



UNIVERSIDAD DE GRANADA

TRABAJO FIN DE GRADO

CalendarUGR - Sistema de gestión personalizada de horarios académicos para la Universidad de Granada

Realizado por
Juan Miguel Acosta Ortega

ETSIIT

Escuela Técnica Superior
de Ingenierías Informática
y de Telecomunicación



**Para la obtención del título de
Grado en Ingeniería Informática**

**Dirigido por
Juan Luis Jiménez Laredo**

**En el departamento de
Dpto. de Ingeniería de Computadores, Automática y Robótica**

Convocatoria de junio, curso 2024/25

Aquí la dedicatoria del trabajo

Agradecimientos

Quiero agradecer a X por...

También quiero agradecer a Y por...

Resumen

Incluya aquí un resumen de los aspectos generales de su trabajo, en español.

Palabras clave: Palabra clave 1, palabra clave 2, ..., palabra clave N

Abstract

This section should contain an English version of the Spanish abstract.

Keywords: Keyword 1, keyword 2, ..., keyword N

Yo, **Juan Miguel Acosta Ortega**, alumno de la titulación Grado en Ingeniería INformática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 54313742R, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Juan Miguel Acosta Ortega

Granada, 30 de marzo de 2025

D. **Juan Luis Jiménez Laredo**, Profesor del Área de XXXX del Departamento de Ingeniería de Computadores, Automática y Robótica de la Universidad de Granada.

Informa:

Que el presente trabajo, titulado *xxxxxx*, ha sido realizado bajo su supervisión por **Juan Miguel Acosta Ortega**, y autorizo la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expide y firma el presente informe en Granada, 30 de marzo de 2025.

El director: Juan Luis Jiménez Laredo

Índice general

1	Introducción	1
1.1.	Motivación	1
1.2.	Estructura de la memoria	1
2	Descripción del problema	2
2.1.	El problema	2
2.2.	La solución	2
2.3.	Restricciones	2
2.4.	Objetivos	2
3	Estado del arte	3
3.1.	Contextualización	3
3.1.1.	Información de horarios académicos en la UGR	3
3.2.	Sistemas de Horarios Académicos Existentes	5
3.3.	Arquitectura de microservicios	5
3.4.	Conclusión	5
4	Especificación de requisitos	6
4.1.	Metodología de desarrollo	6
4.1.1.	Roles y Responsabilidades en este Proyecto	6
4.1.2.	Proceso Scrum Implementado	6
4.1.3.	Gestión de Tareas y Seguimiento del Progreso	7
4.1.4.	Justificación de la Metodología	8
4.2.	Personas	9
4.3.	Escenarios	9
4.4.	Historias de usuario	9
4.4.1.	Estructura de una historia de usuario	9
4.4.2.	Historias de usuario	10
4.5.	Requisitos funcionales	21
4.5.1.	Gestión de usuarios	21
4.5.2.	Gestión de horarios académicos	22
4.6.	Requisitos no funcionales	23
4.6.1.	Rendimiento	23
4.6.2.	Usabilidad	23
4.6.3.	Seguridad	24
4.6.4.	Mantenibilidad	24
4.6.5.	Portabilidad	24
4.6.6.	Disponibilidad	24
4.7.	Requisitos de información	25
4.8.	Validación de los requisitos	25
4.9.	Conclusiones	25

5	Diseño del sistema	26
5.1.	Arquitectura del sistema	26
5.1.1.	Arquitectura de microservicios	26
5.1.2.	Tecnologías y Frameworks	26
5.1.3.	Diseño de la base de datos	26
5.2.	Diseño de la Interfaz de Usuario (UI) y la Experiencia del Usuario (UX)	26
5.3.	Diseño de la API	26
5.4.	Seguridad en el sistema	26
5.5.	Diseño de pruebas	26
5.6.	Conclusiones	26
6	Planificación del proyecto	27
6.1.	Cronograma del proyecto	27
6.1.1.	Presupuesto del proyecto	27
6.1.2.	Gestión de riesgos	27
6.1.3.	Herramientas de gestión	27
7	Implementación	28
7.1.	Iteración 0	28
7.2.	Iteración 1	28
7.3.	Iteración 2	28
7.4.	Iteración 3	28
7.5.	Iteración 4	28
7.6.	Iteración 5	28
8	Despliegue del sistema	29
8.1.	Dockerización de la aplicación	29
9	Conclusiones y trabajos futuros	30
9.1.	Evaluación del proyecto	30
9.2.	Dificultades y resolución	30
9.3.	Mejoras posibles y trabajos futuros	30
	Bibliografía	31
	Anexo A: Glosario	32

Índice de figuras

3.1. Comparación de horarios de diferentes grados: ETSIIT (arriba) y ADE (abajo).	4
4.1. Tablero del 2º Sprint durante su desarrollo.	8
4.2. Segmento del "Roadmap" del proyecto.	8

Índice de tablas

4.1. Estructura de una historia de usuario	10
4.2. Historia de usuario HU-1	10
4.3. Historia de usuario HU-2	11
4.4. Historia de usuario HU-3	11
4.5. Historia de usuario HU-4	12
4.6. Historia de usuario HU-5	12
4.7. Historia de usuario HU-6	13
4.8. Historia de usuario HU-7	13
4.9. Historia de usuario HU-8	13
4.10. Historia de usuario HU-9	14
4.11. Historia de usuario HU-10	14
4.12. Historia de usuario HU-11	15
4.13. Historia de usuario HU-12	15
4.14. Historia de usuario HU-13	16
4.15. Historia de usuario HU-14	17
4.16. Historia de usuario HU-15	18
4.17. Historia de usuario HU-16	19
4.18. Historia de usuario HU-17	19
4.19. Historia de usuario HU-18	20
4.20. Historia de usuario HU-19	20
4.21. Historia de usuario HU-20	21
4.22. Historia de usuario HU-21	21

1. Introducción

1.1. Motivación

1.2. Estructura de la memoria

2. Descripción del problema

2.1. El problema

2.2. La solución

2.3. Restricciones

2.4. Objetivos

3. Estado del arte

3.1. Contextualización

3.1.1. Información de horarios académicos en la UGR

La Universidad de Granada, al igual que muchas otras universidades descentraliza sus sedes, de modo que cada una de ellas tiene su propio sistema de gestión de la información. En este sentido, las facultades cuentan con una serie de sistemas de información propios que se encargan de la generación de horarios académicos, asignación de aulas y profesores a los grupos tanto de teoría como de prácticas de las distintas titulaciones y asignaturas, etc. Y esta información a su vez se le facilita a la Universidad de Granada para la centralización de la información.

Para acceder a la información de los horarios, los estudiantes y docentes pueden hacerlo de diferentes maneras:

- A través de la página propia de su facultad. Poniendo de ejemplo a la ETSIIT, debemos acceder a la página “<https://etsiit.ugr.es>” y buscar la información en la sección de “Calendario de exámenes” en caso de querer saber los días y rangos horarios de estos y visualizándolo con un pdf, o a “Calendario académico y horarios” y a “Grado en Ingeniería Informática” en caso de querer saber los horarios de los diferentes grupos del grado, presentado todo ello en un pdf contenedor de alrededor de 40 tablas.

De esta manera tendremos que buscar el año al que pertenece la asignatura de la que estamos matriculados y el grupo al que pertenecemos. De esta manera obtenemos su franja horaria y aula, pero no profesor que imparte la asignatura.

Sin embargo, el formato de las tablas cambia de un grado a otro, haciendo que el estudiante tenga que buscar la información de manera diferente en cada grado si está matriculado en más de uno, y obteniendo información diferente. En el caso del grado de Administración y Dirección de Empresas por ejemplo, no se muestra el aula en la que se imparte la clase, pero sí las asignaturas bilingües, y los profesores que las imparten.

Esta forma de visualización de horarios es muy poco eficiente, ya que el estudiante tiene que buscar la información de manera manual, es inconsistente entre grados, y no es accesible para personas con discapacidad visual.

1º A Grado en Ingeniería Informática 1er. cuatrimestre									
Lunes		Martes		Miércoles		Jueves		Viernes	
8:30-9:30									
9:30-10:30	ALEM	ALEM	FP	FP	FP	FP	FP	FP	FP
10:30-11:30	CA	CA	FP	FP	FP	FP	FP	FP	FP
11:30-12:30	CA (A1)	FP (A1)	FP (A2)	FP (A2)	FP (A2)	FP (A2)	FP (A2)	FP (A2)	FP (A2)
12:30-13:30	ALEM (A1)	FP (A1)	FP (A2)	FP (A2)	FP (A2)	FP (A2)	FP (A2)	FP (A2)	FP (A2)
13:30-14:30									
15:30-16:30									
16:30-17:30									
17:30-18:30									
18:30-19:30									
19:30-20:30									
20:30-21:30									

























1º A GADE. PRIMER SEMESTRE (D03)					
	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
8:30 a 9:30	MAT  (E20)	MAT  (D03)  (E20)		FDAE	
9:30 a 10:30	MAT  (E20)	MAT  (D03)  (E20)		FDAE	
10:30 a 11:30	IOF	IOF	IMK  (D03)  (D25)	MAT  (D03)	
11:30 a 12:30	IOF	IOF	IMK  (D03)  (D25)	MAT  (D03)	
12:30 a 13:30	EP  (D03)  (E20)	EP  (D03)  (E20)	FDAE	IMK  (D03)  (D25)	
13:30 a 14:30	EP  (D03)  (E20)	EP  (D03)  (E20)	FDAE	IMK  (D03)  (D25)	

Figura 3.1: Comparación de horarios de diferentes grados: ETSIIT (arriba) y ADE (abajo).

- A través de la web “<https://grados.ugr.es/>” se puede buscar la información de los horarios de las asignaturas de los diferentes grados de la Universidad de Granada. Para ello debemos seleccionar rama de conocimiento, grado, curso y asignatura. De esta manera obtenemos un horario semanal con las franjas horarias, aulas, profesores y fechas tanto de inicio como de fin. Este método nos proporcionan una interfaz estándar y más información, pero también es más lento y tedioso para consultar por varias asignaturas o incluso grados.
- A través de las webs de cada departamento. Por ejemplo en la web de “<https://decsai.ugr.es/>” se puede consultar la información de las asignaturas o profesores de este. Ofrece información adicional como asignaturas que imparte “x” profesor y su horario de tutorías y docencia.

Además para acceder a la información de periodos de actividad docente, exámenes finales, periodos de evaluación de convocatorias ... se ha de acceder a la web de “<https://secretariageneral.ugr.es/>” para consultar otro pdf.

En general la información de los horarios académicos de la Universidad de Granada es poco accesible, eficiente y consistente entre grados y facultades, lo que hace que el estudiante tenga que buscar la información de manera manual y tediosa. Además no hay manera de consultar de manera sencilla un calendario

personal que incluya tanto los horarios de las asignaturas como los exámenes y periodos de evaluación, entre otros.

3.2. Sistemas de Horarios Académicos Existentes

3.3. Arquitectura de microservicios

3.4. Conclusión

4. Especificación de requisitos

4.1. Metodología de desarrollo

Para la gestión y desarrollo del proyecto, se ha optado por la metodología ágil [Scrum](#). Esta metodología se caracteriza por su enfoque iterativo e incremental, permitiendo una adaptación flexible a los cambios y una entrega temprana de valor.

4.1.1. Roles y Responsabilidades en este Proyecto

Dada la naturaleza individual de este proyecto, los roles tradicionales de Scrum se han adaptado de la siguiente manera:

- **Equipo de Desarrollo y Scrum Master:** El autor de este TFG ha asumido ambos roles. Esto implica la responsabilidad de llevar a cabo el desarrollo del software, así como de facilitar el proceso Scrum, asegurando que se sigan las prácticas y principios de la metodología. Se ha encargado de la planificación, ejecución y revisión de cada sprint, así como de la identificación y resolución de impedimentos.
- **Product Owner:** El rol de Product Owner ha sido desempeñado tanto por el director del TFG, D. Juan Luis Jiménez Laredo, como por el autor del sistema. En esta función, ambos han sido los responsables de definir la visión del producto, priorizar el Backlog del Producto y asegurar que el desarrollo se alinee con las necesidades y expectativas del proyecto. Los dos participaron activamente en la definición de los requisitos y en la validación de los incrementos de software.

4.1.2. Proceso Scrum Implementado

El proceso Scrum se ha implementado siguiendo los siguientes pasos clave:

- **Backlog del Producto:** Se ha definido un Backlog del Producto inicial, compuesto por las funcionalidades y tareas necesarias para completar el TFG.
- **Sprints:** El desarrollo se ha dividido en sprints de duración determinada (la duración específica de los sprints se definirá en el apartado de planificación [6](#)). Cada sprint ha tenido como objetivo la entrega de un incremento de software funcional y potencialmente entregable.
- **Planificación del Sprint:** Al inicio de cada sprint, se ha llevado a cabo una reunión de planificación en la que, junto con el Product Owner, se han seleccionado los elementos del Backlog del Producto que se abordarían

durante el sprint. Se han estimado las tareas y se ha definido el Sprint Backlog.

- **Desarrollo del Sprint:** Durante el sprint, el autor ha trabajado en el desarrollo de las tareas asignadas, siguiendo las prácticas de desarrollo y asegurando la calidad del código.
- **Reunión Diaria (Daily Scrum):** Aunque adaptada a la naturaleza individual del proyecto, se ha realizado una reflexión diaria sobre el progreso, los impedimentos y las tareas a realizar. Esto ha permitido mantener un seguimiento constante del avance.
- **Revisión del Sprint (Sprint Review):** Al finalizar cada sprint, se ha llevado a cabo una revisión del sprint. Dado que el autor es también el equipo de desarrollo, esta revisión ha consistido en una **introspección personal y un análisis de los resultados del sprint**, evaluando las metas alcanzadas y el incremento de software desarrollado. Se ha realizado una autoevaluación del progreso y la calidad del trabajo.
- **Retrospectiva del Sprint (Sprint Retrospective):** La retrospectiva del sprint se ha realizado en colaboración con el Product Owner (D. Juan Luis Jiménez Laredo). En esta reunión, se ha analizado el sprint finalizado, identificando qué se ha hecho bien, qué se podría mejorar y qué acciones concretas se podrían implementar para el siguiente sprint. Esta colaboración ha permitido obtener una perspectiva externa y valiosa para la mejora continua del proceso.

4.1.3. Gestión de Tareas y Seguimiento del Progreso

Para la gestión de las tareas y el seguimiento del progreso del proyecto, se ha utilizado **GitHub Projects**. Esta herramienta ha permitido:

- **Creación de Tableros por Sprint:** Se han configurado tableros de proyecto en GitHub Projects, utilizando las funcionalidades de **“Iteraciones”** para representar cada sprint. Esto ha facilitado la visualización del trabajo en curso para cada iteración.
- **Gestión del Estado de las Tareas:** Dentro de cada tablero de sprint, se han creado y gestionado las tareas individuales, asignándoles diferentes estados (**“Backlog”**, **“Todo”**, **“In progress”**, **“Testing”**, **“Done”**). Esto ha permitido un seguimiento visual del avance de cada tarea y del sprint en general.
- **Organización y Priorización:** GitHub Projects ha facilitado la organización de las tareas y su priorización dentro de cada sprint, alineándose con el Sprint Backlog definido.
- **Visualización de las tareas en el tiempo:** La herramienta ha permitido visualizar el progreso de las tareas en el tiempo a través de un roadmap, lo que ha facilitado la identificación de posibles retrasos y la toma de decisiones para ajustar el plan si es necesario.

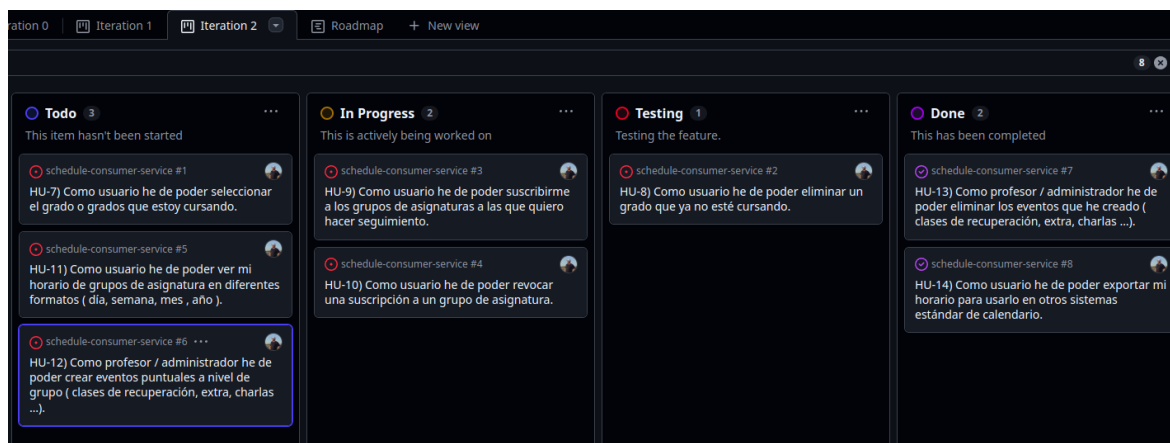


Figura 4.1: Tablero del 2º Sprint durante su desarrollo.

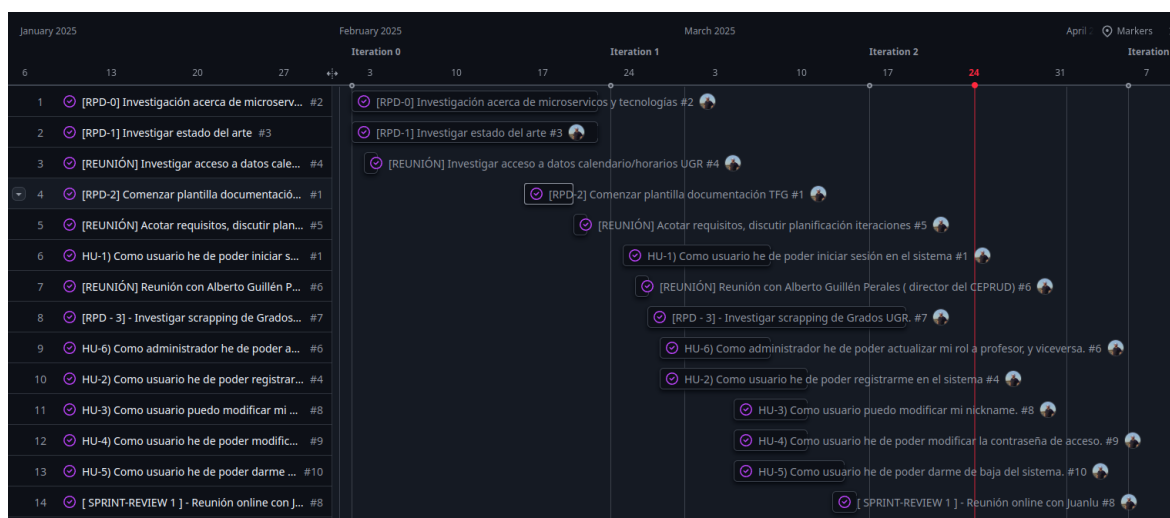


Figura 4.2: Segmento del "Roadmap" del proyecto.

4.1.4. Justificación de la Metodología

La elección de la metodología Scrum se justifica por las siguientes razones:

- **Flexibilidad:** Permite adaptarse a los cambios en los requisitos y a los aprendizajes obtenidos durante el desarrollo. En concreto este sistema dependía en etapas tempranas de desarrollo del posible acceso a datos oficiales de la UGR, sistemas de autenticación internos, datos de matriculaciones, etc. Es por ello que la flexibilidad de Scrum ha sido clave para ajustar el plan a medida que se han ido conociendo más detalles.
- **Entrega Temprana de Valor:** Facilita la entrega de incrementos funcionales de software de forma regular, lo que permite obtener retroalimentación temprana y ajustar el rumbo del proyecto si es necesario.
- **Transparencia:** El uso de herramientas como GitHub Projects y la realización de las reuniones Scrum promueven la transparencia en el progreso del proyecto.

- **Adaptabilidad a un Proyecto Individual:** Aunque tradicionalmente Scrum se aplica a equipos, su estructura iterativa y adaptable se ajusta bien a un proyecto individual como un TFG, permitiendo una organización eficiente del trabajo y una gestión del tiempo efectiva.

Es importante destacar que, dada la naturaleza individual del proyecto, se ha realizado una adaptación de los roles y las ceremonias de Scrum para ajustarse a las necesidades y recursos disponibles. Sin embargo, se han mantenido los principios fundamentales de la metodología para asegurar una gestión eficaz del desarrollo.

4.2. Personas

4.3. Escenarios

4.4. Historias de usuario

Una historia de usuario es una explicación general e informal de una función de software escrita desde la perspectiva del usuario final. Su propósito es articular cómo proporcionará una función de software valor al cliente. [1].

Estas no usan un lenguaje técnico y preciso para definir y acotar los requisitos de un sistema, sino que se enfocan en el usuario final y en cómo este interactuará con el sistema. Por lo tanto, las historias de usuario son una herramienta de comunicación entre el equipo de desarrollo y el cliente.

En **Scrum** las historias de usuario son una parte fundamental del proceso de desarrollo de software. En este marco de trabajo, las historias de usuario son utilizadas para definir los requisitos del sistema y son la base para la planificación y estimación de las tareas a realizar.

¿Por qué son importantes las historias de usuario?

- Centran la atención en el usuario final.
- Permiten la colaboración y comunicación entre el equipo de desarrollo y el cliente.
- Fomentan soluciones creativas y flexibles.

4.4.1. Estructura de una historia de usuario

Las historias de usuario siguen una estructura general simple y clara.

Como [tipo de usuario], **quiero** [realizar una acción], **para** [obtener un beneficio].

- **Como:** describe el tipo de usuario que está interactuando con el sistema.

- **Quiero:** describe la acción que el usuario desea realizar.
- **Para:** describe el beneficio que el usuario obtendrá al realizar la acción.

Además de esta estructura general, las historias de usuario pueden incluir otros elementos como criterios de aceptación, prioridad, estimación de esfuerzo, entre otros.

Para se ha definido la siguiente estructura para las historias de usuario:

ID	Identificador único de la historia de usuario.	Nombre	Nombre de la historia de usuario.
Descripción		Descripción general de la historia de usuario.	
Estimación		Estimación del esfuerzo necesario para completar la historia de usuario. Basado en Planning Poker.	
Prioridad		Acción que el usuario desea realizar. Desde P3 (baja) hasta P0 (alta).	
Criterios de aceptación		Conjunto de condiciones que deben cumplirse para considerar la historia de usuario como completada.	

Tabla 4.1: Estructura de una historia de usuario

4.4.2. Historias de usuario

ID	HU-1	Nombre	Iniciar sesión
Descripción		Como usuario he de poder iniciar sesión en el sistema.	
Estimación		3	
Prioridad		P0	
Criterios de aceptación		<ul style="list-style-type: none"> ■ Para poder iniciar sesión ha de insertar su correo y contraseña. ■ Sólo se puede iniciar sesión con correos de la UGR. 	

Tabla 4.2: Historia de usuario HU-1

ID	HU-2	Nombre	Registrarse
Descripción		Como usuario he de poder registrarme en el sistema.	
Estimación		5	
Prioridad		P0	
Criterios de aceptación		<ul style="list-style-type: none"> ■ El alumno sólo se puede registrar con su correo institucional de la UGR. ■ El alumno debe insertar nickname, correo y contraseña. ■ La contraseña del alumno ha de ser mayor o igual a 9 caracteres, conteniendo esta una mayúscula y un número como mínimo. ■ El registro se ha de completar mediante un link mandado por mail. 	

Tabla 4.3: Historia de usuario HU-2

ID	HU-3	Nombre	Modificar nickname
Descripción		Como usuario puedo modificar mi nickname.	
Estimación		2	
Prioridad		P2	
Criterios de aceptación		<ul style="list-style-type: none"> ■ El alumno no puede cambiar su nickname a otro que exista. ■ El alumno no puede modificar su correo electrónico. 	

Tabla 4.4: Historia de usuario HU-3

ID	HU-4	Nombre	Modificar contraseña
Descripción	Como usuario he de poder modificar la contraseña de acceso.		
Estimación	3		
Prioridad	P1		
Criterios de aceptación	<ul style="list-style-type: none"> ■ Para poder modificar la contraseña ha de insertar la contraseña anterior. ■ Se ha de insertar la nueva contraseña 2 veces, siendo esta mayor o igual a 9 caracteres, y conteniendo una mayúscula y un número como mínimo. 		

Tabla 4.5: Historia de usuario HU-4

ID	HU-5	Nombre	Darse de baja
Descripción	Como usuario he de poder darme de baja del sistema.		
Estimación	1		
Prioridad	P2		
Criterios de aceptación	<ul style="list-style-type: none"> ■ Para poder completar la baja ha de escribir su contraseña en un campo de texto. 		

Tabla 4.6: Historia de usuario HU-5

ID	HU-6	Nombre	Cambiar rol
Descripción	Como administrador he de poder actualizar mi rol a profesor, y viceversa.		
Estimación	2		
Prioridad	P1		
Criterios de aceptación	<ul style="list-style-type: none"> ■ Para poder cambiar el rol a profesor he de ser administrador. ■ Para poder cambiar el rol a profesor he de ser administrador. 		

Tabla 4.7: Historia de usuario HU-6

ID	HU-7	Nombre	Seleccionar grados
Descripción	Como usuario he de poder seleccionar el grado o grados que estoy cursando.		
Estimación	3		
Prioridad	P0		
Criterios de aceptación	<ul style="list-style-type: none"> ■ Se pueden seleccionar un máximo de 4 grados. 		

Tabla 4.8: Historia de usuario HU-7

ID	HU-8	Nombre	Eliminar grado
Descripción	Como usuario he de poder eliminar un grado que ya no esté cursando.		
Estimación	2		
Prioridad	P1		
Criterios de aceptación	<ul style="list-style-type: none"> ■ Si el usuario está suscrito a grupos de asignatura de ese grado, se le recordará que también se revocarán sus suscripciones a estos. 		

Tabla 4.9: Historia de usuario HU-8

ID	HU-9	Nombre	Suscribirse a grupos de asignatura
Descripción		Como usuario he de poder suscribirme a los grupos de asignaturas a las que quiero hacer seguimiento.	
Estimación		4	
Prioridad		P0	
Criterios de aceptación		<ul style="list-style-type: none"> ■ Para hacer seguimiento a un grupo en concreto, el usuario deberá estar cursando el grado al que pertenece. 	

Tabla 4.10: Historia de usuario HU-9

ID	HU-10	Nombre	Revocar suscripción a grupo de asignatura
Descripción		Como usuario he de poder revocar una suscripción a un grupo de asignatura.	
Estimación		2	
Prioridad		P1	
Criterios de aceptación		<ul style="list-style-type: none"> ■ El usuario sólo puede revocar suscripciones de grupos a los que está suscrito. 	

Tabla 4.11: Historia de usuario HU-10

ID	HU-11	Nombre	Ver horario de grupos de asignatura
Descripción		Como usuario he de poder obtener la información de mi horario personalizado conforme a las suscripciones.	
Estimación		5	
Prioridad		P0	
Criterios de aceptación		<ul style="list-style-type: none"> ■ El horario de cada clase debe mostrar la asignatura, grupo, hora de inicio y fin, profesores del grupo, y aula. ■ Las clases sólo deben mostrarse en el rango de fechas en las que se imparten. 	

Tabla 4.12: Historia de usuario HU-11

ID	HU-12	Nombre	Crear evento puntual a nivel de grupo de asignatura
Descripción		Como profesor / administrador he de poder crear eventos puntuales a nivel de grupo (clases de recuperación, extra, charlas ...).	
Estimación		3	
Prioridad		P0	
Criterios de aceptación		<ul style="list-style-type: none"> ■ Se debe especificar la fecha, hora de inicio, hora de fin y tipo de evento. ■ La clase extra no debe coincidir con otra clase existente en horario y aula. ■ El usuario debe ser un profesor o administrador. 	

Tabla 4.13: Historia de usuario HU-12

ID	HU-13	Nombre	Eliminar evento puntual a nivel de grupo de asignatura
Descripción		Como profesor / administrador he de poder eliminar los eventos que he creado (clases de recuperación, extra, charlas ...).	
Estimación		2	
Prioridad		P1	
Criterios de aceptación		<ul style="list-style-type: none"> ■ Se debe especificar la fecha, hora de inicio, hora de fin y tipo de evento. ■ La clase extra no debe coincidir con otra clase existente en horario y aula. ■ El usuario debe ser un profesor o administrador. 	

Tabla 4.14: Historia de usuario HU-13

ID	HU-14	Nombre	Exportar horario a calendario estándar
Descripción		Como usuario he de poder exportar mi horario para usarlo en otros sistemas estándar de calendario.	
Estimación		4	
Prioridad		P0	
Criterios de aceptación		<ul style="list-style-type: none"> ■ El usuario podrá exportar su horario en formatos compatibles con sistemas estándar de calendario (.ics). ■ La exportación debe incluir todas las asignaturas y eventos del usuario. ■ Se debe permitir elegir un rango de fechas para la exportación. ■ El archivo generado debe poder descargarse y ser importable en Google Calendar, Outlook, Apple Calendar, etc. 	

Tabla 4.15: Historia de usuario HU-14

ID	HU-15	Nombre	Sincronizar calendario con Google Calendar
Descripción		Como usuario he de poder sincronizar mi calendario con Google Calendar.	
Estimación		5	
Prioridad		P3	
Criterios de aceptación		<ul style="list-style-type: none"> ■ El usuario podrá vincular su cuenta con Google Calendar mediante OAuth. ■ Los eventos de su horario deben sincronizarse automáticamente con Google Calendar. ■ Se deben reflejar en Google Calendar los cambios realizados en el horario del usuario. ■ El usuario debe poder desactivar la sincronización en cualquier momento. 	

Tabla 4.16: Historia de usuario HU-15

ID	HU-16	Nombre	Ver alertas de clases extra de grupos de asignatura
Descripción		Como alumno he de poder recibir alertas referentes a “clases extra” de mis grupos (clases de recuperación, extra, charlas ...).	
Estimación		3	
Prioridad		P1	
Criterios de aceptación		<ul style="list-style-type: none"> ■ Las clases extra aparecerán, como las clases, en la vista del horario. ■ Se debe mandar un correo electrónico a los alumnos que afecte. 	

Tabla 4.17: Historia de usuario HU-16

ID	HU-17	Nombre	Crear evento a nivel de facultad
Descripción		Como administrador he de poder crear eventos a nivel de facultad (charlas, conferencias, exámenes ...).	
Estimación		5	
Prioridad		P1	
Criterios de aceptación		<ul style="list-style-type: none"> ■ Se debe especificar la fecha, hora de inicio, hora de fin y tipo de evento. ■ El evento no debe coincidir con otra clase existente en horario y aula. ■ El usuario debe ser un administrador. 	

Tabla 4.18: Historia de usuario HU-17

ID	HU-18	Nombre	Registrar días festivos de facultad
Descripción		Como administrador puedo registrar los días festivos de mi facultad.	
Estimación		3	
Prioridad		P1	
Criterios de aceptación		<ul style="list-style-type: none"> ■ El usuario debe ser un administrador. ■ El día a registrar no puede estar ya registrado. 	

Tabla 4.19: Historia de usuario HU-18

ID	HU-19	Nombre	Recibir alertas de cambios en asignaturas suscritas
Descripción		Como alumno debo poder recibir alertas de cambios en las asignaturas a las que estoy suscrito.	
Estimación		4	
Prioridad		P1	
Criterios de aceptación		<ul style="list-style-type: none"> ■ La alerta debe llegar a los usuarios que estén suscritos a ese grupo de asignatura. 	

Tabla 4.20: Historia de usuario HU-19

ID	HU-20	Nombre	Eliminar evento a nivel de facultad
Descripción		Como profesor / administrador he de poder eliminar los eventos que he creado a nivel de facultad (charlas, conferencias, exámenes ...).	
Estimación		3	
Prioridad		P2	
Criterios de aceptación		<ul style="list-style-type: none"> ■ El usuario debe ser un administrador. 	

Tabla 4.21: Historia de usuario HU-20

ID	HU-21	Nombre	Eliminar día festivo de facultad
Descripción		Como administrador he de poder eliminar los días festivos de mi facultad.	
Estimación		3	
Prioridad		P2	
Criterios de aceptación		<ul style="list-style-type: none"> ■ El usuario debe ser un administrador. 	

Tabla 4.22: Historia de usuario HU-21

4.5. Requisitos funcionales

A partir de las historias de usuario, junto a sus criterios de aceptación, se han extraído los siguientes requisitos funcionales:

4.5.1. Gestión de usuarios

- **RF-1) Gestión de usuarios:** El sistema debe poder registrar usuarios para futuros inicio de sesión y seguimiento de su información de suscripciones a grupos de asignaturas.
 - **RF-1.1) Inicio de sesión:** El sistema debe permitir el inicio de sesión de usuarios mediante correo electrónico institucional y contraseña.
 - **RF-2.2) Registro de usuarios:** El sistema debe tener un proceso de registro de usuario.

- **RF-2.3) Completar el registro:** Para completar el registro el sistema debe mandar un mail para confirmar si el usuario se trata de un alumno o de un profesor.
- **RF-2.4) Cambio de nickname:** El sistema debe permitir cambiar el nickname al usuario por uno no usado.
- **RF-2.5) Cambio de contraseña de acceso:** El sistema debe permitir el cambio de la contraseña de acceso.
- **RF-2.6) Dar de baja:** El sistema debe permitir al usuario darse de baja con el objetivo de que no le lleguen más correos relacionados.
- **RF-2.7) Cambio de rol:** EL sistema debe poder facilitar el cambio de rol de profesor a administrador, y viceversa.

4.5.2. Gestión de horarios académicos

- **RF-2) Gestión de horarios académicos:** El sistema debe poder obtener la información relacionada con el horario académico de todos los grados de la UGR, para así poder identificar los horarios personalizados de alumnos y docentes a través de un sistema de suscripción a grupos de asignatura.
 - **RF-2.1) Recopilación de horarios:** El sistema debe recopilar la información de horarios académicos de todos los grados de la UGR.
 - **RF-2.2) Grados del alumno / profesor:** El sistema debe recoger el grado/ grados académicos que está cursando / impartiendo el alumno / profesor.
 - **RF-2.3) Asignaturas del alumno / profesor:** El sistema debe recoger las asignaturas que está cursando / impartiendo el alumno / profesor.
 - **RF-2.4) Grupos del alumno / profesor:** El sistema debe recoger los grupos de las asignaturas que está cursando / impartiendo el alumno / profesor.
 - **RF-2.5) Eliminar grados del alumno / profesor:** El sistema debe poder eliminar el grado/ grados académicos que está cursando / impartiendo el alumno / profesor.
 - **RF-2.6) Eliminar asignaturas del alumno / profesor:** El sistema debe poder eliminar las asignaturas que está cursando / impartiendo el alumno / profesor.
 - **RF-2.7) Eliminar grupos del alumno / profesor:** El sistema debe poder eliminar los grupos de las asignaturas que está cursando / impartiendo el alumno / profesor.
 - **RF-2.8) Horario personalizado:** El usuario ha de poder acceder a la información de horario académico de los grupos de asignaturas a los que esté suscrito.

- **RF-2.9) Crear clases extra:** El sistema debe permitir al profesor / administrador crear clases extra a las oficiales.
- **RF-2.10) Eliminar clases extra:** El sistema debe permitir al profesor / administrador eliminar clases extra a las oficiales.
- **RF-2.11) Exportar horario a estándar:** El sistema debe poder exportar el horario en formato estándar (.ics).
- **RF-2.12) Sincronizar con Google calendar:** El sistema deberá poder sincronizarse con Google Calendar.
- **RF-2.13) Alertas sobre clases extra:** El sistema debe poder mandar alertas sobre clases extra a los alumnos de ese grupo.
- **RF-2.14) Crear eventos a nivel de facultad:** El sistema debe poder crear eventos a nivel de facultad.
- **RF-2.15) Registrar días festivos de facultad:** El sistema debe poder registrar los días festivos de la facultad.
- **RF-2.16) Alertas de cambios en asignaturas suscritas:** El sistema debe poder mandar alertas de cambios en las asignaturas suscritas.
- **RF-2.17) Eliminar eventos a nivel de facultad:** El sistema debe poder eliminar eventos a nivel de facultad.
- **RF-2.18) Eliminar días festivos de facultad:** El sistema debe poder eliminar días festivos de la facultad.

4.6. Requisitos no funcionales

4.6.1. Rendimiento

- **RNF-1.1) Tiempo de respuesta:** El sistema debe responder a las solicitudes de los usuarios en un tiempo máximo de 3 segundos.
- **RNF-1.2) Capacidad de usuarios concurrentes:** El sistema debe soportar un mínimo de 100 usuarios concurrentes sin degradación significativa del rendimiento.

4.6.2. Usabilidad

- **RNF-2.1) Interfaz intuitiva:** La interfaz de usuario debe ser fácil de usar y comprender, incluso para usuarios sin experiencia técnica.
- **RNF-2.2) Accesibilidad:** El sistema debe cumplir con las pautas de accesibilidad web (WCAG) para garantizar que sea utilizable por personas con discapacidades.

- **RNF-2.3) Compatibilidad con dispositivos:** El sistema debe ser compatible con cualquier navegador web, y a cualquier resolución.

4.6.3. Seguridad

- **RNF-3.1) Autenticación segura:** El sistema debe implementar un mecanismo de autenticación y autorización basado en JWT.
- **RNF-3.2) Protección de datos:** El sistema debe proteger los datos de usuario confidenciales (como contraseñas y correos electrónicos) mediante cifrado y otras medidas de seguridad.
- **RNF-3.3) Autorización:** El sistema debe controlar el acceso a las funciones del sistema según los roles de usuario (administrador, profesor, alumno).

4.6.4. Mantenibilidad

- **RNF-4.1) Modularidad:** El sistema debe estar diseñado de forma modular para facilitar el mantenimiento y la actualización.
- **RNF-4.2) Documentación:** El sistema debe estar debidamente documentado para facilitar la comprensión y el mantenimiento del código.
- **RNF-4.3) Pruebas:** El sistema debe incluir pruebas unitarias y de integración para garantizar la calidad del código.
- **RNF-4.4) Descubrimiento:** El sistema ha de tener un servicio de descubrimiento de servicios para facilitar la extensión del sistema.
- **RNF-4.5) Configuración:** El sistema ha de contar con un servidor de configuración para centralizarla.

4.6.5. Portabilidad

- **RNF-5.1) Independencia de plataforma:** El sistema debe ser independiente de la plataforma, lo que significa que debe poder ejecutarse en diferentes sistemas operativos y entornos de servidor.

4.6.6. Disponibilidad

- **RNF-6.1) Tiempo de actividad:** El sistema debe tener un tiempo de actividad del 99.9 %.
- **RNF-6.2) Recuperación ante fallos:** El sistema debe poder recuperarse de fallos de hardware o software sin pérdida de datos.

4.7. Requisitos de información

4.8. Validación de los requisitos

4.9. Conclusiones

En este capítulo concluimos que...

5. Diseño del sistema

5.1. Arquitectura del sistema

5.1.1. Arquitectura de microservicios

5.1.2. Tecnologías y Frameworks

5.1.3. Diseño de la base de datos

5.2. Diseño de la Interfaz de Usuario (UI) y la Experiencia del Usuario (UX)

5.3. Diseño de la API

5.4. Seguridad en el sistema

5.5. Diseño de pruebas

5.6. Conclusiones

En este capítulo concluimos que...

6. Planificación del proyecto

6.1. Cronograma del proyecto

6.1.1. Presupuesto del proyecto

6.1.2. Gestión de riesgos

6.1.3. Herramientas de gestión

7. Implementación

7.1. Iteración 0

7.2. Iteración 1

7.3. Iteración 2

7.4. Iteración 3

7.5. Iteración 4

7.6. Iteración 5

8. Despliegue del sistema

8.1. Dockerización de la aplicación

9. Conclusiones y trabajos futuros

9.1. Evaluación del proyecto

9.2. Dificultades y resolución

9.3. Mejoras posibles y trabajos futuros

Bibliografía

- [1] Atlassian. Historias de usuario, 2024. URL <https://www.atlassian.com/es/agile/project-management/user-stories>.

Anexo A: Glosario

A continuación se presenta un glosario con las definiciones de términos técnicos utilizados a lo largo del trabajo:

SCRUM : es un marco de trabajo ágil para el desarrollo de software. Se basa en la iteración y la colaboración entre los miembros del equipo de desarrollo.