraseña por parte del subdirector.
- **GenericVisualization:** componente desarrollado que muestra la sección dedicada a la visualización de horarios en base a unos criterios fijos, como el grado, semestre o curso, entre otros.

Cuenta con los siguientes componentes hijos:

- **FiltersSection:** componente desarrollado que muestra la sección de filtros dedicada a la selección de los criterios de cara a visualizar un horario. Cuenta, a su vez, con los siguientes componentes hijos:
 - **Draggable:** componente importado orientado a convertir a la sección de filtros en un menú desplazable.
 - **IconButton:** componente desarrollado y reutilizado, consistente en un botón con icono que al ser pulsado desencadena funciones como exportar a pdf un horario escogido.

- **FiltersButton:** componente desarrollado y reutilizado, dedicado al mostrado y selección de opciones en diversos apartados del menú de filtros.
- **IncludeLabsCheckbox:** componente desarrollado y reutilizado, dedicado a mostrar o no las sesiones de laboratorio en el horario.
- **Calendar:** componente importado procedente de la librería “React-Big-Calendar”, dedicado al funcionamiento y mostrado de horarios.
- **CalendarEvent:** componente desarrollado, encargado de actuar como un evento del calendario, adaptado a las necesidades propias del sistema, específicamente a la sección de alumno.
- **CustomVisualization:** componente desarrollado que muestra la sección dedicada a la visualización de horarios personalizada, de manera que el usuario puede crear sus propios horarios.

Cuenta con los siguientes componentes hijos:

- **FiltersSectionCustom:** componente desarrollado que muestra la sección de filtros dedicada a la selección de los criterios de cara a crear un horario personalizado.

Cuenta, a su vez, con los siguientes componentes hijos:

- **Draggable.**
- **IconButton.**
- **FiltersButton.**
- **FiltersSelect:** componente desarrollado que actúa de desplegable con un conjunto de opciones personalizado, en base a los criterios seleccionados en la sección de filtros.
- **IncludeLabsCheckbox.**

- **Calendar.**
- **CalendarEvent.**
- **PrivateRoute:** componente desarrollado que verifica si el administrador cuenta con sus sesión iniciada dentro del contexto.
- **ScheduleCreation:** componente desarrollado que muestra la sección dedicada a la gestión de horarios por parte del subdirector, permitiendo modificar horarios existentes, a través de la modificación, creación y borrado de asignaturas, entre otros aspectos.

Cuenta con los siguientes componentes hijos:

- **LogoutButton:** componente desarrollado destinado a la funcionalidad que permite al subdirector cerrar su sesión en el sistema.
- **FloatingFilterScheduleMenu:** componente desarrollado destinado a la funcionalidad de menú de filtros, con la intención de mostrar un horario que modificar.

Cuenta con los siguientes componentes hijos:

- **ScheduleCreationMenuTabs:** componente importado y posteriormente modificado, enfocado a la funcionalidad de dividir el menú de filtrado en dos pestañas, modificación de horarios y creación de asignaturas.
Cuenta con `ScheduleCreationMenuCalendarTab` y `ScheduleCreationMenuAsigTab` como componentes hijos.
- **ScheduleCreationMenuCalendarTab:** componente desarrollado y enfocado a la funcionalidad de un menú de filtros enfocado a mostrar un horario a modificar en base a criterios seleccionados por el subdirector.

Cuenta con los siguientes componentes hijos:

- **FiltersButton.**
- **ScheduleCreationMenuWarningsField:** componente importado y posteriormente modificado, destinado a la funcionalidad de informar al subdirector de incidencias producidas por incompatibilidades causadas por modificaciones de un horario.
- **ScheduleCreationMenuAsigTab:** componente desarrollado y enfocado a la funcionalidad de formulario, permitiendo proporcionar los datos necesarios para añadir una nueva asignatura dentro del sistema.

Cuenta con los siguientes componentes hijos:

- **ScheduleCreationInput:** componente importado y posteriormente modificado, destinado a servir como un campo de texto a completar con información de la asignatura.
- **ScheduleCreationSelect:** componente importado y posteriormente modificado, destinado a servir como desplegable de opciones a elegir en cuanto a información de la asignatura.
- **ScheduleCreationDropdownSelect:** componente importado y posteriormente modificado, destinado a servir de desplegable, con la posibilidad adicional de búsqueda por palabras clave entre sus opciones.
- **ScheduleCreationMenuWarningsField.**
- **ScheduleCreationAsigActionsButton:** componente importado y posteriormente modificado, destinado a limpiar el formulario o crear una asignatura, en caso de ser pulsado, y separado en dos ocurrencias.

- **DndProvider:** componente importado que proporciona el contexto para permitir el uso de un Calendar con Drag & Drop habilitado.
- **DnDCalendar:** componente importado que supone una modificación al componente base “Calendar”, permitiendo desplazar sus eventos mediante la funcionalidad Drag & Drop.
- **AdminCalendarEvent:** componente desarrollado, encargado de actuar como un evento del calendario, adaptado a las necesidades propias del sistema, específicamente a la sección del subdirector.
- **ModifyCalendarEvent:** componente desarrollado enfocado a servir de formulario, inicialmente compuesto por la información de una asignatura que se desee modificar.

Cuenta con los siguientes componentes hijos:

- **ScheduleCreationInput.**
- **ScheduleCreationSelect.**
- **ScheduleCreationDropdownSelect.**
- **ScheduleCreationMenuWarningsField.**
- **ScheduleCreationAsigActionsButton.**

4.3. Diagrama de componentes

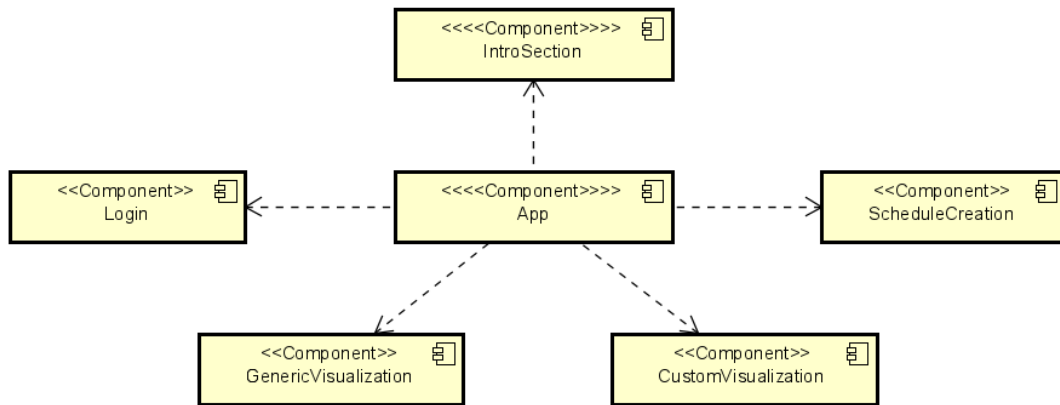


Figura 24: Diagrama de Componentes

4.4. Prototipo de interfaces

A continuación, se muestran los prototipos para las distintas secciones de la aplicación web, creados con la herramienta de diseño gráfico Figma. Se trata de una herramienta de prototipado y creación de interfaces orientadas a diseño web y móvil (*Figma, s.f*).

En la siguiente figura, se muestra un prototipo acerca de la sección introductoria de la aplicación web, en el que pueden apreciarse un dialogo inicial con el usuario, así como las opciones que se ofrecen y las vías de acceso a cada una de ellas.

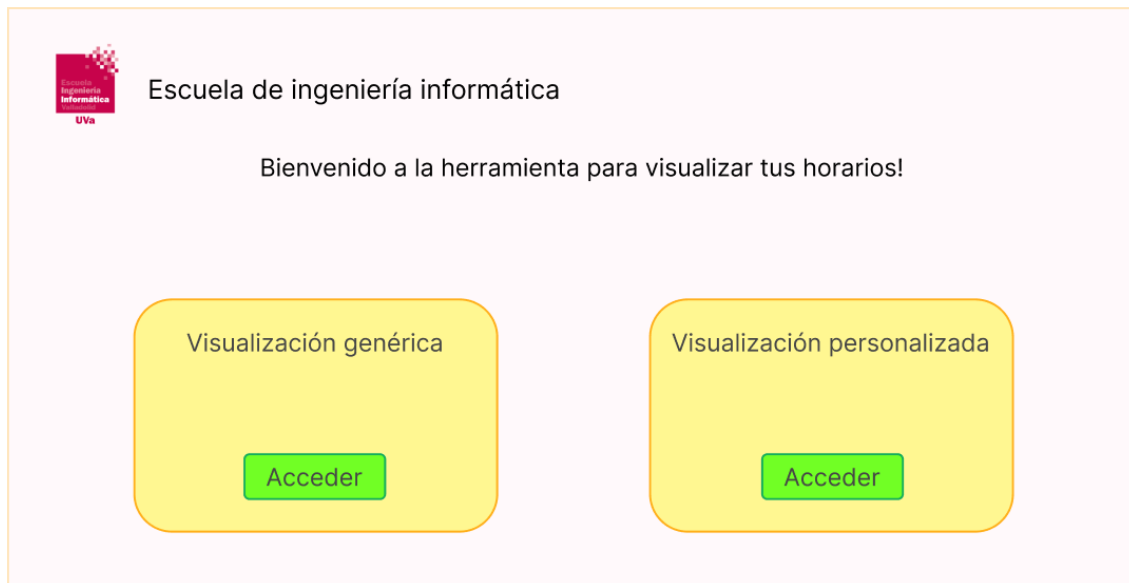


Figura 25: Prototipo de interfaz - Sección introducción

En la siguiente figura, tras elegir la primera opción de la sección introductoria, se muestra un prototipo acerca de la sección correspondiente a las visualizaciones de horarios genéricos. En ella se encuentran los conceptos del menú de filtrado, basados en criterios prefijados como el conjunto de grados, cursos, semestres o menciones seleccionables, y el horario, donde se muestran las asignaturas que cumplen estos criterios.

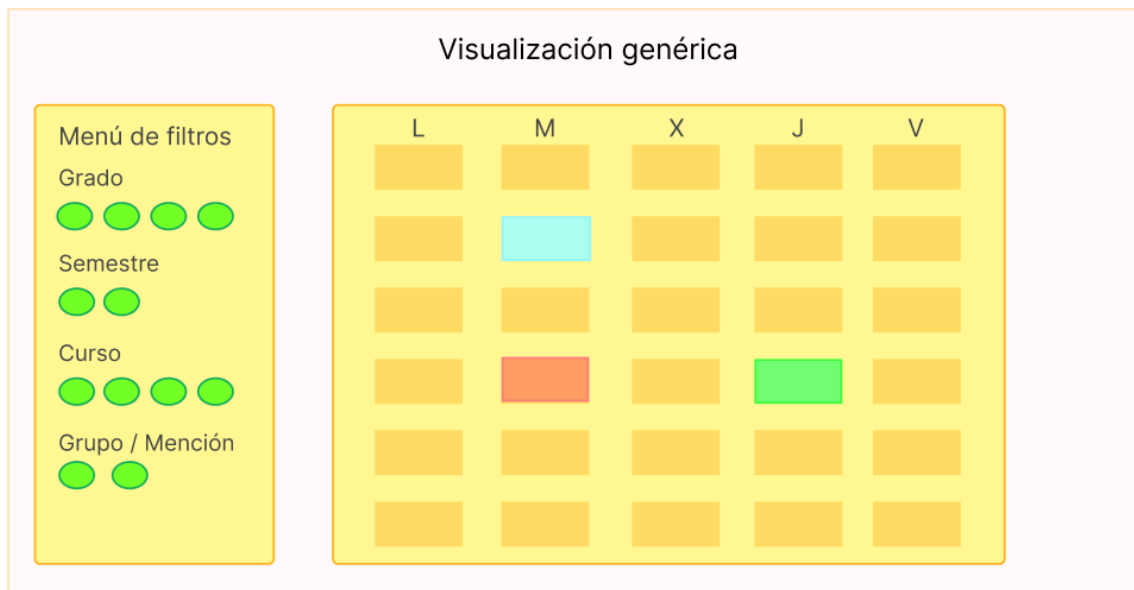


Figura 26: Prototipo de interfaz - Visualización genérica

En la siguiente figura, tras elegir la segunda opción de la sección introductoria, se muestra un prototipo acerca de la sección correspondiente a las visualizaciones de horarios personalizados. En ella se encuentran los conceptos del menú de filtrado, basados en criterios prefijados como el conjunto de grados, cursos, semestres o menciones seleccionables, tras los cuales se muestra un desplegable que permite la elección libre de asignaturas, y el horario adaptado a las necesidades del usuario, donde se muestran las asignaturas que cumplen estos criterios.

A continuación, se muestra el prototipo acerca de la modificación de horario por parte del administrador. Cuenta con los conceptos de menú de filtrado lateral, presentando la pestaña que permite modificar un horario a través de un breve formulario, y el horario, cuyas asignaturas mostradas se adaptan a los criterios del formulario cuya duración, hora de inicio, hora de fin o día, pueden ser modificados a través de Drag & Drop y del redimensionado.

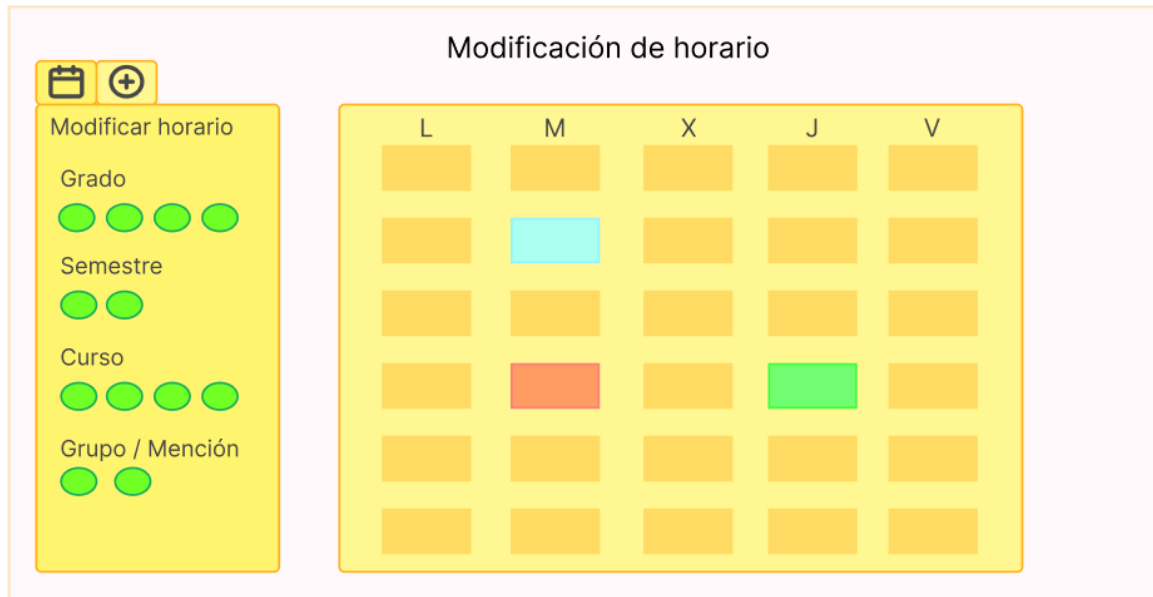


Figura 29: Prototipo de interfaz - Modificación de horario

A continuación, se muestra el prototipo acerca de la creación de asignaturas por parte del administrador. Cuenta con los conceptos de menú de filtrado lateral, presentando la pestaña que permite crear una nueva asignatura a través de un formulario, y el horario, cuyas asignaturas mostradas se adaptan a los criterios del formulario completado en la anterior sección del menú. Una vez completada la información sobre la asignatura a añadir, esta será mostrada en el horario junto con el resto de las asignaturas que coincidan en sus criterios.

Añadir asignatura

Código

Siglas

Nombre

Semestre

Día

Grupo

Subgrupo

Aula

Profesor

L

M

X

J

V

Figura 30: Prototipo de interfaz - Creación de asignaturas

Capítulo 5

Implementación

5. Implementación

En este capítulo se proporciona información detallada acerca del proceso de implementación de la aplicación web, indagando en los aspectos cómo la estructura y funcionamiento del código y las herramientas empleadas en su desarrollo.

5.1. Herramienta de desarrollo

La herramienta escogida para el desarrollo de la aplicación ha sido Visual Studio Code, un editor de código fuente desarrollado por Microsoft. En concreto, ha sido empleado en un equipo informático con sistema operativo Windows 10.

5.2. Tecnologías empleadas

A su vez, se procede a detallar las tecnologías software que cimentan el desarrollo de este proyecto y que han sido utilizadas para su desarrollo.

5.2.1. React

Uno de los grandes pilares dentro de las tecnologías empleadas es React, siendo esta un framework de JavaScript de código abierto enfocado a la construcción de interfaces intuitivas e interactivas, para aplicaciones web y móviles, desarrollado actualmente por Meta Open Source (*Meta Open Source, s.f*). Su funcionamiento se basa en el concepto de componentización o creación de componentes, los cuales por medio del correcto desarrollo software, pretenden ser objetivo de reutilización por parte de los desarrolladores para construir aplicaciones completas (*Imagina Formación, s.f*).

React utiliza una sintaxis denominada JSX, permitiendo mezclar el uso de HTML y JavaScript en cada componente.

Dichos componentes constan de un ciclo de vida con diferentes etapas, desde la creación en la renderización inicial del árbol de componentes hasta su modificación o eliminación (*Codemotion, 2023*). A lo largo de este proceso, los componentes pueden responder a acciones concretas o a lo que se denomina cambios de estado.

Los cambios de estado suponen un elemento claramente identificativo de React. Para su gestión se utilizan los Hooks, funciones que permiten emplear características del ciclo de vida de React dentro de cada componente, facilitando la reutilización de la lógica del estado.

Los Hooks más utilizados en el desarrollo React son `useState` y `useEffect`. `useState` permite añadir un estado a un componente. Al invocar a `useState` inicialmente, se crea una variable de estado y una función para actualizar esa variable, permitiendo desencadenar nuevos renderizados cuando dicho estado sea alterado.

A su vez, otro hooks cuyo comportamiento resulta imprescindible es `useEffect`, el cual se utiliza de forma combinada con los estados definidos en `useState`, puesto que su principal función es la suscripción a eventos cómo el cambio de estados registrados en su lista de dependencias, desencadenando efectos en el componente donde se ubique.

Resulta fundamental esclarecer que los cambios de estado en React son asíncronos, suponiendo que, tras la invocación del Hook correspondiente, React programará dicha actualización de estado para ser efectiva en el siguiente ciclo de renderizado.

5.2.2. Node.js

Node.js se trata de un entorno de ejecución basado en JavaScript multiplataforma, de código abierto y gratuito, creado por Ryan Dahl en 2009 y basado en el motor V8 de JavaScript de Google Chrome (*Node.js, s.f*). Sus características principales se basan en la ejecución de JavaScript fuera del navegador, así como una gran escalabilidad en la gestión de grandes cargas de tráfico, manejando conexiones simultáneas.

Este manejo de conexiones sigue un modelo de evaluación de un único hilo de ejecución, empleando entradas y salidas asíncronas las cuales tienen permitido ser ejecutadas en volúmenes de hasta cientos de miles sin incurrir en costes asociados a un cambio de contexto.

Este diseño de compartición de un único hilo de ejecución entre todo el conjunto de solicitudes soporta adecuadamente las necesidades de aplicaciones de alta concurrencia. Un cambio de contexto consiste en la ejecución de una rutina perteneciente al núcleo del sistema operativo multitarea de un PC, cuyo propósito es parar la ejecución de un hilo o proceso para dar paso a la ejecución de otro distinto.

5.2.3. npm

Node Package Manager (npm) se trata del gestor de paquetes por defecto para Node.js, permitiendo compartir, actualizar y, en general, gestionar bibliotecas y herramientas que se usan en proyectos JavaScript y TypeScript (*npm, s.f*).

Es una fuente de una numerosa amplitud de paquetes y bibliotecas de código abierto, suponiendo una fuente de conocimiento y funcionalidad para un gran número de desarrolladores.

En el entorno de React, esta herramienta es utilizada para gestionar dependencias, facilitando su proceso de incorporación o borrado, resultando crucial para dependencias clave como “React” o “ReactDOM”.

En este proyecto, en concreto, han sido incorporadas nuevas bibliotecas cuyo papel es descrito a continuación:

- **cors:** proporciona herramientas que permiten configurar CORS (Cross Origin Resource Sharing), permitiendo al backend aceptar peticiones de dominios distintos.
- **express:** proporciona un framework sencillo para la creación de servidores web y APIs en Node.js.
- **html2pdf.js:** permite convertir HTML a formato PDF directamente desde el navegador, siendo muy útil para la generación de informes o documentos descargables.
- **lucide-react:** proporciona iconos en formato SVG personalizables para React, es parte del proyecto Lucide.
- **moment:** permite la gestión de fechas y horas, formateo, cálculo de diferencias horarias, conversiones, entre otros aspectos.
- **react:** biblioteca principal para construir interfaces de usuario en React.
- **react-dom:** permite que react emplee el DOM del navegador para renderizar sus componentes.
- **react-router-dom:** permite la navegación entre vistas o páginas en aplicaciones React de una sola página (SPA), aprovechando la página única para mostrar las múltiples vistas sin necesidad de recargar la página completa.
- **react-big-calendar:** biblioteca que proporciona un calendario interactivo para React, facilitando la gestión y visualización de eventos (*jquense s.f*).
- **react-dnd:** biblioteca que proporciona funciones que convierten el calendario estándar de la librería “react-big-calendar” en un calendario que permite realizar Drag & Drop sobre sus eventos.
- **react-dnd-html5-backend:** biblioteca que proporciona el backend html5 y contexto necesario para el correcto funcionamiento del calendario en navegadores modernos.
- **react-draggable:** biblioteca que permite la movilidad de componentes a lo largo del navegador a través de un arrastre.
- **react-resizable:** biblioteca que habilita el redimensionado de componentes en React.

- **semantic-ui-css:** biblioteca que proporciona un css estándar para el estilado de componentes importados desde Semantic UI React.
- **semantic-ui-react:** biblioteca que proporciona componentes basados en Semantic UI.

A su vez, se han empleado una serie de dependencias, necesarias durante el desarrollo del presente proyecto:

- **eslint:** herramienta para detectar errores y aplicar buenas prácticas durante el desarrollo de código.
- **@eslint/js:** configuración base de ESLint para JavaScript puro.
- **eslint-plugin-react:** proporciona reglas específicas de evaluación y validación de componentes React.
- **eslint-plugin-react-hooks:** proporciona reglas específicas de evaluación y verificación de uso correcto de hooks en React.
- **globals:** define variables globales para su uso por ESLint, como window, process, entre otros.
- **vite:** proporciona un empaquetador y servidor de desarrollo rápido.
- **@vitejs/plugin-react-swc:** plugin que permite el uso de react junto con el compilador SWC, más rápido que Babel.

5.2.4. Metodología ágil

La metodología ágil es un enfoque de gestión de proyectos enfocado en entrega continua de valor, a través de versiones incrementales y funcionales del producto. Dichas entregas son guiadas por iteraciones de desarrollo de corta duración, en las que el equipo de proyecto prioriza la estrecha colaboración y comunicación interna y con el cliente, permitiendo el reparto de las responsabilidades y la adaptación eficaz a cambios y necesidades emergentes durante el proceso de desarrollo de cada iteración (*Asana, s.f*).

Las iteraciones son categorizadas como sprints, en los cuales se realiza una evaluación de los objetivos y cómo cumplimentarlos, se desarrolla un incremento funcional y probado del producto, siendo posteriormente mostrado y entregado al cliente con intención de verificar si se ajusta a sus necesidades y por último, se realiza una visión retrospectiva del proceso con la intención de reflexionar sobre el proceso de desarrollo y fomentar la mejora continua (*Atlassian, s.f*).

5.2.5. Jira

Jira es una herramienta de gestión de proyectos basada en la metodología ágil y desarrollada por Atlassian. Permite la planificación, fragmentación y dosificación del desarrollo de proyectos, a través de la creación de conjuntos de tareas que permiten enfocar el flujo de trabajo en pasos y acciones concretas, con el objetivo de visibilizar los avances logrados durante todo el proceso.

A través de la creación de historias de usuario y tareas, las responsabilidades son asignadas entre los miembros del equipo de desarrollo, pudiendo a su vez realizar un seguimiento del progreso de cada tarea, inconvenientes que puedan llegar a surgir o incidencias cómo incumplimientos o dificultades.

Por ello, gracias al marco organizativo proporcionado por la metodología ágil y su gestión a través de esta herramienta, ha sido posible particionar el trabajo en sprints, priorizando las historias de usuario y tareas objetivo de ser completadas en cada uno de ellos (*Atlassian, s.f*).

5.2.6. Unified Modeling Language

El lenguaje de modelado unificado es utilizado en el análisis, representación y documentación de sistemas software complejos. Facilita la comunicación entre miembros del equipo de desarrollo, ayudando a comprender la arquitectura y comportamiento del sistema a través de diagramas que reflejan la interacción del usuario y aportan un enfoque clave en el desarrollo (*ComponentSource, s.f; Lucidchart, s.f*).

5.2.7. Astah

Astah se trata de una herramienta de modelado especialmente utilizada en el ámbito de la Ingeniería de software. Permite la creación y representación de diagramas siguiendo los esquemas y estándares de Unified Modeling Language (UML), permitiendo el análisis, visualización y modelado previo a la implementación de sistemas.

El empleo de esta herramienta ha sido crucial en el desarrollo de diagramas UML, así como en el análisis y documentación de los requisitos y partición del sistema en casos de uso (*ComponentSource, s.f*).

5.2.8. Sistema de gestión de versiones

Durante el proceso de desarrollo, ha sido llevado a cabo una gestión de versiones en base a los progresos logrados en cada una de sus etapas. En dicho control de versiones se ha utilizado la herramienta de Git, junto con GitLab, creando un árbol de versiones cuyo acceso ha sido compartido con el tutor universitario.

Git supone una solución eficaz a la problemática del control de versiones, permitiendo el guardado completo del historial de versiones del proyecto, creación de ramas de desarrollo a elección del usuario y guiadas por hitos u otros criterios. Al mismo tiempo, permite la recuperación frente a errores, permitiendo revertir los cambios aplicados para regresar a versiones anteriores. Por último, también permite la fusión de cambios entre versiones, junto con otras funciones.

5.3. Implementación de funcionalidad

En este apartado se detallarán aspectos claves del desarrollo de código, detrás del funcionamiento de la aplicación web. Cabe remarcar que toda la implementación se basa en el uso de la biblioteca de React.

5.3.1. Carga inicial y enrutamiento de la aplicación web

Durante el arranque de la aplicación web, se renderiza el componente padre, almacenado en el archivo App.jsx, el cual interactúa como componente de enrutamiento a una sección inicial dedicada al uso por parte del alumnado o profesorado universitario.

Para dicho enrutamiento se emplea la biblioteca “react-router-dom” para su configuración.

Cada uno de los componentes principales consta de una ruta definida y asignada, empleando el componente Route o PrivateRoute para la renderización de cada uno de ellos en función de la URL actual. Dichos componentes son IntroSection, Login, GenericVisualization, CustomVisualization y ScheduleCreation, siendo pasados por argumento a su componente Route específico.

También se envían por argumento una serie de elementos requeridos por varias de los componentes, permitiendo el correcto funcionamiento de cada una de sus secciones.

En lo referente a las rutas, se definen el siguiente conjunto de ellas:

- **Ruta base o raíz (“/”):** reservada al renderizado del componente IntroSection mostrado al inicio de la interacción del usuario con el entorno web. Se trata de una sección introductoria en la que se sugiere al usuario tomar una de las dos opciones disponibles para la visualización de horarios.
- **Ruta comodín (“*”):** reservada al renderizado del componente IntroSection en caso de que el usuario introduzca una URL distinta a las registradas, siendo redirigido a la sección de inicio.
- **Ruta de administrador (“/admin”):** reservada al renderizado del componente Login. Se trata de una sección de inicio de sesión, cuyo uso se encuentra reservado al administrador del sistema.
- **Ruta de visualización genérica (“/horarios-genericos”):** reservada al renderizado del componente GenericVisualization. Se trata de la sección dedicada a la visualización genérica de horarios por parte del usuario. A su vez, se envían una serie de parámetros referentes a los grados, semestres y cursos, entre otros aspectos, para el correcto funcionamiento de esta sección.

- **Ruta de visualización personalizada (“/horarios-personalizados”):** reservada al renderizado del componente CustomVisualization. Se trata de la sección dedicada a la visualización y creación de horarios personalizados por parte del usuario. A su vez, este componente recibe como parámetros adicionales una serie de enumeraciones referentes a los grados, semestres y cursos, entre otros, para su correcto funcionamiento.
- **Ruta de creación de horarios (“/creacion-horarios”):** reservada al renderizado del componente ScheduleCreation. Se trata de la sección dedicada a la creación o modificación de horarios, a través del borrado, modificación o creación de asignaturas. Este componente recibe los mismos parámetros adicionales que los anteriores componentes de visualización de horarios y se encuentra encapsulado dentro del componente PrivateRoute, el cual comprueba si el administrador tiene su sesión iniciada dentro del sistema.

5.3.2. Lectura de ficheros y obtención de eventos de calendario

La lectura de ficheros en cada una de las secciones de la aplicación web es realizada a través de la configuración de un hook `useEffect`, realizando la lectura de un fichero predefinido en el servidor, para las secciones de visualización genérica y personalizada de horarios, y en el caso de la sección dedicada al administrador, solicitando el fichero a leer.

En este fichero es almacenado en formato JavaScript Object Notation (JSON) la información de cada una de las asignaturas registradas en el sistema, tratándose de la fuente de información a partir de la cual los eventos del calendario son obtenidos.

Tras dicha lectura, los eventos son creados a partir de cada uno de los campos en cada asignatura dentro del fichero, permitiendo la dotación de diversos valores a sus atributos que convierten a cada evento en distintivo del resto.

```
useEffect(() => {
  const cargarAsignaturas = async () => {
    try {
      const response = await fetch("/asignaturas.json");
      const data = await response.json();

      setSubjects(data); // Guardar asignaturas en el estado

      const eventos = data.map((asignatura) => {
        const diaSemana = diasSemana[asignatura.Dia];
        if (diaSemana === undefined) return null;

        const [hora, minutos] = asignatura.HoraInicio.split(":").map(Number);

        // Obtener el lunes de la semana actual
        const hoy = moment();
        const lunesSemanaActual = hoy.clone().startOf("isoWeek");

        // Calcular la fecha del día de la asignatura dentro de esta semana
        const inicio = lunesSemanaActual.clone().add(diaSemana - 1, "days").set({
          hour: hora,
          minute: minutos,
          second: 0,
        }).toDate();

        const fin = moment(inicio).add(parseInt(asignatura.Duracion), "hours").toDate();

        return {
          id: `${asignatura.Dia} - ${asignatura.Siglas} - ${asignatura.Grupo} - ${asignatura.GrupoLaboratorio} - ${asignatura.Clase} - ${
            title: `${asignatura.Siglas} \n \n ${asignatura.Grupo} - ${asignatura.Clase}`,
            start: inicio,
            end: fin,
            nombre: asignatura.Nombre,
            siglas: asignatura.Siglas,
            grado: asignatura.Grado,
            semestre: asignatura.Semestre,
            curso: asignatura.Curso,
            grupo: asignatura.Grupo,
            grupoLaboratorio: asignatura.GrupoLaboratorio,
            mencion: asignatura.Mencion,
            aula: asignatura.Clase,
            profesor: asignatura.Profesor,
            color: asignatura.Color,
            dia: asignatura.Dia,
            codigo: asignatura.Codigo
          };
        }).filter(Boolean);

      setEvents(eventos);
    } catch (error) {
      console.error("Error cargando los datos del JSON:", error);
    }
  };

  cargarAsignaturas();
}, []);
```

Figura 31: Fragmento de código - Lectura de fichero y creación de eventos

Capítulo 6

Pruebas

6. Pruebas

En el presente capítulo se detalla el proceso de pruebas realizadas sobre el sistema, teniendo por objetivo la verificación de su correcto funcionamiento, teniendo su enfoque en las que suponen un uso de la funcionalidad esencial del entorno web desarrollado.

6.1. Pruebas realizadas sobre el cliente

Con el objetivo de analizar el correcto comportamiento del sistema sobre el cliente y su aptitud frente a las expectativas buscadas, se exponen el conjunto de pruebas a las que el sistema ha sido sometido.

6.1.1. Test de navegabilidad

En la presente sección se muestra la siguiente tabla, conformada por las pruebas de navegabilidad dentro de la aplicación web, reflejando el resultado esperado y el resultado finalmente obtenido.

Nº	Descripción	Entrada url	Resultado esperado	Resultado Obtenido
P01	Usuario accede a la página de inicio de la aplicación	/	Redirección a la página de inicio	Resultado esperado
P02	Usuario accede a la página de visualización genérica de horarios	/horarios-genericos	Redirección a la página de visualización genérica	Resultado esperado
P03	Usuario accede a la página de visualización personalizada de horarios	/horarios-personalizados	Redirección a la página de visualización personalizada	Resultado esperado
P04	Usuario accede a la página de inicio de sesión	/admin	Redirección a la página de inicio de sesión	Resultado esperado
P05	Administrador inicia su sesión y accede a página de gestión de horarios	/creación-horarios	Redirección a la página de gestión de horarios	Resultado esperado

Tabla 22: Test de navegabilidad

6.1.2. Test de visualización genérica de horario troncal general

En este test es evaluada la funcionalidad en el instante de realizar una visualización genérica de un horario troncal, incluyendo solo las sesiones teóricas de asignaturas.

Nº	Descripción	Resultado esperado	Resultado obtenido
P06	Usuario selecciona una opción referente a los grados	La correspondiente elección queda resaltada	Resultado esperado
P07	Usuario selecciona una opción referente a los semestres	La correspondiente elección queda resaltada	Resultado esperado
P08	Usuario selecciona una opción referente a los cursos troncales	La correspondiente elección queda resaltada y el horario mostrado se adapta a los criterios seleccionados	Resultado esperado

Tabla 23: Test de visualización genérica de horario troncal general

6.1.3. Test de visualización genérica de horario troncal general con sesiones de laboratorio

En este test es evaluada la funcionalidad en el instante de realizar una visualización genérica de un horario troncal, incluyendo las sesiones teóricas y prácticas de asignaturas.

Nº	Descripción	Resultado esperado	Resultado obtenido
P06	Usuario selecciona una opción referente a los grados	La correspondiente elección queda resaltada	Resultado esperado
P07	Usuario selecciona una opción referente a los semestres	La correspondiente elección queda resaltada	Resultado esperado
P08	Usuario selecciona una opción referente a los cursos troncales	La correspondiente elección queda resaltada y el horario mostrado se adapta a los criterios seleccionados	Resultado esperado
P09	Usuario activa la opción que hace visibles las sesiones de laboratorio	La correspondiente elección queda resaltada y el horario mostrado se adapta a los criterios seleccionados	Resultado esperado

Tabla 24: Test de visualización genérica de horario troncal general con sesiones de laboratorio

6.1.4. Test de visualización genérica de horario troncal específico

En este test es evaluada la funcionalidad en el instante de realizar una visualización genérica de un horario troncal de un grupo en específico, incluyendo solo las sesiones teóricas de asignaturas.

Nº	Descripción	Resultado esperado	Resultado obtenido
P10	Usuario selecciona una opción referente a los grados	La correspondiente elección queda resaltada	Resultado esperado
P11	Usuario selecciona una opción referente a los semestres	La correspondiente elección queda resaltada	Resultado esperado
P12	Usuario selecciona una opción referente a los cursos troncales	La correspondiente elección queda resaltada	Resultado esperado
P13	Usuario selecciona una opción referente al grupo deseado	La correspondiente elección queda resaltada y el horario mostrado se adapta a los criterios seleccionados	Resultado esperado

Tabla 25: Test de visualización genérica de horario troncal específico

6.1.5. Test de visualización genérica de horario troncal específico con sesiones de laboratorio

En este test es evaluada la funcionalidad en el instante de realizar una visualización genérica de un horario troncal de un grupo en específico, incluyendo las sesiones teóricas y prácticas de asignaturas.

Nº	Descripción	Resultado esperado	Resultado obtenido
P14	Usuario selecciona una opción referente a los grados	La correspondiente elección queda resaltada	Resultado esperado
P15	Usuario selecciona una opción referente a los semestres	La correspondiente elección queda resaltada	Resultado esperado
P16	Usuario selecciona una opción referente a los cursos troncales	La correspondiente elección queda resaltada	Resultado esperado
P17	Usuario selecciona una opción referente al grupo deseado	La correspondiente elección queda resaltada y el horario mostrado se adapta a los criterios seleccionados	Resultado esperado
P18	Usuario activa la opción que hace visibles las sesiones de laboratorio	La correspondiente elección queda resaltada y el horario mostrado se adapta a los criterios seleccionados	Resultado esperado

Tabla 26: Test de visualización genérica de horario troncal específico con sesiones de laboratorio

6.1.6. Test de visualización genérica de horario de mención

En este test es evaluada la funcionalidad en el instante de realizar una visualización genérica de un horario referente a una mención, incluyendo solo las sesiones teóricas de asignaturas.

Nº	Descripción	Resultado esperado	Resultado obtenido
P19	Usuario selecciona una opción referente a los grados	La correspondiente elección queda resaltada	Resultado esperado
P20	Usuario selecciona una opción referente a los semestres	La correspondiente elección queda resaltada	Resultado esperado
P21	Usuario selecciona una opción referente a los cursos de especialidad en menciones	La correspondiente elección queda resaltada	Resultado esperado
P22	Usuario selecciona una opción referente a la mención deseada	La correspondiente elección queda resaltada y el horario mostrado se adapta a los criterios seleccionados	Resultado esperado

Tabla 27: Test de visualización genérica de horario de mención

6.1.7. Test de visualización genérica de horario de mención con sesiones de laboratorio

En este test es evaluada la funcionalidad en el instante de realizar una visualización genérica de un horario referente a una mención, incluyendo las sesiones teóricas y prácticas de asignaturas.

Nº	Descripción	Resultado esperado	Resultado obtenido
P23	Usuario selecciona una opción referente a los grados	La correspondiente elección queda resaltada	Resultado esperado
P24	Usuario selecciona una opción referente a los semestres	La correspondiente elección queda resaltada	Resultado esperado
P25	Usuario selecciona una opción referente a los cursos de especialidad en menciones	La correspondiente elección queda resaltada	Resultado esperado
P26	Usuario selecciona una opción referente a la mención deseada	La correspondiente elección queda resaltada y el horario mostrado se adapta a los criterios seleccionados	Resultado esperado

Tabla 28: Test de visualización genérica de horario de mención con sesiones de laboratorio

6.1.8. Test de visualización personalizada de horario

En este test es evaluada la funcionalidad en el instante de realizar una visualización personalizada de horario, incluyendo solo las sesiones teóricas de asignaturas.

Nº	Descripción	Resultado esperado	Resultado obtenido
P27	Usuario selecciona una opción referente a los grados	La correspondiente elección queda resaltada	Resultado esperado
P28	Usuario selecciona una opción referente a los semestres	La correspondiente elección queda resaltada	Resultado esperado
P29	Usuario selecciona tantas opciones de cursos como desee	La/s correspondiente/s elecciones quedan resaltadas	Resultado esperado
P30	Usuario selecciona el grupo o mención relacionados con cada uno de los cursos elegidos	La/s correspondiente/s elecciones quedan resaltadas	Resultado esperado
P31	Usuario selecciona las asignaturas que cumplen los criterios elegidos	La/s correspondiente/s elecciones quedan resaltadas y el horario refleja la/s asignatura/s	Resultado esperado

Tabla 29: Test de visualización personalizada de horario

6.1.9. Test de visualización personalizada de horario con sesiones de laboratorio

En este test es evaluada la funcionalidad en el instante de realizar una visualización personalizada de horario, incluyendo en las sesiones teóricas y prácticas de asignaturas.

Nº	Descripción	Resultado esperado	Resultado obtenido
P32	Usuario selecciona una opción referente a los grados	La correspondiente elección queda resaltada	Resultado esperado
P33	Usuario selecciona una opción referente a los semestres	La correspondiente elección queda resaltada	Resultado esperado
P34	Usuario selecciona tantas opciones de cursos como desee	La/s correspondiente/s elecciones quedan resaltadas	Resultado esperado
P35	Usuario selecciona el grupo o mención relacionados con cada uno de los cursos elegidos	La/s correspondiente/s elecciones quedan resaltadas	Resultado esperado
P36	Usuario selecciona las asignaturas que cumplen los criterios elegidos	La/s correspondiente/s elecciones quedan resaltadas y el horario refleja la/s asignatura/s	Resultado esperado
P37	Usuario activa la opción que hace visibles las sesiones de laboratorio	La correspondiente elección queda resaltada en el desplegable de asignaturas	Resultado esperado
P38	Usuario selecciona las sesiones de laboratorio deseadas	La/s correspondiente/s elecciones quedan resaltadas y el horario refleja la/s sesiones de laboratorio	Resultado esperado

Tabla 30: Test de visualización personalizada de horario con sesiones de laboratorio

6.1.10. Test de inicio de sesión en sección de administrador

En este test es evaluada la funcionalidad en el instante de iniciar sesión en la sección de administrador.

Nº	Descripción	Resultado esperado	Resultado obtenido
P39	Usuario administrador introduce sus credenciales	El campo de credenciales resulta completado	Resultado esperado
P40	Usuario pulsa el botón de inicio de sesión	El sistema comprueba la validez de credenciales y redirige al usuario a la sección de administrador	Resultado esperado

Tabla 31: Test de inicio de sesión en sección de administrador

6.1.11. Test de carga de fichero en sección de administrador

En este test es evaluada la funcionalidad en el instante de cargar un fichero para proceder a la lectura de su contenido, por parte del administrador.

Nº	Descripción	Resultado esperado	Resultado obtenido
P41	Usuario administrador selecciona la opción de carga de fichero de asignaturas	El sistema abre la pestaña de selección de fichero	Resultado esperado
P42	Usuario administrador selecciona el fichero deseado	El sistema carga el fichero correspondiente	Resultado esperado

Tabla 32: Test de carga de fichero

6.1.12. Test de visualización de horario en sección de administrador

En este test es evaluada la funcionalidad en el instante de visualizar un horario, por parte del administrador.

Nº	Descripción	Resultado esperado	Resultado obtenido
P43	El usuario administrador repite las acciones del test 6.1.11	El sistema refleja las acciones realizadas	Resultado esperado
P44	Usuario administrador selecciona la pestaña de gestión de horarios	El contenido de la pestaña se muestra en el menú	Resultado esperado
P45	Usuario administrador repite las acciones del test 6.1.3 o 6.1.7	El sistema refleja las acciones realizadas	Resultado esperado

Tabla 33: Test de visualización de horario en sección de administrador

6.1.13. Test de manipulación de sesión de asignatura con Drag & Drop

En este test es evaluada la funcionalidad en el instante de realizar el desplazamiento de una sesión de asignatura en particular, por parte del administrador.

Nº	Descripción	Resultado esperado	Resultado obtenido
P46	El usuario administrador repite las Acciones del test 6.1.12	El sistema refleja las acciones realizadas	Resultado esperado
P47	Usuario administrador selecciona la sesión de asignatura a desplazar empleando Drag & Drop	El sistema trata a la sesión de asignatura como un evento desplazable	Resultado esperado
P48	Usuario administrador sitúa la sesión de asignatura en su nueva ubicación	El sistema comprueba la compatibilidad de dicho movimiento dentro del horario y refleja el cambio	Resultado esperado

Tabla 34: Test de manipulación de sesión de asignatura con Drag & Drop

6.1.14. Test de manipulación de sesión de asignatura con Resize

En este test es evaluada la funcionalidad en el instante de realizar la redimensión de una sesión de asignatura en particular, por parte del administrador.

Nº	Descripción	Resultado esperado	Resultado obtenido
P49	El usuario administrador repite las Acciones del test 6.1.12	El sistema refleja las acciones realizadas	Resultado esperado
P50	Usuario administrador selecciona la sesión de asignatura a redimensionar con opción Resize	El sistema trata a la sesión de asignatura como un evento redimensionable	Resultado esperado
P51	Usuario administrador acorta o aumenta la duración de dicha sesión de asignatura	El sistema comprueba la compatibilidad de dicho movimiento dentro del horario y refleja el cambio	Resultado esperado

Tabla 35: Test de manipulación de sesión de asignatura con Resize

6.1.15. Test de guardado de fichero en sección de administrador

En este test es evaluada la funcionalidad en el instante de guardar un nuevo fichero o sobrescribir uno ya existente, por parte del administrador.

Nº	Descripción	Resultado esperado	Resultado obtenido
P52	El usuario administrador selecciona la opción de escritura de fichero	El sistema abre la pestaña de escritura de fichero	Resultado esperado
P53	Usuario administrador selecciona el fichero que crear o sobrescribir	El sistema refleja las acciones realizadas	Resultado esperado

Tabla 36: Test de guardado de fichero en sección de administrador

6.1.16. Test de exportación de horario a formato PDF

En este test es evaluada la funcionalidad en el momento de realizar la exportación a formato PDF de un horario en particular.

Nº	Descripción	Resultado esperado	Resultado obtenido
P54	El usuario selecciona la opción de exportación a PDF	El sistema abre la pestaña de escritura de fichero	Resultado esperado

Tabla 37: Test de exportación de horario a formato PDF

6.1.17. Test de modificación de sesión de asignatura en sección de administrador

En este test es evaluada la funcionalidad en el momento de modificar la sesión de una asignatura en particular, por parte del administrador.

Nº	Descripción	Resultado esperado	Resultado obtenido
P55	El usuario administrador selecciona la sesión de asignatura a modificar	El sistema abre la pestaña de modificación de sesión de la asignatura	Resultado esperado
P56	Usuario administrador modifica los campos deseados	El sistema refleja las acciones realizadas	Resultado esperado
P57	Usuario administrador pulsa el botón de modificación	El sistema refleja el cambio	Resultado esperado

Tabla 38: Test de modificación de sesión de asignatura en sección de administrador

6.1.18. Test de borrado de sesión de asignatura en sección de administrador

En este test es evaluada la funcionalidad en el momento de borrar la sesión de una asignatura en particular, por parte del administrador.

Nº	Descripción	Resultado esperado	Resultado obtenido
P58	El usuario administrador selecciona la sesión de asignatura a modificar	El sistema abre la pestaña de modificación de sesión de la asignatura	Resultado esperado
P59	Usuario administrador pulsa el botón de borrado	El sistema refleja el cambio	Resultado esperado

Tabla 39: Test de borrado de sesión de asignatura en sección de administrador

6.1.19. Test de borrado de incidencias de sesión de asignatura en sección de administrador

En este test es evaluada la funcionalidad en el instante de borrar las incidencias relacionadas con la sesión de una asignatura en particular, por parte del administrador.

Nº	Descripción	Resultado esperado	Resultado obtenido
P60	El usuario administrador selecciona la sesión de asignatura a modificar	El sistema abre la pestaña de modificación de sesión de la asignatura	Resultado esperado
P61	Usuario administrador pulsa el limpieza de incidencias	El sistema refleja el cambio	Resultado esperado

Tabla 40: Test de borrado de incidencias de sesión de asignatura en sección de administrador

6.1.20. Test de creación de sesión de asignatura en sección de administrador

En este test es evaluada la funcionalidad en el instante de creación de una nueva sesión de asignatura por parte del administrador.

Nº	Descripción	Resultado esperado	Resultado obtenido
P62	El usuario administrador Selecciona la pestaña de creación de asignaturas	El sistema abre la pestaña de creación de asignaturas y muestra el formulario a rellenar	Resultado esperado
P63	Usuario administrador rellena el formulario con la información de la sesión de asignatura	El sistema refleja el cambio	Resultado esperado
P64	Usuario administrador pulsa el botón de crear asignaturas	El sistema comprueba la compatibilidad de dicha acción dentro del horario y crea las asignaturas	Resultado esperado

Tabla 41: Test de creación de sesión de asignatura en sección de administrador

6.1.21. Test de limpieza de formulario de creación de sesión de asignatura en sección de administrador

En este test es evaluada la funcionalidad de limpieza del formulario en el instante de creación de una nueva sesión de asignatura por parte del administrador.

Nº	Descripción	Resultado esperado	Resultado obtenido
P65	El usuario administrador selecciona la pestaña de creación de asignaturas	El sistema abre la pestaña de creación de asignaturas y muestra el formulario a rellenar	Resultado esperado
P66	Usuario administrador rellena el formulario con la información de la sesión de asignatura	El sistema refleja el cambio	Resultado esperado
P67	Usuario administrador pulsa el botón de limpieza de formulario	El sistema realiza una limpieza del formulario, dejando cada campo en su valor por defecto	Resultado esperado

Tabla 42: Test de limpieza de formulario de sesión de asignatura en sección de administrador

6.1.22. Test de cierre de sesión en sección de administrador

En este test es evaluada la funcionalidad de cierre de sesión por parte del administrador, una vez se encuentra dentro de su sección.

Nº	Descripción	Resultado esperado	Resultado obtenido
P68	El usuario administrador pulsa el botón de cierre de sesión	El sistema cierra la sesión del administrador y lo envía a la página de inicio de sesión	Resultado esperado
P69	Usuario administrador rellena el formulario con la información de la sesión de asignatura	El sistema refleja el cambio	Resultado esperado
P70	Usuario administrador pulsa el botón de limpieza de formulario	El sistema realiza una limpieza del formulario, dejando cada campo en su valor por defecto	Resultado esperado

Tabla 43: Test de cierre de sesión en sección de administrador

Capítulo 7

Conclusiones y trabajo a futuro

7. Conclusiones y enfoques de trabajo a futuro

En el presente capítulo son especificadas las conclusiones que conllevan el trabajo realizado, así como las posibles líneas de mejora de cara a su continuación en un nuevo desarrollo futuro.

7.1. Conclusiones

Tras el desarrollo del presente Trabajo de Fin de Grado, han sido considerados como cumplimentados los objetivos planteados en sus inicios (Ver 1.3). Es posible afirmar que los frutos de este trabajo han dado como resultado un correcto desarrollo de una aplicación web enfocada en la gestión sobre horarios, cumpliendo los requisitos establecidos inicialmente.

Durante el desarrollo del proyecto especificado, me ha sido posible aplicar los conocimientos adquiridos en las asignaturas cursadas durante estos seis años de mi carrera universitaria, en particular los conceptos adquiridos durante Servicios y Sistemas Web, donde comencé en el desarrollo de aplicación y servicios web, y aquellas enfocadas en la gestión de proyectos y las etapas de diseño de producto, tales como Fundamentos de Ingeniería de Software, Modelado de Software, Diseño de Software y Planificación y Gestión de Proyectos. También han sido fundamentales los conceptos adquiridos en Desarrollo Basado en Componentes y Servicios, enfocada en el desarrollo full – stack y en la creación de componentes.

Es fundamental y necesaria la puesta en consideración del resto de asignaturas de carrera, las cuales, a pesar de no ser mencionadas explícitamente, han supuesto las bases de mi conocimiento actual y de mis objetivos durante estos años.

El desarrollo del presente trabajo ha sido realizado de manera constante, continua y particionada, junto con la comunicación activa de mi tutor universitario D. Joaquín Adiego Rodríguez, fundamental en la verificación del adecuado progreso de este.

Por último, espero que este proyecto sirva de utilidad a las futuras generaciones de alumnos de esta facultad, teniendo firmeza de que hallarán maneras de mejorar y tomar como inspiración este trabajo y tecnologías empleadas para sus propios proyectos futuros y ambiciones personales.

7.2. Enfoques de trabajo futuro

A pesar de haber finalizado el presente proyecto de manera exitosa y satisfactoria, logrando implementar la funcionalidad requerida por el Product Owner y por el cliente, ante la limitación de tiempo impuesta por un proyecto de estas características, es posible continuar su desarrollo para convertir sus funcionalidades en un concepto aún más completo, convirtiéndose en una herramienta más versátil.

Relativo al trabajo a futuro, entre las formas de ampliar la funcionalidad se encuentran las siguientes:

- Permitir a todos los usuarios iniciar sesión en el sistema, con intención de almacenar sus horarios personalizados e información adicional en la nube.
- Introducir la personalización de estilos de interfaz para cada usuario.
- Implementar distintos formatos de búsqueda de asignaturas para su visualización detallada.

Bibliografía

Asana. (s.f.). ¿Qué es la metodología ágil? Visitado el 16 de junio de 2025

<https://asana.com/es/resources/agile-methodology>

Astah. (s.f.). Class diagrams – User guide. Visitado el 16 de junio de 2025

<https://astah.net/support/astah-pro/user-guide/class-diagrams/>

Astah. (s.f.). Sequence diagram – User guide. Visitado el 16 de junio de 2025

<https://astah.net/support/astah-pro/user-guide/sequence-diagram/>

Astah. (s.f.). Usecase diagram – User guide. Visitado el 16 de junio de 2025

<https://astah.net/support/astah-pro/user-guide/usecase-diagram/>

Atlassian. (s.f.). Guía sobre metodologías ágiles. Visitado el 16 de junio de 2025

<https://www.atlassian.com/es/agile>

Atlassian. (s.f.). Introducción a Jira. Visitado el 16 de junio de 2025

<https://www.atlassian.com/es/software/jira/guides/getting-started/introduction>

Codemotion. (2023). ¿Cómo funciona el renderizado en React? Visitado el 16 de junio de 2025

<https://www.codemotion.com/magazine/es/frontend-es/como-funciona-el-renderizado-en-react/>

ComponentSource. (s.f.). Astah UML – Overview. Visitado el 16 de junio de 2025

<https://www.componentsource.com/es/product/astah-uml/about>

DhiWise. (2023). Guide to creating engaging user experiences with React popups. Visitado el 16 de junio de 2025

<https://www.dhiwise.com/post/guide-to-creating-engaging-user-experiences-with-react-popups>

Figma. (s.f.). What is Figma? Visitado el 16 de junio de 2025

<https://www.geeksforgeeks.org/websites-apps/what-is-figma/>

Imagina Formación. (s.f.). ¿Qué es el Virtual DOM de React? Visitado el 16 de junio de 2025

<https://imaginaformacion.com/tutoriales/que-es-el-virtual-dom-de-react>

jquense. (s.f.). React Big Calendar – Examples. Visitado el 16 de junio de 2025

<https://jquense.github.io/react-big-calendar/examples/index.html?path=/story/about-big-calendar--page>

jquense. (s.f.). react-big-calendar. GitHub. Visitado el 16 de junio de 2025

<https://github.com/jquense/react-big-calendar>

Lucidchart. (s.f.). ¿Qué es UML? Visitado el 16 de junio de 2025

<https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>

MDN Web Docs. (s.f.). Modelo vista controlador (MVC). Mozilla. Visitado el 16 de junio de 2025

<https://developer.mozilla.org/es/docs/Glossary/MVC>

Meta Open Source. (s.f.). React. Visitado el 16 de junio de 2025

<https://opensource.fb.com/projects/react/>

Node.js. (s.f.). Acerca de Node.js. Visitado el 16 de junio de 2025

<https://nodejs.org/es/about/>

npm. (s.f.). react-draggable. Visitado el 16 de junio de 2025

<https://www.npmjs.com/package/react-draggable>

React. (s.f.). Hooks – useState [Documentación legacy]. Visitado el 16 de junio de 2025

<https://es.legacy.reactjs.org/docs/hooks-state.html>

React. (s.f.). State and lifecycle [Documentación legacy]. Visitado el 16 de junio de 2025

<https://es.legacy.reactjs.org/docs/state-and-lifecycle.html>

React. (s.f.). useEffect – React Docs. Visitado el 16 de junio de 2025

<https://react.dev/reference/react/useEffect>

Red Hat Developer. (2021). Context switching in operating systems. Visitado el 16 de junio de 2025

<https://www.redhat.com/en/technologies/linux-platforms/enterprise-linux/red-hat-enterprise-linux-for-developers>

Visual Studio Code. (s.f.). Overview – Visual Studio Code. Microsoft Docs. Visitado el 16 de junio de 2025

<https://code.visualstudio.com/docs>

Anexo A

Guía de instalación

A.1. Contenido del repositorio

El repositorio del proyecto se encuentra en GitHub. El nombre del repositorio es “tfg-aplicacion-web-reactjs” (<https://github.com/FrankyCM/tfg-aplicacion-web-reactjs>).

Presenta los siguiente directorios:

- public: directorio en el que residen los ficheros .json oportunos, entre los que se encuentra el fichero de asignaturas, junto con otros elementos como imágenes.
- src: directorio en el que residen los componentes de la aplicación web, junto con sus hojas de estilos “.css”.

A.2. Instalación

En esta sección se presenta el procedimiento de instalación de las herramientas necesarias para la ejecución de la aplicación web. En relación con la descarga del código fuente, es necesario acceder al repositorio donde se ubica y descomprimir el archivo .zip con el contenido de dicho código.

El conjunto de pasos de instalación será enfocado en un entorno de ejecución con sistema operativo Windows.

A.2.1. Instalación de NVM

En este apartado se explica el procedimiento de instalación de Node.js y npm, paso a paso, siendo esenciales para el funcionamiento de la aplicación y de React, así como la instalación de las dependencias necesarias para el funcionamiento conjunto del proyecto.

Toda este proceso de instalación parte de la premisa de que el usuario cuenta con un editor de código cuyo entorno se haya previamente configurado.

1. Descargar e instalar Node.js desde su sitio web oficial: <https://nodejs.org/es>. Es recomendable descargar su última versión LTS soportada (Long – Term Support).
2. Revisar la correcta instalación de Node.js y npm, abriendo una terminal y ejecutando los siguientes comandos:



```
npm -v  
node -v
```

Figura 32: Comandos de verificación de versiones

A.2.2. Ejecución del proyecto

Tras la instalación y configuración requeridas previamente para el correcto funcionamiento de la aplicación web, se procede a explicar su procedimiento de ejecución.

1. Abrir en una terminal de comandos el directorio raíz del proyecto.

2. Descargar las dependencias reflejadas en el archivo package.json, ejecutando el siguiente comando:

```
npm install
```

Figura 33: comando de instalación de Node Package Manager

3. En el mismo directorio, ejecutar el siguiente comando para inicial la aplicación:

```
npm run dev
```

Figura 34: comando de ejecución de proyecto

4. Tras unos segundos, en la terminal se informará al usuario que la aplicación web se encuentra ejecutándose en el siguiente puerto, siendo necesario pegar la URL en un navegador para comenzar la interacción con la app:

```
VITE v6.2.3 ready in 316 ms
  Local: http://localhost:5173/
  Network: use --host to expose
  press h + enter to show help
```

Figura 35: Retroalimentación de ejecución

Anexo B

Manual de usuario

En este apartado será descrito el proceso de visualización y modificación de horarios, así como la creación, modificación o borrado de asignaturas, enfocado desde el punto de vista de un usuario real y final.

El objetivo de este apartado trata de servir como manual de usuario para todo aquel que desee interactuar con la aplicación.

B.1. Sección de inicio

Tras acceder a la aplicación web, se muestra una pantalla de inicio explicativa de las posibilidades ofrecidas al usuario.

En ella se muestra en su cabecera el logo de la Universidad de Valladolid, así como un breve dialogo introductorio dedicado al usuario y las dos opciones de visualización de horarios, cada una de ellas contando también con su propio diálogo. Al hacer click sobre el botón situado en la parte inferior de cada sección, el usuario será redirigido al apartado correspondiente, el cual cumple con la finalidad y características descritas previamente.

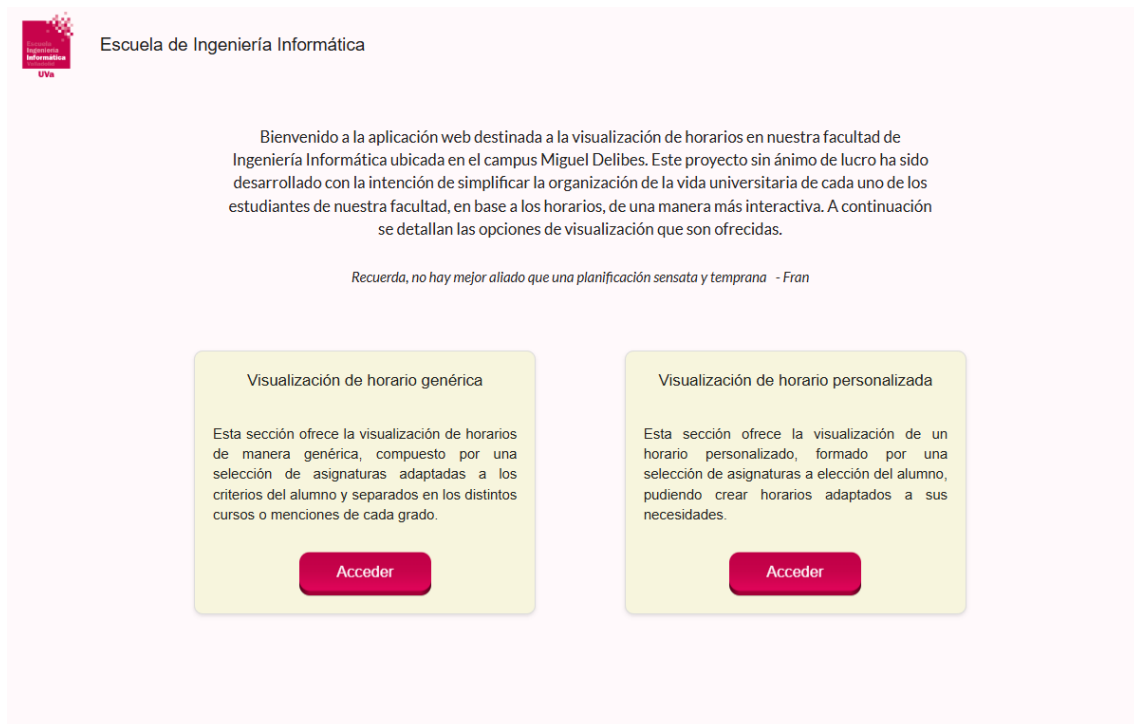


Figura 36: Manual de usuario - Sección introducción

B.2 Sección para visualización genérica

Si se pulsa el botón dedicado a este tipo de visualización en el apartado introductorio inicial, el usuario es redirigido a su sección correspondiente, la cual muestra inicialmente un menú desplazable con los diferentes grados contemplados como opciones junto con una plantilla de horario vacío, a la espera de que el usuario decida los criterios de visualización. Todo ello se muestra reflejado en la siguiente imagen:

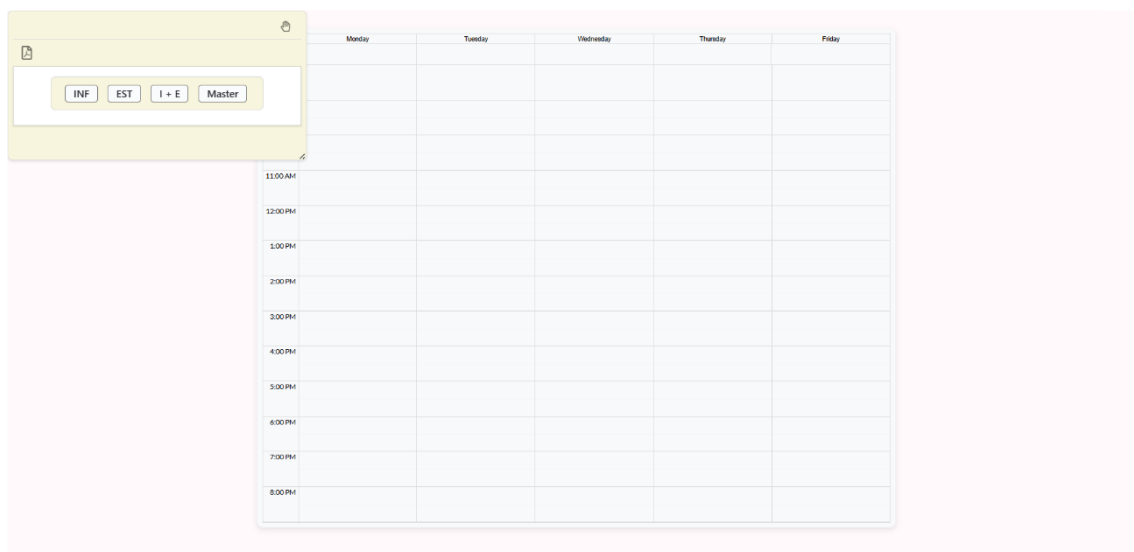


Figura 37: Manual de usuario - Visualización genérica - estándar

Una vez el usuario ha escogido los criterios, la aplicación web mostrará sobre la plantilla de horario el conjunto de asignaturas que cumplen dichas características y que constituyen dicho horario. A su vez, se ofrece al usuario la posibilidad de mostrar solamente asignaturas con sesiones teóricas o mostrar también las asignaturas con sesiones prácticas, tal como se muestra en las siguientes imágenes:

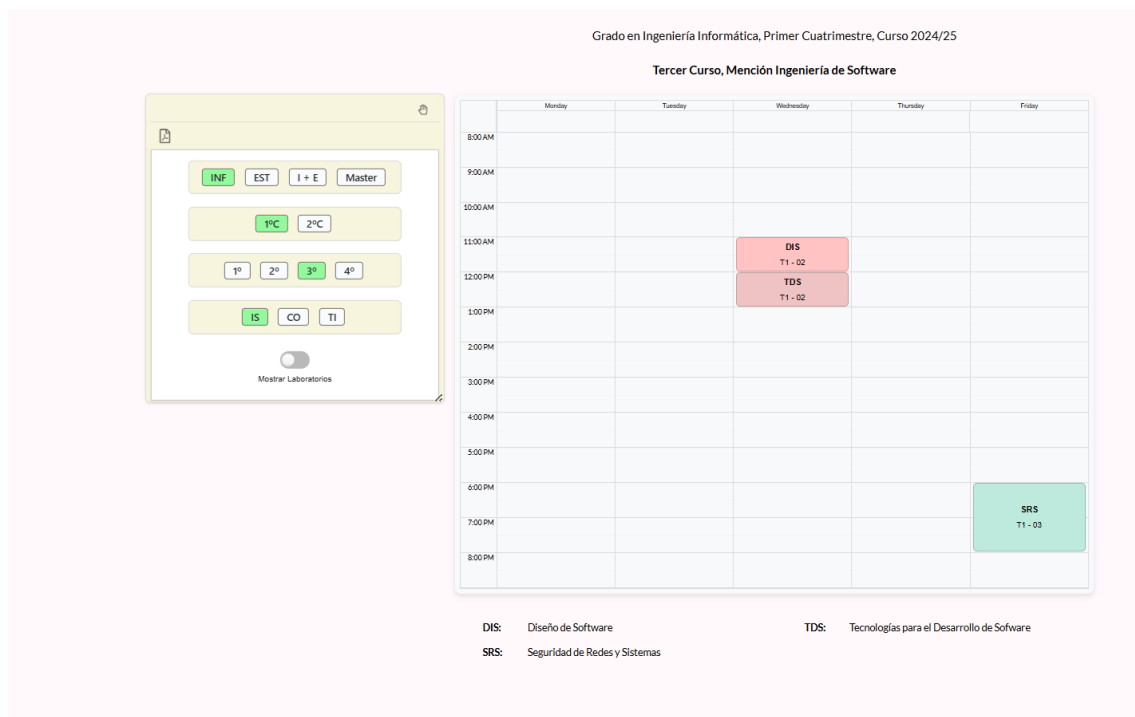


Figura 38: Manual de usuario - Visualización genérica - criterios

Junto a las opciones seleccionadas en la anterior imagen, es posible habilitar un aspecto adicional que permite la visualización de las clases prácticas de laboratorio.

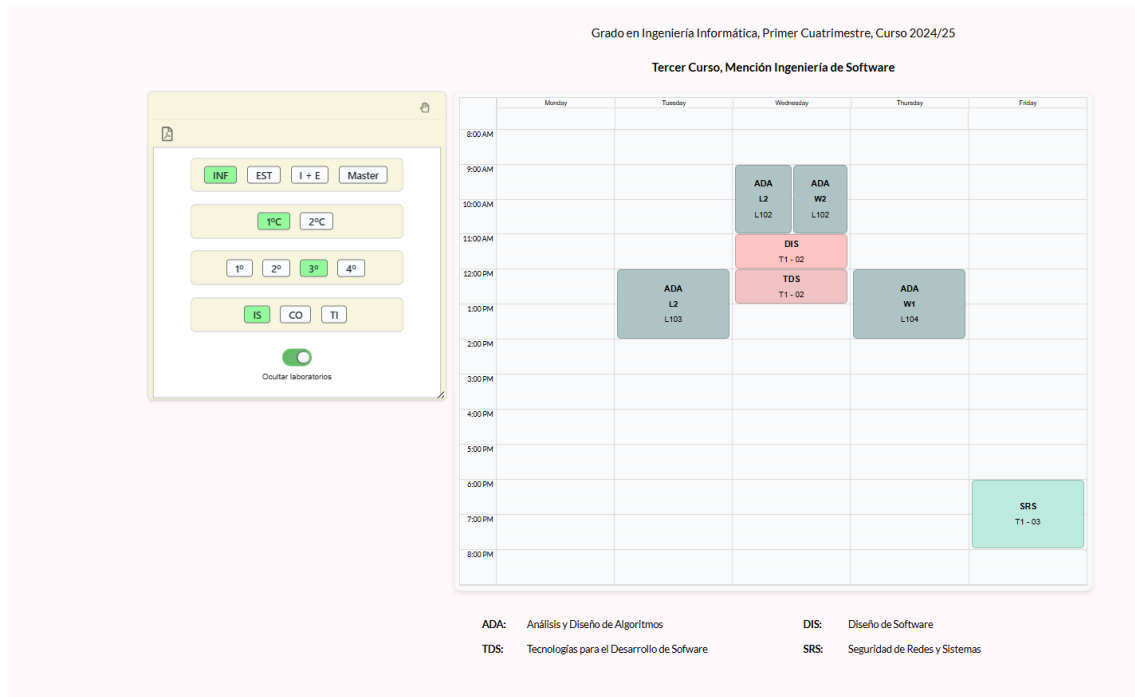


Figura 39: Manual de usuario - Visualización genérica - laboratorios

B.3 Sección para visualización personalizada

Si se pulsa el botón dedicado a este tipo de visualización en el apartado introductorio inicial, el usuario es redirigido a su sección correspondiente, la cual muestra inicialmente un menú desplazable con los diferentes grados contemplados como opciones junto con una plantilla de horario vacío, a la espera de que el usuario decida los criterios de visualización y sesiones de asignatura que desea mostrar en su propio horario. Todo ello se muestra reflejado en la siguiente imagen:

Grado en Ingeniería Informática, Primer Cuatrimestre, Curso 2024/25

Tercer Curso, Mención Ingeniería de Software

Monday Tuesday Wednesday Thursday Friday

8:00 AM

9:00 AM

10:00 AM

11:00 AM

12:00 PM

1:00 PM

2:00 PM

3:00 PM

4:00 PM

5:00 PM

6:00 PM

7:00 PM

INF EST I + E Master

1ºC 2ºC

1º 2º 3º 4º

Menciones de tercer curso

IS CO TI

DIS - T1 - IS

DIS - T1 - IS

TDS - T1 - IS

SRS - T1 - IS

Figura 40: Manual de usuario - Visualización personalizada - estándar

A su vez, también se permite al usuario elegir entre sesiones prácticas de laboratorio, en caso de habilitar la opción correspondiente, tal cómo se muestra en la siguiente imagen:

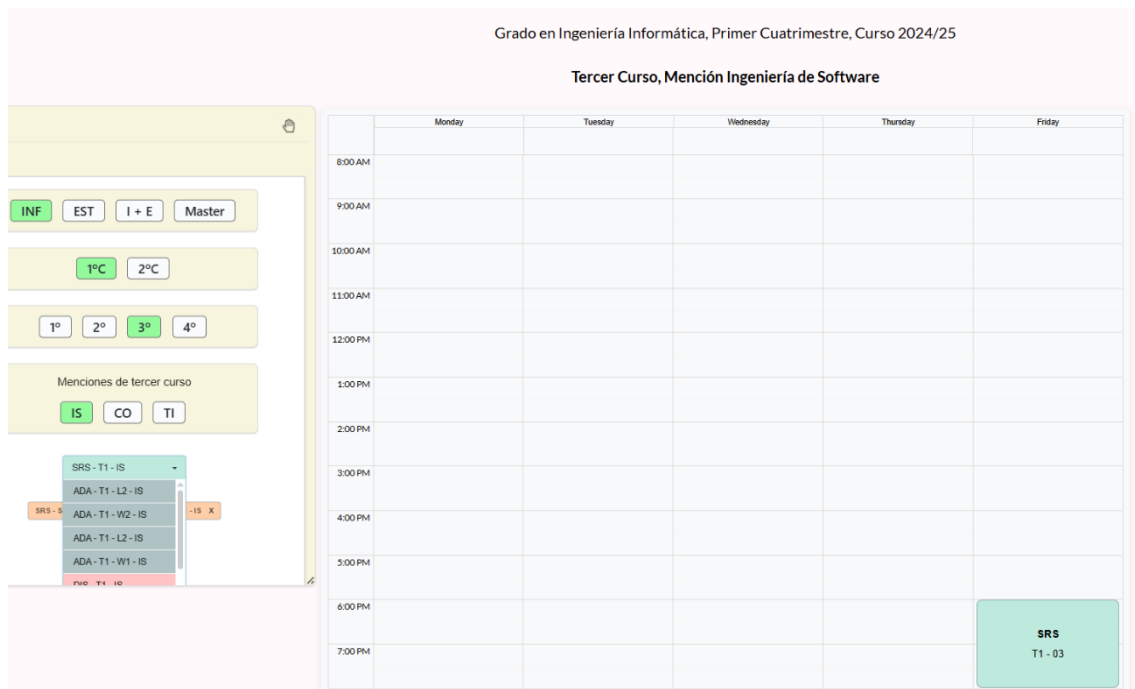


Figura 41: Manual de usuario - Visualización personalizada - criterios, asignaturas

B.4 Pop – ups de asignaturas

En cada una de ambas secciones, si el usuario desea acceder a información adicional sobre una sesión de asignatura, se ofrece la posibilidad de conocer su grupo y aula de laboratorio o teoría, así como el profesor ocupado de impartirla. Un pop-up se trata de un componente sencillo que permite la visualización de contenido adicional y la creación de “modales” y menús (DhiWise, 2023).

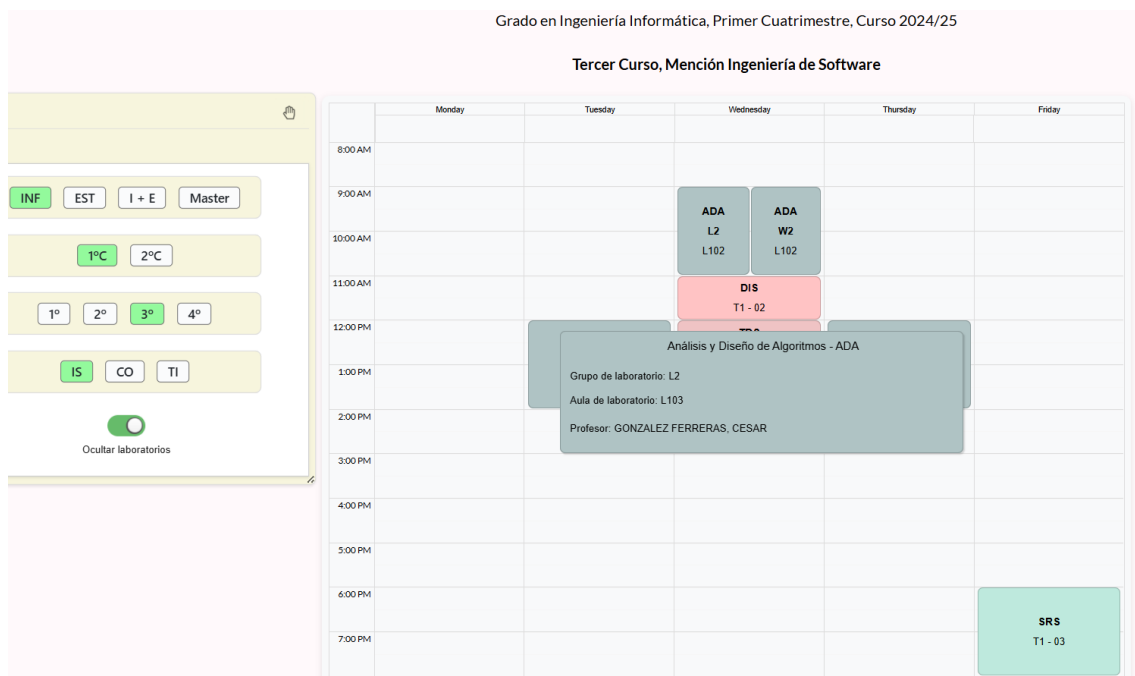


Figura 42: Manual de usuario - Visualización genérica - pop-ups

B.5 Información adicional de asignaturas

A su vez, si el usuario hace click sobre la sesión teórica o práctica de una de las asignaturas del horario, se mostrará por pantalla un cuadro con información adicional sobre el grupo y aula de dicha sesión, así como un mapa del plano del edificio y sus aulas ubicadas en él, orientativo para el usuario. Dicho mapa puede ser mostrado u ocultado a través de la pulsación del botón situado en su parte superior.

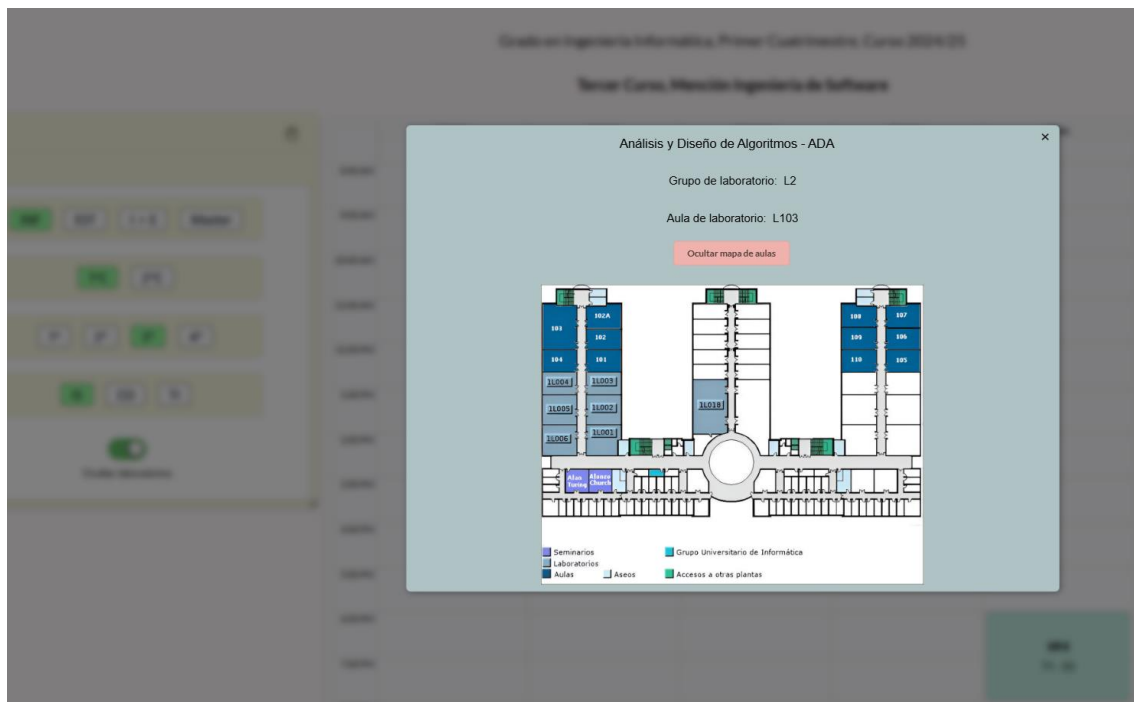


Figura 43: Manual de usuario - Visualización genérica - información adicional

B.6 Inicio de sesión

A su vez, la aplicación web cuenta con una sección dedicada al inicio de sesión .

Se trata de un cuestionario en la que se solicitan las credenciales a introducir por parte del administrador de los horarios del sistema, tal cómo se muestra en la siguiente imagen:

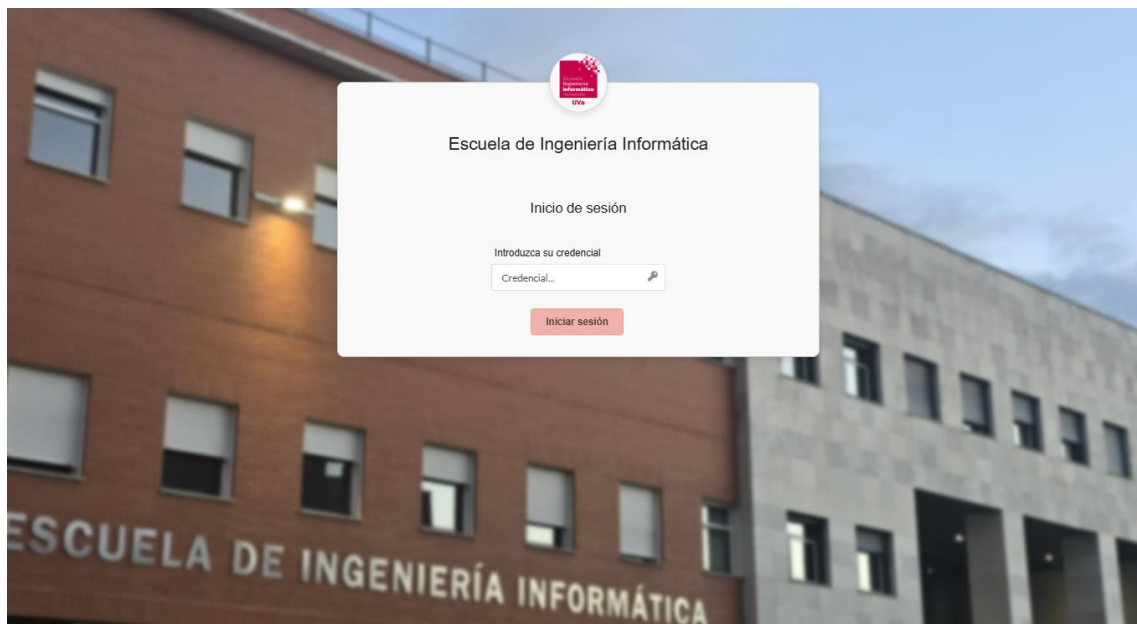


Figura 44: Manual de usuario - Inicio de sesión

Esta sección cuenta con un control de acceso en caso de proporcionar credenciales erróneas, tal cómo se muestra en la siguiente imagen:

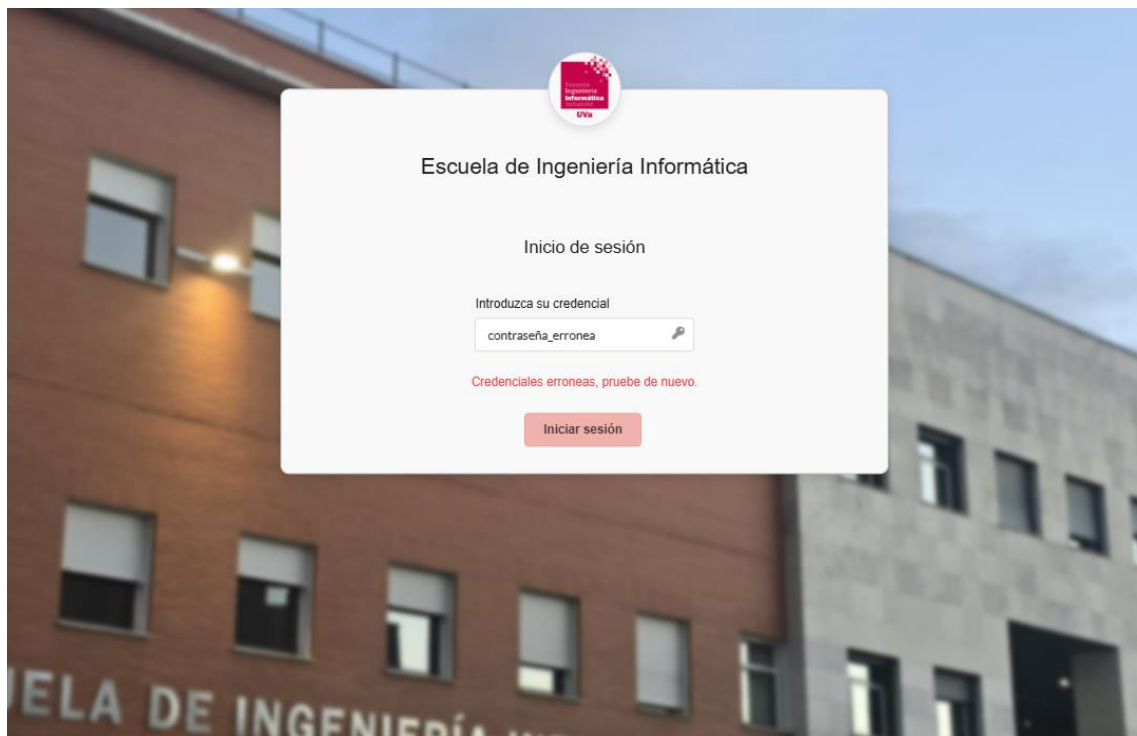


Figura 45: Manual de usuario - Inicio de sesión - credenciales erróneas

B.7 Cierre de sesión

Una vez dentro del sistema, si el administrador desea cerrar su sesión, solamente el necesario hacer click en el botón situado en la parte superior derecha, tal cómo se muestra en la siguiente imagen:

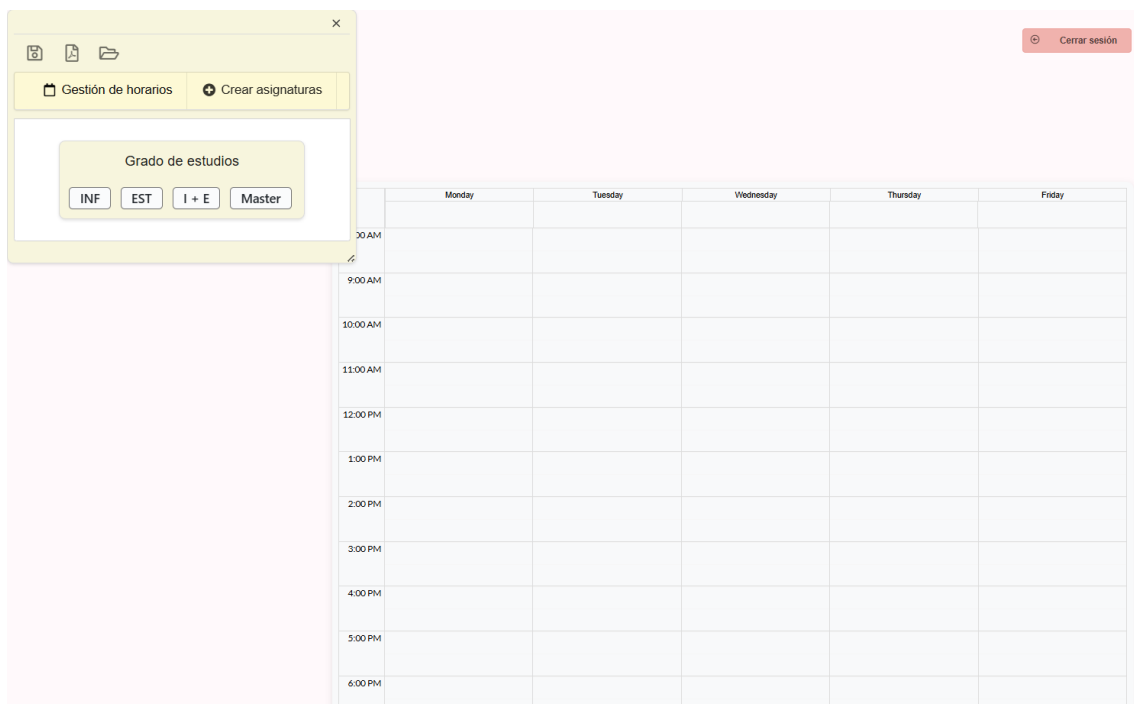


Figura 46: Manual de usuario - Cierre de sesión

B.8 Modificación de horario y asignaturas

En la sección dedicada al administrador es posible modificar un horario a través del uso de la tecnología Drag & Drop para desplazar sesiones teóricas o prácticas, tal cómo se muestra en la siguiente imagen:

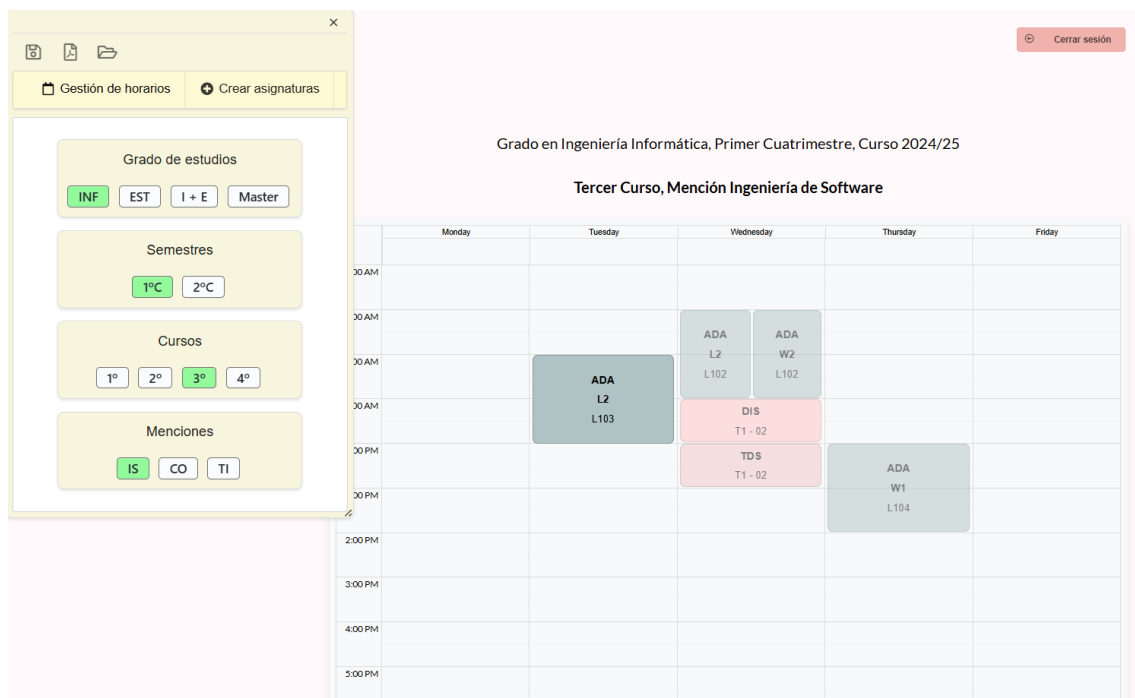


Figura 47: Manual de usuario - Modificación de asignatura - Drag & Drop

A su vez, también es posible redimensionar la sesión teórica o práctica elegida, modificando su duración en término de horas, tal cómo es posible visualizar en la siguiente imagen:

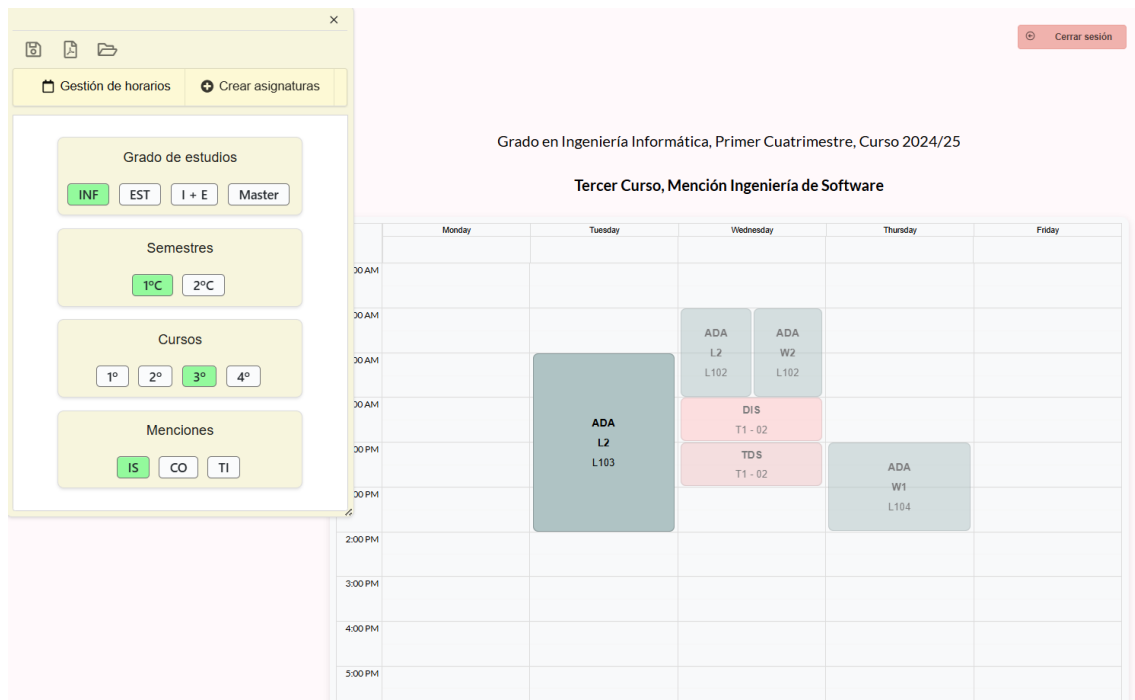


Figura 48: Manual de usuario - Modificación de asignatura - Resize

B.9 Creación de asignaturas

En la sección dedicada al administrador también se ofrece la posibilidad de crear una nueva sesión de asignatura, siendo necesario completar un formulario con características básicas sobre la misma. A su vez, también se permite la limpieza del formulario, en caso de ser necesario.

Ambos aspectos son mostrados en la siguiente imagen:

Figura 49: Manual de usuario - Creación de asignatura

B.10 Modificación de asignatura

En esta sección destinada al administrador, también es posible modificar una sesión de asignatura, siendo necesario modificar los apartados deseados del siguiente formulario, el cual es mostrado en la siguiente imagen:

Formulario de modificación de asignatura: **Análisis y Diseño de Algoritmos - ADA**

Código	Siglas	Nombre completo	Día	Hora de inicio	Color
45678	ADA	Análisis y Diseño de Algoritmos	M	12:00	#BBE9FF

Semestre	Grupo	Subgrupo	Tipo	Duración	Aula
1º	Grupo 1	L2	T	2	L104

Estudios donde se imparte, cursos y optatividad					
GII-IS	GII-TI	GII-CO	EST	INDat	Master
3º	-	-	-	-	-

Profesores	
Docente asignado	GONZALEZ FERRERAS, CESAR

Borrar asignatura **Modificar asignatura**

Figura 50: Manual de usuario - Modificación de asignatura - formulario

B.11 Guardado, impresión en formato PDF y apertura de fichero

Destinados a su uso por parte del administrador, esta sección cuenta a su vez con las opciones de guardado, impresión en formato PDF y apertura de horarios, presionando sobre los iconos mostrados en la siguiente imagen y situados en la parte superior del menú desplazable lateral, ordenados de izquierda a derecha, respectivamente:

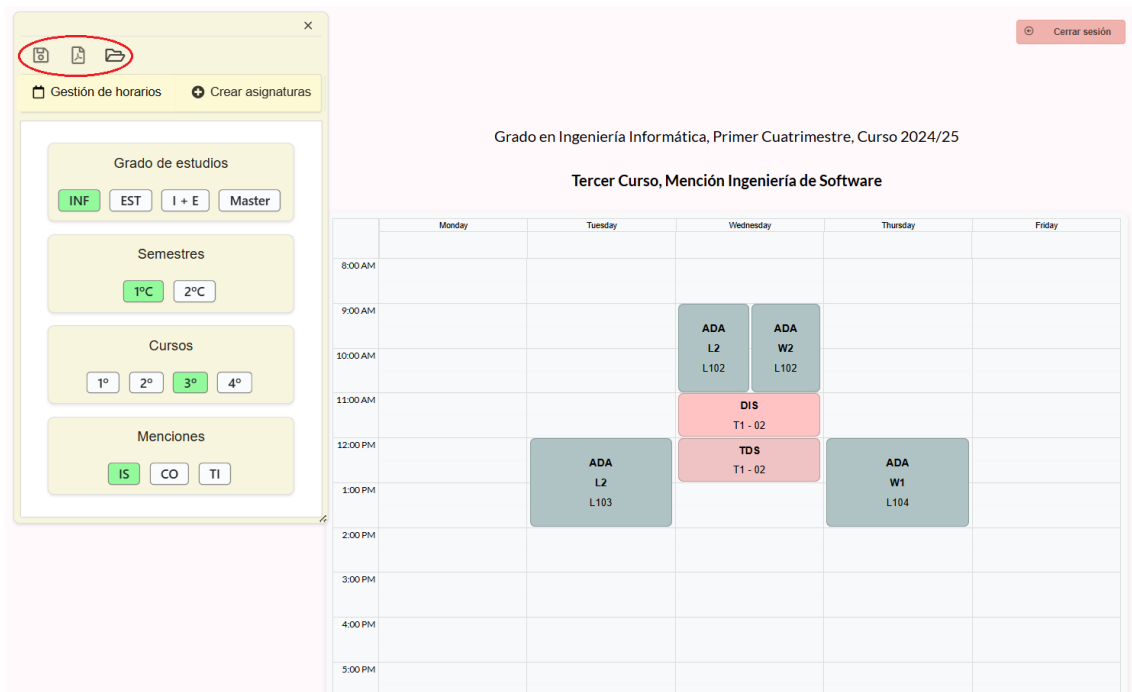


Figura 51: Manual de usuario - Guardado, Impresión PDF y apertura de fichero