



---

**Universidad de Valladolid**

**Escuela de Ingeniería Informática**

**TRABAJO FIN DE GRADO**

Grado en Ingeniería Informática  
Mención Ingeniería de Software

**Calculadora personal online  
de horarios para los alumnos**

Autor:  
**Miguel Rojo Revenga**





---

**Universidad de Valladolid**

**Escuela de Ingeniería Informática**

**TRABAJO FIN DE GRADO**

Grado en Ingeniería Informática  
Mención Ingeniería de Software

**Calculadora personal online  
de horarios para los alumnos**

Autor:

**Miguel Rojo Revenga**

Tutor:

**Dr. Joaquín Adiego Rodríguez**



---

*A quienes creyeron.*



# Agradecimientos

A mi familia, porque sin su ayuda y confianza no habría llegado hasta aquí.

A mi pareja, Noelia, porque sin ella esto no sería posible.

A los profesores y profesoras que me han enseñado tanto a lo largo de estos años en la universidad.

A Yania y a Carmen, por haber confiado en mí y descubrirme el maravilloso mundo de la enseñanza, y por estar pendientes de que terminara este trabajo de manera continua.

A mi tutor Joaquín, por aguantar todas las idas y venidas de este proyecto y haberme ayudado y apoyado siempre.





# Resumen

Los estudiantes universitarios se matriculan todos los años en diferentes asignaturas, eligiendo entre los grupos de clase que se ofertan. A la hora de realizar esa elección muchas veces no se analizan posibles coincidencias horarias, dando lugar a situaciones en las que los alumnos dejan de asistir a determinadas horas de clase desde el principio del curso. Esta ausencia a las clases presenciales es uno de los factores que influyen negativamente en el rendimiento académico.

El objetivo de este trabajo es desarrollar una aplicación web que sirva de herramienta para evitar ese abandono de asignaturas. La herramienta se encargará de calcular todas las combinaciones posibles y los estudiantes obtendrán un horario con el menor número de horas solapadas y lo más compacto posible.



# Abstract

University students enroll every year in different subjects, choosing among the class groups that are offered. By the time they make that choice they usually don't analyze possible coincidences, leading to situations in which students stop attending certain hours of class from the beginning of the course. This absence to the classroom is one of the factors that negatively influence academic performance.

The objective of this project is to develop a web application that serves as a tool to avoid that abandonment of subjects. The tool will be responsible for calculating all possible combinations and students will get a schedule with the least number of overlapping hours and as compact as possible.



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Contexto . . . . .	1
1.2. Motivación . . . . .	1
1.3. Objetivos . . . . .	2
1.4. Estructura del documento . . . . .	2
<b>2. Contexto</b>	<b>5</b>
2.1. Asistencia a clase y rendimiento académico . . . . .	5
2.2. Aplicaciones existentes . . . . .	8
2.2.1. Aplicaciones para alumnos . . . . .	8
2.2.2. Aplicaciones para centros educativos . . . . .	9
2.3. Desarrollo web . . . . .	14
<b>3. Planificación</b>	<b>17</b>
3.1. Propósito y alcance . . . . .	17
3.2. Metodología . . . . .	17
3.2.1. Scrum . . . . .	18
3.3. Plan temporal . . . . .	22
3.3.1. Product Backlog . . . . .	23
3.4. Gestión de riesgos . . . . .	24
3.5. Gestión de configuraciones . . . . .	24
3.6. Seguimiento . . . . .	24
3.6.1. Sprint Planning y Sprint Reviews . . . . .	24
<b>4. Análisis del sistema</b>	<b>33</b>
4.1. Especificación de la aplicación . . . . .	33
4.2. Modelo de dominio . . . . .	37
4.3. Historias de usuario . . . . .	39
4.3.1. Tareas . . . . .	43

<b>5. Diseño del sistema</b>	<b>47</b>
5.1. Arquitectura . . . . .	47
5.1.1. Visión global . . . . .	47
5.1.2. Spring Web MVC . . . . .	50
5.2. Patrones de Diseño . . . . .	52
5.2.1. Modelo Vista Controlador . . . . .	52
5.2.2. Data Access Object . . . . .	53
5.2.3. Template . . . . .	54
5.3. Despliegue . . . . .	55
5.4. Diseño de la base de datos . . . . .	57
5.4.1. Modelo Entidad-Relación . . . . .	57
5.4.2. Modelo Relacional . . . . .	57
<b>6. Implementación</b>	<b>61</b>
6.1. Herramientas y tecnologías utilizadas . . . . .	61
6.2. Interfaz de usuario . . . . .	64
6.3. Código de la aplicación . . . . .	68
6.4. Cálculo de horarios . . . . .	71
6.5. Base de datos . . . . .	72
6.6. Pruebas . . . . .	73
<b>7. Conclusiones</b>	<b>83</b>
7.1. Conclusiones . . . . .	83
7.2. Trabajo futuro . . . . .	83
7.2.1. En la misma aplicación . . . . .	83
7.2.2. En la línea de aplicaciones complementarias . . . . .	84
<b>Referencias</b>	<b>85</b>
<b>A. Manual de usuario</b>	<b>87</b>
A.1. Despliegue . . . . .	87
A.1.1. Requisitos . . . . .	87
A.1.2. Heroku . . . . .	87
A.1.3. Otras formas de despliegue . . . . .	88
A.2. Manual de usuario . . . . .	88
A.2.1. Introducción . . . . .	88
A.2.2. Administrador . . . . .	89
A.2.3. Estudiante . . . . .	96
<b>B. Contenido del CD</b>	<b>101</b>

# Índice de figuras

2.1. Evolución del número de grupos en algunas asignaturas del Grado en Ingeniería Informática de la Universidad de Valladolid. . . . .	7
2.2. Captura de pantalla del menú principal de la herramienta Agenda del estudiante. . . . .	8
2.3. Captura de pantalla del menú de gestión de calendario de la herramienta Agenda del estudiante. . . . .	9
2.4. Captura de pantalla de la vista de un horario de la herramienta Agenda del estudiante. . . . .	9
2.5. Captura de pantalla del menú principal de la herramienta ascHorarios. . . . .	10
2.6. Captura de pantalla del menú de gestión de profesores de la herramienta ascHorarios. . . . .	11
2.7. Captura de pantalla de la vista de un horario de la herramienta ascHorarios. . . . .	11
2.8. Captura de pantalla de la vista de un horario con restricciones ignoradas de la herramienta ascHorarios. . . . .	12
2.9. Captura de pantalla del menú principal de la herramienta FET. . . . .	12
2.10. Captura de pantalla del menú para añadir una actividad de la herramienta FET. . . . .	13
2.11. Captura de pantalla del menú de restricciones aplicables a alumnos de la herramienta FET. . . . .	13
2.12. Captura de pantalla de la vista de horarios de profesores de la herramienta FET. . . . .	14
2.13. Captura de pantalla de la vista de horarios de estudiantes de la herramienta FET. . . . .	14
3.1. Diagrama del ciclo de vida de un proyecto en Scrum. . . . .	19
3.2. <i>Burndown</i> del primer <i>sprint</i> . . . . .	26
3.3. <i>Burndown</i> del segundo <i>sprint</i> . . . . .	27
3.4. <i>Burndown</i> del tercer <i>sprint</i> . . . . .	28
3.5. <i>Burndown</i> del cuarto <i>sprint</i> . . . . .	29
3.6. <i>Burndown</i> del quinto <i>sprint</i> . . . . .	30
3.7. Desviaciones sobre las estimaciones realizadas. . . . .	31
4.1. Horario del segundo cuatrimestre del curso 2018/2019 del Grado en Ingeniería Informática de la Universidad de Valladolid, mención Ingeniería del Software, tercer curso. . . . .	35
4.2. Horario del segundo cuatrimestre del curso 2018/2019 del Grado en Ingeniería Informática de la Universidad de Valladolid, mención Tecnologías de la Información, tercer curso. . . . .	36
4.3. Horario del segundo cuatrimestre del curso 2018/2019 del Grado en Ingeniería Informática de la Universidad de Valladolid, mención Computación, tercer curso. . . . .	37
4.4. Modelo de dominio. . . . .	38
5.1. Representación simplificada de una arquitectura de 3 capas. . . . .	47

---

5.2. Arquitectura en tres capas de la aplicación. . . . .	48
5.3. Capa de presentación de la aplicación. . . . .	49
5.4. Capa de negocio de la aplicación. . . . .	50
5.5. Capa de persistencia de la aplicación. . . . .	50
5.6. Stack de Spring Framework. . . . .	51
5.7. Flujo de trabajo del procesamiento de peticiones en Spring Web MVC a alto nivel. . . . .	52
5.8. Relación de componentes en el patrón Modelo-Vista-Controlador. . . . .	53
5.9. Ejemplo de diagrama de patrón DAO. . . . .	54
5.10. Ejemplo de funcionamiento del patrón Template para Spring JdbcTemplate. . . . .	55
5.11. Diagrama de despliegue remoto de la aplicación. . . . .	56
5.12. Diagrama de despliegue local de la aplicación. . . . .	56
5.13. Modelo Entidad-Relación de la aplicación. . . . .	57
5.14. Modelo relacional de la base de datos. . . . .	58
6.1. Interfaz del usuario desde un navegador Firefox. . . . .	64
6.2. Interfaz del usuario desde un iPad y un navegador Safari. . . . .	65
6.3. Interfaz del usuario desde un navegador de un smartphone Android. . . . .	65
6.4. Interfaz del usuario de la pantalla horario reducida horizontalmente. . . . .	66
6.5. Ejemplo de panel utilizado para operaciones habituales. . . . .	67
6.6. Ejemplo de panel utilizado para operaciones de edición. . . . .	67
6.7. Ejemplo de panel utilizado para operaciones de borrado. . . . .	68
6.8. Código de ejemplo validación de campos correspondiente al formulario de registro. . . . .	69
6.9. Código de la búsqueda dinámica en la tabla de Asignaturas. . . . .	70
6.10. Código para obtener los horarios en formato PDF. . . . .	70
6.11. Ejemplo de cálculo del grado de compactación de un horario. . . . .	72
6.12. Script de creación de la base de datos. . . . .	73
A.1. Página de bienvenida de la aplicación. . . . .	88
A.2. Panel de administración de la aplicación. . . . .	89
A.3. Página de gestión de titulaciones de la aplicación. . . . .	90
A.4. Formulario de la página de edición de titulación. . . . .	90
A.5. Formulario de la página de borrado de titulación. . . . .	91
A.6. Formulario de la página de vista de titulación. . . . .	91
A.7. Página de gestión de asignaturas. . . . .	92
A.8. Formulario de la página de edición de asignatura. . . . .	92
A.9. Formulario de la página de borrado de asignatura. . . . .	93
A.10. Formulario de la página de vista de asignatura. . . . .	93



---

A.11.Página de gestión de grupos de la aplicación. . . . .	94
A.12.Formulario de la página de edición de grupo. . . . .	94
A.13.Formulario de la página de borrado de grupo. . . . .	95
A.14.Formulario de la página de vista de grupo. . . . .	95
A.15.Panel del estudiante en la aplicación. . . . .	96
A.16.Formulario de selección de titulación y periodo. . . . .	96
A.17.Formulario de selección de asignaturas para el cálculo de horarios. . . . .	97
A.18.Página de horario generado. . . . .	97
A.19.Operaciones adicionales de la página de horario. . . . .	98
A.20.Página de horarios guardados del estudiante. . . . .	99
A.21.Formulario de ajustes de perfil. . . . .	99



# Índice de tablas

2.1. Tabla resumen de la evolución del número de grupos en algunas asignaturas del Grado en Ingeniería Informática de la Universidad de Valladolid. . . . .	6
3.1. Distribución del horario de trabajo. . . . .	19
3.2. Product Backlog del proyecto. . . . .	23
3.3. Planificación temporal del proyecto. . . . .	23
3.4. <i>Sprint Backlog</i> e información de seguimiento del primer <i>sprint</i> . . . . .	25
3.5. <i>Sprint Backlog</i> e información de seguimiento del segundo <i>sprint</i> . . . . .	27
3.6. <i>Sprint Backlog</i> e información de seguimiento del tercer <i>sprint</i> . . . . .	28
3.7. <i>Sprint Backlog</i> e información de seguimiento del cuarto <i>sprint</i> . . . . .	29
3.8. <i>Sprint Backlog</i> e información de seguimiento del quinto <i>sprint</i> . . . . .	30
3.9. <i>Sprint Backlog</i> e información de seguimiento del cuarto <i>sprint</i> . . . . .	31
4.1. Modelo de tarjeta para las historias de usuario. . . . .	40
4.2. Historia de usuario HU01. . . . .	41
4.3. Historia de usuario HU02. . . . .	41
4.4. Historia de usuario HU03. . . . .	41
4.5. Historia de usuario HU04. . . . .	42
4.6. Historia de usuario HU05. . . . .	42
4.7. Historia de usuario HU06. . . . .	42
4.8. Historia de usuario HU07. . . . .	43
6.1. Información del servidor Tomcat utilizado. . . . .	62
6.2. Información del servidor Tomcat utilizado. . . . .	63
6.3. Plantilla utilizada para documentar las pruebas realizadas. . . . .	74



# Capítulo 1

## Introducción

### 1.1. Contexto

Hoy en día no es raro encontrar algunas titulaciones en la Universidad en las que en ciertas asignaturas hay una cantidad elevada de estudiantes matriculados. Independientemente de las causas de esta situación, muchas facultades, centros y escuelas universitarias se ven obligadas a crear múltiples grupos, tanto de teoría como de prácticas y/o laboratorios a la hora de configurar los horarios lectivos de cada curso y cuatrimestre.

Esta situación se debe, entre otros factores, a que muchos de los alumnos y alumnas que se matriculan en dichas asignaturas lo hacen porque no las superaron en cursos pasados y, además de en estas, se matriculan de otras de cursos diferentes.

El resultado es que muchos de esos estudiantes que se matriculan en asignaturas de diferentes cursos o que combinan grupos entre asignaturas del mismo curso, lo hacen renunciando desde el primer día a la asistencia a dichas clases sin detenerse a evaluar las alternativas que tienen para diseñar un horario en el que no se produzcan solapamientos de horas de clase o en el que los solapamientos sean mínimos.

El hecho de no asistir a horas de clase se ha relacionado con el fracaso escolar y el rendimiento académico, demostrando que la asistencia regular a clase favorece la obtención de mejores resultados por parte de los estudiantes.

### 1.2. Motivación

La motivación para desarrollar este proyecto surge de la experiencia personal diseñando múltiples combinaciones de horarios de clase antes de realizar la matrícula en mis últimos años como estudiante del Grado de Ingeniería Informática en la Universidad de Valladolid, en los que he compatibilizado estudios en asignaturas de diferentes cursos con otras actividades. De esa experiencia personal también he podido llegar a la conclusión de que la mía no era una situación excepcional, y que había compañeros y compañeras que lidiaban con la misma problemática en cada período de matrícula.

Además, a todo ello se le une el hecho de que no se ha encontrado en el mercado ninguna herramienta que ofrezca la automatización en el proceso de elección de grupos que aquí se pretende desarrollar.

---

## 1.3. Objetivos

El principal objetivo del presente Trabajo de Fin de Grado es desarrollar una aplicación web que permita a los usuarios obtener de manera automática los horarios de clase en los que se produzcan menos solapamientos de horas entre clases de diferentes asignaturas.

A partir de ese objetivo principal surgen otros de carácter secundario, como son:

- Ser capaz de analizar un problema real y de diseñar una solución óptima.
- Aprender a planificar un proyecto de desarrollo de software.
- Familiarizarse con la metodología de trabajo Scrum.
- Afianzar conocimientos de desarrollo en Java y aprender a utilizar uno de los frameworks existentes asociados a dicho lenguaje.
- Ampliar los conocimientos adquiridos de desarrollo web para diseñar una aplicación cuya interaz sea fácil de usar y su diseño sea adaptativo.

Por último se busca alcanzar los objetivos fijados en el apartado 3 del Proyecto Docente de Trabajo de Fin de Grado para la mención de Ingeniería del Software [1] y que son los siguientes:

- Buscar, ordenar, analizar y estructurar la información para desarrollar una aplicación informática.
- Trabajar asumiendo diferentes roles.
- Elaborar una memoria del proyecto desarrollado que incluya el contexto en el que se realiza, los objetivos, la metodología de trabajo, información sobre el proceso de análisis y diseño, conclusiones y líneas de trabajo futuras.
- En último término plantear y defender una presentación pública del trabajo realizado.

## 1.4. Estructura del documento

La estructura de la memoria, a partir de este apartado, se detalla a continuación.

En el segundo capítulo se presenta el contexto actual desde dos puntos de vista. Por un lado analizando la relación entre el rendimiento académico y la asistencia a clase, y por otro, presentando un breve análisis que se ha realizado sobre algunas de las aplicaciones que se pueden encontrar hoy en día y que permiten calcular horarios de manera automática. También se incluye una breve reseña sobre desarrollo web.

El tercer capítulo incluye todo lo relativo a la planificación del proyecto, la metodología utilizada y un apartado de seguimiento del proyecto.

En el cuarto capítulo se puede encontrar información sobre la fase de análisis del problema al que se trata de dar una solución y que constituye una de las partes más importantes de este trabajo.

El quinto capítulo hace referencia a la fase de diseño de la aplicación.

El sexto capítulo detalla la implementación de la aplicación, incluyendo las tecnologías utilizadas y las pruebas realizadas.

---

En el séptimo capítulo se pueden encontrar las conclusiones obtenidas como resultado de la realización del Trabajo Fin de Grado respectivamente y algunas líneas de trabajo futuro que se han planteado en relación al presente proyecto.

Por último se encuentran un capítulo de referencias en el que se incluyen todos los recursos consultados y siguiendo la notación IEEE, y los apéndices, que son los siguientes:

**Apéndice A** Manual de usuario.

**Apéndice B** Contenido del CD-ROM.





# Capítulo 2

## Contexto

En este capítulo se describirá el contexto actual en el que se plantea el desarrollo de una calculadora online de horarios para los alumnos. Se ha realizado primero un breve análisis sobre la relación entre asistencia a clase y el rendimiento académico prestando especial atención a las situaciones de solapamiento de clases. En una segunda parte del capítulo se han buscado, probado y analizado algunas de las herramientas que se han encontrado en el mercado y que guardan similitudes con la aplicación que se pretende desarrollar.

### 2.1. Asistencia a clase y rendimiento académico

Dado que uno de los objetivos marcados para el desarrollo de este trabajo era el de intentar facilitar a los alumnos la posibilidad de asistir al mayor número de clases posibles de las asignaturas que estén cursando, parece apropiado realizar un breve análisis de la asistencia a clase como factor a tener en cuenta en el rendimiento académico de los estudiantes.

Este tema se ha intentado explicar a través de diferentes estudios, analizando multitud de conjuntos de datos y atendiendo a diferentes variables. La asistencia a clase aparece una y otra vez como una de esas variables, y lo hace como una de las más relevantes en muchos casos.

Tal y como se explica en [2] el bajo rendimiento académico es un problema social cuyos efectos podrían mitigarse poniendo medios que redujesen el fracaso en la etapa universitaria. En dicha publicación se analiza el impacto de diferentes variables en el rendimiento académico universitario, y se llega a la conclusión de que para la aproximación planteada, la asistencia regular a las clases es una de las primeras variables en importancia, por delante de otras como por ejemplo el nivel de satisfacción de los estudiantes ante la carrera elegida o el rendimiento académico en cursos inferiores.

En [3] se presenta la idea de que para una implantación exitosa del EEES será necesario el desarrollo de estrategias de motivación para que aumente el porcentaje de asistencia a las clases. Todo ello para seguir las asignaturas a través de una evaluación continuada, siguiendo la estructura fijada en el EEES. En sus conclusiones constatan que la asistencia a clase influye en los resultados obtenidos por los estudiantes.

Se puede concluir hasta este punto que el absentismo de los estudiantes universitarios es un problema relevante en la sociedad actual. Los motivos de ese absentismo son muy diversos y en [4] se plantean algunos como pueden ser que se considera que la clase presencial no aporta valor, el/la profesor/a de la asignatura, la proximidad de alguna entrega o examen, la posibilidad de seguir la asignatura mediante el uso de diapositivas y/o libros o el solapamiento de horarios con otras asignaturas.

Para el presente trabajo nos centraremos en el último motivo mencionado, el solapamiento de horarios con otras asignaturas. En el estudio citado se clasifica a los estudiantes de acuerdo a su grado de asistencia a las clases en 3 franjas: los que más asisten, los que asisten más o menos a la mitad de clases y los que asisten a menos clases. Aunque en los dos primeros grupos el solapamiento de horarios representa como máximo un 5% de las respuestas dadas como motivo para faltar a clase, en el tercer grupo de estudiantes, los que asisten a menos clases, los porcentajes crecen hasta un 25% en las clases de teoría y un 12,5% en las clases de problemas. Una conclusión posible para esta situación es que esos alumnos precisamente faltan a tantas clases porque el solapamiento de horarios que tienen se prolonga a lo largo del período lectivo y no se trata de solapamientos puntuales.

Otra de las referencias consultadas es [5], donde se analizan los motivos del absentismo y apunta al solapamiento de horarios con otras asignaturas como uno de los más problemáticos y propone como posible solución la modificación de horarios de aquellas asignaturas que produzcan mayores solapamientos. Si bien se trata de una solución viable, también es cierto que esto probablemente derivase en la aparición de solapamientos en otras zonas horarias o que se tuvieran que trasladar determinadas horas a las primeras y últimas franjas horarias, lo que seguramente terminaría provocando absentismo debido a que las clases se celebran demasiado pronto o demasiado tarde, motivo que aparece tanto en este estudio como en alguna de las referencias citadas anteriormente, como es el caso de [4].

Por último cabe destacar que existen publicaciones en las que el solapamiento de horarios llega a ser considerado como el motivo más importante para no asistir a clase, como en [6]. En este caso es considerado por encima de motivos como ser repetidor/a o la existencia de material complementario que hacen la asistencia menos útil.

Además, también se han encontrado estudios más generalistas, como por ejemplo [7], en los que el solapamiento se incluye como motivo dentro de otras categorías más amplias, donde se estipula que más de la mitad de los alumnos que no asisten a clase por motivos de organización, entre los que considera el exceso de clases, horarios muy cargados, solapamiento de clases, clases aisladas en ciertos días, etc.

Determinadas asignaturas tienen tasas de abandono elevado y esto hace que en muchas de ellas los resultados obtenidos por los estudiantes sean malos. Esto provoca unos porcentajes bajos de superación de dichas asignaturas y a su vez que su docencia tenga que desdoblarse en diferentes grupos para cursos posteriores. En los estudios del Grado en Ingeniería Informática de la Universidad de Valladolid se puede apreciar esta tendencia, aunque también han influido en ella otros factores como el aumento de plazas de acceso a la titulación. A continuación se muestran la Tabla 2.1 y la Figura 2.1, donde se puede ver la evolución del número de grupos de algunas asignaturas de los primeros cursos.

Asignatura	2013-2014		2014-2015		2015-2016		2016-2017		2017-2018		2018-2019	
	T	L	T	L	T	L	T	L	T	L	T	L
Fundamentos de Matemáticas (FMAT)	2	6	3	9	3	9	3	9	3	12	4	12
Fundamentos de Organización de Empresas (FOE)	1	2	3	8	3	8	3	8	3	8	3	8
Matemática Discreta (MDIS)	2	6	3	8	3	8	3	8	3	8	3	8
Sistemas Digitales (SDIG)	2	5	3	6	3	6	3	6	3	6	3	6
Ampliación de Matemáticas (AMAT)	2	6	3	8	3	9	3	9	3	10	3	10
Física (FIS)	2	6	3	8	3	8	3	8	3	8	3	10
Fundamentos de Redes de Computadoras (FRC)	2	5	3	5	3	6	3	6	3	8	3	8

Tabla 2.1: Tabla resumen de la evolución del número de grupos en algunas asignaturas del Grado en Ingeniería Informática de la Universidad de Valladolid.

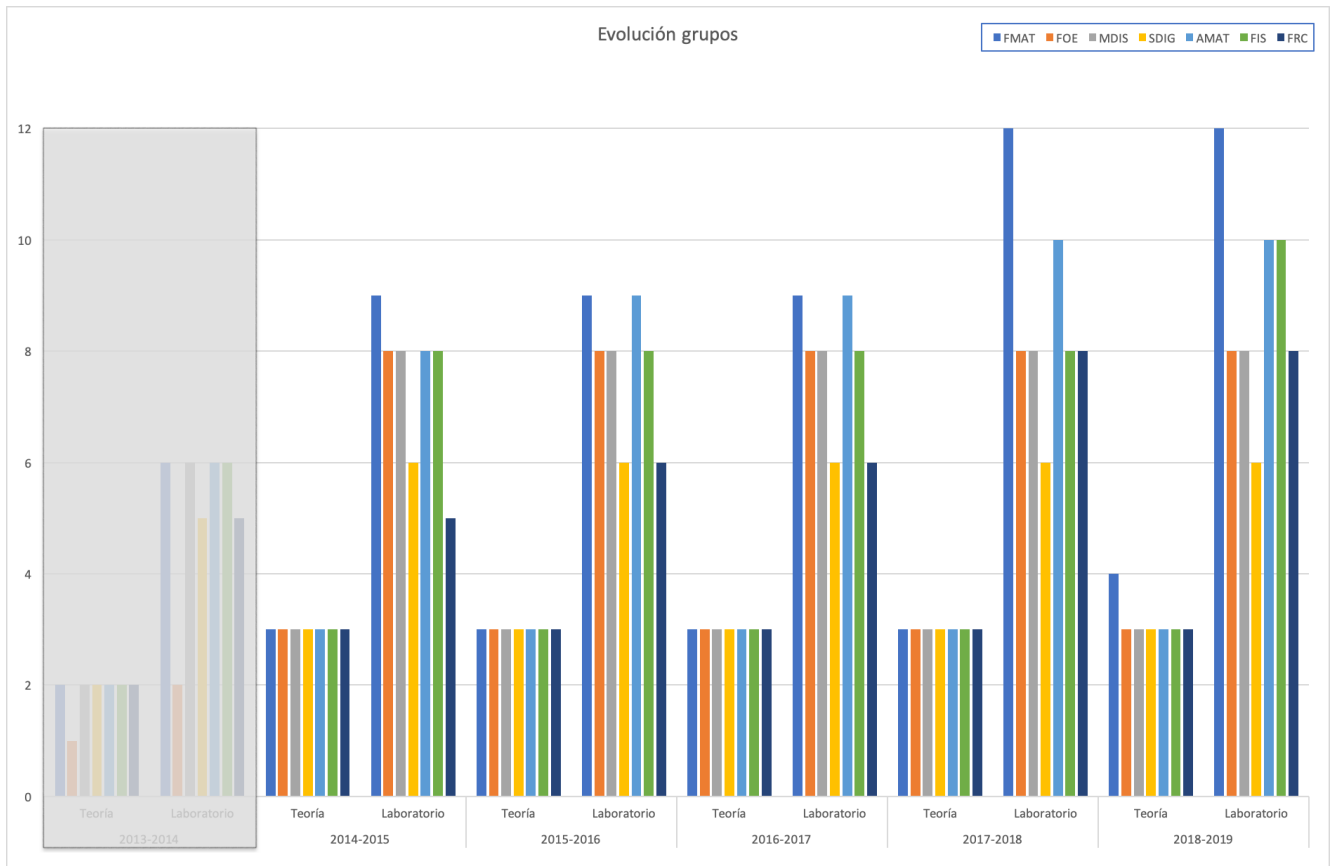


Figura 2.1: Evolución del número de grupos en algunas asignaturas del Grado en Ingeniería Informática de la Universidad de Valladolid.

Los datos presentados en la tabla y que se representan en la gráfica posterior se han obtenido de la página web de la Escuela de Ingeniería Informática de la Universidad de Valladolid [8].

Lo primero que hay que decir es que el curso 2013/2014 se ha marcado en un color grisáceo tanto en la tabla como en la gráfica puesto que en dicho curso existían dos titulaciones, la de Grado en Ingeniería Informática y la de Grado en Ingeniería Informática de Sistemas y los datos pueden no reflejar la realidad con respecto a otros cursos en los que se unificaron ambas titulaciones en una.

Analizando los datos en detalle se puede observar que los grupos de laboratorio han crecido más en número que los de teoría en casi todas las asignaturas de las que se presentan datos. Es una realidad que a los grupos de teoría se suele poder acudir aunque no se esté matriculado en ese grupo, pero en los laboratorios los y las docentes suelen ser más estrictos a la hora de controlar la pertenencia o no al grupo que se está impartiendo. Esto suele provocar que los estudiantes realicen su matrícula teniendo más en cuenta las clases de laboratorio que las de teoría. Desde ese punto de vista, si se analizan los datos en conjunto de las asignaturas escogidas, se puede ver que un estudiante en el curso 2014-2015 que se matriculase de esas asignaturas podía elegir entre un total de 52 grupos la combinación que más le interesara, lo que genera un total de 1.105.920 combinaciones posibles. El mismo estudiante, en el curso 2018-2019 tiene a su disposición, para esas mismas asignaturas, 62 grupos. Esto generaría 3.686.400 combinaciones posibles entre todos los grupos.

Al final lo que provoca la aparición de tantos grupos es que el problema del solapamiento de horas entre grupos de unas asignaturas con grupos de otras se vaya reproduciendo de tal manera que, como ya se ha mencionado antes, los estudiantes optan por no asistir a algunas clases antes que analizar si existe alguna opción en la que pudieran compatibilizar el horarios de todas sus asignaturas.

---

A modo de conclusión de este apartado se podría decir que el rendimiento académico y la asistencia a clase están estrechamente relacionados. Que dicha asistencia se ve afectada en un porcentaje considerable de casos por el solapamiento de horarios con otras asignaturas. Y que si se quiere mejorar el rendimiento académico hay que desarrollar estrategias y herramientas que disminuyan, en la medida de lo posible, el absentismo escolar.

## 2.2. Aplicaciones existentes

Tras una investigación preliminar se ha constatado que en la actualidad existen herramientas que realizan tareas similares pero ninguna hace exactamente lo que se plantea en este trabajo. Las aplicaciones que calculan los mejores horarios no están orientadas a los alumnos, sino que están enfocadas a que los centros educativos generen las combinaciones posibles de horarios para ofrecer a los alumnos.

Un listado bastante completo sobre aplicaciones de este tipo se puede consultar en [9].

### 2.2.1. Aplicaciones para alumnos

Se han localizado diversas herramientas orientadas a alumnos, pero estas no automatizan el proceso de elección del horarios, sino que permiten a los usuarios configurarlos paso a paso, es decir, son los alumnos los que deben ir añadiendo sus horas de clase, actividades extraescolares, trabajo, citas médicas, etc.

La mayoría tienen las mismas funcionalidades, por lo que aquí sólo se presenta una de las aplicaciones revisadas.

#### Agenda del estudiante

Se trata de una aplicación desarrollada para estudiantes cuyo principal objetivo es ayudar al usuario en la organización de clases y tareas para tener un mejor desempeño en sus estudios.

Permite gestionar horarios personalizados que incluyan horas de clase, actividades extraescolares, tareas pendientes, recados personales, etc.

Se han obtenido las siguientes capturas de pantalla de su página en la Aptoide [10], una tienda de aplicaciones para Android.

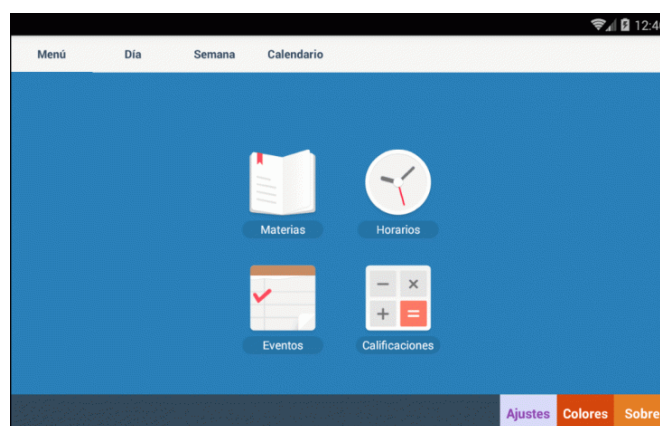


Figura 2.2: Captura de pantalla del menú principal de la herramienta Agenda del estudiante.

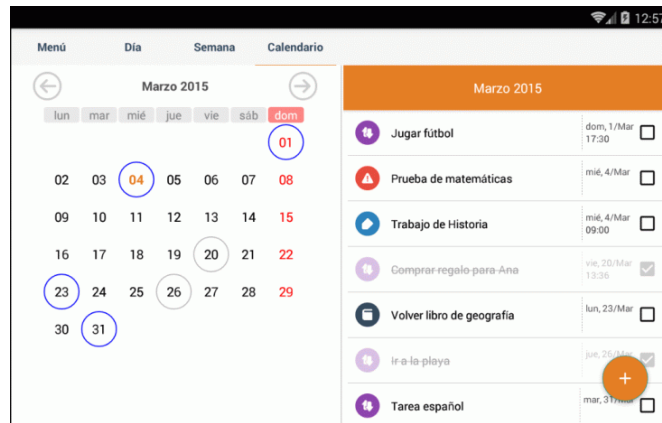


Figura 2.3: Captura de pantalla del menú de gestión de calendario de la herramienta Agenda del estudiante.



Figura 2.4: Captura de pantalla de la vista de un horario de la herramienta Agenda del estudiante.

## 2.2.2. Aplicaciones para centros educativos

En este punto las aplicaciones encontradas sí que generan horarios de manera automática, aunque como ya se ha mencionado anteriormente, lo hacen calculando los posibles horarios que los centros educativos pueden ofrecer a sus estudiantes. Se han encontrado herramientas que abarcan diferentes grados de complejidad, aunque la mayoría admiten la introducción de múltiples variables a la hora de generar los cálculos, entre las que se incluyen:

- Número de grupos de cada asignatura.
- Aulas.
- Categoría y/o tipo de contrato de los profesores.
- Tipo de clase (teórica, prácticas de laboratorio, taller, etc.).
- Restricciones personalizadas.
- Criterios pedagógicos.

## ascHorarios

Se trata de un software empresarial que asegura no sólo cumplir los requisitos indicados por los usuarios, sino que realiza el proceso de cálculo de horarios de una manera rápida y sencilla.

Con respecto a otras herramientas similares de las analizadas destaca por permitir introducir cambios y adaptar los horarios a partir de ellos y además, en su versión avanzada, permite dividir clases, tener en cuenta semanas pares e impares, clases optativas, clases con horarios fuera del habitual (fines de semana), etc. Es también muy destacable el hecho de que permite la posibilidad de saltarse algunas reglas de las establecidas inicialmente a la hora de calcular los horarios, permitiendo que algunos ajustes se hagan de manera más manual a posteriori.

Ofrece funcionalidades adicionales como la posibilidad de programar las sustituciones de profesores (incluyendo notificaciones e impresión) o herramientas para crear el sitio web del centro educativo.

Dispone de una versión de prueba gratuita, además de diversas licencias de pago que pueden encontrarse en la sección “Precios” de su página web [11]. Cabe destacar que la versión PRO ofrece un generador automático del horario de alumnos, que es lo más parecido que se ha encontrado a la aplicación que se presenta en este Trabajo de Fin de Grado.

Se presentan a continuación algunas capturas de pantalla:

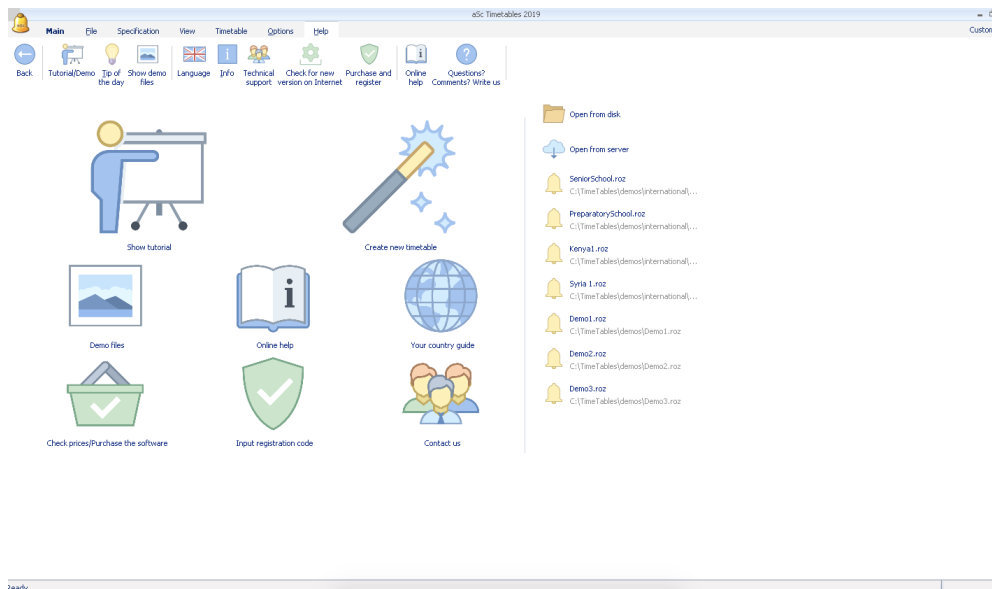


Figura 2.5: Captura de pantalla del menú principal de la herramienta ascHorarios.

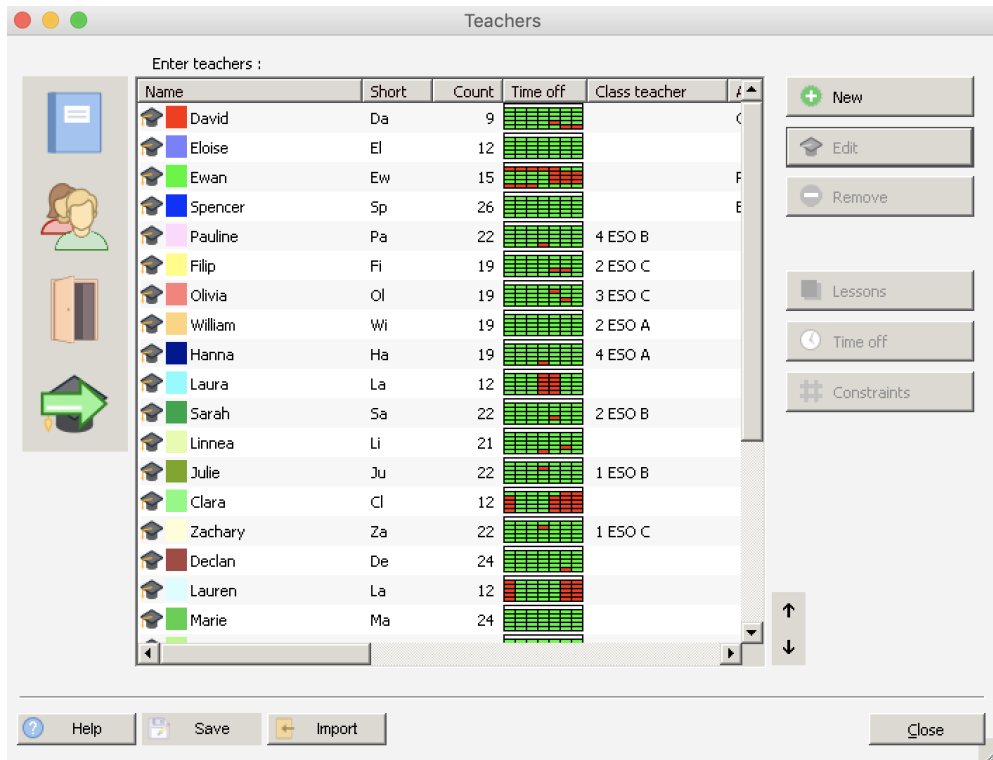


Figura 2.6: Captura de pantalla del menú de gestión de profesores de la herramienta ascHorarios.

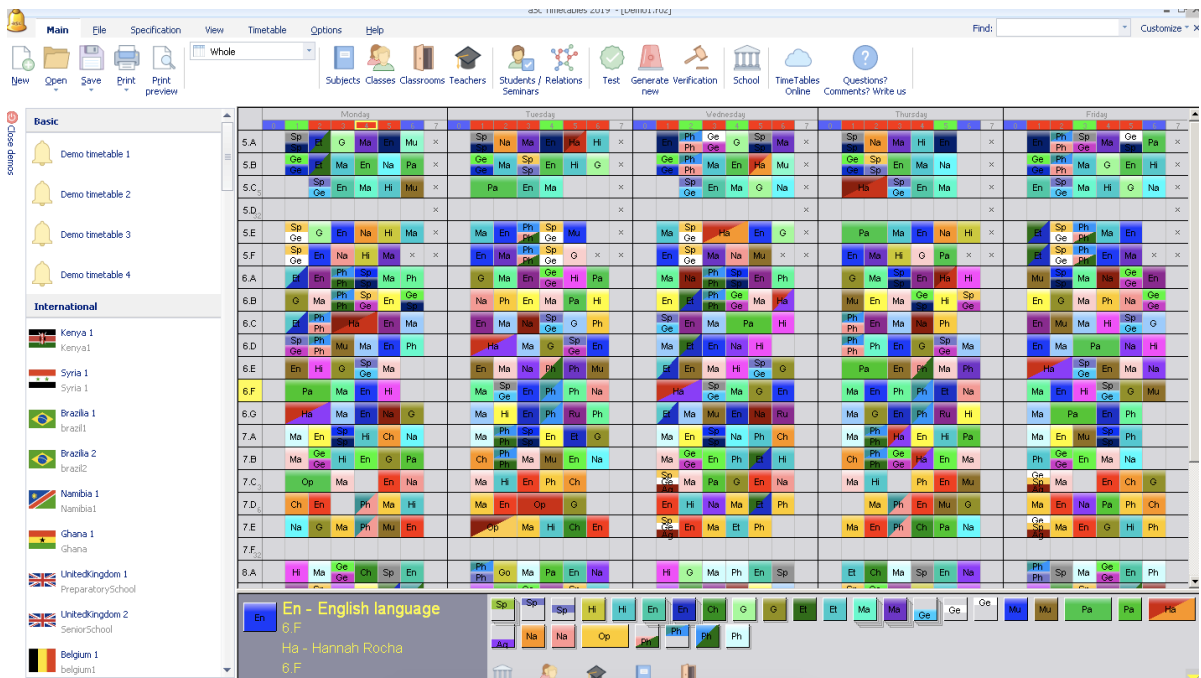


Figura 2.7: Captura de pantalla de la vista de un horario de la herramienta ascHorarios.

	Monday							Tuesday							Wednesday								
	0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1
1 ESO A	x	Ang	1M2Ct 34Cs	1Ct2C s34M	Edf	CV1	1Cs2 M34Ct	Cgc	1Ct2C s34M	Tec	Cn	Rel	Ang	Soc	Pla	1M2Ct 34Cs	Tdc	Soc	CV1	Esc	1Cs2 M34Ct	x	1M2Ct 34Cs
1 ESO B	x	Edf	1M2Ct 34Cs	1Ct2C s34M	Tec	Cn	1Cs2 M34Ct	Ang	1Ct2C s34M	Soc	Tdc	Mus	Esc	Rel	Tec	1M2Ct 34Cs	Cn	Edf	Mus	Soc	1Cs2 M34Ct	x	1M2Ct 34Cs
1 ESO C	x	Tdc	1M2Ct 34Cs	1Ct2C s34M	Soc	CV1	1Cs2 M34Ct	Soc	1Ct2C s34M	Ang	Edf	Pla	Cn	Cgc	Esc	1M2Ct 34Cs	Ang	Rel	CV1	Tec	1Cs2 M34Ct	x	1M2Ct 34Cs
2 ESO A	x	CV2	Edf	Ang	1Cs2 M34Ct	PDro	Cn	Ssc	Ang	1M2Ct 34Cs	Pla	1Ct2C s34M	Soc	Tec	CV2	Rel	Edf	1M2Ct 34Cs	Tec	1Cs2 M34Ct	Cn	x	Cn
2 ESO B	x	CV2	Mus	Soc	1Cs2 M34Ct	Ang	Soc	Edf	Mus	1M2Ct 34Cs	Ang	1Ct2C s34M	Cn	Cgc	CV2	Pla	PDro	1M2Ct 34Cs	Cn	1Cs2 M34Ct	Soc	x	Soc
2 ESO C	x	CV2	Tec	Cn	1Cs2 M34Ct	Soc	Edf	Cn	Edf	1M2Ct 34Cs	Ssc	1Ct2C s34M	Tec	Ang	CV2	PDro	Rel	1M2Ct 34Cs	Soc	1Cs2 M34Ct	Ang	x	Pla
3 ESO A	x	1M2Ct s3Ct4	1Cs2 Q34M/GN	1S/2A Q34M/GN	Tec	ca	CV3	1A/2N 1Ct2M 1N/2S 3S	1Ct2M 1N/2S 3S	1M2Ct s3Ct4	Epals	Tec	CV3	Rel	1M2Ct s3Ct4	1S/2A Q34M/GN	1Cs2 Q34M	Jp	1N/2S 3A	Edf	1Ct2M 1Cs2 3C4	Q34M	
3 ESO B	x	1M2Ct s3Ct4	1Cs2 Q34M/GN	1S/2A Q34M/GN	Mus	Edf	CV3	1A/2N 1Ct2M 1N/2S 3S	1Ct2M 1N/2S 3S	1M2Ct s3Ct4	ca	Mus	CV3	Pla	1M2Ct s3Ct4	1S/2A Q34M/GN	1Cs2 Q34M	Edf	1N/2S 3A	Rel	1Ct2M 1Cs2 3C4	Q34M	
3 ESO C	x	1M2Ct s3Ct4	1Cs2 Q34M/GN	1S/2A Q34M/GN	Epals	Pla	CV3	1A/2N 1Ct2M 1N/2S 3S	1Ct2M 1N/2S 3S	1M2Ct s3Ct4	Tec	ca	CV3	Edf	1M2Ct s3Ct4	1S/2A Q34M/GN	1Cs2 Q34M	Rel	1N/2S 3A	Jp	1Ct2M 1Cs2 3C4	Q34M	
4 ESO A	x	1A/2N 1Ct2M 1S/3S	1Ct2M 1N/2S 3S	1S/2A 1M2Ct s34Ct	CV4	1M2Ct s34Ct	Tec	ComN 1N/2S 3A	Edf	1S/2A 1Ct2M 1Cs2 34Cs	1Ct2M 1Cs2 34Cs	CV4	Tec	Fm	Tec	CV4	1Cs2 1A/2N Q34M/GS	1A/2N Q34M/GS	Edf	1N/2S 3A	Ss	1Ct2M 1Cs2 34Cs	
4 ESO B	x	1A/2N 1Ct2M 1S/3S	1Ct2M 1N/2S 3S	1S/2A 1M2Ct s34Ct	CV4	1M2Ct s34Ct	Ss	Mus	1N/2S 3A	CV42	1S/2A 1Ct2M 1Cs2 34Cs	1Ct2M 1Cs2 34Cs	CV4	Fm	CV42	Mus	CV4	1Cs2 1A/2N Q34M/GS	ComN 1N/2S 3A	1N/2S 3A	Pla	1Ct2M 1Cs2 34Cs	
4 ESO C	x	1A/2N 1Ct2M 1S/3S	1Ct2M 1N/2S 3S	1S/2A 1M2Ct s34Ct	CV4	1M2Ct s34Ct	Pla	Tec	1N/2S 3A	ComN 1N/2S 3A	1S/2A 1Ct2M 1Cs2 34Cs	1Ct2M 1Cs2 34Cs	CV4	Edf	Tec	Ss	CV4	1Cs2 1A/2N Q34M/GS	Rel	1N/2S 3A	Edf	1Ct2M 1Cs2 34Cs	
1 BAT	Mat Hdmc	Filo	Ang	Cas	Fis La	Fin MCS Tec	x	Cat	Di Gui Ecn	Cas	Fin MCS Tec	Edf	Mat Hdmc	Filo	Tut	Fis La	Ang	Rel	Cat	x	x	Fis La	Ang
2 BAT	Cas	Fin MCS Tec	Ang	Fis La	Mat Geo	Filo	Hist	Di Gui Ecn	Cas	Fis La	Cat	Filo	x	x	Hist	Mat Geo	Tut	Di Gui Ecn	Ang	x	x	Filo	Hist

Grouped constraints		Object	Description
Expand/Collapse		13x	Card distribution over the week
		(1 ESO B:Mus) 1 ...	Card distribution over the week: Max consecutive days 3>2
		(2 ESO A:Tec) 2 ...	Card distribution over the week: Max consecutive days 3>2
		(2 ESO C:Edf) 2 ...	Card distribution over the week: Max consecutive days 2>1
		(3 ESO A:Edf) 3 ...	Card distribution over the week: Max consecutive days 2>1
		(3 ESO A:1M2Cs...	Card distribution over the week: Max consecutive days 3>2
		(3 ESO B:Rel) 3 ...	Card distribution over the week: Max consecutive days 2>1
		(3 ESO B:1M2Cs...	Card distribution over the week: Max consecutive days 3>2

Figura 2.8: Captura de pantalla de la vista de un horario con restricciones ignoradas de la herramienta ascHorarios.

## Free Timetabling Software

Free Timetabling Software (FET) es un software de código abierto para planificar de manera automática los horarios de colegios, institutos y/o universidades.

En su página web [12] informan de que la herramienta es capaz de generar horarios de complejidad normal en tiempos de 5 a 20 minutos. Para casos de extrema dificultad puede tener que estar ejecutándose durante horas.

Se trata de una aplicación que se ejecuta de manera local y permite exportar los resultados a diferentes formatos como por ejemplo HTML, XML y CSV. La última versión del software (FET-5.37.5) fue lanzada el 10 de enero de 2019.

A continuación se muestran algunas capturas de pantalla obtenidas de la propia página web del proyecto y que está accesible desde su página web.



Figura 2.9: Captura de pantalla del menú principal de la herramienta FET.



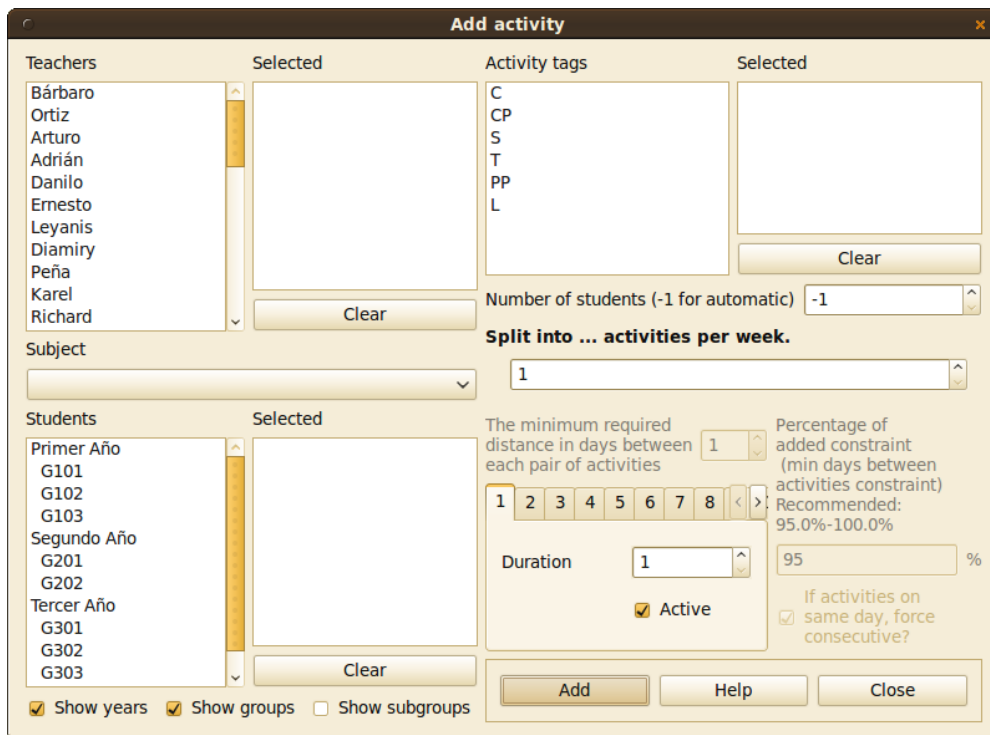


Figura 2.10: Captura de pantalla del menú para añadir una actividad de la herramienta FET.

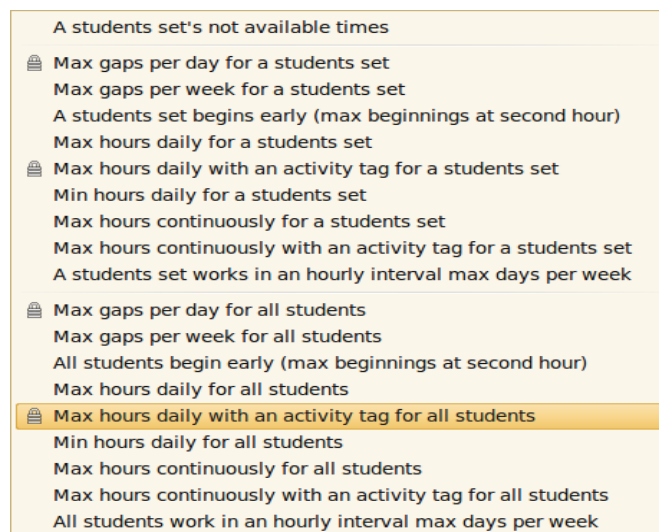


Figura 2.11: Captura de pantalla del menú de restricciones aplicables a alumnos de la herramienta FET.

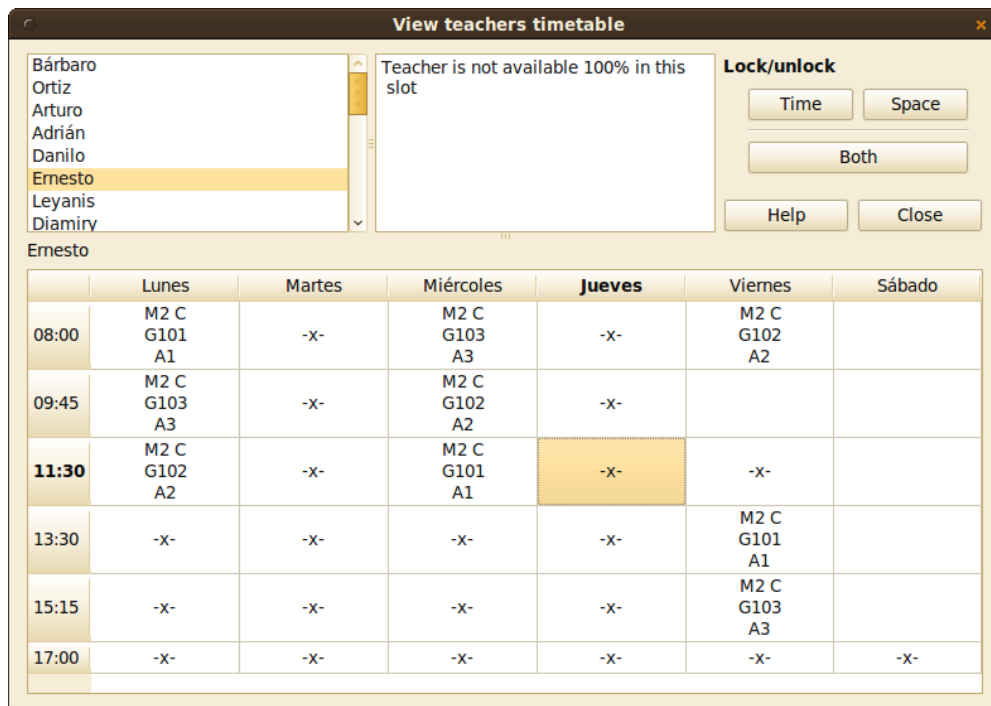


Figura 2.12: Captura de pantalla de la vista de horarios de profesores de la herramienta FET.

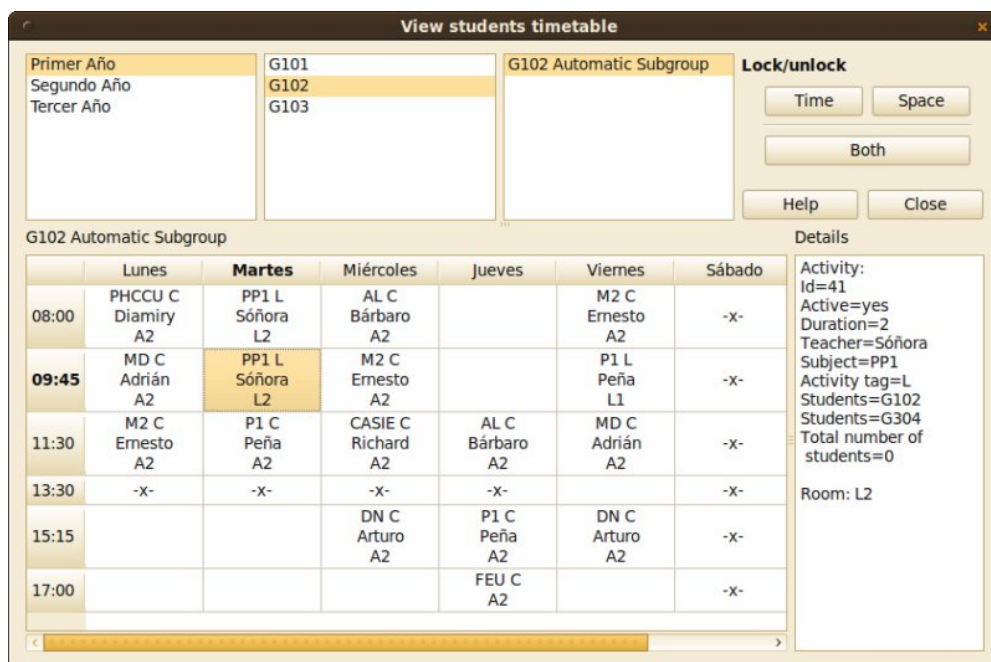


Figura 2.13: Captura de pantalla de la vista de horarios de estudiantes de la herramienta FET.

## 2.3. Desarrollo web

Aunque es verdad que hoy en día la mayoría de aplicaciones que se desarrollan son para entornos móviles, principalmente para dispositivos con sistema operativo iOS o Android, no es menos cierto que se siguen desarrollando muchas aplicaciones web.

---

El principal motivo de elegir este tipo de desarrollo es que se tiene la percepción de que los estudiantes formalizan su matrícula en la universidad en ordenadores más que en dispositivos móviles. Además las aplicaciones web cuentan con una serie de ventajas sobre las aplicaciones móviles:

- No requieren instalación por parte de los usuarios, ya que está alojada en un servidor. Lo único que necesita el usuario es un navegador.
- Se pueden ejecutar tanto en ordenadores como en otros dispositivos como tablets o smartphones.
- El desarrollo suele ser más rápido que en aplicaciones móviles.
- Las actualizaciones se realizan más rápido, ya que no se depende de versiones de dispositivos ni sistemas operativos.



## Capítulo 3

# Planificación

### 3.1. Propósito y alcance

El principal propósito del proyecto es desarrollar un sistema que permita a los estudiantes universitarios obtener el horario con menos solapamiento de clases y más compacto posible a partir de las asignaturas que ellos o ellas elijan.

La aplicación permitirá elegir la titulación, el período o semestre y las asignaturas para las que se quiere calcular el horario y generará la mejor opción posible priorizando que no haya conflictos horarios y que sea lo más compacto posible, es decir, que haya las menos horas libres posibles entre varias clases en una misma mañana o tarde.

Se plantea que la aplicación permita diferentes operaciones, dependiendo del rol del usuario que acceda al sistema, así tendremos:

- **Administrador de la aplicación**

- Gestionar las titulaciones, asignaturas y grupos de la aplicación.

- **Estudiante**

- Registrarse en el sistema y acceder a él.
- Gestionar su perfil en la aplicación.
- Generar el mejor horario posible atendiendo a la selección de asignaturas realizada.
- Guardar los horarios generados.
- Gestionar los horarios que tiene guardados.
- Descargar en formato PDF los horarios generados.

### 3.2. Metodología

Desde un principio se consideró la necesidad de darle al proyecto un enfoque de desarrollo ágil que permitiese flexibilidad y facilidad de organización. Al tratarse de un producto del que no se tienen referencias previas, y que es muy posible que se introduzcan cambios durante el proceso de desarrollo, el hecho de elegir una metodología de

---

este tipo nos permitirá hacerlo sin que sea traumático para el desarrollo normal del trabajo. Además, esto permitirá al equipo desarrollar un producto que se parezca más a lo que el cliente necesita.

De las estrategias de desarrollo ágil la elegida ha sido Scrum por considerar que es fácilmente adaptable a las necesidades del proyecto. No se utilizará de manera estricta, sino que se utilizarán algunos de los elementos y técnicas que establece para el desarrollo del proyecto.

### 3.2.1. Scrum

En los próximos apartados se presenta información relativa a Scrum y al final de cada uno se hará un apunte sobre los elementos que se van tomando o descartando de este marco de trabajo.

#### Qué es

Scrum nace como una estrategia de desarrollo de producto ágil en la que el equipo de desarrollo trabaja como una unidad para alcanzar un objetivo común [13]. Años más tarde, en 1995, Ken Schwaber y Jeff Sutherland adaptan dicha estrategia como procedimiento de desarrollo de software y en la última guía que publican de Scrum ([14]) lo definen de la siguiente forma:

*“Scrum es un marco de trabajo a través del cual las personas pueden abordar problemas complejos adaptativos, a la vez que se entregan productos de forma eficiente y creativa con el máximo valor”.*

Scrum plantea una forma de organizarse y trabajar cuyo principal elemento es el equipo que realiza el trabajo. Dotándole de herramientas y libertad para autoorganizarse se busca promover al equipo, pero también a los individuos que lo integran y conseguir así un trabajo final de calidad.

#### Características

Tal y como se menciona en [15], para conseguir el éxito en el desarrollo de productos Scrum propone una serie de principios que deben cumplir las personas que participan en el proyecto así como el propio proyecto. Son los siguientes:

- **Satisfacción del cliente:** es lo que se busca en último término, que el cliente esté contento con el producto que recibe y que sea exactamente lo que desea.
- **Respuesta al cambio:** los eventos que propone Scrum, así como sus artefactos y la forma de organizar el trabajo permiten responder a los cambios de manera normal y sin provocar grandes prejuicios en el trabajo desarrollado.
- **Equipos autoorganizados:** se funciona como un equipo a todos los niveles, y esto permite que se pueda hacer frente a las diferentes tareas, asumiendo cada miembro los aspectos necesarios para completar el trabajo.
- **Simplicidad:** hace referencia a que sólo es necesario centrarse en las posibles necesidades del cliente, sin pensar en funcionalidades o incrementos del producto que, aun aportándole valor, no vayan a ser utilizados por el cliente.
- **Desarrollo incremental:** el desarrollo es incremental desde el primer momento permitiendo ofrecerle al cliente un producto mínimo viable desde las primeras iteraciones. Se trata de que al final de cada ciclo del proceso que se sigue, se le ofrezca al cliente un producto susceptible de desplegarse.

- **Trabajo enfocado en el producto:** la finalidad es desarrollar un producto útil de calidad por encima de los procesos o métodos empleados para hacerlo.

## El proceso

El proceso se suele llevar a cabo en períodos cortos de tiempo que van de 1 a 4 semanas y cuya duración suele ser fija, aunque por necesidades del proyecto puede adaptarse. Al final de cada iteración hay que disponer de un producto final susceptible de ser entregado al cliente y que incremente el valor sobre lo entregado anteriormente.

Una representación gráfica del proceso puede verse en la Figura 3.1, obtenida de [16].

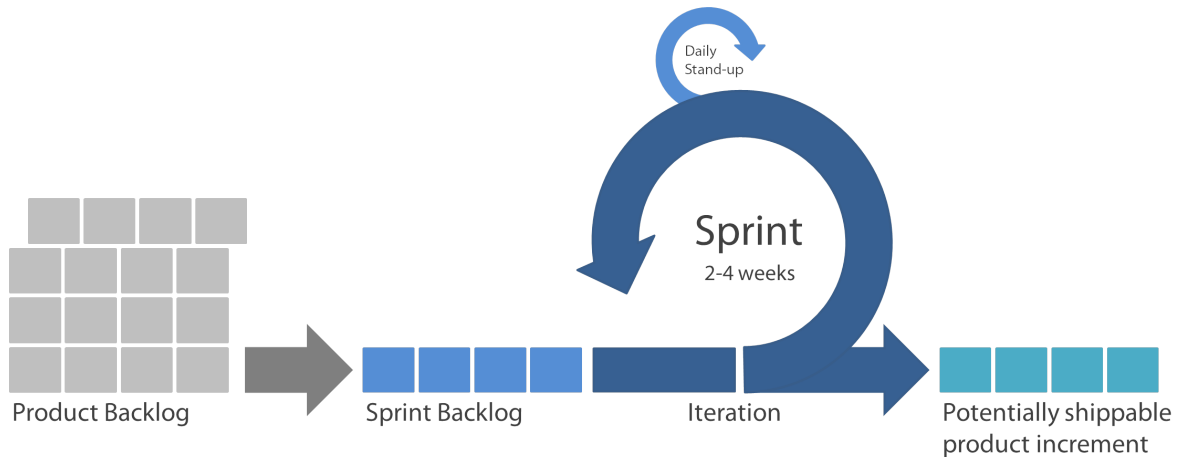


Figura 3.1: Diagrama del ciclo de vida de un proyecto en Scrum.

Lo primero que tiene que producirse es la definición del *Product Backlog* por parte del *product owner*. Se añaden las historias de usuario, que vienen a representar funcionalidades que se desea que tenga el producto final. Después se pasa a las fases denominadas *sprints*, en las que el equipo decide qué historias de usuario de las definidas en el *Product Backlog* se añaden al *sprint backlog*. En ese momento el equipo de desarrollo define tareas para completar el *sprint*. Durante la duración del *sprint*, y con frecuencia diaria, se llevan a cabo reuniones (*daily meetings*) en las que se analiza el estado de las tareas asignadas.

Al final de cada *sprint* debe tenerse un producto con un incremento funcional con respecto a la anterior versión y que debe ser fácilmente puesto en producción si así se desea.

Para el proyecto que se va a desarrollar se seguirá el proceso marcado por Scrum con alguna de las características adaptadas a las necesidades del proyecto:

- En primer lugar los *sprints* tendrán una duración de dos semanas.
- El trabajo no será uniforme todos los días, ya que la situación laboral del autor de este trabajo lo impide. Sí se ha fijado un ritmo de trabajo de 25 horas semanales distribuidas de la siguiente manera:

Día de la semana	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
Horas de trabajo	5	2	2	3	3	5	5

Tabla 3.1: Distribución del horario de trabajo.

- No se realizarán los *daily meetings* ya que hay días en los que el trabajo va a ser escaso y no se considera que dichas reuniones vayan a ser muy productivas.

---

## Roles

Definen el conjunto de responsabilidades que es necesario asumir en un proyecto que siga el marco de trabajo Scrum. En su conjunto garantizan cubrir las necesidades del proyecto a nivel de información, desarrollo y comunicación. Los roles propuestos son los siguientes:

- **Product Owner:** se trata de la persona que actúa de intermediario entre el equipo de desarrollo y el cliente. Es el responsable del éxito o fracaso del proyecto en último término. Sus principales funciones son:
  - Negociar con el cliente el alcance del proyecto.
  - Definir la estrategia y objetivos.
  - Definir y mantener el *Product Backlog*.
  - Ayudar al *Scrum Master* y al *Development Team* a resolver cualquier duda o problema que tenga que ver con el proyecto, el producto y su funcionalidad.
- **Scrum Master:** es el/la responsable de que los valores, reglas y prácticas de Scrum se lleven a cabo de manera adecuada. Tiene especial relevancia en gestionar las interacciones dentro del equipo para maximizar el valor creado. Aunque suele dar lugar a confusión hay que indicar que no es el *project manager* y no cumple sus deberes ni asume sus obligaciones. Puede darse el caso de que sea un desarrollador del *Development Team*, pero una persona nunca puede ser al mismo tiempo y en un mismo proyecto *Scrum Master* y *Product Owner*.
- **Development Team:** es el equipo de desarrollo y lo componen las personas encargadas de trabajar en los incrementos de producto que se producen a lo largo del ciclo de vida de Scrum. Asume la responsabilidad de gestionar el *sprint backlog*, determinando el detalle de cada elemento del *Product Backlog* y dividiéndolo en las tareas que considere. Es responsable de entregar el producto terminado y de que este cumpla con los criterios de aceptación marcados. Se recomienda que esté formado por entre tres y nueve personas. Para lograr el éxito debe cumplir las siguientes características:
  - **Ser flexible:** cada persona puede ocupar varios roles.
  - **Estar autoorganizado:** el mismo equipo define sus métodos de trabajo.
  - **Ser multidisciplinario:** el conjunto del equipo cuenta con la capacidad para hacer frente al proyecto con garantías.

Para el desarrollo del proyecto se ha decidido asignar los roles de la siguiente manera:

- **Product Owner:** estará representado por Joaquín Nicolás Adiego, tutor de este Trabajo de Fin de Grado y actuará como cliente.
- **Scrum Master:** será Miguel Rojo y se encargará de intentar eliminar los obstáculos que impidan alcanzar los objetivos de cada *sprint*, además de velar por que se sigan las técnicas y guías que se consideren de Scrum.
- **Development Team:** estará formado sólo por una persona, Miguel Rojo, como el responsable de desarrollar el producto.

## Artefactos

Representan las herramientas que propone Scrum para maximizar la transparencia de la información y que todo el equipo tenga una visión única del producto que se quiere conseguir. Ayuda a los diferentes roles a coordinarse y trabajar juntos.



---

Hay diversidad de opiniones sobre lo que se consideran artefactos en Scrum, pero aquí sólo se detallan los más habituales y que aparecen en la Guía de Scrum ([14]) como artefactos propios del marco de trabajo:

- **Product Backlog:** es la lista de funcionalidades que deberá tener el producto que se va a desarrollar y están representadas en forma de historias de usuario. Las características que debe tener se pueden resumir en las siguientes (obtenidas a partir de [17]):
  - **Detallado:** las historias del *Product Backlog* tienen que estar descritas en lenguaje que el cliente entienda, pero con el suficiente nivel de detalle como para que se puedan estimar y priorizar.
  - **Estimado:** las historias de usuario tienen que estar estimadas de alguna forma que permita al equipo saber el esfuerzo que le requerirá completarlas.
  - **Priorizado:** las historias de usuario tienen que estar priorizadas para determinar el orden en el que se completarán.
  - **Dinámico:** las historias de usuario pueden eliminarse o añadirse a medida que avance el proyecto. Además son susceptibles de ser estimadas y priorizadas de nuevo en cualquier momento del ciclo de vida de Scrum.
- **Sprint Backlog:** se define al principio del *sprint* y es la lista de funcionalidades que se saca del *product backlog* que se llevarán a cabo durante el *sprint*. Tiene las mismas características que se han indicado para el *product backlog*. La lista de funcionalidades elegidas tiene que adaptarse a la velocidad del equipo, y en caso de no ser posible se pueden dividir esas funcionalidades.
- **Incremento:** está compuesto por las historias de usuario completadas durante un *sprint* y el valor del incremento de los *sprints* anteriores. Representa el estado mejorado del producto tras el *sprint* y debe estar en condiciones de utilizarse sin importar si se decide sacar a producción o no.

Para el proyecto a desarrollar se decide utilizar los tres artefactos que se acaban de definir, un *product backlog* con la lista de funcionalidades del producto y que se definirá al comienzo del proyecto, los *sprint backlogs* correspondientes a cada uno de los *sprints* del proyecto y que se definirán al comienzo de cada uno, y los incrementos que serán las historias de usuario completadas en cada *sprint*, y además se utilizará uno más denominado *Sprint Burndown*. Este artefacto consiste en una gráfica que se actualiza diariamente y que tiene dos métricas: la previsión de horas que va a conllevar el *sprint* y la cantidad real de horas consumidas en dicho periodo.

## Eventos

Son hitos o momentos en los que hay que ejecutar alguna acción relevante para el proyecto. Están predefinidos, tienen una duración máxima y el objetivo es crear cierta regularidad. Se trata de evitar tener que llevar a cabo reuniones que no estén definidas.

Es fácil encontrar diferentes guías o manuales en los que aparecen unos eventos u otros. Aquí se exponen los que aparecen en la Guía de Scrum ([14]):

- **Sprint:** es la unidad de tiempo que determina un ciclo de desarrollo en Scrum. Se debe crear un incremento sobre el producto que sea utilizable y potencialmente desplegable al final del *sprint*. Durante este intervalo de tiempo tienen lugar otros de los eventos más relevantes, como son el *sprint planning*, los *daily scrums* o *daily meetings*, el *sprint review* y un *sprint retrospective*.
- **Sprint Planning:** es la reunión de planificación del *sprint* en la que se determina el trabajo a realizar y tiene lugar siempre al inicio del *sprint*.

---

Se recoge la funcionalidad que hay que desarrollar, se resuelven dudas, se crean las historias de usuario en caso de que fuera necesario, se estiman las historias de usuario que se van a completar en ese periodo dividiéndolas en tareas y se determinan los criterios de aceptación.

Se divide en dos partes que deben responder a las preguntas ¿Qué se hará? y ¿Cómo se hará?.

Se recomienda una duración máxima de 8 horas para *sprints* de un mes.

- **Daily Scrum:** reunión diaria del equipo de desarrollo para planear el trabajo que se realizará las próximas 24 horas. Tiene una duración máxima de 15 minutos y siempre se realiza en el mismo lugar y a la misma hora.

De estas reuniones se suele obtener información suficiente para evaluar el progreso dentro del *sprint* ya que en ellas todos los miembros del equipo de desarrollo deben exponer cómo llevan el trabajo asignado, si lo acabarán a tiempo, qué tareas asumen una vez han terminado las que tenían asignadas y cualquier problema que les haya podido surgir.

- **Sprint Review:** se trata de una revisión que se hace al final del *sprint* con el objetivo de evaluar el incremento entregado y reorganizar el *product backlog* en caso de que fuese necesario. En esta reunión es donde se produce la aceptación del incremento según los criterios establecidos para cada historia de usuario. Aunque no es obligatorio, sí es muy recomendable que cuente con los siguientes elementos:

- Breve explicación de los objetivos del *sprint*.
- Repaso de los criterios de aceptación.
- Resumen del desarrollo del *sprint*.
- Demostración del incremento entregado.

- **Sprint Retrospective:** reunión cuyo principal objetivo es que el equipo de Scrum se evalúe a sí mismo e intente corregir los errores que se pudieran haber cometido. Tiene lugar entre el *Sprint Review* y el siguiente *Sprint Planning*.

En esta reunión se evalúan las personas, los procesos, las relaciones y las herramientas para mantener lo que está funcionando bien y tratar de buscar soluciones a lo que esté funcionando mal.

De los eventos presentados sólo se van a utilizar tres y con las siguientes características:

- **Sprint:** se realizarán cinco con una duración de dos semanas cada uno. El tiempo de trabajo dedicado semanalmente se puede consultar en la Tabla 3.1.
- **Sprint Planning:** serán reuniones de corta duración mantenidas con el tutor al final de cada *sprint review* con el objetivo de preparar el siguiente *sprint*.
- **Sprint Review:** breve reunión que se mantendrá al finalizar cada *sprint* y cuyo objetivo principal será evaluar si se van cumpliendo los plazos establecidos.

### 3.3. Plan temporal

En este apartado se presenta la planificación temporal realizada una vez que se tenía definida la primera versión del *Product Backlog*. Se trata de una estimación inicial y que atiende a dos criterios:

- Team velocity: fijada en 25 horas por semana de trabajo.
- Información asociada a las historias de usuario (riesgo, puntuación, etc.).

El seguimiento del proyecto así como el cumplimiento de la plan temporal inicial se analizan en detalle en el apartado 3.6 del presente documento.

### 3.3.1. Product Backlog

En este apartado se presenta el *product backlog* completo, con la correspondiente priorización de las historias de usuario (el detalle de cada una de las historias se puede consultar en el apartado 4.3 del presente documento)

Prioridad	ID Historia de usuario	Historia de usuario	Estimación (horas)
1	HU06	Como usuario quiero poder utilizar la aplicación en diferentes tipos de dispositivos y/o navegadores para acceder a ella en cualquier momento y lugar.	29
2	HU01	Como usuario quiero poder registrarme en el sistema para poder acceder a él.	16
3	HU02	Como usuario quiero poder acceder a la aplicación para poder hacer uso de ella.	50
4	HU03	Como usuario quiero poder generar un horario de acuerdo a las asignaturas que elija para poder asistir al mayor número de clases posible.	23
5	HU04	Como usuario quiero guardar los horarios que genere para poder tener acceso a ellos en cualquier momento.	17
6	HU05	Como usuario quiero poder exportar mis horarios en diferentes formatos para después imprimirlos y/o almacenarlos en mis dispositivos.	50
7	HU07	Como administrador quiero gestionar la información de la aplicación en lo relativo a titulaciones, asignaturas, grupos, etc., para mantenerla actualizada.	49

Tabla 3.2: Product Backlog del proyecto.

Se decidió que los *sprints* ocupasen dos semanas con una estimación de trabajo de 25 horas por semana, haciendo que en cada *sprint* se incluyan historias de usuario atendiendo a la prioridad fijada en el product backlog e intentando que se ajusten a las 50 horas de trabajo establecido para cada *sprint*.

La planificación temporal que se realiza inicialmente es la siguiente:

Nº Sprint	Fecha inicio	Fecha fin	Historias de usuario a realizar
<i>Sprint</i> 1	25 de febrero	10 de marzo	HU06
<i>Sprint</i> 2	11 de marzo	24 de marzo	HU01 y HU02
<i>Sprint</i> 3	25 de marzo	7 de abril	HU03
<i>Sprint</i> 4	8 de abril	21 de abril	HU04 y HU05
<i>Sprint</i> 5	22 de abril	5 de mayo	HU07

Tabla 3.3: Planificación temporal del proyecto.

---

## 3.4. Gestión de riesgos

El PM-BOK ([18]) define riesgo de la siguiente forma: “*Un riesgo es un evento o una condición inciertos que, si ocurren, tienen un efecto positivo o negativo sobre los objetivos del proyecto.*”. Cuando hablamos de gestionar riesgos nos referimos a estimar e intentar controlar los riesgos que afectan al proyecto, incluyendo el proceso y el producto final.

Aunque Scrum no propone una gestión de riesgos de manera explícita como sí lo hacen otros marcos de trabajo y estrategias, lo cierto es que la propia definición y muchos de los elementos van enfocados a controlar y minimizar los riesgos en el desarrollo del proyecto. Según [14] Scrum se basa en la teoría de control de procesos empírica, que se basa en la transparencia, inspección y adaptación. Esto, junto con el enfoque iterativo e incremental que se propone, permite controlar el riesgo en todo momento.

Tal y como se recomienda en [13] y al estar trabajando en un entorno ágil y adaptativo los riesgos se considerarán al inicio de cada iteración, y mientras dure la misma se analizarán y gestionarán para minimizar el impacto que puedan tener en el proyecto.

## 3.5. Gestión de configuraciones

En el glosario estándar de términos utilizados en Ingeniería del Software ([19]) se define como una disciplina que aplica la supervisión y dirección técnica y administrativa para:

- Identificar y documentar las características físicas y funcionales de un elemento de configuración.
- Controlar los cambios de esas características.
- Registrar y gestionar el proceso de cambios y el estado de la implementación.
- Verificar el cumplimiento de los requisitos especificados.

En proyectos de desarrollo de software se asocia al control de versiones y para este proyecto, si bien no se ha manejado a un nivel profesional, sí que se ha utilizado la herramienta Github para tener actualizado en todo momento el código desarrollado.

## 3.6. Seguimiento

Tal y como se ha explicado antes el seguimiento del proyecto se va a realizar tomando algunos de los elementos que propone el marco de trabajo Scrum, entre los que se encuentran el *Sprint Planning* y el *Sprint Review*. Son las dos actividades principales que se han realizado para llevar un seguimiento del proyecto y que se explican en la siguiente sección.

### 3.6.1. Sprint Planning y Sprint Reviews

En esta parte del trabajo y para cada *sprint* se explica brevemente en qué consistió, se presenta la planificación del trabajo y se analiza el seguimiento con

---

## Sprint 0

El *Sprint 0* en Scrum es una fase previa que sirve de preparación al proyecto. Aunque no se realiza siempre, en casos como este en los que es la primera vez que se trabaja con una metodología ágil es muy recomendable. Lo que se realice en este *sprint* permitirá establecer una base sobre la que trabajar, tanto a nivel de metodología como a nivel de desarrollo del proyecto.

Siguiendo algunas de las indicaciones encontradas en [15] se ha intentado que en este *sprint* quedaran definidos 2 aspectos muy importantes del proyecto y que son los siguientes:

- **Características del producto:** hay que reunir toda la información necesaria para poder dejar claro lo que hará el producto una vez esté terminado. La mayor parte de la fase de análisis se centra en esto.
- **Estructura del *Product Backlog*:** se dejan definidas las historias de usuario. Es posible que varíen pero el hecho de tenerlas delimitadas ayudará a no perder demasiado tiempo en los siguientes *sprints*.

Aunque lo recomendable es que esta fase dure una semana, para este proyecto se empleó aproximadamente un mes ya que se consideró desde el inicio la fase de análisis y definición del producto como algo esencial en el proyecto. Además el tiempo disponible del autor del presente trabajo fue una característica determinante a la hora de alargar este *sprint*.

## Primer sprint

Para el primer *sprint* se propone completar la historia de usuario con identificador HU06, que corresponde al funcionamiento de la aplicación en diferentes navegadores y dispositivos.

Aunque en este *sprint* no se aplican apenas elementos de la fase de análisis se ha estimado teniendo en cuenta que es posible encontrarse con bastantes dificultades a la hora de desplegar la aplicación y en la parte de programación de la interfaz de usuario para que sea adaptativa. En la siguiente tabla se muestra información relativa al *Sprint Backlog* (3 primeras columnas) junto con los resultados obtenidos:

ID HU	Tarea	Estimación (horas)	Tiempo real (horas)	Desviación
HU06	Programación UI	25	20	-5
HU06	Configuración despliegue remoto	15	12	-3
HU06	Configuración despliegue local	3	3	0
HU06	Pruebas	2	2	0
HU06	Documentación	5	5	0
Totales		50	42	-8

Tabla 3.4: *Sprint Backlog* e información de seguimiento del primer *sprint*.

Como se puede ver la estimación de las tareas fue aceptable excepto en la parte de la programación de la interfaz de usuario, donde se invirtió sólo el 80% del tiempo estimado. La conclusión obtenida es que muchos elementos de los utilizados ya eran conocidos y quizá debió realizarse una estimación menos precavida.

La evolución de las horas de trabajo se puede ver en el siguiente gráfico de *burndown*:

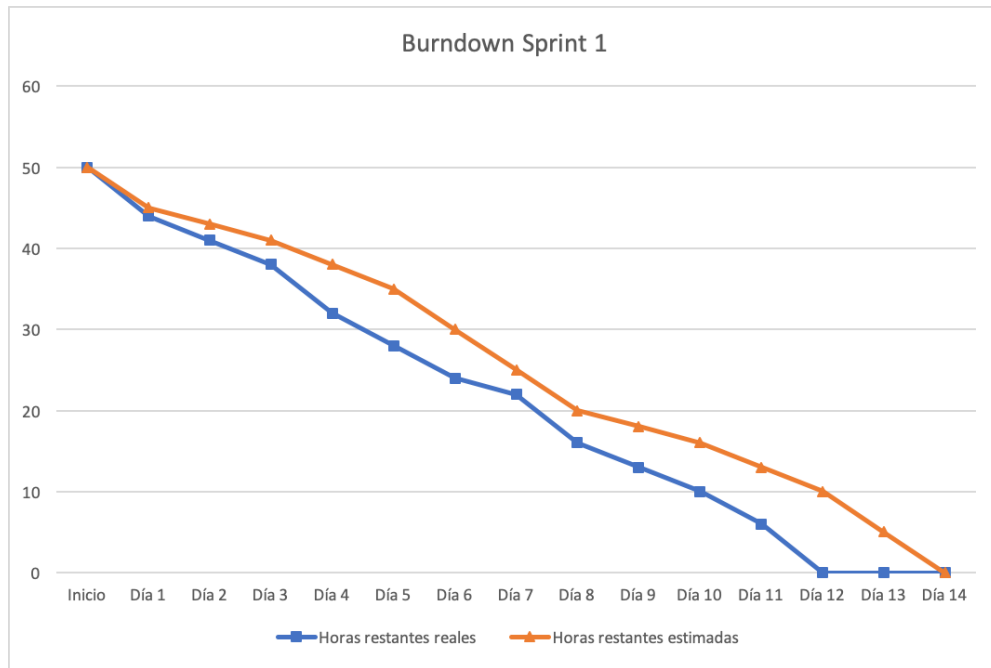


Figura 3.2: *Burndown* del primer *sprint*.

La Figura 3.2 viene a demostrar lo que ya se comentó antes de que la estimación inicial fue pesimista, terminando el trabajo asignado al *sprint* dos días antes de que este finalizara.

## Segundo sprint

Teniendo en cuenta los resultados del primer *sprint*, para este segundo se intenta ser más realista ajustando algo más el tiempo de cada una de las tareas. Este *sprint* tiene dos puntos que se consideran críticos y que se han tenido muy en cuenta a la hora de hacer las estimaciones:

- El framework de desarrollo elegido: aunque se ha obtenido mucha información en la fase de análisis y se han realizado pequeñas pruebas, es la primera vez que se va a trabajar con el framework dentro del proyecto.
- La conexión de la capa de persistencia: la experiencia en otras asignaturas y prácticas ha demostrado que la instalación y conexión a la base de datos puede ser un foco de problemas.

En este segundo *sprint* se han planificado tareas para un total de 45 horas de trabajo. Se decidió dejarlo de esta forma ya que las tareas de la siguiente historia de usuario del *product backlog* tenían demasiada carga horaria como para introducirlas en este.

ID HU	Tarea	Estimación (horas)	Tiempo real (horas)	Desviación
HU01	Configuración Base de Datos	7	13	+6
HU01	Programación UI	4	2	-2
HU01	Programación negocio	8	10	+2
HU01	Programación persistencia	3	1	-2
HU01	Pruebas	3	2	-1
HU01	Documentación	4	3	-1
HU02	Programación UI	3	2	-1
HU02	Programación negocio	6	7	+1
HU02	Pruebas	3	1	-2
HU02	Documentación	4	3	-1
Totales		45	44	-1

Tabla 3.5: *Sprint Backlog* e información de seguimiento del segundo *sprint*.

Aunque la estimación final del *sprint* fue mejor que en el primero, se ve en la Tabla 3.5 que en todas las tareas hubo desviación, especialmente en aquellas que ya se consideraron susceptibles de generar riesgos (“Configuración de la Base de Datos” y “Programación de la capa de negocio”).

Se esperaba que el *sprint* finalizase el día 13, como así fue y se puede ver en el *burndown* de la Figura 3.3

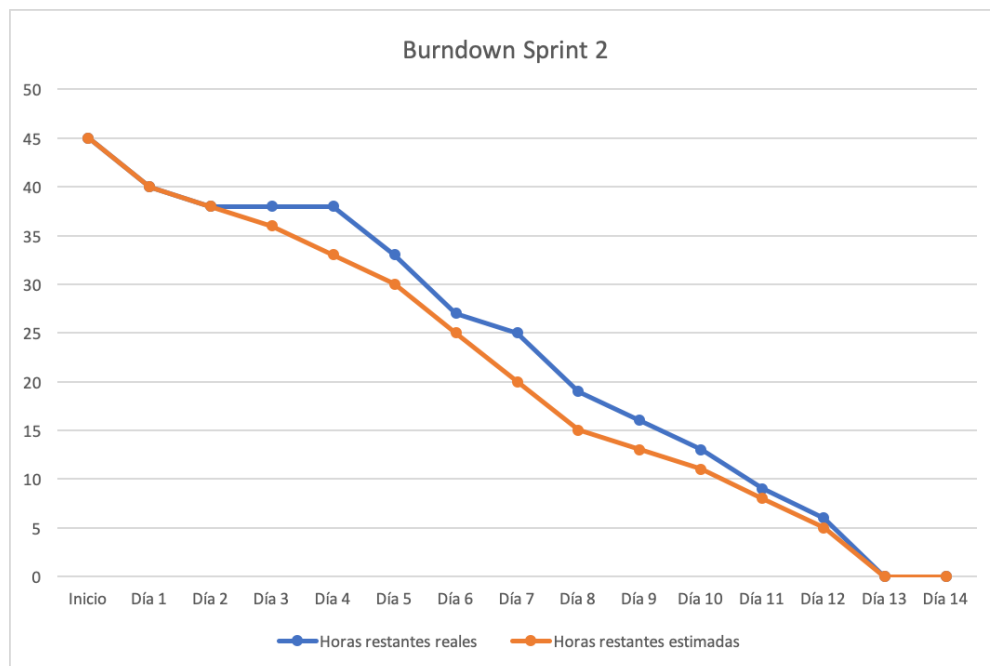


Figura 3.3: *Burndown* del segundo *sprint*.

### Tercer sprint

Se trata del más importante dentro del proyecto, ya que la historia de usuario que se va a desarrollar corresponde a la funcionalidad más importante del producto, que es la de generar el mejor horario de manera automática.

Al igual que en el segundo *sprint* la estimación que se hace es “generosa”, ya que se considera que la parte de programación de la lógica de negocio puede ser complicada y además puede provocar cambios en el diseño de la

capa de persistencia. Aunque no se ha incluido de manera explícita en la Tabla 3.6, la primera tarea de este *sprint* incluyó el cambio de la página de inicio de la aplicación. Tras la reunión con el tutor previa al desarrollo del *sprint*, y teniendo en cuenta que el uso de la aplicación implica estar registrado en el sistema, se decide incluir información en la propia página de inicio que facilite dicha información al usuario.

ID HU	Tarea	Estimación (horas)	Tiempo real (horas)	Desviación
HU03	Programación UI	10	8	-2
HU03	Programación negocio	15	30	+15
HU03	Programación persistencia	15	22	+7
HU03	Pruebas	5	6	+1
HU03	Documentación	5	6	+1
Totales		50	72	+22

Tabla 3.6: *Sprint Backlog* e información de seguimiento del tercer *sprint*.

Las estimaciones realizadas fueron muy malas, desviándose más de un 40% al final del *sprint*. Los principales problemas que se encontraron fueron los que se preveía, pero de mayor magnitud. La dificultad de la programación de la lógica provocó que se reconsiderara el modelo planteado en la fase de análisis y se realizaron algunos cambios. Esto retrasó bastante el desarrollo, tal y como se puede ver en la Tabla 3.6. Lo recomendable hubiera sido negociar de nuevo el *sprint* con el *product owner*, pero se toma la decisión de invertir más horas en los últimos días del *sprint* con el objetivo de completar todas las tareas. Esto se ve claramente en la Figura 3.4, donde a partir del día 8 se empieza a reducir drásticamente la diferencia entre horas estimadas restantes y horas reales restantes.

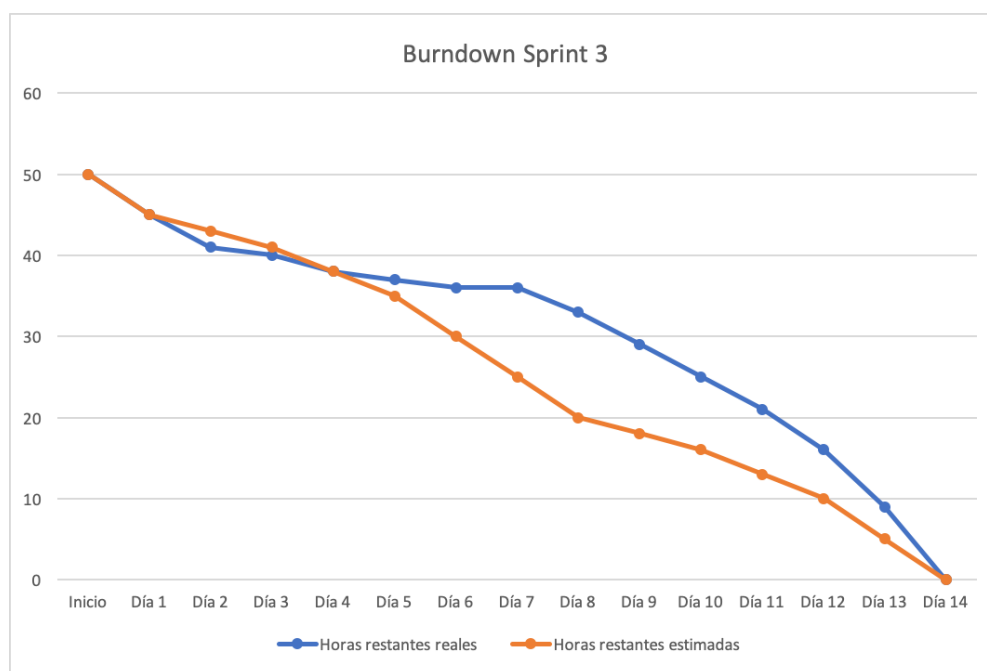


Figura 3.4: *Burndown* del tercer *sprint*.

#### Cuarto sprint

En este *sprint* se van a completar las historias de usuario referentes a que el usuario guarde sus horarios y los exporte en diversos formatos. Como se puede ver en la Tabla 3.7 las tareas de programación de la lógica de negocio



son las más relevantes, ocupando entre las dos la mitad del tiempo estimado para completar el *sprint*. El resto de tareas, teniendo en cuenta que algunas son similares a las de anteriores *sprints* y que hay cosas que se pueden reutilizar, tienen valores bajos, y como se puede ver en la misma tabla en varias de ellas el tiempo real empleado fue menor que el estimado inicialmente.

ID HU	Tarea	Estimación (horas)	Tiempo real (horas)	Desviación
HU04	Programación UI	2	3	+1
HU04	Programación negocio	10	12	+2
HU04	Programación persistencia	5	3	-2
HU04	Pruebas	3	3	0
HU04	Documentación	3	3	0
HU05	Programación UI	2	1	-1
HU05	Programación negocio	10	8	-2
HU05	Pruebas	2	2	0
HU05	Documentación	3	2	-1
Totales		40	37	-3

Tabla 3.7: *Sprint Backlog* e información de seguimiento del cuarto *sprint*.

La historia de usuario HU04, con pequeñas desviaciones en algunas tareas, se completó tal y como estaba previsto. En cuanto a la historia de usuario HU05 se completó en 4 horas menos de las 17 estimadas, es decir, aproximadamente se completó en un 75% del tiempo que se había previsto, haciendo de nuevo las estimaciones iniciales malas. Esto se debe a que al final el uso de librerías externas para exportar los documentos a formato PDF no resultó tan complicado como se esperaba, y tras probar diferentes opciones se escogió la que mejor se adaptaba al proyecto.

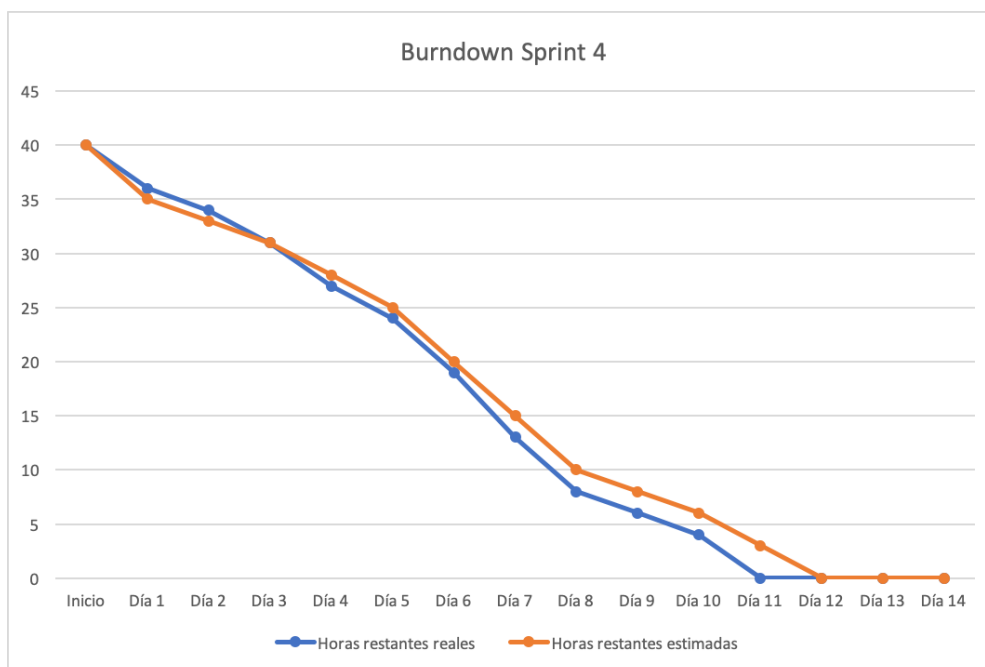


Figura 3.5: *Burndown* del cuarto *sprint*.

La Figura 3.5 muestra que, hasta el momento, el cuarto *sprint* ha sido el que más se ha ajustado a las estimaciones iniciales, aunque terminara un día antes de lo previsto.

## Quinto sprint

En el quinto *sprint* se va a completar la historia de usuario HU07, que hace referencia a las tareas de administración de la aplicación web. Se hicieron estimaciones teniendo en cuenta que habría que hacer el trabajo para administrar titulaciones, asignaturas y grupos.

ID HU	Tarea	Estimación (horas)	Tiempo real (horas)	Desviación
HU07	Programación UI	14	17	+3
HU07	Programación negocio	20	24	+4
HU07	Programación persistencia	5	3	-2
HU07	Pruebas	5	7	+2
HU07	Documentación	5	3	-2
Totales		49	54	+5

Tabla 3.8: *Sprint Backlog* e información de seguimiento del quinto *sprint*.

Como se desprende de la Tabla 3.8 en este quinto *sprint* tampoco fueron buenas las estimaciones, y terminaron dedicándose más horas de las establecidas para completarlo, como ya sucediera en el segundo y tercer *sprint*. Tan solo se pudo recuperar parte del tiempo en tareas que fueron muy parecidas a las de *sprints* anteriores.

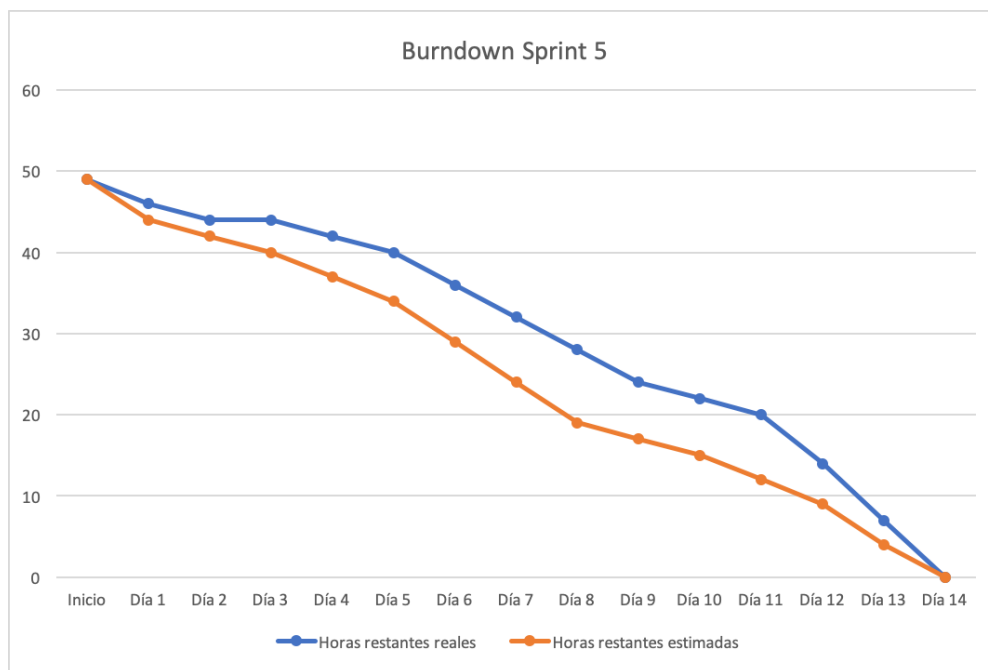


Figura 3.6: *Burndown* del quinto *sprint*.

La Figura 3.6 muestra lo comentado anteriormente sobre que es sólo al final del *sprint* cuando se consigue igualar el trabajo restante estimado con el real.

Para tener una visión global de lo que ha sido la evolución del trabajo en horas y las variaciones que se produjeron sobre las estimaciones realizadas se presentan la Tabla 3.9 y la Figura 3.7.

Nº Sprint	Estimación (horas)	Tiempo real (horas)	Desviación sprint	Desviación acumulada
Sprint 1	50	42	-8	-8
Sprint 2	45	44	-1	-9
Sprint 3	50	72	+22	+13
Sprint 4	40	37	-3	+10
Sprint 5	49	54	+5	+15
Totales	234	249	-	+15

Tabla 3.9: *Sprint Backlog* e información de seguimiento del cuarto sprint.

En la tabla se puede observar que al final del proyecto se tiene una desviación de 15 horas sobre lo estimado inicialmente, aproximadamente un 6% de la duración que se preveía.

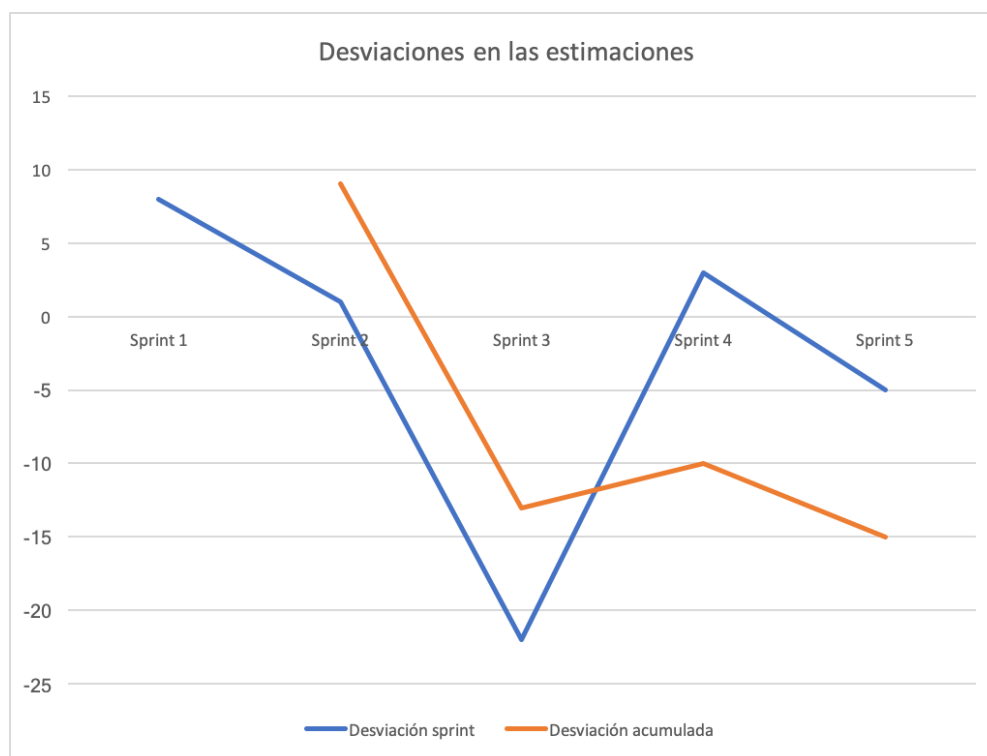


Figura 3.7: Desviaciones sobre las estimaciones realizadas.

Y en la figura se comprueba que sólo hubo dos *sprints* de los cinco que se acercaron realmente al número de horas estimadas, que son el segundo y el cuarto. Además la desviación acumulada del proyecto nunca se acercó a lo que debería haber sido la evolución normal, pasando de un registro positivo de 9 horas a otro ya negativo de 13 entre el segundo y tercer *sprint*.

A modo de conclusión sobre el apartado de planificación cabe indicar lo siguiente:

- Se han utilizado algunos de los elementos más relevantes del marco de trabajo Scrum.
- Las estimaciones realizadas no han sido buenas, teniendo en todos los *sprints* desviaciones, y en algunos bastante importantes.
- Dedicando horas adicionales de trabajo se ha conseguido completar todas las tareas marcadas para cada uno de los *sprints*.



## Capítulo 4

# Análisis del sistema

En este capítulo se presentarán todas las actividades correspondientes de la fase de análisis del proyecto.

Si bien es cierto que el problema aparentemente no tiene un grado de complejidad demasiado elevado, esta fase del trabajo reveló que tenía más peculiaridades de las aparentaba en un principio.

Por todo ello se dedicó una cantidad de tiempo razonable al análisis, con el objetivo de que las decisiones tomadas fueran adecuadas y no hubiera que hacer cambios posteriores que retrasaran el desarrollo del proyecto.

### 4.1. Especificación de la aplicación

Lo primero y más importante que se ha tenido en cuenta a la hora de desarrollar esta aplicación es la normativa existente en la Universidad de Valladolid con respecto a los horarios y su distribución.

Si bien es cierto que algunos centros, como por ejemplo la Facultad de Educación y Trabajo Social, han cerrado acuerdos a nivel de Junta de Facultad para aprobar su propia distribución horaria, ésta se ajusta siempre a lo marcado en el Reglamento de Ordenación Académica de la Universidad de Valladolid [20], que establece lo siguiente en su artículo 15:

***“Artículo 15. Distribución de horarios***

***15.1.*** *Los estudiantes podrán matricularse de asignaturas de cursos distintos, siempre que lo permita el plan de estudios que estén cursando. Se facilitará la asistencia de los estudiantes a las actividades presenciales de carácter obligatorio evitando, en la medida de lo posible, el solapamiento de horarios entre cursos sucesivos de la misma titulación.*

***15.2.*** *Las actividades docentes presenciales se distribuirán de lunes a viernes, de 8h a 15h en horario de mañana y de 15h a 22h en horario de tarde. En ambos casos, las primeras y últimas horas se consideran franjas de desbordamiento y serán empleadas sólo en el caso en que sea imprescindible por coherencia interna de los horarios y por disponibilidad de recursos. Las actividades presenciales, por tanto, se situarán normalmente dentro de la franja formada por las cinco horas restantes. Excepcionalmente, podrán proponerse actividades presenciales docentes en sábado siempre que se justifique adecuadamente dicha propuesta.”*

---

De esta normativa se han obtenido las franjas en las que las clases podrán tener lugar. Para el trabajo que aquí se presenta se han obviado las posibles actividades presenciales docentes que pudieran tener lugar en sábado, ya que el problema que se trata de resolver es el del solapamiento de horarios y la posibilidad de que esas actividades en sábado se solapen es demasiado remota como para ser tenida en cuenta.

Así pues, las horas consideradas en los posibles horarios que se generarán serán las comprendidas en la franja de 8 de la mañana a 10 de la noche en intervalos de una hora y de lunes a viernes.

Además de lo establecido en la normativa vigente, la primera fase del análisis centrada en la búsqueda de información proporcionó muchos datos que incrementaron la complejidad del problema y que serían tenidos en cuenta más adelante. La información recopilada se resume en los siguientes puntos:

- En los últimos años han aparecido titulaciones que se denominan “Programas de Estudios Conjunto” y que son titulaciones que están compuestas por asignaturas de otras dos titulaciones diferentes.
- Existen asignaturas que se imparten en varias titulaciones y el curso en el que se imparten varía de unas a otras. Esto sucede también con asignaturas de los “Programas de Estudios Conjunto”. En el caso del Grado en Ingeniería Informática y del Programa de estudios conjunto de Grado en Estadística y Grado en Ingeniería Informática sucede con varias asignaturas. Algunos ejemplos son:
  - Ingeniería del Conocimiento: se imparte en el cuarto curso del programa de estudios conjunto y en tercer curso del Grado en Ingeniería Informática (sólo para las menciones de Ingeniería del Software y Computación).
  - Fundamentos de Ingeniería del Software: se imparte en segundo curso del Grado en Ingeniería Informática y en el tercer curso del programa de estudios conjunto.
  - Sistemas Distribuidos: aparece como asignatura de segundo curso del Grado en Ingeniería Informática y asociada al cuarto curso del programa de estudios conjunto.
- Existen asignaturas que se imparten en titulaciones que constan de varias menciones. Algunas de estas asignaturas aparecen en diferentes cursos dependiendo de la asociación con la mención correspondiente. Por ejemplo, la asignatura “Arquitectura de Redes y Servicios” del Grado en Ingeniería Informática está relacionada con la titulación a través de dos menciones:
  - Mención de Ingeniería del Software: aparece como optativa de cuarto curso.
  - Mención en Tecnologías de la Información: aparece como optativa de tercer curso.
- Las asignaturas que tienen carácter de “anuales” pueden variar la carga lectiva entre semestres y su horario también.
- El Reglamento de Ordenación Académica diferencia entre docencia teórica y práctica, aunque en los horarios consultados de diferentes facultades de la Universidad de Valladolid, la diferenciación es mayor, encontrando clases de los siguientes tipos:
  - Teórica
  - Práctica
  - Seminario
- Algunas horas lectivas de algunas asignaturas tienen lugar sólo en determinadas semanas y/o días. Por lo general esta características se suele indicar en los horarios que publican los diferentes Centros de la Universidad de Valladolid. Además, en algunos de estos casos, las fechas en las que tienen lugar esas horas lectivas vienen indicadas en los correspondientes Proyectos Docentes.

- Aunque el Reglamento de Ordenación Académica lo establece en su Artículo 13, “*Concepto de calendario de actividades docentes*”, hay centros en la Universidad de Valladolid que no establecen unos horarios de manera clara antes del primer período de matrícula. Un ejemplo de ello es la Escuela de Ingeniería Informática de dicha Universidad, donde existen asignaturas para las que no se publican los horarios de manera completa. Un ejemplo de ello es la asignatura de Servicios y Sistemas Web. Se puede ver en las Figuras 4.1, 4.2, 4.3. cómo el grupo de laboratorio X3 sólo aparece en el horario de la mención de Tecnologías de la Información, cuando en realidad es accesible para estudiantes de las tres menciones del Grado en Ingeniería Informática. Se han indicado en rojo los grupos de laboratorio de dicha asignatura.

Grado en Ingeniería Informática, Segundo Cuatrimestre, Curso 2018-19

Tercer Curso, Mención *Ingeniería de Software*

	L	M	X	J	V	
9:00	ERSS T1 07	LP T1 07	TAA T1 07		LP T1 07	
10:00	DIS L2 L102	CPAR L1 07   L105	ERSS T1 04	CPAR T1 07   L105	TAA T1 07	SSW T1 07
11:00		DIS T1 04		LP L2 I+D	TAA L1 L106	DIS T1 07
12:00	DIS L1 L102	CPAR L2 07   L105	LP L1 L103			TAA L2 I+D
13:00			Charla de los Miércoles	SSW T1 07		
14:00						
15:00						
16:00	ECT T1 101	CPAR L3 07   L105	SSW X1 L101	ECT L 101	ERSS L1 L106	
17:00						
18:00		SSW X2 L101		ERSS L2 L106		
19:00						

Figura 4.1: Horario del segundo cuatrimestre del curso 2018/2019 del Grado en Ingeniería Informática de la Universidad de Valladolid, mención Ingeniería del Software, tercer curso.

Grado en Ingeniería Informática, Segundo Cuatrimestre, Curso 2018-19

**Tercer Curso, Mención *Tecnologías de la Información***

	L	M	X	J	V
9:00			ESI T1 05	SSW X3 L103	
10:00	CPAR L1 07   L105	SEMP T1 L105	CPAR T1 07   L105		SSW T1 07
11:00				DIAS T1 04	
12:00	CPAR L2 07   L105	ESI T1 03	Charla de los Miércoles		SEMP L1 L105
13:00				SSW T1 07	SEMP L2 L102
14:00					
15:00					
16:00	ESI L1 L102	CPAR L3 07   L105	SSW X1 L101	DIAS L2 L106	
17:00					
18:00	ESI L2 L102	SSW X2 L101	DIAS L1 L106		
19:00					

Figura 4.2: Horario del segundo cuatrimestre del curso 2018/2019 del Grado en Ingeniería Informática de la Universidad de Valladolid, mención Tecnologías de la Información, tercer curso.



**Tercer Curso, Mención *Computación***

	L	M	X	J	V
8:00		INFE1 L 08	INFE1 L 08	GLF T1 104	
9:00	INFE1 L 08	MTD T1 104	TAA T1 07		INFE1 L 08
10:00	CPAR L1 07   L105	SAII T1 104	CPAR T1 07   L105	TAA T1 07	SSW T1 07
11:00		MTD L 104		TAA L1 L106	MTD T1 104
12:00	CPAR L2 07   L105		Charla de los Miércoles		TAA L2 I+D
13:00				SSW T1 07	
14:00					
15:00					
16:00		CPAR L3 07   L105	SSW X1 L101	SAII L L101	GLF L 102   I+D
17:00					
18:00		SSW X2 L101			
19:00					

Figura 4.3: Horario del segundo cuatrimestre del curso 2018/2019 del Grado en Ingeniería Informática de la Universidad de Valladolid, mención Computación, tercer curso.

## 4.2. Modelo de dominio

Para realizar el modelo de dominio se ha tenido en cuenta la naturaleza del problema que se pretendía resolver, la normativa actual y se han tomado algunas decisiones para intentar que fuera lo más fiel posible a la realidad.

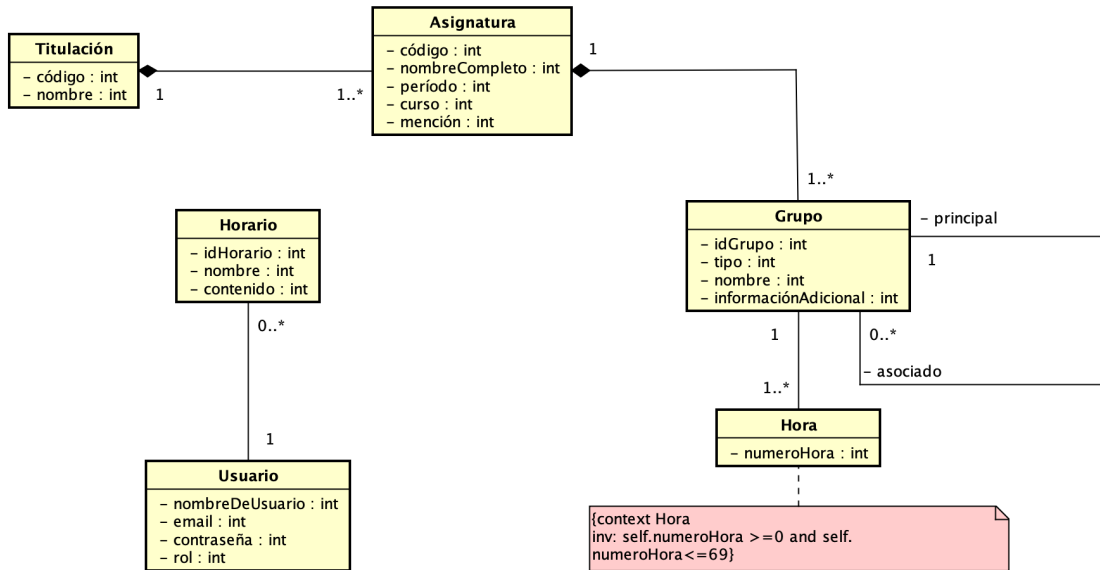


Figura 4.4: Modelo de dominio.

Los elementos que aparecen en la figura 4.4 se describen a continuación:

- **Titulación:** Representa las titulaciones introducidas en la base de datos de la aplicación y de las que se podrán obtener los horarios. Tiene los siguientes atributos:
  - *Código:* código único de la titulación.
  - *Nombre:* nombre de la titulación.
- **Asignatura:** Se refiere a las asignaturas que cursan los alumnos. Sus atributos son:
  - *Código:* código que identifica a la asignatura.
  - *Curso:* curso en el que se imparte la asignatura.
  - *NombreCompleto:* nombre de la asignatura.
  - *Mención:* mención o menciones a las que pertenece la asignatura.
  - *Período:* período en el que se imparte la asignatura (primer o segundo semestre).
- **Grupo:** Cada uno de los grupos de teoría o de prácticas que tienen las diferentes asignaturas.
  - *IdGrupo:* código del grupo.
  - *Tipo:* tipo de grupo entre teoría (T) y laboratorio/prácticas (L).
  - *Nombre:* nombre abreviado del grupo y que suele ser el que aparece reflejado en los horarios publicados por los diferentes centro de la universidad.
  - *InformaciónAdicional:* información de relevancia sobre el grupo y que deberían conocer los alumnos a la hora de configurar su horario, como por ejemplo las fechas en las que tienen lugar las clases de ese grupo si no tiene docencia todas las semanas.
- **Hora:** Almacena las horas en las que tienen lugar las clases de los diferentes grupos.
  - *NumeroHora:* número entero entre 0 y 69 que representa la hora dentro del horario lectivo considerado (lunes a viernes de 8 de la mañana a 10 de la noche) en la que tiene lugar la clase del grupo cuyo ID corresponde con el atributo anterior.

- 
- **Horario:** Hace referencia a los horarios que los alumnos guardan en el sistema.
    - *IdHorario:* identificador único del horario.
    - *Nombre:* nombre que el usuario propietario del horario le da al guardarlo.
    - *Contenido:* texto que almacena los identificadores de los grupos (ID\_Grupo) que conforman el horario.
  - **Usuario:** Representa a los usuarios registrados en el sistema, tanto alumnos como administradores.
    - *NombreDeUsuario:* identificador único del usuario en el sistema.
    - *Contraseña:* contraseña establecida por el usuario al registrarse.
    - *Email:* correo electrónico del usuario.
    - *Rol:* rol o categoría del usuario en el sistema.

A la hora de diseñar el modelo del dominio, y pensando ya en algunas partes del diseño como será la elaboración del diagrama entidad-relación, se decide adoptar las siguientes decisiones que afectan a la forma de interpretar el problema y que influirán en la solución propuesta:

- Los grupos son únicos para cada asignatura, es decir, que no podrá haber grupos que compartan dos asignaturas diferentes. Además se ha establecido una relación de composición, así que en caso de modificación o eliminación de asignaturas, los grupos asociados a dichas asignaturas se verán afectados.
- Las asignaturas pertenecen sólo a una titulación y, al igual que en el caso anterior, la relación establecida es de composición.
- Los grupos pueden estar asociados a otros grupos si son de tipo “Laboratorio”.
- Las titulaciones que se conocen como “Programas de Estudios Conjunto” se modelarán como titulaciones propias, y no como la suma de varias titulaciones. Esto provocará que tengan sus propias asignaturas.
- Para las asignaturas que se cursan en diferentes menciones, y además lo hacen en diferentes cursos de una misma titulación, el modelo contempla que sólo se almacene el curso inferior.

### 4.3. Historias de usuario

Las historias de usuario son la representación en forma escrita y en lenguaje común de los requisitos del sistema. Se ha optado por esta forma de definir los requisitos ya que es rápida y evitará el tener que crear una gran cantidad de documentos formales. Además, es probable que a medida que avance el desarrollo nos encontremos con problemas que no se conocían al principio del proyecto, y las historias de usuario permiten responder de forma rápida a esos requisitos cambiantes, por lo que son la mejor opción para abordarlo.

Entre las características más destacables que deben cumplir las historias de usuario se encuentran las siguientes:

- Ser independientes unas de otras.
- Ser negociables.
- Ser valoradas por los clientes o los usuarios.
- Estimables.
- Pequeñas.

- Verificables.

Atendiendo a dichas características y a la propia definición de lo que es una historia de usuario, se elaboró la lista para el presente proyecto. Se intentó que las historias quedasen lo menos abiertas posibles a interpretaciones, para evitar retrasos en la fase de desarrollo.

Por último, y tras analizar diferentes alternativas se ha decidido dejar reflejadas las historias de usuario en forma de tarjetas que contengan toda la información asociada a cada una de ellas. El diseño de la tarjeta es el siguiente:

ID		Nombre	
Descripción			
Nº tareas		Prioridad	Baja ○○○○○ Alta
Valor		Riesgo	
Criterios de aceptación			

Tabla 4.1: Modelo de tarjeta para las historias de usuario.

Y los campos que aparecen en ella son:

- **ID:** Identificador de la historia de usuario. Sigue el siguiente formato: HUXX, dónde XX representa un número asignado de forma progresiva entre 1 y 99.
- **Nombre:** Nombre corto utilizado para referirse de manera unívoca a la historia de usuario.
- **Descripción:** Breve explicación de la funcionalidad que se desea implementar a través de la historia de usuario.
- **Nº tareas:** El número de tareas que tiene asociada la historia de usuario.
- **Prioridad:** representa la prioridad de la historia de usuario con respecto al resto de las que aparecen en el product backlog. Se ha optado por un sistema de puntuación de gradiente cerrado para tener una mejor percepción de la prioridad entre historias.
- **Valor:** Puntuación que se otorga a la historia de usuario siguiendo el criterio establecido de valorarlas en función de la estimación de horas que va a llevar realizarlas. En varias historias de usuario la valoración ha variado a la hora de definir sus tareas, por lo que si aparecen dos cifras, la tachada se corresponde con la valoración inicial que se hizo y la otra es la valoración final.
- **Riesgo:** refleja la interpretación que hace el equipo de desarrollo sobre los posibles inconvenientes que pueden aparecer a la hora de desarrollar la historia y tiene en cuenta su grado de complejidad. Para este caso se ha decidido utilizar una escala de tres valores: bajo, medio y alto, siendo éste último indicativo de que la tarea implica mucho riesgo.
- **Criterios de aceptación:** Descripción de los criterios de aceptación que deberá cumplir la historia de usuario para poder marcarla como completada.

Las historias de usuario del proyecto a desarrollar se presentan a continuación con el formato de la Tabla 4.1:

ID_HU	HU01	Nombre	Registro en el sistema
Descripción	Como usuario quiero poder registrarme en el sistema para poder acceder a él		
Nº tareas	6	Prioridad	Baja ●●●●○ Alta
Valor	29	Riesgo	Medio
Criterios de aceptación	<ul style="list-style-type: none"> <li>- El formulario de registro tiene que ser accesible.</li> <li>- El usuario tiene que poder registrarse en el sistema.</li> <li>- En caso de que el nombre de usuario o el correo electrónico ya estén utilizados debe denegarse el registro e informar al usuario.</li> </ul>		

Tabla 4.2: Historia de usuario HU01.

ID_HU	HU02	Nombre	Acceso al sistema
Descripción	Como usuario quiero poder acceder a la aplicación para poder hacer uso de ella		
Nº tareas	4	Prioridad	Baja ●●●●○ Alta
Valor	16	Riesgo	Medio
Criterios de aceptación	<ul style="list-style-type: none"> <li>- El formulario de inicio de sesión tiene que ser accesible.</li> <li>- El usuario tiene que poder iniciar sesión en el sistema.</li> <li>- Si los datos introducidos no son correctos el usuario no podrá acceder al sistema y se le informará.</li> </ul>		

Tabla 4.3: Historia de usuario HU02.

ID_HU	HU03	Nombre	Generar horario
Descripción	Como usuario quiero poder generar un horario de acuerdo a las asignaturas que elija para poder asistir al mayor número de clases posible		
Nº tareas	5	Prioridad	Baja ●●●●○ Alta
Valor	50	Riesgo	Alto
Criterios de aceptación	<ul style="list-style-type: none"> <li>- El usuario tiene que poder elegir la titulación, el período y las asignaturas para las que quiere obtener el horario.</li> <li>- El sistema tiene que mostrar la mejor combinación encontrada atendiendo a dos criterios: el menor número de solapamientos de clase posibles y el mayor grado de compactación.</li> <li>- El sistema tiene que mostrar enlaces a otras posibles soluciones con las mismas características de solapamientos y grado de compactación.</li> </ul>		

Tabla 4.4: Historia de usuario HU03.

ID_HU	HU04	Nombre	Guardar horario
Descripción	Como usuario quiero guardar los horarios que genere para poder tener acceso a ellos en cualquier momento		
Nº tareas	5	Prioridad	Baja ●●●○○ Alta
Valor	23	Riesgo	Bajo
Criterios de aceptación	<ul style="list-style-type: none"> <li>- Al generar un horario nuevo abajo aparecerá un formulario al usuario con la opción de guardar el horario.</li> <li>- Si el usuario no rellena el formulario se informará del error al usuario.</li> <li>- El horario guardado aparecerá en el listado de horarios del usuario.</li> <li>- El usuario podrá cargar cualquier horario de los guardados si así lo desea.</li> </ul>		

Tabla 4.5: Historia de usuario HU04.

ID_HU	HU05	Nombre	Exportar horarios
Descripción	Como usuario quiero poder exportar mis horarios en diferentes formatos para después imprimirlos y/o almacenarlos en mis dispositivos		
Nº tareas	4	Prioridad	Baja ●●○○○ Alta
Valor	17	Riesgo	Medio
Criterios de aceptación	<ul style="list-style-type: none"> <li>- Al generar un horario nuevo abajo aparecerá un formulario al usuario con la opción de exportar a PDF el horario.</li> <li>- El horario se descargará en formato PDF y contendrá la tabla con el mismo estilo que el utilizado en la aplicación.</li> </ul>		

Tabla 4.6: Historia de usuario HU05.

ID_HU	HU06	Nombre	Funcionamiento de la aplicación
Descripción	Como usuario quiero poder utilizar la aplicación en diferentes tipos de dispositivos y/o navegadores para acceder a ella en cualquier momento y lugar		
Nº tareas	5	Prioridad	Baja ●●●●● Alta
Valor	50	Riesgo	Alto
Criterios de aceptación	<ul style="list-style-type: none"> <li>- La aplicación se ejecuta con normalidad en los principales navegadores web (Chrome, Safari, Firefox, Internet Explorer).</li> <li>- La aplicación es accesible desde dispositivos móviles (tablets y smartphones) y el diseño se ajusta a las pantallas de los dispositivos.</li> </ul>		

Tabla 4.7: Historia de usuario HU06.

ID_HU	HU07	Nombre	Administración
Descripción	Como administrador quiero gestionar la información de la aplicación en lo relativo a titulaciones, asignaturas, grupos, etc., para mantenerla actualizada.		
Nº tareas	5	Prioridad	Baja ● ○ ○ ○ ○ Alta
Valor	49	Riesgo	Bajo
Criterios de aceptación	<ul style="list-style-type: none"> <li>- Al iniciar sesión como administrador aparecerá un menú de administración para gestionar titulaciones, asignaturas y grupos.</li> <li>- El usuario administrador podrá realizar algunas modificaciones como cambios de denominación a través de los diferentes menús de gestión, pero habrá otras operaciones limitadas (por ejemplo el cambio de códigos de asignaturas y/o grupos).</li> </ul>		

Tabla 4.8: Historia de usuario HU07.

### 4.3.1. Tareas

Siguiendo la filosofía de Scrum, las historias de usuario del apartado anterior se han dividido en tareas. Como ya se indicó en el apartado 4.3, la definición de las tareas para cada elemento del *product backlog* se realizó al comienzo del sprint correspondiente.

En el caso de las tareas no se necesita tanto detalle como a la hora de definir las historias de usuario, y más siendo un proyecto en el que solo una persona está desarrollando la aplicación. Es por ello que se decidió definir las con una breve descripción además del nombre de las propias tareas.

A continuación se presentan relacionadas con cada una de las historias de usuario:

#### Historia de usuario HU01: Registro en el sistema

- **Configuración Base de Datos:** es la primera historia de usuario en la que se va a realizar interacción con la base de datos, por lo que hay que instalar el servidor, crear la base de datos y poblar la tabla “Users” que se utilizará para las pruebas.
- **Programación UI:** desarrollar el formulario de registro y la página inicial que verán los usuarios.
- **Programación negocio:** programar las clases involucradas en el registro de usuarios.
- **Programación persistencia:** además de programar las clases DAO, y debido a que es la primera vez que se utiliza la base de datos, hay que configurar la aplicación para que se conecte correctamente a dicha base de datos.
- **Pruebas:** comprobar que se cumplen los criterios de aceptación.
- **Documentación:** documentar el seguimiento del *sprint* y la memoria del proyecto.

#### Historia de usuario HU02: Acceso al sistema

- **Programación UI:** desarrollo del formulario de inicio de sesión en el sistema y programación del menú inicial dependiendo del rol del usuario que inicia sesión.

- 
- **Programación negocio:** programar las clases involucradas en el inicio de sesión. Se decide ampliar el modelo con una clase Login para seguir de una manera más estricta la arquitectura elegida.
  - **Pruebas:** comprobar que se cumplen los criterios de aceptación.
  - **Documentación:** documentar el seguimiento del *sprint* y la memoria del proyecto.

#### Historia de usuario HU03: Generar horario

- **Programación UI:** incluye el desarrollo de los menús de selección de titulación y período, el de elección de asignaturas y la salida del horario generado.
- **Programación negocio:** programación de todas las clases involucradas en el proceso de cálculo de posibilidades y generación del horario final.
- **Programación persistencia:** ampliar la base de datos creada y poblarla con datos suficientes como para poder llevar a cabo las pruebas.
- **Pruebas:** comprobar que se cumplen los criterios de aceptación.
- **Documentación:** documentar el seguimiento del *sprint* y la memoria del proyecto.

#### Historia de usuario HU04: Guardar horario

- **Programación UI:** diseño de la vista “Mis horarios” además de añadir un formulario para guardar el horario a la vista del horario ya creada.
- **Programación negocio:** programación de las clases involucradas en el proceso de guardar horarios. Se modifican algunos servicios para guardar el horario en formato texto.
- **Programación persistencia:** ampliación de la base de datos con la tabla “Timetables”.
- **Pruebas:** comprobar que se cumplen los criterios de aceptación.
- **Documentación:** documentar el seguimiento del *sprint* y la memoria del proyecto.

#### Historia de usuario HU05: Exportar horarios

- **Programación UI:** programación de formulario para descargar el fichero en formato PDF.
- **Programación negocio:** programar en javascript la función para descargar el fichero, lo que incluye documentarse sobre las librerías que se utilizan para estas tareas.
- **Pruebas:** comprobar que se cumplen los criterios de aceptación.
- **Documentación:** documentar el seguimiento del *sprint* y la memoria del proyecto.

#### Historia de usuario HU06: Funcionamiento de la aplicación

- **Programación UI:** diseño de la página de inicio de la aplicación.
- **Configuración despliegue remoto:** creación de cuenta en el servicio de alojamiento Heroku y puesta en marcha de una aplicación de prueba para comprobar que funciona.



- 
- **Configuración despliegue local:** instalación de servidor local con Apache Tomcat y despliegue de una aplicación de prueba para comprobar el buen funcionamiento.
  - **Pruebas:** comprobar que se cumplen los criterios de aceptación.
  - **Documentación:** documentar el seguimiento del *sprint*.

#### Historia de usuario HU07: Administración

- **Programación UI:** creación de los diferentes menús de administración.
- **Programación negocio:** programación de las clases involucradas. Aunque se espera que sea mucho trabajo se trata de algo repetitivo a la hora de gestionar asignaturas, grupos y titulaciones.
- **Programación persistencia:** corregir errores en el diseño de la base de datos además de programar las clases DAO que intervienen en la historia de usuario.
- **Pruebas:** comprobar que se cumplen los criterios de aceptación.
- **Documentación:** documentar el seguimiento del *sprint*.



## Capítulo 5

# Diseño del sistema

### 5.1. Arquitectura

#### 5.1.1. Visión global

La aplicación se ha construido siguiendo una arquitectura basada en capas. Sigue un modelo de desarrollo de software en el que el objetivo es desacoplar las partes que componen la aplicación ya que éstas están destinadas a operaciones muy diferenciadas. Se trata de separar roles y dividir responsabilidades entre la capa de presentación, la de negocio y la de persistencia.

Los componentes de las diferentes capas se comunican con los de otras a través de interfaces y la mayoría de la interacción sólo ocurre entre capas contiguas, tal y como muestra la Figura 5.7.

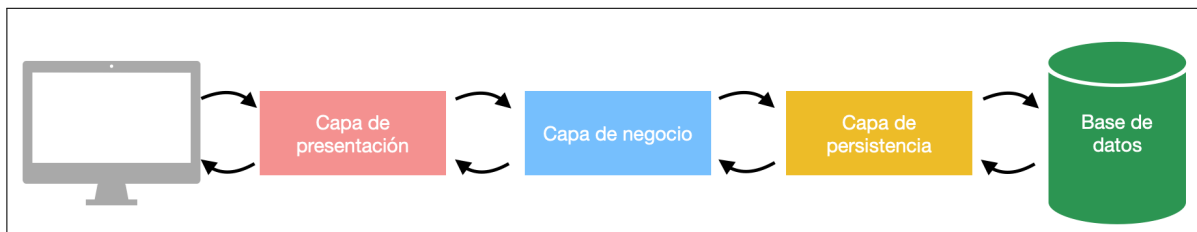


Figura 5.1: Representación simplificada de una arquitectura de 3 capas.

Aunque existen otras variantes, para el proyecto se ha elegido la de tres capas, cuyas características son las siguientes:

- Capa de presentación: es la capa con la que interacciona el usuario directamente. Se le presentan los resultados y se capturan sus acciones. Sólo se debe comunicar con la capa de negocio.
- Capa de negocio: es donde reside la lógica de los programas. Se reciben las peticiones procesadas en la capa de presentación y, en caso de necesitarlo, se envían las solicitudes correspondientes a la capa de persistencia. Una vez se ha procesado la solicitud se devuelve a la capa de presentación el resultado.
- Capa de persistencia: es la capa encargada del acceso a datos. Recibe las peticiones desde la capa de negocio y las realiza al sistema gestor de la base de datos elegida.

A la hora de aplicar esta arquitectura se busca que el sistema construido cumpla con una serie de principios de desarrollo de software, entre los que están:

- Alta cohesión: cada una de las capas construidas tiene sólo la funcionalidad correspondiente a esa capa.
- Reutilización: las capas de negocio y persistencia no tienen dependencia con la capa de presentación, por lo que su contenido puede reutilizarse en otros entornos.
- Abstracción: se abstrae la vista del modelo, pero facilitando el suficiente nivel de detalle como para entender la comunicación entre capas.
- Desacoplamiento: la comunicación entre las tres capas se basa en la abstracción, lo que facilita que no exista acoplamiento entre ellas.

En la aplicación desarrollada la estructura es la siguiente:

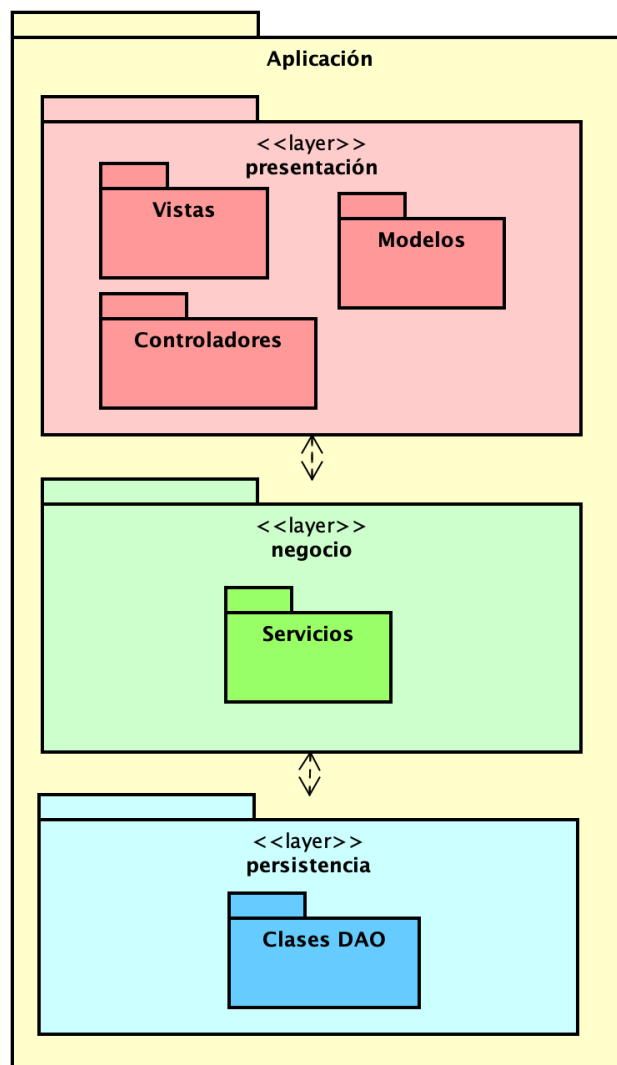


Figura 5.2: Arquitectura en tres capas de la aplicación.

A continuación se presenta la estructura seguida en las diferentes capas que forman la arquitectura de la aplicación. Con el fin de obtener una mayor claridad en los esquemas se ha prescindido de añadir los atributos y/o métodos de cada una de las clases que aparecen.

En la Figura 5.3 se puede observar la estructura de la capa de presentación. Se ve como la vista y el controlador hacen uso de los modelos desarrollados. Además es el controlador el que actúa de intermediario para atender peticiones y servir la información.

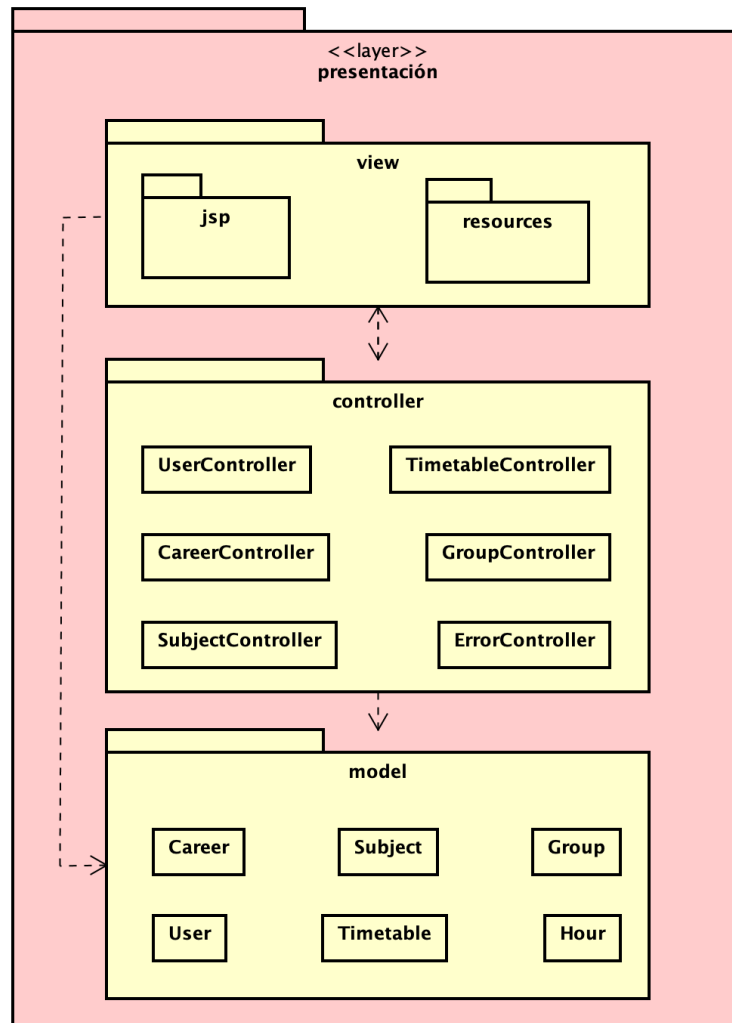


Figura 5.3: Capa de presentación de la aplicación.

En la siguiente figura, la 5.4 se muestra la estructura de la capa de negocio. Todas las clases implementadas tienen asociadas sus interfaces que es a través de las cuáles se entablarán las comunicaciones con otras capas.

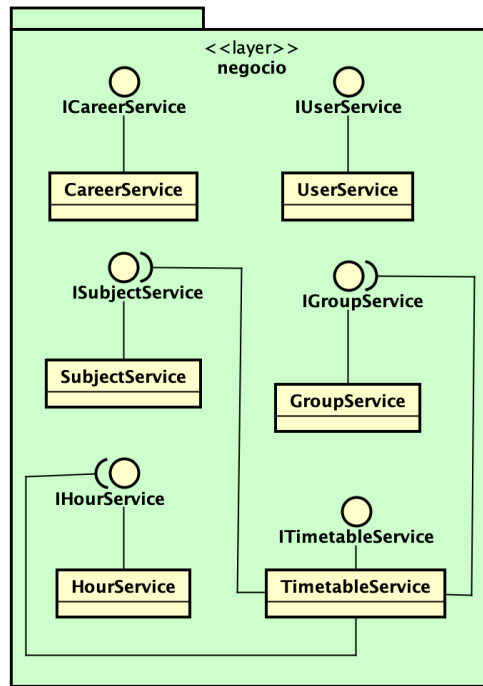


Figura 5.4: Capa de negocio de la aplicación.

Y la tercera capa de la arquitectura se muestra en la Figura 5.5. Al igual que en la capa de negocio las clases cuentan todas con una interfaz que es a través de la que se accede desde el exterior.

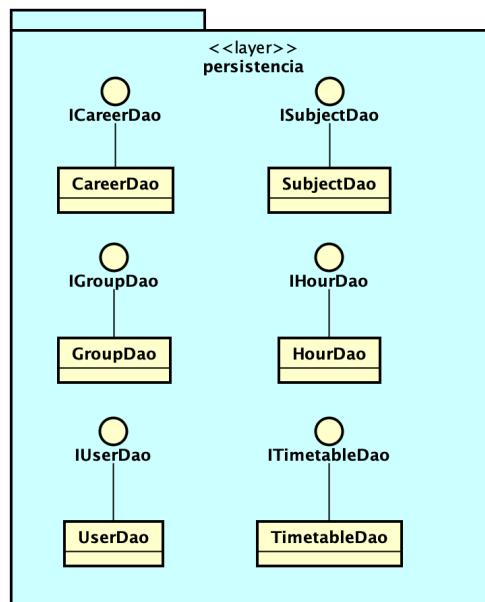


Figura 5.5: Capa de persistencia de la aplicación.

### 5.1.2. Spring Web MVC

Es el framework original para desarrollo web y se encuentra dentro de lo que se conoce como “Spring Framework”. En la página oficial [21] se clasifica en el “Servlet Stack” tal y como muestra la Figura 5.6.

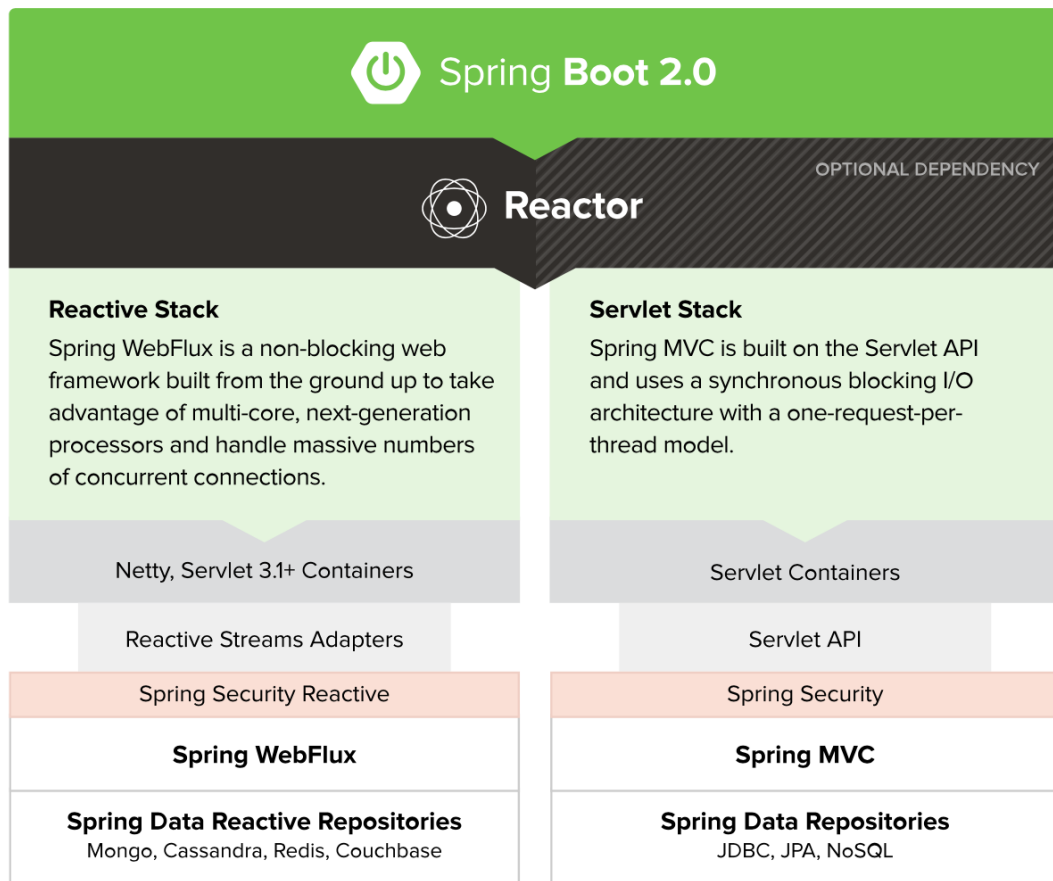


Figura 5.6: Stack de Spring Framework.

El framework Spring hace uso del principio de diseño de software denominado inversión de control, a través del cual se invierte el flujo de ejecución de un programa. Se reciben las peticiones y se establecen las respuestas que hay que darlas, pero no se establece un orden de respuesta fijo.

Spring Web MVC está diseñado en base al patrón “Front Controller” donde un servlet central, que en este caso se denomina DispatcherServlet, proporciona la posibilidad de gestionar las peticiones que llegan, enviándolas a los componentes que corresponda. Este modelo de trabajo ofrece mucha flexibilidad, permitiendo diferentes flujos de trabajo dentro de una misma aplicación [Referencia Docs Spring io](#).

Entre sus características cabe destacar las siguientes:

- Separación de roles.
- Adaptable y flexible.
- Reutilización de código de negocio.
- Soporte para JSP y JSTL.

Spring Web MVC es un marco de trabajo dirigido por peticiones, y en el centro se encuentra el ya mencionado DispatcherServlet. El flujo de trabajo que siguen las peticiones de manera habitual se puede ver en la Figura 5.7, extraída de [22].

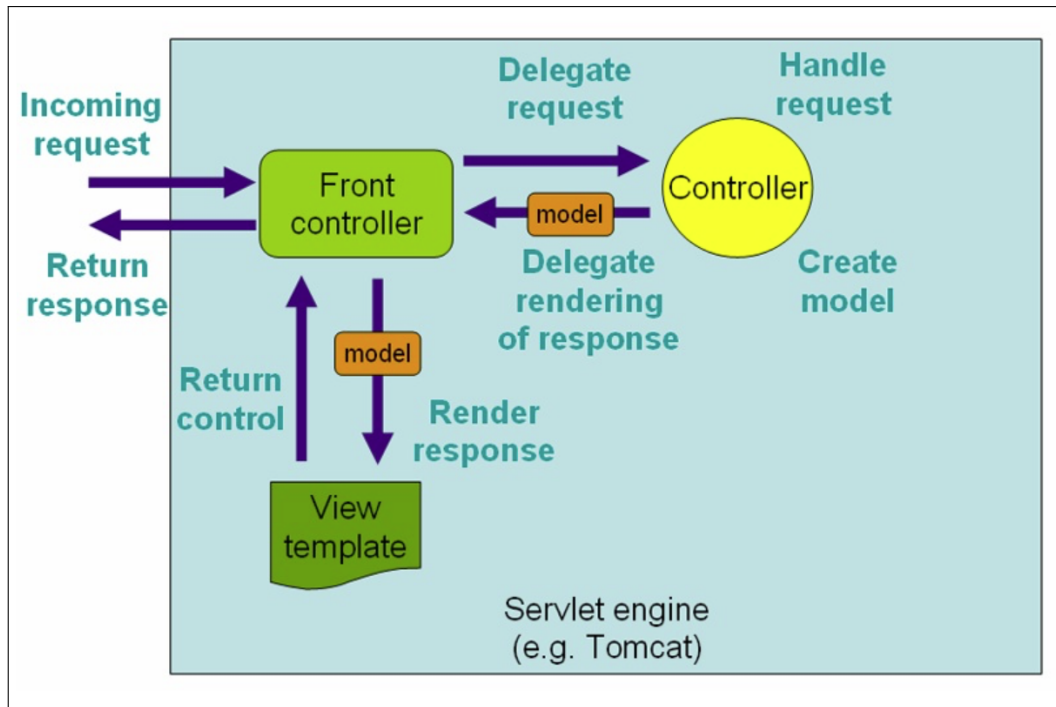


Figura 5.7: Flujo de trabajo del procesamiento de peticiones en Spring Web MVC a alto nivel.

Cualquier petición que llega se envía al “Front Controller”, que decide a qué controlador le corresponde resolver la petición. El controlador encargado de la petición se la envía a la clase de servicio que corresponda, y una vez que se han realizado las tareas correspondientes para procesar la petición, el propio controlador recibe, de la capa de servicio o de la de datos, el modelo que corresponda. Por último el controlador envía ese modelo al “Front Controller”, que se encarga de añadir el modelo a la vista que corresponda y se la envía al navegador para que se muestre al usuario.

## 5.2. Patrones de Diseño

### 5.2.1. Modelo Vista Controlador

Se trata de un patrón de arquitectura de software. Basándose en la idea de separación de conceptos lo que hace es separar los datos de la lógica de negocio y de la presentación de la información a los usuarios. Para ello propone la construcción de tres componentes que son el modelo, la vista y el controlador. En la Figura 5.8, obtenida de [23] se puede ver la relación entre los componentes de este patrón.



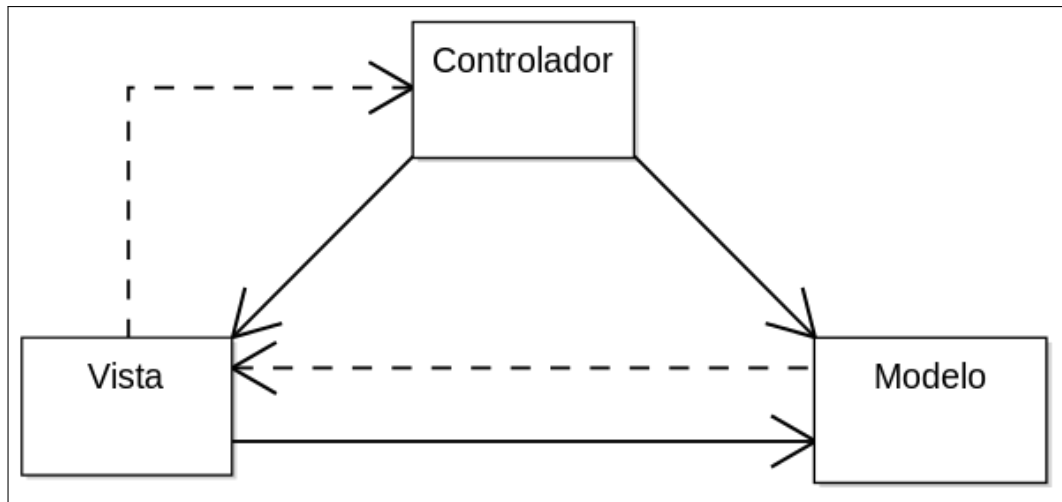


Figura 5.8: Relación de componentes en el patrón Modelo-Vista-Controlador.

Las funciones que desempeñan son, de manera genérica, las siguientes:

- **Modelo:** representa la información que maneja el sistema y gestiona el acceso a la misma.
- **Vista:** es la encargada de presentar el modelo al usuario para que pueda interactuar con la aplicación.
- **Controlador:** es el encargado de “controlar”, en el sentido de que procesa las peticiones que el usuario realiza a través de la vista y solicita datos al modelo para atender esas peticiones.

### 5.2.2. Data Access Object

Es un patrón que nos permite aislar la capa persistencia de la de negocio abstrayendo la forma de acceso a la base de datos. Los principales elementos son estos:

- **Intefaz Data Access Object:** define las operaciones estándar que se realizarán en el modelo de objeto.
- **Clase Data Access Object:** es la clase que implementa la interfaz DAO. Es la responsable del acceso a los datos y de la lógica de los métodos definidos.
- **Modelo de objeto:** objeto que tiene los métodos para almacenar los datos que se han obtenido utilizando la clase DAO.

La siguiente figura muestra un ejemplo de cómo se comunican las clases participantes en este patrón [24]:

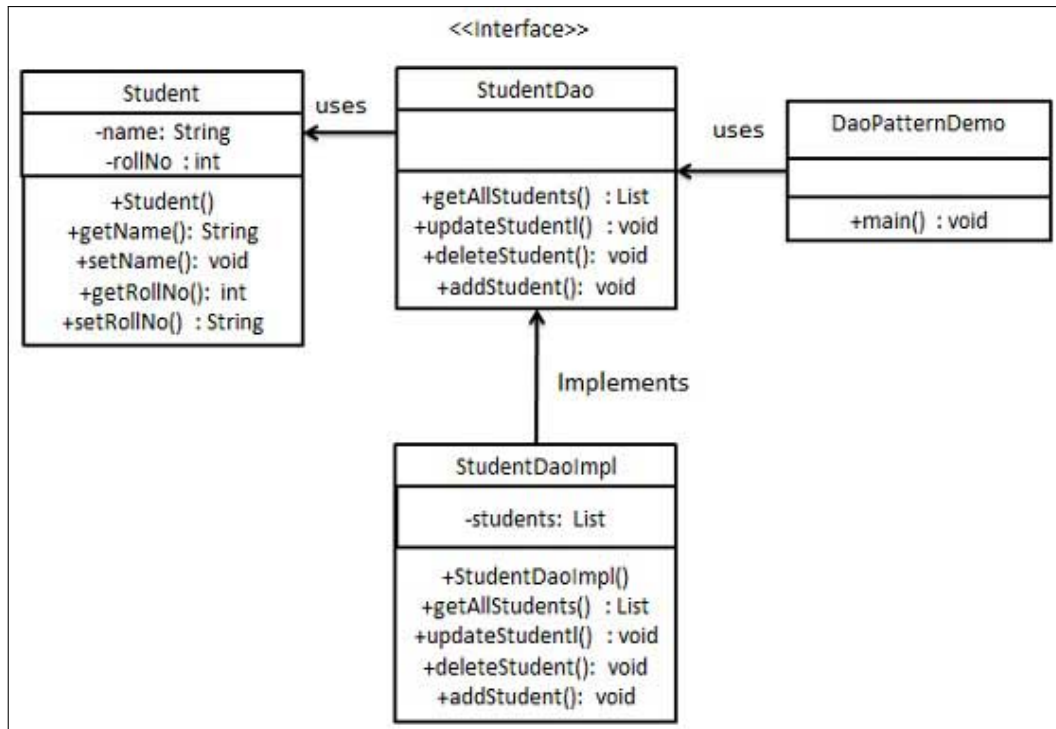


Figura 5.9: Ejemplo de diagrama de patrón DAO.

El uso de este patrón para gestionar el acceso a los datos dentro de una aplicación proporciona una serie de ventajas:

- Disminuye el acoplamiento entre los diferentes componentes de la aplicación.
- En caso de que la capa de persistencia se modifique, la capa de servicio no tendrá que modificarse, ya que simplemente recibe los datos, sin preocuparse del origen.
- Será más fácil desarrollar test unitarios para la aplicación.

### 5.2.3. Template

Sin desarrollarse de manera específica para la aplicación, sí que se ha utilizado en las clases que hacen uso de Spring JdbcTemplate. Este patrón permite dividir una funcionalidad en partes reutilizables y realizar una implementación por defecto de la mayoría de ellos. Un ejemplo de cómo funciona se ve en la Figura 5.10, sacada de [25]. Los JdbcTemplates permiten eliminar la mayor parte del código que habría que desarrollar cada vez que se realiza la consulta, recogiendo los datos y construyendo los objetos.

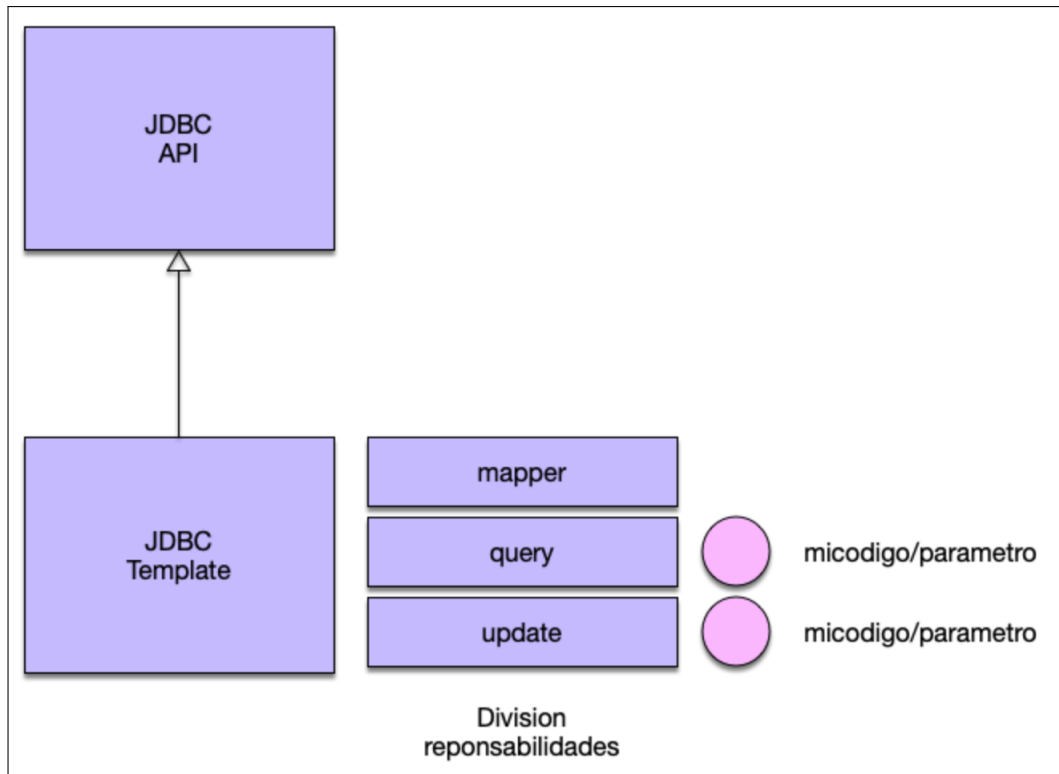


Figura 5.10: Ejemplo de funcionamiento del patrón Template para Spring JdbcTemplate.

### 5.3. Despliegue

El despliegue de la aplicación se realizó de dos maneras diferentes, en local y en remoto. Aunque la aplicación funcionaba en ambas versiones, se decidió optar por el despliegue local ya que era el que menos problemas suponía en cuanto a tiempos de respuesta del servidor de la base de datos.

#### Despliegue remoto

En el despliegue remoto intervienen dos máquinas. Una es el servidor de Heroku, servicio que se detalla en la sección 6.1, y otra una máquina virtual facilitada por los técnicos de la Escuela de Ingeniería Informática de la Universidad de Valladolid donde se instala el servidor MySQL.

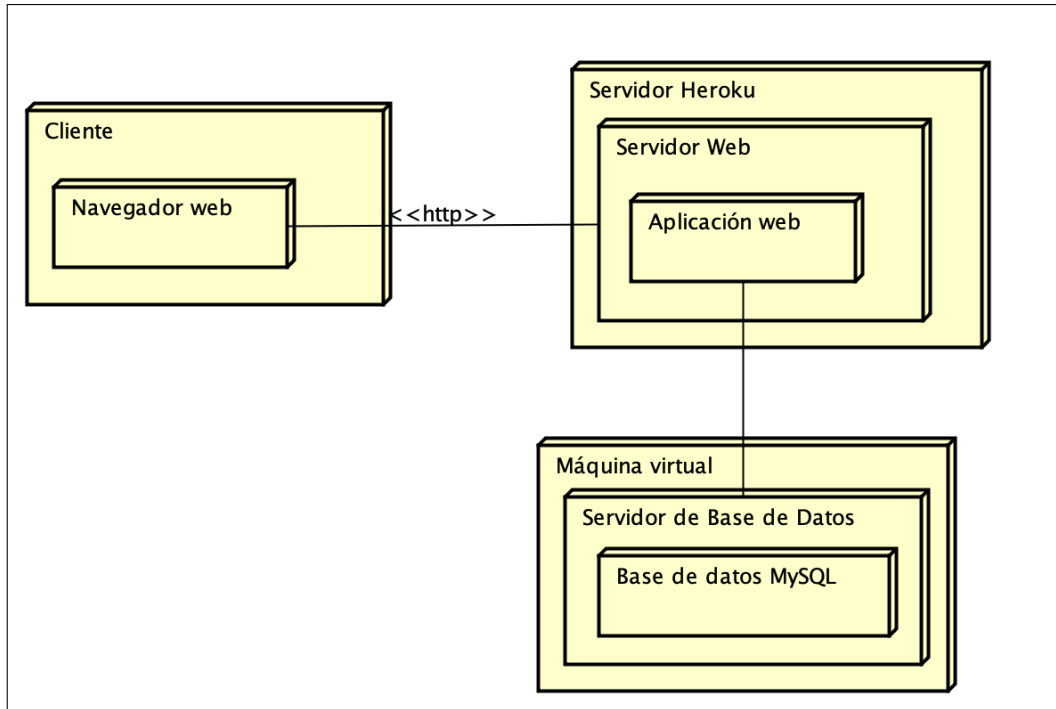


Figura 5.11: Diagrama de despliegue remoto de la aplicación.

### Despliegue local

En esta versión el despliegue se hace en una sola máquina que funciona como servidor de aplicaciones web, servidor de base de datos y además desde la que se ejecuta el navegador en el que se carga la aplicación.

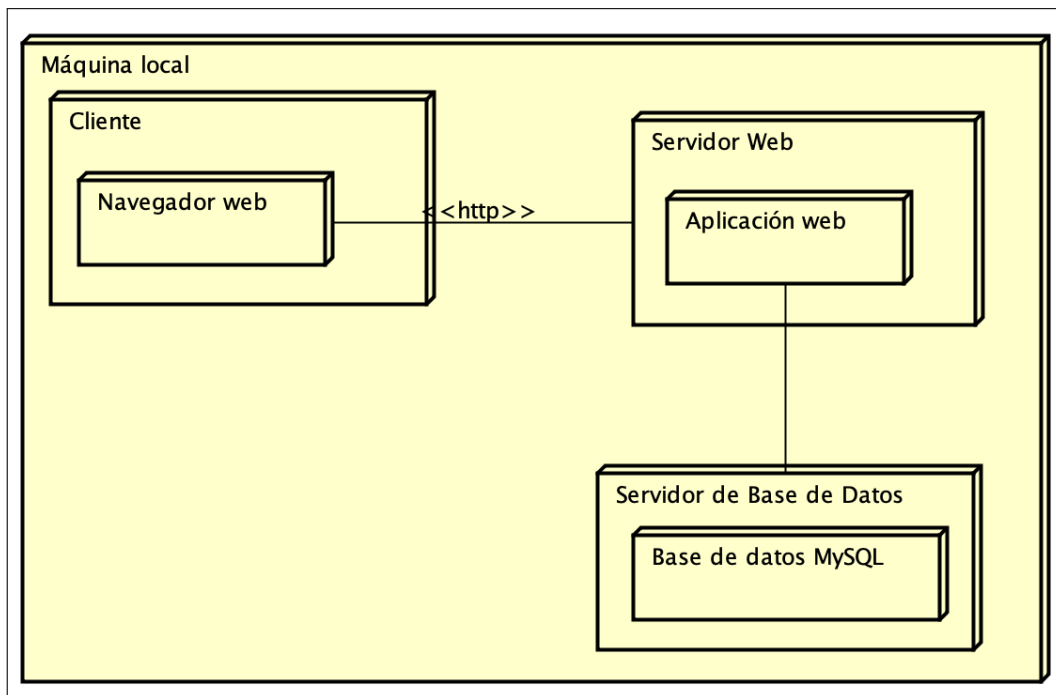


Figura 5.12: Diagrama de despliegue local de la aplicación.

## 5.4. Diseño de la base de datos

### 5.4.1. Modelo Entidad-Relación

Se trata de un modelo que está basado en la percepción del mundo real y que sirve para diseñar esquemas que posteriormente implementaremos en algún sistema gestor de bases de datos. Dicho diagrama se ha elaborado nombrando las diferentes entidades y sus atributos de la manera más ajustada posible a la realidad para que luego sean fácilmente identificables en la base de datos. Las claves primarias de las diferentes entidades se muestran con el estereotipo «PK».

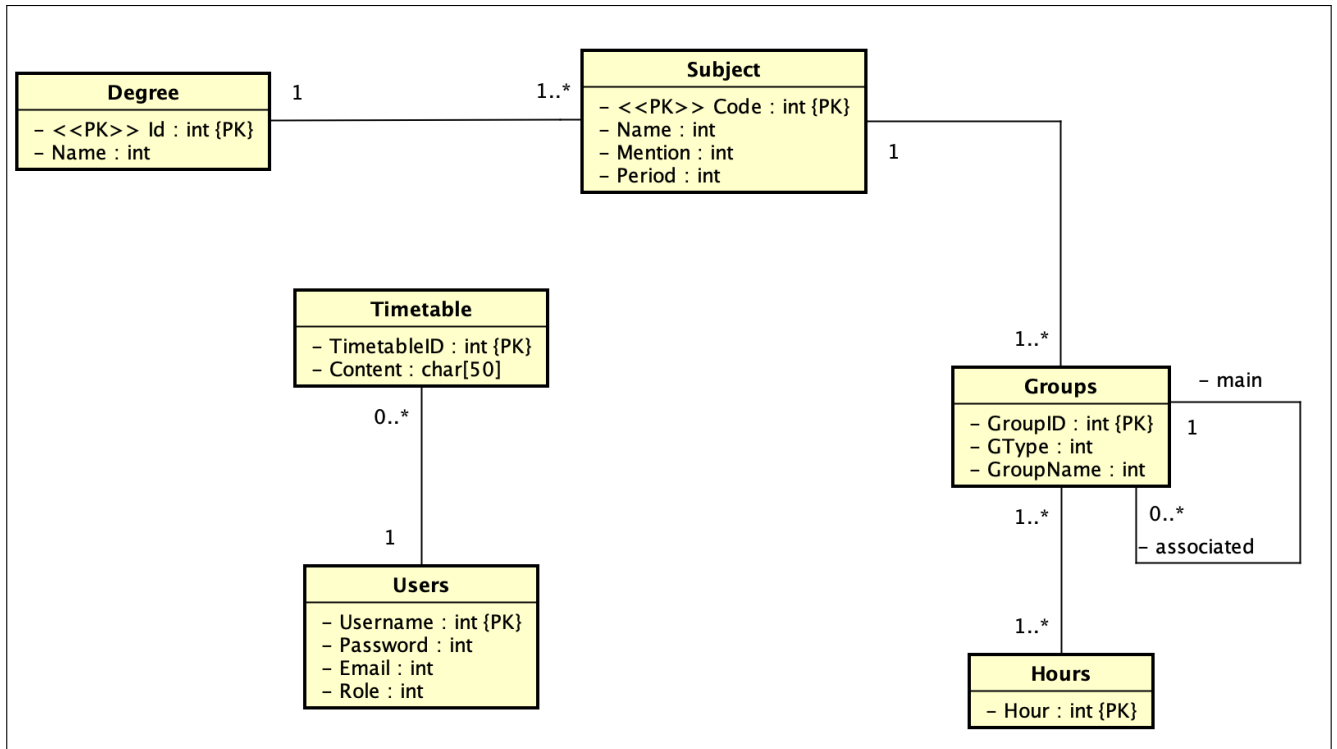


Figura 5.13: Modelo Entidad-Relación de la aplicación.

### 5.4.2. Modelo Relacional

El modelo relacional es un modelo de datos basado en la lógica de predicados y en la teoría de conjuntos. Según [26] es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente y cuenta con las siguientes ventajas:

- Provee herramientas que permiten evitar la duplicidad de registros.
- Garantiza la integridad referencial.
- Favorece la normalización.

Muestra la estructura de la base de datos, incluyendo las tablas, las columnas de cada una, las claves primarias (aparecen subrayadas en el diagrama) y las claves foráneas, que son aquellas de las que salen las flechas del diagrama. El modelo relacional construido para la base de datos del proyecto puede verse en la Figura 5.14.

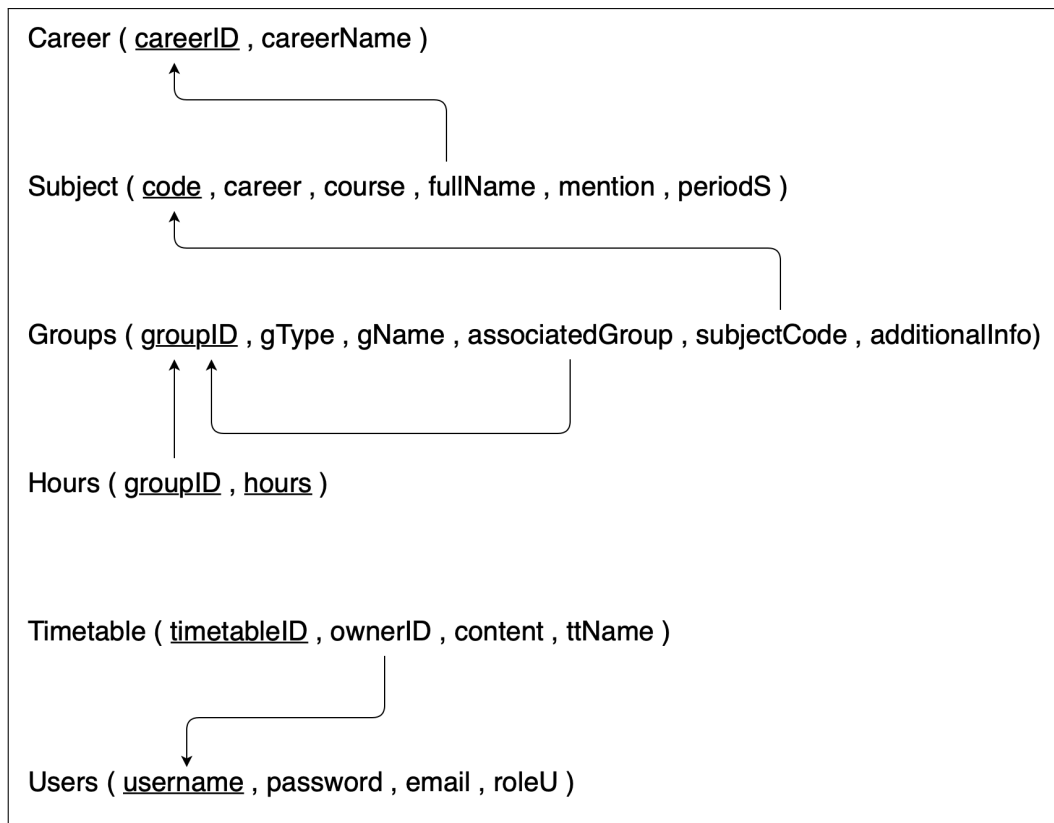


Figura 5.14: Modelo relacional de la base de datos.

El detalle sobre las tablas, sus columnas y la manera en que están relacionadas se explica a continuación:

- **Career:** tabla que representa las titulaciones almacenadas en la base de datos. Consta de dos columnas:
  - *careerID*: es el identificador de la titulación y que se corresponde con el número asociado al Plan de Estudios. Es la clave primaria de la tabla.
  - *careerName*: el nombre de la titulación.
  
- **Subject:** tabla reservada para representar las asignaturas. Tiene 6 columnas:
  - *code*: código único para identificar cada asignatura.
  - *career*: clave foránea asociada al campo *careerID* de la tabla *Career* y que representa la titulación a la que está asociada la asignatura.
  - *course*: curso en el que se imparte la asignatura.
  - *fullName*: nombre completo de la asignatura.
  - *mention*: mención o menciones en las que se imparte la asignatura. Si hay varias se representan como una cadena separada por comas.
  - *periodS*: período en el que se imparte la asignatura. Puede ser un número entero entre 1 y 2.
  
- **Groups:** contiene la información de cada uno de los grupos que existen para cada asignatura.

- 
- *groupID*: es la clave primaria de la tabla y representa el identificador único del grupo.
  - *gType*: tipo de grupo. Puede ser “T” si el grupo es de teoría o “L” si el grupo es de laboratorio.
  - *gName*: nombre corto del grupo que se mostrará en la salida al usuario para que lo identifique fácilmente.
  - *associatedGroup*: para los grupos de teoría este campo es nulo, y para los de laboratorio contiene el identificador del grupo de teoría al que están asociados. Es clave foránea del campo *groupID* de esta misma tabla.
  - *subjectCode*: el código de asignatura a la que pertenece el grupo.
  - *additionalInfo*: información adicional que puede ser de interés para los usuarios. Por ejemplo si el grupo sólo se imparte en determinadas fechas.
- **Hours**: tabla cuyas tuplas representan las horas en las que tienen lugar las clases de los diferentes grupos. La clave primaria de esta tabla está compuesta por los campos *groupID* y *hours*.
- *groupID*: clave foránea que hace referencia al campo *groupID* de la tabla *Groups* y que representa el identificador del grupo para el que se imparte esta hora.
  - *hours*: número comprendido entre 0 y 69 que representa la hora de la franja considerada en la que se imparte el grupo.
- **Timetable**: tabla que almacena la información de los horarios que los usuarios guardan en la base de datos.
- *timetableID*: identificador único del horario en la base de datos. Es la clave primaria de la tabla.
  - *ownerID*: clave foránea de la tabla asociada al campo “username” de la tabla “Users” y que representa al usuario al que pertenece el horario.
  - *content*: cadena que contiene los identificadores de los grupos que conforman el horario.
  - *ttName*: nombre con el que el usuario guarda el horario en el sistema.
- **Users**: representa la información de los usuarios registrados en el sistema.
- *username*: clave primaria de la tabla que representa el nombre de usuario en el sistema de cada usuario.
  - *password*: cadena que contiene la contraseña.
  - *email*: correo electrónico asociado a la cuenta de usuario.
  - *roleU*: número entero que representa el rol del usuario en la aplicación.





# Capítulo 6

## Implementación

### 6.1. Herramientas y tecnologías utilizadas

En este apartado se dividen en diferentes categorías y se describen brevemente las herramientas y tecnologías utilizadas durante el desarrollo de todo el Trabajo de Fin de Grado.

#### Gestión del proyecto

##### Git

Software de control de versiones diseñado para que las tareas de mantenimiento del código fuente en proyectos de cierto calibre fuese más fácil y eficaz. Se trata de un software libre distribuible bajo la Licencia Pública General de GNU.

##### GitHub

Plataforma online para almacenar proyectos utilizando el sistema de control de versiones Git. Se ha utilizado la cuenta personal del autor del presente trabajo y se crearon dos repositorios, uno para la aplicación y otro para los scripts de creación y población de la base de datos.

#### Diseño y desarrollo

##### Eclipse IDE

Es el Entorno de Desarrollo Integrado de Eclipse, que es una plataforma de desarrollo de software compuesta por multitud de herramientas de programación de código abierto. En lugar de incluir todas las funciones por defecto, el entorno proporciona sólo la que el usuario requiera a través de la instalación de módulos .

En el proyecto se ha utilizado para desarrollar todo el código fuente de la aplicación, tanto la parte del servidor como la del cliente. Los detalles de la versión utilizada se muestran en la siguiente tabla:

---

Versión	2018-12 (4.10.0)
Build id	20181214-0600

Tabla 6.1: Información del servidor Tomcat utilizado.

## Astah UML

Herramienta de modelado UML que también se conoce por el nombre de JUDE (Java and UML Developers' Environment) que se ha utilizado para realizar los diagramas que aparecen en el presente documento. Se trata de la herramienta con la que ha trabajado el alumno durante sus estudios de Grado en Ingeniería Informática, y aunque no se disponía de licencia al haber terminado el acuerdo con la Universidad de Valladolid, se solicitó una de carácter temporal.

## Visual Studio Code

Editor de código fuente gratuito y de código abierto. Se puede personalizar a través de módulos para dar soporte a multitud de lenguajes de programación. Permite la depuración de código, control integrado de Git y refactorización de código, entre otras características.

## JSFiddle

Editor de código online que permite utilizar diferentes lenguajes y frameworks de programación. Se ha utilizado para realizar pruebas del código javascript y de la parte de diseño de la interfaz con CSS.

## Draw.io

Herramienta online para realizar diseños y diagramas de todo tipo, entre los que se incluyen diagramas de clases, de casos de uso, de secuencia, de tipo entidad-relación, etc. Draw.io [referencia](#) permite además exportar los diagramas realizados en multitud de formatos. En este caso se utilizó para diseñar el modelo relacional porque ya se había trabajado con dicha herramienta en ocasiones anteriores y, además, para este tipo de diagrama es más sencilla y fácil de usar que otras.

## Back-end

### Spring Web MVC

Tal y como se indicó en el apartado 5.1.2 del capítulo anterior se trata del framework de desarrollo web utilizado para la programación de la aplicación. Es el marco de trabajo original para desarrollo web, está construido sobre la API Servlet y se ha incluido desde un inicio en lo que se denomina Framework Spring.

### MySQL

Sistema de gestión de bases de datos relacional. Es un sistema ideal para aplicaciones web, ya que habitualmente estas tienen una baja concurrencia en la modificación de datos y la lectura de datos suele ser mucho más frecuente,

---

y es en la lectura de datos donde el entorno actúa mejor. Se ha utilizado para implementar la base de datos de la aplicación.

## Maven

Es una herramienta para construir y gestionar proyectos en Java. Se ha utilizado para gestionar las dependencias externas.

## Front-end

### Bootstrap

Es un framework multiplataforma y de código abierto que se ocupa de la parte de desarrollo *front-end*. Cuenta con multitud de elementos de diseño basados en HTML y CSS, además de extensiones JavaScript. Es compatible con la mayoría de navegadores web y desde su versión 2.0 soporta diseños adaptativos. Para el desarrollo de la aplicación se utilizó la versión 3.3.7.

### JavaScript

Es un lenguaje de programación interpretado y orientado a objetos. Se suele utilizar en la parte del cliente y es interpretado por todos los navegadores web actuales. Se utiliza para obtener mejoras en el apartado de la interfaz de usuario y añadir elementos dinámicos a las páginas web. En este trabajo se ha utilizado para validación de formularios, búsqueda dinámica en tablas y para generar ficheros PDF a partir de cierto contenido web.

## Otras herramientas

### Apache Tomcat

También conocido como Jakarta Tomcat o sólo como Tomcat, es un contenedor web y no un servidor de aplicaciones. Puede funcionar como un servidor web y funciona en cualquier máquina que tenga un sistema operativo que disponga de la máquina virtual Java (JVM).

En este caso se ha utilizado como servidor web para desplegar de manera local la aplicación desarrollada. La información de la versión utilizada es la siguiente:

Versión de Tomcat	Apache Tomcat/9.0.4
Versión JVM	1.8.0_144-b01

Tabla 6.2: Información del servidor Tomcat utilizado.

### Heroku

Es una plataforma de computación en la nube que soporta diferentes lenguajes como Ruby, Java, Node.js o Python, entre otros. En este caso se utilizó para realizar pruebas de despliegue en un entorno real y probar la aplicación fuera de la máquina local.

---

## Microsoft Excel

Aplicación de hojas de cálculo que se incluye dentro de la suite de aplicaciones Microsoft Office. Se utilizó solo para las tareas de población de la base de datos, permitiendo introducir los datos en tablas y generar automáticamente las consultas SQL. Se utilizó la suscripción a Office 365 proporcionada por la Universidad de Valladolid y la versión utilizada fue la “15.38”.

## 6.2. Interfaz de usuario

Aunque se tenía una idea previa sobre lo que sería la interfaz del usuario y la manera de presentar la información no se realizaron bocetos previos, sino que se fue construyendo a medida que el proyecto avanzaba.

Lo más destacable del diseño de la interfaz del usuario es que se ha realizado siguiendo algunas directrices para que dicho diseño fuese adaptativo. En la actualidad la gente accede a aplicaciones web desde multitud de dispositivos como ordenadores, tablets, smartphones, etc. Para que la experiencia de usuario a la hora de utilizar la aplicación sea prácticamente idéntica independientemente del dispositivo en la que la esté ejecutando es necesario que la interfaz se adapte a ese dispositivo. A continuación se muestran ejemplos de cómo se ve la aplicación en diferentes dispositivos y navegadores:

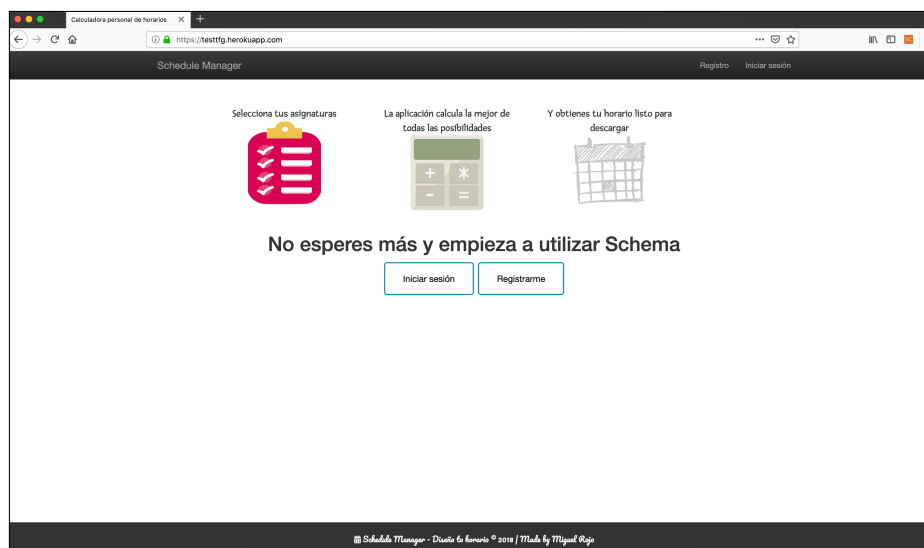


Figura 6.1: Interfaz del usuario desde un navegador Firefox.



Figura 6.2: Interfaz del usuario desde un iPad y un navegador Safari.



Figura 6.3: Interfaz del usuario desde un navegador de un smartphone Android.

Se han adaptado la mayoría de elementos, y otros se han fijado para que la información sea de utilidad al usuario. Nos estamos refiriendo a la salida por pantalla del horario, que no se puede adaptar del todo cuando la pantalla es demasiado estrecha, ya que sino el usuario tendría casi que leer en vertical. En dicho caso se ajustará hasta un determinado momento y el usuario, para poder consultar todo el horario, tendrá que desplazarse por la página lateralmente, tal y como se puede ver en la Figura ??.

Schedule Manager

todas en la parte inferior de la página.

#	Lunes	Martes	Miércoles	Jueves
8:00				
9:00	Planificación y Gestión de Proyectos (PGP Teoría 1)	Análisis y Diseño de Algoritmos (ADA Teoría 1)	Planificación y Gestión de Proyectos (PGP Teoría 1) Análisis y Diseño de Bases de Datos (ADB Teoría 1)	
10:00	Desarrollo Basado en Componentes y Servicios (DBCS Teoría 1) Ingeniería del Conocimiento (ICO Teoría 1)	Análisis y Diseño de Bases de Datos (ADB Teoría 1)	Desarrollo Basado en Componentes y Servicios (DBCS Teoría 1)	
11:00		Análisis y Diseño de Bases de Datos (ADB Lab 2)		Modelado de Sistemas Software (MOD Teoría 1)

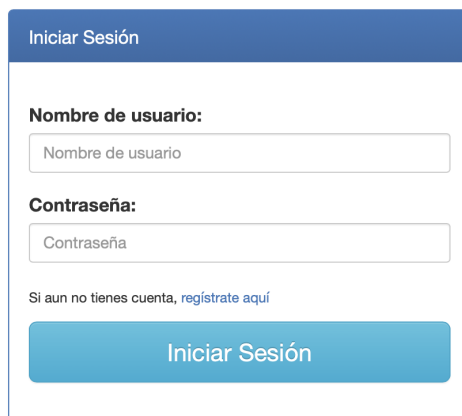
Schedule Manager - Diseño tu horario © 2018 / Made by

Figura 6.4: Interfaz del usuario de la pantalla horario reducida horizontalmente.

Otra parte importante del diseño, y que se utiliza en multitud vistas de la aplicación, son los paneles. Se trata de unos elementos que ofrece Bootstrap y que permiten presentar la información en formato de cajas, a las que se les puede añadir un título, un cuerpo y un pie. No sólo nos permite agrupar información relacionada, como por ejemplo todos los campos de un formulario, sino que además se pueden presentar con diferentes estilos para darle al usuario, de una manera más visual, una referencia sobre la operación que está llevando a cabo.

En ese sentido se han configurado los paneles para que se mostrasen con un aspecto u otro dependiendo del tipo de operación que el usuario o el administrador iban a hacer:

- **Operaciones habituales:** se refiere a los formularios de registro e inicio de sesión, o a los menús principales. El panel que se muestra tiene color azulado.



Panel de inicio de sesión con un encabezado azul que dice "Iniciar Sesión". Contiene dos campos de entrada: "Nombre de usuario:" con el texto "Nombre de usuario" y "Contraseña:" con el texto "Contraseña". Debajo de los campos hay un enlace que dice "Si aun no tienes cuenta, [regístrate aquí](#)". En la parte inferior hay un botón azul que dice "Iniciar Sesión".

Figura 6.5: Ejemplo de panel utilizado para operaciones habituales.

- **Operaciones de edición:** para las tareas de administración en las que se va a modificar la información permitida de una titulación, asignatura y/o grupo. Se han puesto paneles en color amarillo para identificarlos con una apariencia de “advertencia”.



Panel de edición de asignatura con un encabezado amarillo que dice "Editar asignatura". Contiene varios campos de entrada: "ID Titulación:" con el valor "1", "ID Asignatura:" con el valor "1000000", "Nombre completo:" con el texto "Asignatura de pruebas 2", "Curso:" con el valor "4", "Período:" con el valor "2", y "Mención/especialidad:" con el valor "TI". En la parte inferior hay dos botones azules: "Actualizar asignatura" con un icono de flecha y "Volver a asignaturas" con un icono de flecha curva.

Figura 6.6: Ejemplo de panel utilizado para operaciones de edición.

- **Operaciones de borrado:** cuando el usuario va a borrar alguno de los horarios que tiene guardados o cuando el administrador va a eliminar una titulación, asignatura y/o grupo. Para este caso se ha optado por un panel con apariencia de “peligro” que se identifica con el color rojo.

Eliminar horario

**Nombre horario:**  
Horario de pruebas

**ID Horario:**  
11

**Propietario:**  
miguel

**Contenido:**  
[34, 47]

Eliminar horario      Volver a horarios

No podrá recuperar el horario una vez lo haya eliminado.

Figura 6.7: Ejemplo de panel utilizado para operaciones de borrado.

### 6.3. Código de la aplicación

Cabe indicar que hay ciertas partes del código de la aplicación que se han reutilizado de otras aplicaciones ya existentes y que se indican a continuación.

#### Clases involucradas en el registro e inicio de sesión de la aplicación

Por ser la primera vez que se utilizaba el framework Spring MVC se consultaron multitud de ejemplos de aplicaciones en las que se implementaba. La mayoría de ellas eran simples y sólo contaban con el código necesario para llevar a cabo el registro y el inicio de sesión, pero con eso resultó suficiente para obtener un ejemplo real de cómo desarrollar la aplicación utilizando el citado framework a la vez que se implementaba la arquitectura de tres capas. La aplicación de referencia que se siguió se puede encontrar en [27].

#### Métodos auxiliares para el cálculo de probabilidades

Uno de los problemas a la hora de calcular las posibles combinaciones entre grupos de asignaturas era obtener una visión global de esas combinaciones. No hay manera de reducir el número de casos a evaluar, sino que hay que tenerlos en cuenta todos y después ir analizando el número de solapamientos de horas y el grado de compactación de cada combinación.

Esto se realiza en el método “getValidCombinations” de la clase “TimetableService.java”. Este método hace uso de otros dos métodos auxiliares que se tomaron de [28], y que son “combinationUtil” y “printCombination”.



---

## JavaScript

JavaScript es un lenguaje de programación que se ha utilizado para determinadas funciones y en partes muy concretas del proyecto. Aprovechando que se trata de un lenguaje interpretado se ha incluido en el código de las vistas, por lo que se ejecutará en los navegadores, es decir, en la parte del cliente. De esta manera se consigue disminuir algo la carga de trabajo del servidor donde esté alojada la aplicación. Por el contrario se depende de que el navegador donde se cargue la aplicación pueda ejecutar javascript, aunque en la actualidad es lo más común.

Se han utilizado algunas librerías ya existentes y se han desarrollado porciones de código para estas partes de la aplicación:

### Validación de formularios

Utilizarlo para la validación de los formularios de la aplicación permite que estos sólo se envíen cuando se hayan cumplido las condiciones. En la mayoría de los casos se evalúa que los campos no estuvieran vacíos. Se ha utilizado en diferentes partes de la aplicación, pero un ejemplo utilizado en el formulario de registro se muestra en la Figura 6.8.

```
function validateForm(){
    var a=document.forms["RegisterForm"]["username"].value;
    var b=document.forms["RegisterForm"]["email"].value;
    var c=document.forms["RegisterForm"]["password"].value;
    if (a==null || a=="" || b==null || b=="" || c==null || c=="")
    {
        alert("Debe completar todos los campos");
        return false;
    }
}
```

Figura 6.8: Código de ejemplo validación de campos correspondiente al formulario de registro.

### Búsquedas dinámicas

Estas búsquedas se realizan como administrador de la aplicación, y es muy importante que se realice de manera dinámica para que no haya que estar recargando la página con el contenido todo el tiempo. Lo que se consigue con la función es que sólo se muestren las filas de la tabla que cumplan las condiciones de búsqueda a la vez.

```

function dynamicSearchAdmin() {
    var idCareer, idSubject, nameSubject, filterIDC, filterIDS, filterNameS, trIDC,
        trIDS, trNS, tdIDC, tdIDS, tdNS, i, textIDC, textIDS, textNS;
    idCareer = document.getElementById("idCareer");
    idSubject = document.getElementById("idSubject");
    nameSubject = document.getElementById("nameSubject");
    filterIDC = idCareer.value.toUpperCase();
    filterIDS = idSubject.value.toUpperCase();
    filterNameS = nameSubject.value.toUpperCase();
    table = document.getElementById("tableSubjects");
    tr = table.getElementsByTagName("tr");
    for (i = 0; i < tr.length; i++) {
        tdIDC = tr[i].getElementsByTagName("td")[0];
        tdIDS = tr[i].getElementsByTagName("td")[1];
        tdNS = tr[i].getElementsByTagName("td")[2];
        if (tdIDC) {
            textIDS = tdIDS.textContent || tdIDS.innerText;
            textIDC = tdIDC.textContent || tdIDC.innerText;
            textNS = tdNS.textContent || tdNS.innerText;
            if ((textIDC.toUpperCase().indexOf(filterIDC) > -1)
                && (textIDS.toUpperCase().indexOf(filterIDS) > -1)
                && (textNS.toUpperCase().indexOf(filterNameS) > -1)) {
                tr[i].style.display = "";
            } else {
                tr[i].style.display = "none";
            }
        }
    }
}

```

Figura 6.9: Código de la búsqueda dinámica en la tabla de Asignaturas.

## Generación de ficheros PDF

Esta parte se implementó utilizando la librería “html2canvas”, que se utiliza para capturar partes de la pantalla utilizando el lenguaje javascript. En la siguiente porción de código se muestra

```

function generatePDF() {
    html2canvas(document.getElementById('timetableTable')).then(
        function(canvas) {
            var wid = 3000;
            var hgt = 3000;
            var img = canvas.toDataURL("image/png",
                wid = canvas.width, hgt = canvas.height);
            var hratio = hgt / wid;
            var doc = new jsPDF('p', 'pt', 'a4');
            var width = doc.internal.pageSize.width;
            var height = width * hratio;
            doc.addImage(img, 'JPEG', 0, 0, width, height);
            doc.save('HorarioSchema.pdf');
        });
}

```

Figura 6.10: Código para obtener los horarios en formato PDF.

---

## 6.4. Cálculo de horarios

Como ya se ha explicado en apartados anteriores de este trabajo, el objetivo fundamental era desarrollar una aplicación que permita a los usuarios calcular el mejor horario posible una vez que hubiesen elegido las asignaturas que quieren cursar.

Lo que aquí se define como “el mejor horario posible” se obtiene aplicando dos filtros sobre todas las combinaciones posibles:

- El primero es obtener aquellos horarios con el menor número de solapamiento de horas de clase posible.
- De los horarios anteriores filtrar aquellos que tienen un mejor grado de compactación, es decir, que tienen menos horas libres entre clases en una misma parte del día.

### Primera fase

La estrategia utilizada para diseñar el algoritmo en una primera fase es la fuerza bruta, ya que el modelo planteado no permite otra forma de obtener el número de solapamiento de horas. Así pues en esa primera parte se obtienen todos los horarios posibles, guardando para cada uno la información de los grupos que lo conforman (a través del identificador del grupo) y el número de conflictos que plantean, siendo los mejores aquellos cuyo número de solapamientos sea menor.

### Segunda fase

Una vez se tienen esos horarios se pasa a evaluar el grado de compactación de los horarios, pero sólo de aquellos que se consideraron mejores en la primera fase.

Para implementar esta parte del algoritmo se decidió dividir en dos partes los días, una primera que iría de las 8 de la mañana a las 3 de la tarde, y una segunda que iría desde las 3 de la tarde hasta las 10 de la noche. De esta forma el grado de compactación se evalúa en 10 intervalos, 2 por cada día de la semana de lunes a viernes. Un ejemplo de cómo se calcula se puede ver en la Figura 6.11.

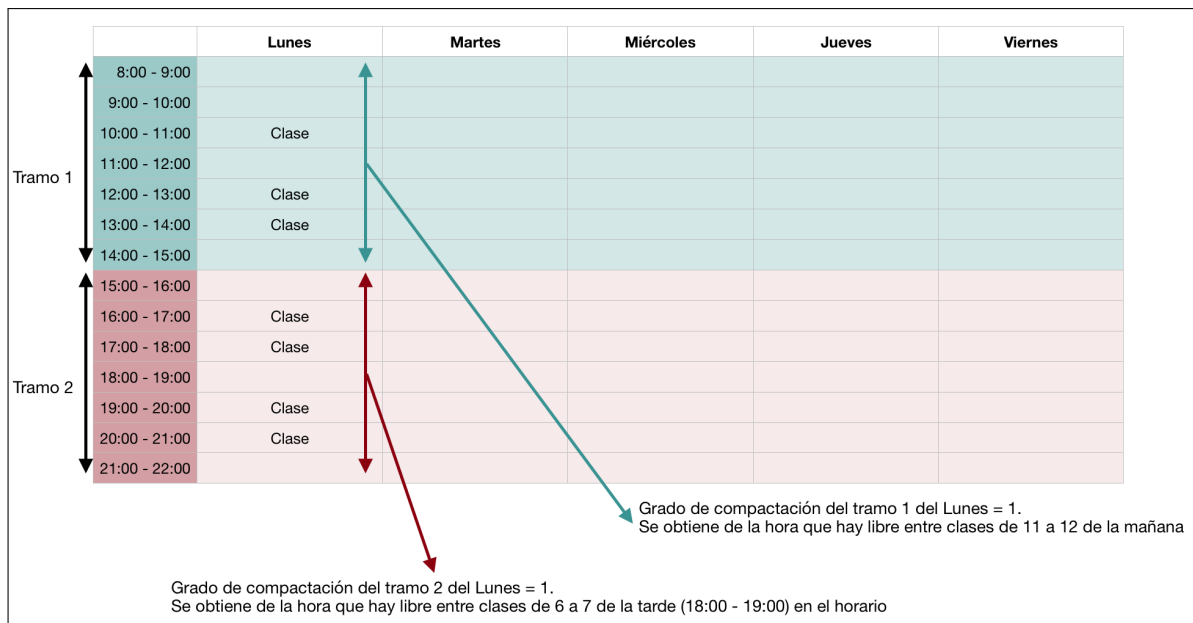


Figura 6.11: Ejemplo de cálculo del grado de compactación de un horario.

Como se puede ver en la anterior figura, el grado de compactación resultante para el Lunes sería de 2, que corresponde a la suma de los dos tramos horarios de ese día. El horario con mejor grado de compactación es aquel en el que su valor es menor, es decir, tiene menos horas libres entre clases.

## 6.5. Base de datos

La estructura de la base de datos es la indicada en el apartado 5.4.2. El script de creación es el siguiente:

```

CREATE TABLE Users (
  username VARCHAR(20),
  password VARCHAR(40),
  email VARCHAR(40),
  roleU INTEGER,
  PRIMARY KEY (username)
);

CREATE TABLE Career (
  careerID INTEGER,
  careerName VARCHAR(300),
  PRIMARY KEY (careerID)
);

CREATE TABLE Timetable (
  timetableID INTEGER NOT NULL AUTO_INCREMENT,
  ownerID VARCHAR(20),
  ttName VARCHAR(200),
  content VARCHAR(3000),
  PRIMARY KEY (timetableID),
  FOREIGN KEY (ownerID) REFERENCES Users (username) ON DELETE CASCADE
);

CREATE TABLE Subject (
  code INTEGER,
  career INTEGER,
  course INTEGER,
  fullName VARCHAR(100),
  mention VARCHAR(8),
  periodS INTEGER,
  PRIMARY KEY (code),
  FOREIGN KEY (career) REFERENCES Career (careerID) ON DELETE CASCADE
);

CREATE TABLE Groups (
  groupID INTEGER,
  gType VARCHAR(1),
  gName VARCHAR(20),
  associatedGroup INTEGER,
  subjectCode INTEGER,
  additionalInfo VARCHAR(3000),
  PRIMARY KEY (groupID),
  FOREIGN KEY (associatedGroup) REFERENCES Groups (groupID) ON DELETE CASCADE,
  FOREIGN KEY (subjectCode) REFERENCES Subject (code) ON DELETE CASCADE
);

CREATE TABLE Hours (
  groupID INTEGER,
  hours INTEGER,
  PRIMARY KEY (groupID, hours),
  FOREIGN KEY (groupID) REFERENCES Groups (groupID) ON DELETE CASCADE
);

```

Figura 6.12: Script de creación de la base de datos.

## 6.6. Pruebas

Scrum establece que las pruebas las realicen algunos miembros del equipo de desarrollo dentro del mismo sprint. Siguiendo esa misma filosofía las pruebas se han ido realizando a medidas que se completaban las historias de usuario, para que una vez finalizadas y probadas pudieran ser consideradas ya un incremento sobre el producto anterior y susceptibles de ponerse en producción.

Las pruebas se orientan a comprobar si el software cumple con los criterios de aceptación de cada historias

de usuario y que se detallaron en el apartado 4.3. No se ha encontrado apenas información sobre como realizar pruebas de caja negra en proyectos relacionados con metodologías ágiles, por lo que ha optado por seguir el ejemplo indicado en [29], que consiste en analizar las condiciones iniciales, evaluar la acción del usuario y documentar el resultado obtenido, todo ello en forma de tarjeta como muestra la Tabla 6.3:

<b>Condición inicial</b>	
<b>Acción del usuario</b>	
<b>Resultado</b>	

Tabla 6.3: Plantilla utilizada para documentar las pruebas realizadas.

Se presentan a continuación las pruebas realizadas asociadas con cada historia de usuario.

### Historia de usuario HU01: Registro en el sistema

- El formulario de registro tiene que ser accesible.

<b>Condición inicial</b>	El usuario ha cargado la aplicación en un navegador web y se encuentra en la página de inicio.
<b>Acción del usuario</b>	El usuario hace clic en el botón “Registrarme” de la página de inicio o en el enlace “Registro” de la barra de navegación superior.
<b>Resultado</b>	Correcto. Se carga adecuadamente el formulario de registro en la aplicación.

- El usuario tiene que poder registrarse en el sistema.

<b>Condición inicial</b>	El usuario se encuentra en la página con el formulario de registro.
<b>Acción del usuario</b>	Completa los campos solicitados (nombre de usuario, correo electrónico y contraseña) con datos sin repetir sobre los ya existentes en la base de datos.
<b>Resultado</b>	Correcto. El usuario queda registrado en la aplicación y automáticamente se le redirige a la página de inicio de los usuarios que han iniciado sesión en el sistema.

- En caso de que el nombre de usuario o el correo electrónico ya estén utilizados debe denegarse el registro e informar al usuario.

<b>Condición inicial</b>	El usuario se encuentra en la página con el formulario de registro.
<b>Acción del usuario</b>	Completa los campos solicitados (nombre de usuario, correo electrónico y contraseña) repitiendo el nombre de usuario y/o el correo electrónico de algún usuario que ya estuviera registrado en el sistema.
<b>Resultado</b>	Correcto.

### Historia de usuario HU02: Acceso al sistema

- El formulario de inicio de sesión tiene que ser accesible.

<b>Condición inicial</b>	El usuario ha cargado la aplicación en un navegador web y se encuentra en la página de inicio.
<b>Acción del usuario</b>	El usuario hace clic en el botón “Iniciar Sesión” de la página de inicio o en el enlace “Iniciar Sesión” de la barra de navegación superior.
<b>Resultado</b>	Correcto. Se carga adecuadamente el formulario de inicio de sesión en la aplicación.

- El usuario tiene que poder iniciar sesión en el sistema.

<b>Condición inicial</b>	El usuario se encuentra en la página con el formulario de inicio de sesión.
<b>Acción del usuario</b>	Completa los campos solicitados (nombre de usuario y contraseña) con unos datos válidos.
<b>Resultado</b>	Correcto. Se inicia sesión en la aplicación de manera correcta. Dependiendo del rol del usuario en la aplicación se carga una página de inicio u otra, y la barra de navegación superior correspondiente.

- Si los datos introducidos no son correctos el usuario no podrá acceder al sistema y se le informará.

<b>Condición inicial</b>	El usuario se encuentra en la página con el formulario de inicio de sesión.
<b>Acción del usuario</b>	Completa los campos solicitados (nombre de usuario y contraseña) con unos datos no válidos.
<b>Resultado</b>	Correcto. El usuario no inicia sesión en el sistema, y se le indica mediante un mensaje en el propio formulario de que los datos introducidos son incorrectos.

### Historia de usuario HU03: Generar horario

- El usuario tiene que poder elegir la titulación, el período y las asignaturas para las que quiere obtener el horario.

<b>Condición inicial</b>	El usuario ha iniciado sesión con el rol de usuario normal. En la página de inicio ha hecho clic en el botón “Generar nuevo horario” y se encuentra en la el formulario para elegir titulación y período.
<b>Acción del usuario</b>	El usuario elige una de las titulaciones y un período de los disponibles en las dos listas desplegadas.
<b>Resultado</b>	Correcto. La aplicación pasa a un segundo formulario donde el usuario elegirá las asignaturas para las que quiere configurar el horario.

<b>Condición inicial</b>	El usuario ha iniciado sesión con el rol de usuario normal. En la página de inicio ha hecho clic en el botón “Generar nuevo horario” y se encuentra en la el formulario para elegir titulación y período.
<b>Acción del usuario</b>	El usuario deja sin elegir alguno de los dos campos disponibles y hace clic en el botón “Elegir asignaturas”.
<b>Resultado</b>	Correcto. Sale un mensaje emergente indicando al usuario que debe elegir una opción de los dos menús. Además, el primer campo que haya dejado sin elegir se marcará con el borde en color rojo para indicar al usuario donde se ha localizado su error. Este error se producirá hasta que el usuario complete el formulario de manera correcta.

<b>Condición inicial</b>	El usuario ha completado satisfactoriamente el paso de elegir titulación y curso. Ahora se encuentra en la pantalla con el formulario para elegir asignaturas.
<b>Acción del usuario</b>	Elige, al menos, una asignatura en el formulario de selección de asignaturas y hace clic en “Calcular horario”.
<b>Resultado</b>	Correcto. Se genera un horario de acuerdo a las asignaturas elegidas.

<b>Condición inicial</b>	El usuario ha completado satisfactoriamente el paso de elegir titulación y curso. Ahora se encuentra en la pantalla con el formulario para elegir asignaturas.
<b>Acción del usuario</b>	El usuario intenta enviar el formulario sin elegir ninguna asignatura.
<b>Resultado</b>	Correcto. Se muestra un mensaje emergente de error informando al usuario de que debe elegir al menos una asignatura.

- El sistema tiene que mostrar la mejor combinación encontrada atendiendo a dos criterios: el menor número de solapamientos de clase posibles y el mayor grado de compactación.

<b>Condición inicial</b>	El usuario ha completado satisfactoriamente el paso de elegir titulación y curso. En la siguiente página, el usuario ha seleccionado una o más asignaturas.
<b>Acción del usuario</b>	El usuario envía el formulario haciendo clic en el botón “Calcular horario”.
<b>Resultado</b>	Incorrecto. Se observa que si bien el horario generado tiene el menor número de solapamiento de horas posible, no es el más compacto. Tras revisar el algoritmo se localiza el error, que consistía en que se estaba generando el horario menos compacto en lugar de lo contrario. Se corrige el error y la salida es ahora la correcta.

- El sistema tiene que mostrar enlaces a otras posibles soluciones con las mismas características de solapamientos y grado de compactación.



<b>Condición inicial</b>	El usuario ha completado satisfactoriamente el paso de elegir titulación y curso. En la siguiente página, el usuario ha seleccionado una o más asignaturas.
<b>Acción del usuario</b>	El usuario envía el formulario haciendo clic en el botón "Calcular horario".
<b>Resultado</b>	Correcto. Se le indica al usuario en un mensaje que aparece en la parte superior de la pantalla las opciones que se han encontrado con el mismo número de solapamientos y el mismo grado de compactación. Se muestran en la parte inferior de la pantalla botones de acceso a las demás versiones.

<b>Condición inicial</b>	El usuario ha completado satisfactoriamente el paso de elegir titulación y curso. En la siguiente página, el usuario ha seleccionado una o más asignaturas. Además ha enviado el formulario haciendo clic en el botón "Calcular horario".
<b>Acción del usuario</b>	Hace clic en alguna de las demás opciones generadas.
<b>Resultado</b>	Correcto. Se muestra de manera correcta el nuevo horario.

#### Historia de usuario HU04: Guardar horario

- Al generar un horario nuevo abajo aparecerá un formulario al usuario con la opción de guardar el horario.

<b>Condición inicial</b>	El usuario completa los pasos para generar un horario de selección de titulación, período y asignaturas.
<b>Acción del usuario</b>	El usuario envía el formulario haciendo clic en el botón "Calcular horario".
<b>Resultado</b>	Correcto. En la página donde se genera el horario aparece un formulario en la parte inferior para poder guardar un horario dándole el nombre que desee el usuario.

- Si el usuario no rellena el formulario se informará del error al usuario.

<b>Condición inicial</b>	El usuario se encuentra en la página en la que se muestra el horario que se acaba de generar.
<b>Acción del usuario</b>	No introduce ningún nombre para guardar el horario y hace clic en "Guardar horario".
<b>Resultado</b>	Correcto. Aparece una ventana emergente informando al usuario de que debe darle un nombre al horario antes de guardarlo.

- El horario guardado aparecerá en el listado de horarios del usuario.

<b>Condición inicial</b>	El usuario está en la página donde se carga el horario. Ha introducido un nombre para guardarlo.
<b>Acción del usuario</b>	El usuario hace clic en el botón "Guardar horario".
<b>Resultado</b>	Correcto. El horario se guarda correctamente y el usuario es redirigido a la página de horarios, donde hay un listado de todos los horarios que tiene guardados. El horario que acaba de guardar aparece en la lista.

<b>Condición inicial</b>	El usuario está en cualquier página de la aplicación.
<b>Acción del usuario</b>	El usuario accede hace clic en el enlace “Mis horarios” que aparece en la barra superior de navegación.
<b>Resultado</b>	Correcto. Se muestra un listado de los horarios guardados.

- El usuario podrá cargar cualquier horario de los guardados si así lo desea.

<b>Condición inicial</b>	El usuario se encuentra en la página donde se muestran sus horarios guardados.
<b>Acción del usuario</b>	El usuario hace clic en la opción “Ver” de alguno de ellos, que corresponde al icono del ojo.
<b>Resultado</b>	Correcto. Se carga el horario guardado correctamente.

### Historia de usuario HU05: Exportar horarios

- Al generar un horario nuevo abajo aparecerá un formulario al usuario con la opción de exportar a PDF el horario.

<b>Condición inicial</b>	El usuario completa los pasos para generar un horario de selección de titulación, período y asignaturas.
<b>Acción del usuario</b>	El usuario envía el formulario haciendo clic en el botón “Calcular horario”.
<b>Resultado</b>	Correcto. En la página donde se genera el horario aparece un formulario en la parte inferior para poder descargar el horario en formato de fichero PDF.

- El horario se descargará en formato PDF y contendrá la tabla con el mismo estilo que el utilizado en la aplicación.

<b>Condición inicial</b>	El usuario está en la página donde se carga el horario.
<b>Acción del usuario</b>	Hace clic en el botón “Descargar como PDF”.
<b>Resultado</b>	Correcto. El fichero se descarga en formato PDF con el nombre “HorarioSchema.pdf” que es el que se ha especificado en la función javascript. El PDF muestra el horario con el mismo aspecto visual que el que tiene en la aplicación web.

### Historia de usuario HU06: Funcionamiento de la aplicación

- La aplicación se ejecuta con normalidad en los principales navegadores web (Chrome, Safari, Firefox, Internet Explorer).

<b>Condición inicial</b>	El usuario tiene un navegador Safari abierto.
<b>Acción del usuario</b>	El usuario introduce la URL de la aplicación.
<b>Resultado</b>	Correcto. La aplicación se visualiza de manera correcta.

<b>Condición inicial</b>	El usuario tiene un navegador Google Chrome abierto.
<b>Acción del usuario</b>	El usuario introduce la URL de la aplicación.
<b>Resultado</b>	Correcto. La aplicación se visualiza de manera correcta.

<b>Condición inicial</b>	El usuario tiene un navegador Firefox abierto.
<b>Acción del usuario</b>	El usuario introduce la URL de la aplicación.
<b>Resultado</b>	Correcto. La aplicación se visualiza de manera correcta.

<b>Condición inicial</b>	El usuario tiene un navegador Internet Explorer abierto.
<b>Acción del usuario</b>	El usuario introduce la URL de la aplicación.
<b>Resultado</b>	Correcto. La aplicación se visualiza de manera correcta.

- La aplicación es accesible desde dispositivos móviles (tablets y smartphones) y el diseño se ajusta a las pantallas de los dispositivos.

<b>Condición inicial</b>	El usuario tiene un navegador Safari abierto en un iPad.
<b>Acción del usuario</b>	El usuario introduce la URL de la aplicación.
<b>Resultado</b>	Correcto. La aplicación se visualiza de manera correcta ajustándose a las dimensiones de la pantalla.

<b>Condición inicial</b>	El usuario tiene un navegador abierto en un smartphone con el sistema operativo Android.
<b>Acción del usuario</b>	El usuario introduce la URL de la aplicación.
<b>Resultado</b>	orrecto. La aplicación se visualiza de manera correcta ajustándose a las dimensiones de la pantalla.

### Historia de usuario HU07: Administración

- Al iniciar sesión como administrador aparecerá un menú de administración para gestionar titulaciones, asignaturas y grupos.

<b>Condición inicial</b>	El usuario se encuentra en la página con el formulario de inicio de sesión.
<b>Acción del usuario</b>	Completa los campos solicitados (nombre de usuario y contraseña) con unos datos válidos de usuario con rol de administrador.
<b>Resultado</b>	Correcto. Se inicia sesión en la aplicación de manera correcta. Se carga el menú de administración.

- El usuario administrador podrá realizar algunas modificaciones como cambios de denominación a través de los diferentes menús de gestión, pero habrá otras operaciones limitadas (por ejemplo el cambio de códigos de asignaturas y/o grupos).

<b>Condición inicial</b>	El administrador se encuentra en el menú de gestión de asignaturas.
<b>Acción del usuario</b>	Sobre una de las asignaturas hace clic en el icono de editar asignatura.
<b>Resultado</b>	Correcto. Se carga un formulario con los datos de la asignatura, en el que sólo es posible modificar algunos campos.

<b>Condición inicial</b>	El administrador se encuentra en la página de edición de una asignatura.
<b>Acción del usuario</b>	Deja alguno de los campos vacíos que no sean el de “Mención/especialidad” y hace clic en “Actualizar asignatura”.
<b>Resultado</b>	Correcto. Se muestra una ventana emergente en la que se le indica al usuario que debe rellenar todos los campos del formulario.

<b>Condición inicial</b>	El administrador se encuentra en la página de edición de una asignatura.
<b>Acción del usuario</b>	Cambia los campos que desea sin introducir un nombre de asignatura ya existente para esa titulación y hace clic en “Actualizar asignatura”.
<b>Resultado</b>	Correcto. Se recarga el formulario con los datos de la nueva asignatura y abajo se muestra un mensaje indicando al usuario que se han guardado los cambios.

<b>Condición inicial</b>	El administrador se encuentra en la página de edición de una asignatura.
<b>Acción del usuario</b>	Cambia los campos que desea e introduce un nombre de asignatura ya existente para esa titulación y hace clic en “Actualizar asignatura”.
<b>Resultado</b>	Correcto. Se recarga el formulario con los datos y abajo se muestra un mensaje indicando al usuario que se ya existe una asignatura en esa titulación y con ese nombre. No se produce ninguna actualización en la base de datos.

<b>Condición inicial</b>	El administrador se encuentra en el menú de gestión de asignaturas.
<b>Acción del usuario</b>	Sobre una de las asignaturas hace clic en el icono de eliminar asignatura.
<b>Resultado</b>	Correcto. Se carga un formulario con los datos de la asignatura, en el que no se puede modificar ningún cambio y en el que se le informa al usuario de los grupos que se eliminarán si elimina esa asignatura.

<b>Condición inicial</b>	El administrador se encuentra en la página de eliminación de asignatura.
<b>Acción del usuario</b>	El administrador hace clic en “Eliminar asignatura”.
<b>Resultado</b>	Correcto. Se vuelve a cargar la página de gestión de asignaturas, donde ya no aparece la asignatura que se acaba de eliminar.

<b>Condición inicial</b>	El administrador se encuentra en la página de eliminación de asignatura.
<b>Acción del usuario</b>	El administrador hace clic en “Volver a asignaturas”.
<b>Resultado</b>	Correcto. Se vuelve a cargar la página de gestión de asignaturas.

<b>Condición inicial</b>	El administrador se encuentra en el menú de gestión de asignaturas.
<b>Acción del usuario</b>	Sobre una de las asignaturas hace clic en en el icono de ver la asignatura.
<b>Resultado</b>	Correcto. Se carga un formulario con los datos de la asignatura, en el que no es posible editar nada.

<b>Condición inicial</b>	El administrador se encuentra en la página de vista de asignatura.
<b>Acción del usuario</b>	El administrador hace clic en el botón “Editar asignatura”.
<b>Resultado</b>	Correcto. Se carga una página con un formulario de edición de la asignatura.

<b>Condición inicial</b>	El administrador se encuentra en la página de vista de asignatura.
<b>Acción del usuario</b>	El administrador hace clic en el botón “Volver a asignaturas”.
<b>Resultado</b>	Correcto. Se carga la página de gestión de asignaturas.

<b>Condición inicial</b>	El administrador se encuentra en el menú de gestión de asignaturas.
<b>Acción del usuario</b>	Sobre una de las asignaturas hace clic en en el icono de ver grupos de asignatura.
<b>Resultado</b>	Correcto. Se carga la página de gestión de grupos, pero sólo aparecen aquellos que corresponden a la asignatura elegida.

Las tablas que se han mostrado en este último apartado se refieren a los procesos de consulta, edición y borrado de asignaturas, pero estas pruebas son también válidas para el caso de las titulaciones y los grupos de asignaturas.



# Capítulo 7

## Conclusiones

### 7.1. Conclusiones

Se ha desarrollado una aplicación web que permita a los estudiantes universitarios calcular sus horarios antes de matricularse, que era el objetivo principal de este Trabajo de Fin de Grado.

Realizar un proyecto de estas dimensiones de forma individual ha puesto de manifiesto las dificultades que entraña, por lo que cabe imaginarse que proyectos más grandes y con más personas implicadas requieren de un esfuerzo elevado para ser bien gestionados.

Además considero de especial relevancia el hecho de haber podido aplicar los conocimientos obtenidos en multitud de asignaturas de la titulación de Grado en Ingeniería Informática. Esto reafirma mi opinión personal de que todas las asignaturas tienen cierta importancia, aunque a veces se antoje difícil la idea de aplicar los contenidos de esta o aquella asignatura.

El hecho de trabajar con una metodología ágil utilizando algunos de los elementos que propone Scrum me ha servido para gestionar mejor el tiempo y adaptarme a un método de trabajo que muchas empresas ponen en práctica, por lo que lo considero de gran valor para mi futuro profesional.

Lo que me ha resultado más difícil del proyecto ha sido la fase de análisis del problema, por la complejidad que entrañaba y por la dificultad para modelarlo después en la aplicación.

Una vez terminado el proyecto, y analizando los resultados obtenidos habría cambiado el tiempo de dedicación tanto a la parte de administración como a la interfaz gráfica y lo hubiera dedicado a implementar algunas de las líneas de trabajo futuro que se mencionan en el siguiente apartado.

### 7.2. Trabajo futuro

#### 7.2.1. En la misma aplicación

Como estudiante del Grado en Ingeniería Informática en la Universidad de Valladolid, ha habido cursos en los que he tenido que compaginar mi actividad académica con otras actividades, como por ejemplo el trabajo y/o clases de idiomas. Del desarrollo de este trabajo y de mi propia situación personal se plantean algunas mejoras y complementos que podrían introducirse a la aplicación como son:

- 
- La posibilidad de que los usuarios, de forma previa a la elección de asignaturas, puedan seleccionar franjas horarias para bloquearlas y que la aplicación obtenga aquellas combinaciones horarias que no incluyan horas de clase de ningún tipo, en la medida de lo posible, en esos periodos.
  - Aunque en el desarrollo de la aplicación no se ha contemplado ya que en las guías docentes de las asignaturas no aparece, es cierto que en la titulación de Grado en Ingeniería Informática, existen algunas asignaturas en las que se permite conservar la nota obtenida en la parte de prácticas de cursos inmediatamente anteriores. Esto abre la posibilidad a que se pudiera modificar la aplicación para que el estudiante pudiera elegir no sólo las asignaturas, sino también la parte (teórica y/o práctica) en base a la cual quiere calcular su horario.
  - Como ampliación ideal para la aplicación desarrollada se plantea la inclusión de todas las titulaciones de la universidad. En este punto, y dado que en las distintas facultades se siguen unas pautas a la hora de clasificar asignaturas y grupos, se tendrían que revisar los modelos propuestos, ya que en algunos casos pueden no ser del todo válidos.
  - En la línea de trabajo del análisis masivo de datos y tratamiento de grandes cantidades de información se plantea además la posibilidad de que se permita a los usuarios guardar un horario por cuatrimestre. Partiendo de la idea de que los usuarios calcularán sus horarios antes de completar el proceso de matrícula, un análisis exhaustivo de esos datos permitiría a la Escuela hacer una estimación previa más aproximada de los recursos que debe proveer para el siguiente periodo lectivo.
  - Como ya se expuso en el segundo capítulo de este trabajo, hay múltiples factores que influyen en la asistencia o falta de asistencia de los estudiantes a clase, por lo que se plantea la posibilidad de analizarlos en detalle y añadir las heurísticas necesarias para ofrecer mejores horarios a los alumnos.
  - Se propone también la adaptación de la aplicación al lenguaje de programación adecuado para tratar de integrarlo en forma de plug-in a la plataforma Moodle de la Escuela de Ingeniería Informática.

### 7.2.2. En la línea de aplicaciones complementarias

Los estudiantes que pasan por la Universidad cursan multitud de asignaturas, participan en diversas actividades complementarias, trabajan en diferentes grupos de prácticas, etc. Siguiendo la idea original del proyecto de automatizar tareas que faciliten la asistencia a clase y mejoren el rendimiento académico se propone la creación de aplicaciones similares que permitan al usuario automatizar el tipo de tareas mencionadas anteriormente. Sirvan de ejemplo:

- Una aplicación para permita gestionar los grupos en los que está matriculado el/la estudiante y que permita operaciones como:
  - Solicitar el cambio de grupo.
  - Apuntarse a grupos de prácticas de manera individual o colectiva.
  - Que el profesor o la profesora establezca fechas límite para las operaciones anteriores.

Esta aplicación debería poder gestionar de manera automática el trasvase de alumnos entre grupos teniendo en cuenta los tamaños, el orden de solicitud, la prioridad desde el punto de vista de solapamiento con otras clases, etc.

- Una aplicación que permita gestionar la participación en actividades complementarias con reconocimiento de créditos ECTS y que permita tanto al usuario como a la Universidad saber, en todo momento, el número de créditos que tiene reconocidos cada alumno.



---

# Referencias

- [1] U. de Valladolid. (2018) Proyecto Docente de Trabajo de Fin de grado, Mención Ingeniería del Software. Fecha de último acceso: 2019-04-17. [Online]. Disponible en: [https://alojamientos.uva.es/guia\\_docente/uploads/2018/545/46976/1/Documento.pdf](https://alojamientos.uva.es/guia_docente/uploads/2018/545/46976/1/Documento.pdf)
- [2] M. E. H. García, S. N. Martín, M. J. R. Conde, y M. C. S. Gómez, “Factores implicados en el rendimiento académico de los alumnos Universidad de Salamanca,” *Revista de investigación educativa*, vol. 17, no. 2, pp. 413–421, 1999.
- [3] D. Albalate del Sol, X. Fageda Sanjuan, y J. Perdiguero, “Éxito académico, características personales y proceso de bolonia: una aplicación econométrica,” *Revista d’innovació docent universitària*, no. 3, pp. 11–25, 2011. [Online]. Disponible en: <http://revistes.ub.edu/index.php/RIDU/article/view/105.000001655>
- [4] F. Sánchez Carracedo, C. Álvarez Martínez, A. Fernández Jiménez, y J. F. Llosa Espuny, “¿Por qué faltan a clase los alumnos?” in *JENUI 2017: XXIII Jornadas sobre la Enseñanza Universitaria de la Informática: Cáceres, del 5 al 7 de julio de 2017: actas*. Asociación de Enseñantes Universitarios de la Informática (AENUI), 2017, pp. 165–172.
- [5] I. Sanz, M. J. Aramburu, L. Museros, M. Pérez, y C. Barrachina, “En busca del estudiante perdido: caracterización de los ‘no presentados’,” *JENUI 2011: XVIII Jornadas de Enseñanza Universitaria de la Informática (2011)*, p. 403–410, 2011.
- [6] E. G. Expósito y M. Villasol, “Absentismo entre los estudiantes de teoría económica. un análisis cuantitativo,” *CD Dávila Quintana, 8 e. all., Investigaciones de Economía de la Educación*, pp. 231–240, 2007.
- [7] R. Rodríguez González, J. Hernández García, A. M. Alonso Gutiérrez, y E. Díez Itza, “El absentismo en la universidad: resultados de una encuesta sobre motivos que señalan los estudiantes para no asistir a clase,” *Aula Abierta*, 82, 2003.
- [8] (2019) Escuela de ingeniería informática de valladolid. Fecha de último acceso: 2019-05-26. [Online]. Disponible en: <https://www.inf.uva.es>
- [9] P. Mármol. (2018) 20 herramientas para elaborar los horarios del centro escolar. Fecha de último acceso: 2019-03-02. [Online]. Disponible en: <https://www.educaciontrespuntocero.com/recursos/herramientas-elaborar-horarios/34971.html>
- [10] (2019) Agenda del estudiante. Fecha de último acceso: 2019-03-03. [Online]. Disponible en: <https://student-agenda-pro.es.aptoide.com>
- [11] (2019) aschorarios. Fecha de último acceso: 2019-03-03. [Online]. Disponible en: <https://www.asctimetables.com>
- [12] (2019) Free Timetabling Software. Fecha de último acceso: 2019-03-02. [Online]. Disponible en: <https://www.lalescu.ro/liviu/fet/>

- 
- [13] P. de la Fuente Redondo, “Gestión de riesgos,” Apuntes de clase de la asignatura Planificación y Gestión de Proyectos, curso 2017/2018, Universidad de Valladolid, 2017.
- [14] K. Schwaber y J. Sutherland. (2017) La Guía de Scrum. Fecha de último acceso: 2019-05-12. [Online]. Disponible en: <https://www.scrumguides.org/scrum-guide.html>
- [15] J. L. M. Galiano, *Implantar SCRUM con éxito*. Editorial UOC, 2016.
- [16] (2018) Qué es Scrum. Fecha de último acceso: 2019-05-12. [Online]. Disponible en: <https://www.masterditec.com/wp-content/uploads/2018/05/Qu%C3%A9-es-SCRUM.png>
- [17] M. Cohn. (2009) Make the Product Backlog DEEP. Fecha de último acceso: 2019-05-12. [Online]. Disponible en: <https://www.mountangoatsoftware.com/blog/make-the-product-backlog-deep>
- [18] A. Guide, “Project management body of knowledge (pmbok® guide),” in *Project Management Institute*, 2001.
- [19] IEEE, *IEEE Standard Glossary of Software Engineering Terminology*, IEEE Std., 1990.
- [20] U. de Valladolid. (2014) Reglamento de Ordenación Académica de la Universidad de Valladolid. Fecha de último acceso: 2019-05-08. [Online]. Disponible en: [http://www.uva.es/export/sites/uva/1.lauva/1.19.normativa/\\_documentos/VII.4.Reglamento-de-Ordenacion-Academica.pdf](http://www.uva.es/export/sites/uva/1.lauva/1.19.normativa/_documentos/VII.4.Reglamento-de-Ordenacion-Academica.pdf)
- [21] (2019) Spring. Fecha de último acceso: 2019-05-20. [Online]. Disponible en: <https://spring.io>
- [22] Web mvc framework. Fecha de último acceso: 2019-05-19. [Online]. Disponible en: <https://docs.spring.io/spring/docs/3.2.x/spring-framework-reference/html/mvc.html>
- [23] Wikipedia, “Modelo-vista-controlador — wikipedia, la enciclopedia libre,” 2019, [Internet; descargado 19-mayo-2019]. [Online]. Disponible en: <https://es.wikipedia.org/w/index.php?title=Modelo%E2%80%93vista%E2%80%93controlador&oldid=115244051>
- [24] (2019) Data access object pattern. Fecha de último acceso: 2019-05-19. [Online]. Disponible en: [https://www.tutorialspoint.com/design\\_pattern/data\\_access\\_object\\_pattern.htm](https://www.tutorialspoint.com/design_pattern/data_access_object_pattern.htm)
- [25] C. Álvarez Caules. Spring jdbc-template y el principio dry. Fecha de último acceso: 2019-05-19. [Online]. Disponible en: <https://www.arquitecturajava.com/spring-jdbctemplate-y-el-principio-dry/>
- [26] Wikipedia, “Modelo relacional — wikipedia, la enciclopedia libre,” 2019, [Internet; descargado 20-mayo-2019]. [Online]. Disponible en: [https://es.wikipedia.org/w/index.php?title=Modelo\\_relacional&oldid=114895946](https://es.wikipedia.org/w/index.php?title=Modelo_relacional&oldid=114895946)
- [27] R. Sekar. (2018) Repository for spring framework mvc. Fecha de último acceso: 2019-03-12. [Online]. Disponible en: <https://github.com/javabyranjith/spring-framework-mvc>
- [28] (2018) Print all possible combinations of r elements in a given array of size n. Fecha de último acceso: 2019-03-27. [Online]. Disponible en: <https://www.geeksforgeeks.org/print-all-possible-combinations-of-r-elements-in-a-given-array-of-size-n/>
- [29] V. Osetskyi. (2017) Acceptance criteria in scrum: Explanation, examples, and template. Fecha de último acceso: 2019-05-10. [Online]. Disponible en: <https://dzone.com/articles/acceptance-criteria-in-software-explanation-exampl>

---

# Apéndice A

## Manual de usuario

### A.1. Despliegue

#### A.1.1. Requisitos

Hay que tener en cuenta que los datos para el acceso a la base de datos están dentro de uno de los fichero de la aplicación, por lo que si se desea modificar hay que hacerlo antes de obtener el fichero WAR de la misma.

Ese fichero se encuentra en el directorio *src/main/resources/com.tfg/config* y se llama *app-beans.xml*. Las líneas a modificar se encuentran al final del mismo, y son las que contienen la siguiente información:

```
<property name="url" value="DIRECCION_SERVIDOR:PUERTO/NOMBRE_BASE_DE_DATOS" />
<property name="username" value="NOMBRE_DE_USUARIO" />
<property name="password" value="CONTRASENA" />
```

#### A.1.2. Heroku

El primer requisito para desplegar la aplicación en Heroku es tener creada una cuenta. A continuación hay que abrir un terminal en el que se tengan permisos de administrador y seguir los siguientes pasos:

- **Primer paso:** instalar la interfaz de línea de comandos de heroku con el siguiente comando:

```
heroku plugins:install heroku-cli-deploy
```

- **Segundo paso (opcional):** aunque no es necesario, se recomienda instalar el pack de construcción de Heroku para aplicaciones Java. Se puede hacer ejecutando:

```
heroku buildpacks:add https://github.com/jkutner/heroku-buildpack-javafx --app NOMBRE\_APL
```

- **Tercer paso:** crear la aplicación en Heroku:

```
heroku create NOMBRE\_APLICACION
```

- **Cuarto paso:** desplegar el fichero WAR en el servidor:

```
heroku war:deploy RUTA\FICHERO\_WAR/FICHERO\_WAR --app NOMBRE\_APLICACION
```

- **Quinto paso:** abrir la aplicación en un navegador con la siguiente instrucción:

```
heroku open --app NOMBRE\_APLICACION
```

### A.1.3. Otras formas de despliegue

Se puede desplegar también en un servidor de aplicaciones como Apache Tomcat. Para ello los pasos a seguir son los siguientes:

- Copiar el fichero WAR de la aplicación en el directorio *webapps* que se encuentra dentro del directorio de instalación de Tomcat en la máquina que actúa como servidor.
- Acceder a la aplicación a través de la URL: *direccion\_del\_servidor:puerto/nombre\_de\_la\_aplicacion*.

## A.2. Manual de usuario

### A.2.1. Introducción

Al iniciar la aplicación se encontrará con la página de bienvenida, tal y como se muestra en la Figura A.1. Desde allí podrá acceder a los formularios de registro e inicio de sesión a través de los dos botones que aparecen en el centro de la misma y a través de la barra de navegación superior.

Para poder utilizar la aplicación es necesario estar registrado en ella, y además, cuando complete el proceso de registro se iniciará sesión de manera automática en la aplicación.



Figura A.1: Página de bienvenida de la aplicación.

---

## A.2.2. Administrador

### Introducción

Una vez haya iniciado sesión con un perfil de usuario administrador será redirigido al panel de administración, tal y como muestra la Figura A.2.

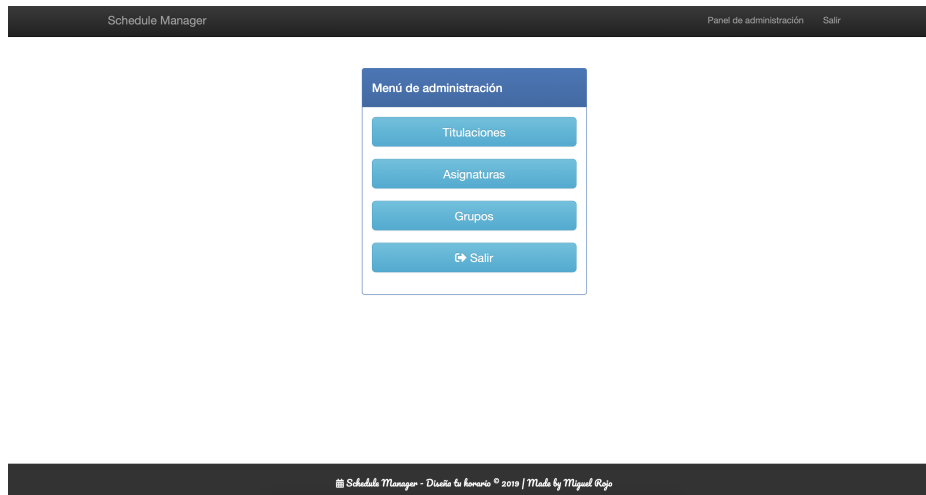


Figura A.2: Panel de administración de la aplicación.

A través del panel de administración, que tendrá accesible en todo momento a través del enlace en la barra de navegación superior, podrá gestionar titulaciones, asignaturas y grupos. Sólo tendrá que hacer clic en cualquiera de los botones del panel mencionado.

### Gestión de titulaciones

Cuando acceda al menú “Titulaciones” accederá a la página de gestión de titulaciones, que tiene un aspecto como el que refleja la Figura A.3

## Gestión de titulaciones





Búsqueda por ID titulación	Búsqueda por nombre de titulación	
ID Titulación	Nombre Titulación	Acciones
1	Grado en Ingeniería Informática	   

Figura A.3: Página de gestión de titulaciones de la aplicación.

En la parte superior de la página dispone de campos de búsqueda para filtrar las titulaciones que se muestran en la tabla. Las limitaciones que introduzca en esos campos de búsqueda se tendrán en cuenta de manera conjunta.


Para cada titulación tendrá cuatro opciones, representadas por iconos y que son las siguientes (en el orden en el que se muestran en la imagen):

- **Editar titulación:** la aplicación mostrará la vista de edición de asignatura, en la que encontrará un formulario para modificar los datos permitidos. Si pulsa “Editar” en dicha página se recargará y se le mostrará un mensaje de éxito o fracaso en la parte inferior del formulario. Para volver a la gestión de titulaciones tendrá que hacerlo a través del botón “Volver a titulaciones”. Un ejemplo del formulario de edición se muestra en la siguiente figura.

### Editar titulación

**ID Titulación:**

**Nombre de la Titulación:**

 Actualizar titulación


 Volver a titulaciones

Figura A.4: Formulario de la página de edición de titulación.

- **Eliminar titulación:** si elige esta opción será redirigido a una página en la que se mostrará un formulario para eliminar la titulación seleccionada. Se le mostrará información sobre las asignaturas que se verán afectadas al eliminar una titulación. Un ejemplo de este formulario se muestra a continuación.

Eliminar titulación

**ID Titulación:**  
1

**Nombre de la Titulación:**  
Grado en Ingeniería Informática

Eliminar titulación Volver a titulaciones

Eliminar esta titulación implicará eliminar también las siguientes asignaturas asociadas:

- Fundamentos de Organización de Empresas
- Fundamentos de Matemáticas
- Matemática Discreta
- Sistemas Digitales
- Fundamentos de Programación
- Fundamentos de Redes de Computadoras
- Ampliación de Matemáticas
- Física

Figura A.5: Formulario de la página de borrado de titulación.

Si confirma la eliminación de la titulación ésta será borrada de la base de datos y se le redirigirá a la página de gestión de titulaciones en la que ya no aparecerá la titulación eliminada.

- **Ver titulación:** accederá a la página de vista de la titulación, que consiste en un formulario con la información de la titulación pero que no se podrá editar. Desde ese menú podrá acceder a la edición de la titulación o volver a la gestión de titulaciones, tal y como muestra la siguiente figura.

Detalle de titulación

**ID Titulación:**  
1

**Nombre de la Titulación:**  
Grado en Ingeniería Informática

Editar titulación Volver a titulaciones













































Figura A.6: Formulario de la página de vista de titulación.

- **Ver asignaturas de la titulación:** al hacer clic en este enlace será redirigido a la página de gestión de asignaturas, pero sólo aparecerán aquellas que estén relacionadas con la titulación que esté gestionando.

## Gestión de asignaturas

Cuando acceda al menú “Asignaturas” accederá a la página de gestión de asignaturas, cuya vista se muestra en la Figura A.7.

## Gestión de asignaturas

ID Titulación	ID Asignatura	Nombre asignatura	Acciones
1	46900	Fundamentos de Organización de Empresas	   
1	46901	Fundamentos de Matemáticas	   
1	46902	Matemática Discreta	   
1	46903	Sistemas Digitales	   
1	46904	Fundamentos de Programación	   
1	46905	Fundamentos de Redes de Computadoras	   
1	46906	Ampliación de Matemáticas	   
1	46907	Física	   
1	46908	Fundamentos de Computadoras	   
1	46909	Paradigmas de Programación	   
1	46910	Estadística	   

 Schedule Manager - Diseña tu horario © 2019 / Made by Miguel Rojo

Figura A.7: Página de gestión de asignaturas.

En la parte superior de la página dispone de campos de búsqueda para filtrar las asignaturas que se muestran en la tabla. Los filtros de búsqueda se aplican de manera conjunta.

Para cada asignatura, igual que para las titulaciones, tendrá cuatro opciones, representadas por iconos y que son las siguientes (en el orden en el que se muestran en la imagen):

- **Editar asignatura:**

Editar asignatura

**ID Titulación:**

**ID Asignatura:**

**Nombre completo:**

**Curso:**

**Período:**

**Mención/especialidad:**

Figura A.8: Formulario de la página de edición de asignatura.



- **Eliminar asignatura:** se mostrará la página para borrar asignaturas. En el formulario los campos no serán editables, y se le informará de los grupos afectados si finalmente decide eliminar la asignatura.



Eliminar asignatura

<b>ID Titulación:</b> 1	<b>ID Asignatura:</b> 46954
<b>Nombre completo:</b> Desarrollo Basado en Componentes y Servicios	<b>Mención/especialidad:</b> IS
<b>Curso:</b> 4	<b>Período:</b> 1

Eliminar asignatura

Volver a asignaturas

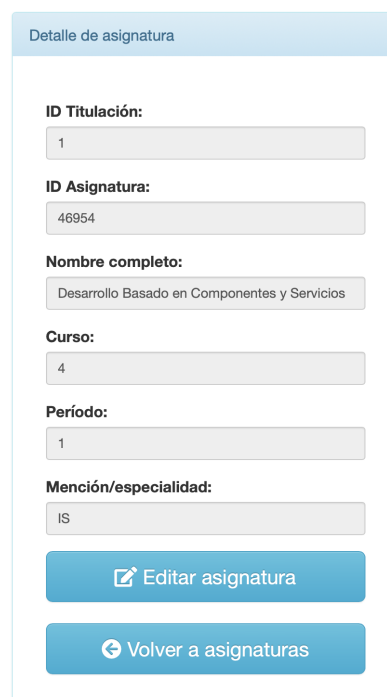
Eliminar esta asignatura implicará eliminar también los siguientes grupos asociados a ella:

- DBCS Teoría 1
- DBCS Lab 1

Figura A.9: Formulario de la página de borrado de asignatura.

Si confirma la eliminación de la asignatura será redirigido a la página de gestión de asignaturas, donde ya no aparecerá la que acaba de eliminar.

- **Ver asignatura:** accederá a la página de vista de la titulación, que consiste en un formulario con la información de la titulación pero que no se podrá editar. Desde ese menú podrá acceder a la edición de la titulación o volver a la gestión de titulaciones, tal y como muestra la siguiente figura.



Detalle de asignatura

<b>ID Titulación:</b> 1
<b>ID Asignatura:</b> 46954
<b>Nombre completo:</b> Desarrollo Basado en Componentes y Servicios
<b>Curso:</b> 4
<b>Período:</b> 1
<b>Mención/especialidad:</b> IS

Editar asignatura

Volver a asignaturas

Figura A.10: Formulario de la página de vista de asignatura.

- **Ver grupos de la asignatura:** al hacer clic en este enlace será redirigido a la página de gestión de grupos, donde ahora sólo aparecerán los grupos asociados a la asignatura seleccionada.

## Gestión de grupos

Cuando acceda al menú “Grupos” accederá a la página de gestión de grupos, que se muestra en la Figura A.11.













































ID Asignatura	ID Grupo	Nombre grupo	Acciones
46920	31	ADA Teoría 1	   
46921	32	ADBBD Teoría 1	   
46922	33	ICO Teoría 1	   
46923	34	MOD Teoría 1	   
46954	35	DBCS Teoría 1	   
46971	36	PGP Teoría 1	   
46974	37	SMOV Teoría 1	   
46952	38	PYS Teoría 1	   
46920	39	ADA Lab 1	   
46920	40	ADA Lab 2	   
46921	41	ADBBD Lab 1	   

Figura A.11: Página de gestión de grupos de la aplicación.

En la parte superior de la página dispone de campos de búsqueda para filtrar los grupos que se muestran en la tabla. Los filtros de búsqueda se aplican de manera conjunta.

Para cada grupo, igual que para las asignaturas y las titulaciones, tendrá cuatro opciones, representadas por iconos y que son las siguientes (en el orden en el que se muestran en la imagen):

- **Editar grupo:** accederá al formulario de edición del grupo, donde podrá cambiar sólo algunos campos.

Editar grupo

**ID Asignatura:**

**ID Grupo:**

**Nombre:**

**Tipo de grupo:**

Figura A.12: Formulario de la página de edición de grupo.

- **Eliminar grupo:** se mostrará un formulario para eliminar el grupo en el que se informará al usuario de los grupos asociados que también se verán afectados por la operación de borrado.

Figura A.13: Formulario de la página de borrado de grupo.

Si confirma la eliminación de la asignatura será redirigido a la página de gestión de asignaturas, donde ya no aparecerá la que acaba de eliminar.

- **Ver grupo:** accederá a la página de vista del grupo, tal y como se muestra en la siguiente imagen. Los campos no podrán editarse y podrá acceder a la edición del grupo o volver a la gestión de grupos.

Figura A.14: Formulario de la página de vista de grupo.

- **Ver grupos asociados:** al hacer clic en este enlace será redirigido a la página de gestión de grupos, donde ahora sólo aparecerán los grupos asociados al grupo seleccionado, en caso de que los haya. Si no tuviera grupos asociados aparecería una lista vacía.

## Otras operaciones

Para realizar otras operaciones como puede ser la inserción de datos, el registro de más administradores, o algunas operaciones de edición se ha optado por que lo haga un administrador del sistema conectándose directamente a la base de datos.

---

## A.2.3. Estudiante

### Introducción

Al iniciar sesión como estudiante en la aplicación se le mostrará el menú que se muestra en la Figura A.15. A través de dicho menú podrá realizar las siguientes operaciones en la aplicación: consultar sus horarios guardados, generar un nuevo horario, acceder y modificar su perfil. También tiene un botón habilitado para salir de la aplicación.

Podrá acceder en cualquier momento a este menú a través del enlace “Inicio” que se muestra en la barra de navegación superior.

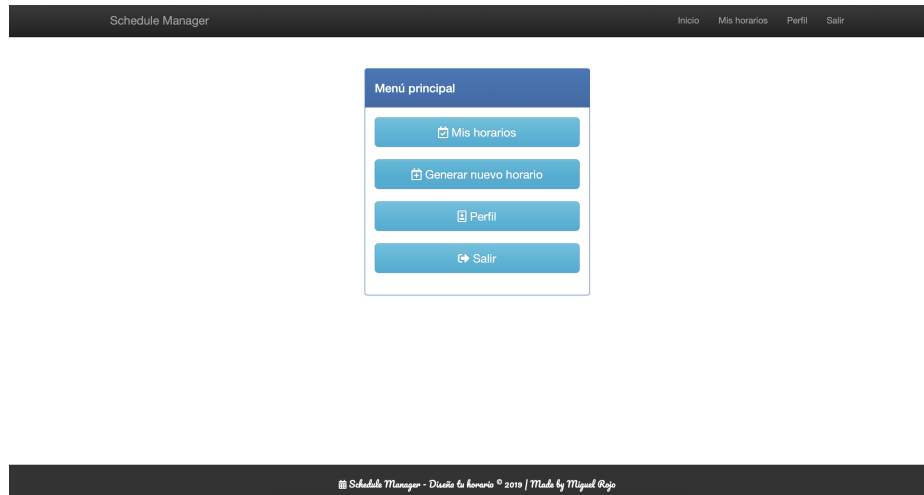


Figura A.15: Panel del estudiante en la aplicación.

### Generación de horarios

Para generar un nuevo horario tendrá que completar dos formularios. El primero se muestra en la Figura A.16 y en él tendrá que elegir la titulación y el periodo para los que desea generar el horario. Después accederá a la selección de asignaturas.

The image shows a screenshot of a web form. It features two dropdown menus. The first dropdown menu is labeled "Grado en Ingeniería Informática" and has a downward arrow. The second dropdown menu is labeled "1º cuatrimestre" and also has a downward arrow. Below these two menus is a blue button with a white icon of a list and the text "Elegir asignaturas". At the bottom of the form, there is a light blue box containing the text: "NOTA: Las asignaturas de tipo 'anual' se mostrarán en la selección de asignaturas independientemente del periodo elegido."

Figura A.16: Formulario de selección de titulación y periodo.

Cuando se encuentre en la pantalla de selección de asignaturas (Figura A.17) deberá añadir al menos una para calcular su horario. Puede añadir y eliminar asignaturas de la lista.

Curso: 3 Asignatura: Ingeniería del Conocimiento-CO,IS

Resetear Añadir asignatura

	Asignaturas	Acciones
1	Análisis y Diseño de Algoritmos	Eliminar
2	Análisis y Diseño de Bases de Datos	Eliminar
3	Ingeniería del Conocimiento	Eliminar

Calcular horario

Figura A.17: Formulario de selección de asignaturas para el cálculo de horarios.

Cuando haya finalizado debe pulsar “Calcular horario” y será redirigido a una pantalla en la que se mostrará el horario generado (Figura A.18).

Schedule Manager Inicio Mis horarios Perfil Salir

### Horario:

Mostrando la primera opción de un total de 5. Puede acceder a todas en la parte inferior de la página.

#	Lunes	Martes	Miércoles	Jueves	Viernes
8:00					
9:00		Análisis y Diseño de Algoritmos (ADA Teoría 1)	Análisis y Diseño de Bases de Datos (ADBBD Teoría 1)		
10:00	Ingeniería del Conocimiento (ICO Teoría 1)	Análisis y Diseño de Bases de Datos (ADBBD Teoría 1)	Análisis y Diseño de Algoritmos (ADA Lab 1)		
11:00		Análisis y Diseño de Bases de Datos (ADBBD Lab 2)	Análisis y Diseño de Algoritmos (ADA Lab 1)		
12:00		Análisis y Diseño de Bases de Datos (ADBBD Lab 2)		Análisis y Diseño de Algoritmos (ADA Teoría 1)	
13:00		Ingeniería del Conocimiento (ICO Lab 1)		Ingeniería del Conocimiento (ICO Teoría 1)	
14:00		Ingeniería del Conocimiento (ICO Lab 1)			
15:00					

Schedule Manager - Diseña tu horario © 2019 | Made by Miguel Rojo

Figura A.18: Página de horario generado.

Además, en la parte inferior de la pantalla tendrá disponibles las opciones para guardar el horario en la base de datos y para descargar el horario en formato PDF. También encontrará disponibles el resto de opciones generadas, si las hay, con las mismas características de solapamiento de horas y grado de compactación. Verá algo similar a lo mostrado en la siguiente imagen.

Schedule Manager		Inicio	Mis horarios	Perfil	Salir
20:00	Algoritmos (ADA Teoría 1)				
21:00	Análisis y Diseño de Bases de Datos (ADB Teoría 1)				

**Opciones de horario**

[Descargar como PDF](#)

---

[Guardar horario](#)

**Nombre del horario:**

Horario 2

Horario 3

Horario 4

Horario 5

Schedule Manager - Diseña tu horario © 2019 | Made by Miguel Rojo

Figura A.19: Operaciones adicionales de la página de horario.

### Consulta de horarios guardados

Podrá acceder en todo momento a aquellos horarios que haya guardado accediendo al menú “Mis horarios” que tendrá accesible desde la barra de navegación superior y desde el menú principal de estudiante. Al acceder verá un listado con sus horarios (Figura A.20) y dos posibles acciones para cada uno.

Si hace clic en la opción de visualización se cargará la página de horario con los datos del horarios guardado. Si decide eliminar el horario pasará a un formulario de confirmación del borrado del horario.

### Mis horarios



Horario	Acciones
Horario de pruebas	 

Figura A.20: Página de horarios guardados del estudiante.

## Perfil

La consulta y gestión de su perfil estará accesible a través del menú principal o del enlace correspondiente en la barra de navegación superior. Al acceder se mostrará un formulario como el que aparece en la siguiente figura.

Ajustes de perfil

**Nombre de usuario:**

**Correo electrónico:**

**Contraseña:**

Figura A.21: Formulario de ajustes de perfil.

En el caso de que quiera modificar algún campo podrá hacerlo, y para que tenga efecto deberá pulsar el botón “Actualizar perfil”. Si existe algún problema de duplicado de nombre de usuario y/o de cuenta de correo electrónico se le informará al enviar el formulario.





---

## Apéndice B

# Contenido del CD

El contenido del disco que acompaña a este trabajo es el siguiente:

- **Código fuente:** carpeta que contiene el código fuente de la aplicación desarrollada.
- **Memoria:** carpeta con la presente memoria en formato PDF.