

**ESCUELA POLITÉCNICA
SUPERIOR DE CÓRDOBA**
Universidad de Córdoba



TRABAJO FIN DE GRADO
Grado en Ingeniería Informática

UCOTrack: Aplicación Web para el seguimiento de TFGs.

Autor: Irene López Jiménez
Directores: María Luque Rodríguez, Antonio Araúzo Azofra

agosto, 2024



UNIVERSIDAD DE CÓRDOBA



*Dedicado a
mi familia*

Agradecimientos

Me gustaría expresar mi más sincero agradecimiento a todas las personas que han hecho posible la realización de este Trabajo de Fin de Grado.

En primer lugar, quiero dar las gracias a todos los profesores que he tenido durante estos cuatro años de carrera. Sus enseñanzas y conocimientos han sido esenciales para enfrentar este proyecto de TFG.

Quiero hacer un agradecimiento especial a mis profesores y tutores, Antonio y María. Gracias por estar siempre disponibles para escucharme y por compartir su experiencia y sabiduría. Su apoyo y orientación han sido clave para el avance y éxito de este proyecto.

Mi más sincero agradecimiento va dirigido a mi familia, quienes han sido una fuente constante de apoyo y motivación. A mis padres, por su amor incondicional, paciencia y aliento durante todo este proceso. Su comprensión y sacrificio han sido un pilar fundamental en mi camino académico. A mi hermano, por su apoyo y comprensión en momentos de estrés y dedicación. Y un agradecimiento especial a mi novio. Gracias por tu paciencia, por tu apoyo constante y por estar siempre a mi lado, incluso en los momentos más complicados; gracias por no hacer que me rinda.

También quiero agradecer a todos los amigos que he hecho durante estos 4 años en la universidad, por su apoyo y por estar siempre ahí para escucharme y animarme. Sobre todo a Álvaro, por ser un gran amigo y compañero.

A todos ustedes, muchas gracias por hacer posible este logro. Sin su ayuda, este proyecto no habría sido posible.

Índice general

Agradecimientos	II
Índice de figuras	VII
Índice de tablas	IX
1. Introducción	1
2. Descripción de problema	2
2.1. Descripción del problema actual	2
2.2. Descripción del problema técnico	3
2.2.1. Entorno	3
2.2.2. Vida esperada	3
2.2.3. Ciclo de mantenimiento	3
2.2.4. Programa de tareas	3
3. Objetivos	6
4. Antecedentes	7
4.1. Moodle	7
4.2. Trello	8
4.3. Tabla comparativa	8
5. Planificación temporal	11
6. Restricciones y estudios de alternativas	13
6.1. Factores dato	13
6.2. Factores estratégicos	14
6.2.1. Tipo de aplicación	14
6.2.2. Fronted	14
6.2.3. Backend	14
6.2.4. Otras tecnologías	14



7. Recursos	16
7.1. Recursos humanos	16
7.2. Recursos hardware	16
7.3. Recursos software	17
8. Especificación de requisitos	18
8.1. Introducción	18
8.1.1. Requisitos funcionales	18
8.1.2. Requisitos no funcionales	19
8.1.3. Requisitos de información	19
8.2. Casos de uso	19
8.2.1. Diagrama de casos de uso	20
8.2.2. Actores del sistema	20
8.2.3. CU01 - Registrar usuario	21
8.2.4. CU02 - Iniciar sesión	22
8.2.5. CU03 - Añadir TFG	23
8.2.6. CU04 - Añadir descripción TFG	24
8.2.7. CU05 - Cambiar estado TFG	25
8.2.8. CU06 - Crear seguimiento TFG	26
8.2.9. CU07 - Generar resumen seguimientos	27
8.2.10. CU08 - Listar TFGs por estado	28
8.2.11. CU9 - Modificar detalles TFG	29
8.2.12. CU10 - Eliminar TFG	30
8.3. Matriz de trazabilidad	31
9. Análisis de la información	32
9.1. Descripción del modelo conceptual de datos	32
9.2. Descripción de las clases	33
9.2.1. Clase TFG	33
9.2.2. Clase Usuario	33
9.2.3. Clase Seguimientos	34
9.3. Diagrama de actividad de los casos de uso	34
9.3.1. Iniciar sesión	34
9.3.2. Añadir TFG	35
9.3.3. Eliminar TFG	36
9.3.4. Modificar TFG	37
9.3.5. Crear seguimiento TFG	38



10. Diseño	40
10.1. Diagramas de interacción	40
10.1.1. Diagramas de secuencia	40
10.2. Diseño de la interfaz	42
10.2.1. Maquetas	43
10.3. Arquitectura	46
10.3.1. Backend	46
10.3.2. Frontend	48
10.3.3. Base de datos	49
11. Pruebas	52
11.1. Pruebas de caja blanca	52
11.2. Pruebas de caja negra	52
11.2.1. CP1: RegistrarNuevoUsuario	53
11.2.2. CP2: IniciarSesión	53
11.2.3. CP3: CrearTFG	53
11.2.4. CP4: EditarTFG	54
11.2.5. CP5: EliminarTFG	54
11.2.6. CP6: FiltrarTFGs	54
11.2.7. CP7: IniciarSesionCredencialesIncorrectas	55
11.2.8. CP8: RegistrarUsuarioConDatosInvalidos	55
11.2.9. CP9: CerrarSesión	55
11.3. Problemas encontrados	55
12. Conclusiones y Futuras Mejoras	57
12.1. Objetivos conseguidos	57
12.2. Objetivos no conseguidos	58
12.3. Aprendizaje	58
12.3.1. Next.js	59
12.3.2. NestJS	59
12.3.3. TypeScript	59
12.3.4. Estructuración de Proyectos	59
12.4. Futuras Mejoras	59
Bibliografía	61
A. Manual de Usuario	62
A.1. Requisitos previos	62
A.1.1. Instalación de Node.js	62
A.1.2. Instalación de Prisma	62
A.2. Instalación de la aplicación	63



A.2.1. Levantar la base de datos	63
A.2.2. Clonación del repositorio	63
A.3. Configuración del entorno	63
A.3.1. Variables de entorno	63
A.4. Ejecución de la aplicación	63
A.4.1. Arranque del frontend	64
A.4.2. Arranque del backend	64
A.5. Descripción de la Interfaz	64
A.5.1. Inicio de Sesión	64
A.5.2. Página de Inicio	66
A.5.3. Creación de TFG	67
A.5.4. Listado de TFGs	67
A.5.5. Confirmación de Eliminación de TFG	68
A.5.6. Filtrado de TFGs	69
A.5.7. Añadir Notas a un TFG	70
A.5.8. Descarga de Notas de TFG	71
A.5.9. Edición de TFG	71
A.5.10. Cierre de Sesión	71
B. Manual de Código	72
B.1. Descripción del Proyecto	72
B.2. Tecnologías y lenguajes utilizados	72
B.3. Requisitos y dependencias	73
B.4. Estructura del Código	74

Índice de figuras

2.1. WBS de la aplicación	4
4.1. Logotipo de Moodle.	7
4.2. Logotipo de Trello	8
8.1. Diagrama de casos de uso	20
9.1. Diagrama de clases	32
9.2. Diagrama de actividad: Iniciar sesión	35
9.3. Diagrama de actividad: Añadir TFG	36
9.4. Diagrama de actividad: Eliminar TFG	37
9.5. Diagrama de actividad: Modificar TFG	38
9.6. Diagrama de actividad: Crear seguimiento TFG	39
10.1. Diagrama de secuencia: Añadir TFG	41
10.2. Diagrama de secuencia: Listar TFG	42
10.3. MockUp Inicio de sesión	43
10.4. MockUp Página principal.	44
10.5. MockUp Creación TFG	45
10.6. MockUp Listado TFG.	45
10.7. Domain Driven Design (DDD)	47
10.8. Arquitectura hexagonal	48
10.9. Diagrama E-R	50
A.1. Página principal	65
A.2. Página de inicio de sesión	65
A.3. Página de Registro	66
A.4. Página de inicio	66
A.5. Formulario de creación de TFG	67
A.6. Listado de TFGs	68
A.7. Ventana de confirmación de eliminación de TFG	69
A.8. Menú desplegable de filtrado de TFGs	70



A.9. Pantalla para añadir notas a un TFG	70
A.10. Página de edición de un TFG	71

Índice de tablas

4.1. Comparación de funcionalidades entre UCOTrack, Trello y Moodle . .	9
5.1. Distribución aproximada y temporal de las fases del proyecto	12
5.2. Cronograma estimando cuantas semanas serán ocupadas para la rea- lización de cada tarea.	12
8.1. Matriz de trazabilidad	31

Capítulo 1

Introducción

Hoy en día, en el ámbito académico, la coordinación y seguimiento eficiente de los Trabajos de Fin de Grado (TFGs) se ha convertido en una tarea cada vez más compleja y crucial. Esta complejidad se debe a la necesidad de gestionar múltiples proyectos de diferentes estudiantes, así como en la demanda creciente de una gestión ágil y estructurada en el entorno universitario.

En respuesta a esta necesidad, ha surgido la iniciativa de desarrollar una aplicación web dedicada al seguimiento personalizado de TFGs. Actualmente, la gestión de estas actividades se realiza en gran medida de manera manual o mediante herramientas digitales limitadas, lo que dificulta la organización y el acceso a la información relevante.

La aplicación propuesta tiene como objetivo proporcionar una plataforma integral que permita a los profesores universitarios llevar un seguimiento detallado de cada TFG, desde su creación hasta su conclusión. Al igual que otras aplicaciones que utilizamos en nuestra vida diaria, esta aplicación busca aprovechar las tecnologías actuales para hacer más accesible y eficiente la gestión académica.

El propósito fundamental de este Trabajo de Fin de Grado es diseñar e implementar una herramienta que mejore significativamente la gestión de TFGs, ofreciendo funcionalidades que faciliten la planificación, seguimiento y evaluación de estos proyectos. A través de una interfaz intuitiva y fácil de usar, la aplicación permitirá a los profesores acceder a información actualizada sobre el estado y progreso de cada TFG, así como facilitar la comunicación y colaboración entre los diferentes participantes en el proceso.

Capítulo 2

Descripción de problema

2.1. Descripción del problema actual

A día de hoy, la gestión de los Trabajos de Fin de Grado (TFGs) dirigidos por profesores universitarios se realizan de manera manual y descentralizada, lo cual conlleva una serie de desafíos y limitaciones. Los profesores universitarios, encargados de supervisar múltiples TFGs, se enfrentan a dificultades para mantener un seguimiento eficiente de cada proyecto debido a la falta de una plataforma integral que centralice la información relevante. A todo esto, hay que añadir la creciente demanda de una gestión más ágil y estructurada en el ámbito académico.

En la mayoría de los casos, la gestión de TFGs se lleva a cabo utilizando métodos tradicionales, como el intercambio de correos electrónicos, documentos en papel o incluso hojas de cálculo compartidas. Estos enfoques son propensos a errores, pérdida de información y, además, dificultan la colaboración y comunicación entre los distintos participantes en el proceso.

La falta de una herramienta específica y dedicada para la gestión de TFGs dificulta el seguimiento del progreso de cada proyecto, la asignación de tareas y la comunicación entre los involucrados. Además, la ausencia de un sistema estructurado para registrar seguimientos, actualizar estados de los proyectos y generar resúmenes de actividad conlleva una pérdida de eficiencia y transparencia en el proceso de supervisión.

La gestión manual de los TFGs también puede resultar en problemas de organización y coordinación, especialmente cuando se trata de proyectos múltiples o cuando varios profesores tienen que supervisar un mismo proyecto. La falta de una plataforma centralizada dificulta la identificación y seguimiento de estos proyectos.

Además, la necesidad de mantener un registro detallado de cada TFG, incluyendo información sobre el alumno, el estado del proyecto, las etapas de desarrollo y los seguimientos realizados, representa una carga adicional para los profesores y el personal administrativo de las instituciones educativas.

Entonces, podríamos decir que el problema actual radica en la falta de una herramienta dedicada y eficiente para la gestión de TFGs, lo que resulta en una supervisión ineficiente, falta de organización, dificultades en la comunicación y colaboración, así como una pérdida de tiempo y recursos para los profesores y estudiantes involucrados en el proceso. Es por eso que surgió UCOTrack, con la intención de resolver todos estos problemas.

2.2. Descripción del problema técnico

2.2.1. Entorno

UCOTrack está centrado al seguimiento eficiente de TFGs en el ámbito académico, por lo tanto, este proyecto está dirigido a los profesores universitarios que se encarguen de supervisar y gestionar cada TFG, desde su creación hasta su conclusión. No es necesario que estos usuarios tengan conocimientos tecnológicos avanzados, ya que esta aplicación contará con un funcionamiento sencillo e intuitivo.

2.2.2. Vida esperada

La vida esperada de este proyecto será indefinida, ya que el tema a tratar (la gestión de TFGs) no es cambiante en el panorama actual académico.

2.2.3. Ciclo de mantenimiento

El sistema no contará con ciclo de mantenimiento, a excepción de bugs y errores que se encuentren cuando se ponga el sistema en producción.

2.2.4. Programa de tareas

Las tareas del proyecto se muestran en la figura 2.1

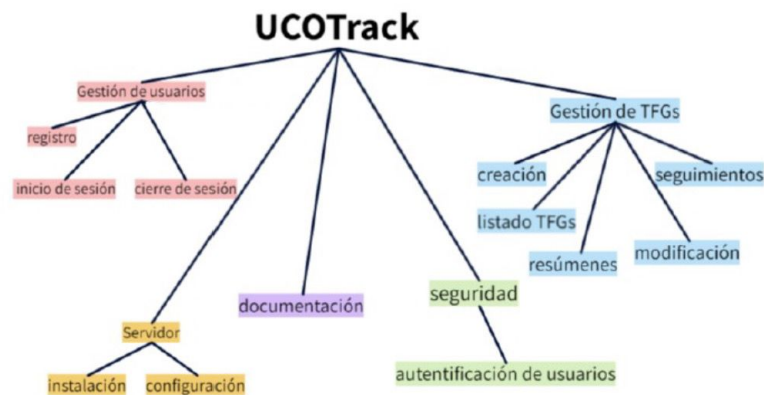


Figura 2.1: WBS de la aplicación

1. Servidor:

- a) Instalación y Configuración del Servidor: incluirá el proceso de instalación y configuración del servidor necesario para el funcionamiento de la aplicación web, asegurando su correcta operación.

2. Gestión de Usuarios:

- a) Registro de usuario: desarrollo de la funcionalidad que permitirá a los usuarios crear cuentas en la aplicación.
- b) Inicio de sesión: implementación de la opción de inicio de sesión para los usuarios registrados.
- c) Cierre de sesión: desarrollo de la opción para que los usuarios puedan cerrar sesión de manera segura.

3. Gestión de TFGs:

- a) Creación y modificación de TFGs: implementación de funcionalidades para la creación, modificación y eliminación de Trabajos de Fin de Grado.
- b) Listado de TFGs: desarrollo de una sección que mostrará un listado completo de todos los TFGs registrados.
- c) Creación de seguimientos: inclusión de la funcionalidad para crear seguimientos asociados a cada TFG.
- d) Generación de resúmenes de seguimiento: implementación de la opción para generar resúmenes detallados de seguimientos realizados.

4. Seguridad:

- a)* Autenticación de usuarios: uso de credenciales seguras, como correo electrónico y contraseña, para el proceso de inicio de sesión.

5. Documentación:

- a)* Manual de Usuario: creación de una guía detallada para los usuarios finales, explicando cómo utilizar cada función de la aplicación.
- b)* Manual de Código: detalles técnicos sobre el código fuente de la aplicación.

Capítulo 3

Objetivos

El objetivo principal de este Trabajo Fin de Grado es diseñar e implementar una aplicación web que permita una gestión eficiente de los TFGs que dirigen en cada curso los profesores universitarios. Este objetivo será realizado mediante otros objetivos más específicos enumerados a continuación:

- Acceder al sistema mediante usuario y contraseña.
- Crear TFGs y asignarlos a cursos académicos.
- Añadir información detallada a cada TFG, incluyendo el título, el alumno y sus datos de contacto.
- Enlazar el TFG con la petición de tema una vez aprobada
- Permitir la asignación y el cambio de estados a los TFGs (pendiente de petición, petición aprobada, desarrollo, entregado, presentado, ...).
- Crear seguimientos para cada TFG, registrando fechas y proporcionando un espacio para información relevante (qué se ha hecho, que se ha corregido, que se ha hablado...). La fecha que se asigna es, por defecto, la del sistema.
- Generar un documento txt con todos los seguimientos de un TFG a modo de resumen
- Listar los TFGs por tipo/estado para una mejor organización.
- Diseñar una interfaz intuitiva y amigable para que los profesores puedan gestionar TFGs.

Capítulo 4

Antecedentes

La idea de este proyecto surge a raíz de la carencia de una plataforma que permita a los docentes universitarios realizar un seguimiento detallado de cada Trabajo de Fin de Grado y gestionar la información de manera efectiva y ágil.

Las aplicaciones que ofrecen unas prestaciones más completas y similares a este proyecto son Moodle [1] y Trello [2].

4.1. Moodle



Figura 4.1: Logotipo de Moodle.

Por un lado, Moodle (figura 4.1), se destaca por ser un sistema de gestión de aprendizaje de código abierto que abarca una variedad de funciones, como la creación y administración de cursos en línea, evaluaciones, seguimiento del proceso del estudiante y colaboración en línea. Sin embargo, su enfoque generalizado no aborda las particularidades y necesidades específicas de la gestión detallada de TFGs.

4.2. Trello



Figura 4.2: Logotipo de Trello

Y, por otro lado, Trello (figura 4.2) es una herramienta de gestión de proyectos que permite la creación de tableros, listas y tarjetas para organizar tareas y colaborar en proyectos. En Trello, los usuarios pueden crear tarjetas para representar tareas o elementos específicos a seguir. A pesar de la versatilidad de Trello como herramienta de gestión de proyectos, presenta limitaciones específicas que pueden hacer que no sea la opción más adecuada para la gestión detallada de TFGs en un entorno académico universitario.

4.3. Tabla comparativa

A continuación, se muestra una tabla comparativa de las principales funcionalidades que ofrecen las alternativas comentadas anteriormente, junto con las que ofrece UCOTrack (figura 4.1).

Tabla 4.1: Comparación de funcionalidades entre UCOTrack, Trello y Moodle

Funcionalidad / Característica	UCOTrack	Trello	Moodle
Acceder al sistema mediante usuario y contraseña.	Sí	Sí	Sí
Creación y asignación de TFGs a cursos	Sí	Sí	Sí
Añadir información detallada a cada TFG, incluyendo el título, el alumno y sus datos de contacto.	Sí	Sí	Sí
Enlazar el TFG con la petición de tema una vez aprobada	Sí	No	Sí
Permitir asignación y cambio de estado del TFG	Sí	Sí	No
Creación de seguimientos detallados	Sí	Sí	Sí
Generación de resumen de seguimientos	Sí	No	No
Listado de TFGs por tipo	Sí	No	No
Interfaz intuitiva y amigable	Sí	Sí	Sí



Merece la pena comentar los resultados obtenidos tras el análisis y comparativa ofrecida entre los antecedentes y este TFG. Como podemos ver, UCOTrack destaca al ofrecer una solución integral que cumple con todos los objetivos específicos para la gestión de TFGs, proporcionando una interfaz centrada en estas necesidades y ofreciendo funcionalidades adicionales que la hacen la elección más adecuada.

Capítulo 5

Planificación temporal

El trabajo de Fin de Grado del grado de Ingeniería Informática corresponde a 12 créditos ECTS, y al corresponder una asignación de 25 horas por crédito en el plan de estudio, la suma total debe ser igual a 300 horas.

A continuación, mostraré dos tablas. La primera 5.1 mostrará una distribución aproximada y temporal de las fases del proyecto, y la segunda 5.2 mostrará un cronograma estimando cuantas semanas serán ocupadas para la realización de cada tarea.



Tabla 5.1: Distribución aproximada y temporal de las fases del proyecto

Tareas	Tiempo (h)
Aprendizaje y familiarización de las tecnologías a usar en el proyecto	25
- Nest.js	10
- GitHub	5
- Next.js	10
Especificación de requisitos	10
- Extracción de requisitos	5
- Modelado de requisitos	5
Análisis y diseño	55
- Análisis de requisitos	5
- Diseño	50
- Diseño de la base de datos	5
- Diseño de la aplicación web	20
- Diseño de la interfaz de usuarios	15
- Diseño del logo	10
Desarrollo e implementación	130
- Creación de la base de datos	5
- Configuración del servidor	30
- Codificación del problema	95

Tabla 5.2: Cronograma estimando cuantas semanas serán ocupadas para la realización de cada tarea.

	Marzo			Abril				Mayo			Junio			
Semana	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Aprendizaje y familiarización de las tecnologías a usar en el proyecto														
Especificación de requisitos														
Análisis y diseño														
Desarrollo e implementación														
Pruebas														
Documentación														

Capítulo 6

Restricciones y estudios de alternativas

A continuación, vamos a tratar la identificación de los factores limitativos. Todo proyecto viene condicionado por ciertas restricciones. En este apartado se abordarán todas aquellas que son relevantes para el diseño del sistema. Como factores condicionantes que son, habrá decisiones que se tomen basándose en estas restricciones. Dichos factores son de dos tipos:

- Factores dato: son aquellos inherentes a la naturaleza del problema. Suelen venir impuestos por el cliente y no pueden ser modificados.
- Factores estratégicos: Son variables de diseño seleccionadas de entre dos o más posibilidades. Es necesario justificar dichas restricciones, ya que no son impuestas, sino elegidas. Estas son las restricciones impuestas para la realización del proyecto, escogidas por acuerdo entre cliente y programador.

6.1. Factores dato

Los factores dato de este proyecto son:

- Restricciones de personal: Al ser un Trabajo Fin de Grado, este proyecto solo contará con un autor, junto con la ayuda de los dos tutores correspondientes.
- Restricciones temporales: Es necesario cumplir con los plazos establecidos para un Trabajo Fin de Grado, el cual corresponde a 300 horas de trabajo.
- Restricciones económicas: Los costes de desarrollo deben ser, en todo lo posible, inexistente. Esto es debido a que el proyecto es un Trabajo Fin de Grado, por

lo cual no posee ningún tipo de ayudas ni presupuesto. Se utilizarán en mayor o total medida herramientas gratuitas y de software libre.

6.2. Factores estratégicos

Pasemos ahora a hablar de los factores estratégicos de este proyecto:

6.2.1. Tipo de aplicación

Se ha decidido realizar una aplicación web, ya que estas son accesibles desde cualquier dispositivo que tenga conexión a internet y un navegador web, además de que las aplicaciones web no necesitan ser descargadas e instaladas en el dispositivo del usuario.

6.2.2. Frontend

A la hora de seleccionar la tecnología con la que desarrollar el lado frontend del proyecto, la autora ha optado por usar Next.js [3], un framework construido sobre React.js [4] de código abierto desarrollado por Vercel [5]. Este framework tiene las funcionalidades y los recursos necesarios para cubrir todos los objetivos de este Trabajo Fin de Grado.

6.2.3. Backend

La autora ha tomado la decisión de utilizar NestJS [6] como framework para el Backend de este proyecto, ya que es uno de los más utilizados para el lenguaje de programación JavaScript. Su documentación es muy completa, además de contar con una gran comunidad que lo respalda. Cuenta con un gran número de funcionalidades que permiten implementar diferentes metodologías como son librerías GraphQL, inyección de dependencias para facilitar la arquitectura dividida en capas, etc.

6.2.4. Otras tecnologías

- Latex [7] para la documentación: Se ha elegido utilizar este editor de textos para la realización de la documentación del proyecto. Este editor ofrece un gran número de herramientas que facilitan la creación de documentos de apariencia técnica y profesional.



- Visual Studio Code [8] como editor de código: Este editor posee una gran cantidad de extensiones que facilitan enormemente el desarrollo, además de ser muy intuitivo y contar con una gran comunidad a sus espaldas.

Capítulo 7

Recursos

7.1. Recursos humanos

Los recursos humanos presentes en este proyecto son:

- Directores: Son los encargados de realizar un seguimiento, supervisión y orientación del proyecto en todas sus fases.
 - Profesora Dra. María Luque Rodríguez.
 - Profesor Dr. Antonio Araúzo Azofra.
- Diseño y desarrollo: Encargado de diseñar e implementar todo el proyecto.
 - Estudiante: Irene López Jiménez

7.2. Recursos hardware

Sobre los recursos hardware, se dispondrá de un ordenador con las siguientes especificaciones:

- CPU: Intel® Core™ i7-7500U CPU @ 2.70GHz 2.90 GHz
- RAM: 12GB
- Almacenamiento: 446GB (SSD)

7.3. Recursos software

El proyecto contará con los siguientes recursos software:

- Sistema operativo: Microsoft Windows 10 Home [?]
- Lenguajes de programación:
 - Nest.js [6]
 - Next.js [3]
- Administración de la base de datos
 - MySQL [9]. Para levantarlo usaré Docker. [10]
- Elaboración de la documentación
 - LaTeX [7].
- Entorno de desarrollo:
 - Visual Studio Code [8]
- Realización de diagramas
 - Draw.io [11]
- Control de versiones
 - Git

Capítulo 8

Especificación de requisitos

8.1. Introducción

Como se ha mencionado en anteriormente, el principal objetivo de este proyecto es el desarrollo de una aplicación de gestión de Trabajos Fin de Grado. A continuación, pondremos los requisitos funcionales, no funcionales y de información.

8.1.1. Requisitos funcionales

Conjunto de funcionalidades o servicios que el sistema debe ofrecer. Sirven para expresar qué debe hacer el sistema, es decir, como debe reaccionar ante determinadas entradas.

RF1 El sistema deberá permitir el acceso mediante usuario y contraseña.

RF2 El sistema deberá posibilitar la creación de Trabajos de Fin de Grado (TFGs) y su asignación a cursos académicos.

RF3 El sistema deberá permitir añadir información detallada a cada TFG, incluyendo título, alumno y sus datos de contacto.

RF4 El sistema deberá ser capaz de enlazar cada TFG con la petición de tema una vez que esta sea aprobada.

RF5 El sistema deberá facilitar la asignación y el cambio de estados de los TFGs (pendiente de petición, petición aprobada, desarrollo, entregado, presentado, etc.).

RF6 El sistema deberá permitir la creación de seguimientos para cada TFG, registrando fechas e información relevante (acciones realizadas, conversaciones...).



RF7 El sistema deberá generar un documento con todos los seguimientos de un TFG a modo de resumen.

RF8 El sistema deberá listar los TFGs por tipo/estado para una mejor organización.

8.1.2. Requisitos no funcionales

Permiten indicar restricciones al sistema que pueden afectar a la calidad del servicio (fiabilidad, tiempo de respuesta, etc.). Se centran en expresar como debe comportarse el sistema. También pueden hacer referencia.

RNF1 El sistema deberá garantizar la seguridad de los datos y la protección de la información confidencial de los usuarios. Se comprobará que las contraseñas estén cifradas en la base de datos.

RNF2 El sistema deberá mantener un alto rendimiento incluso bajo cargas de trabajo elevadas, asegurando tiempos de respuesta rápidos. El sistema debe responder a una solicitud de página dentro de 5 segundos.

RNF3 El sistema deberá diseñar una interfaz intuitiva y amigable para que los profesores puedan gestionar los TFGs.

RNF4 El sistema deberá ser compatible con diferentes navegadores web populares, como Google, Firefox... Para ello, una vez que este la aplicación, se abrirá en los diferentes navegadores.

8.1.3. Requisitos de información

RI 1 La aplicación debe almacenar información detallada sobre cada trabajo de fin de grado, incluyendo título, alumno, estado, seguimientos, etc.

RI 2 Se deben almacenar los datos de inicio de sesión y la información de contacto de los usuarios, como nombre, dirección de correo electrónico, etc.

RI 3 La aplicación debe tener información sobre los cursos académicos a los que se asignan los trabajos de fin de grado.

8.2. Casos de uso

Los casos de uso consisten en descripciones que pretenden definir las interacciones entre un sistema y sus actores principales.

8.2.1. Diagrama de casos de uso

Tal y como podemos ver en la figura 8.1 esos serían todos los casos de uso del sistema.

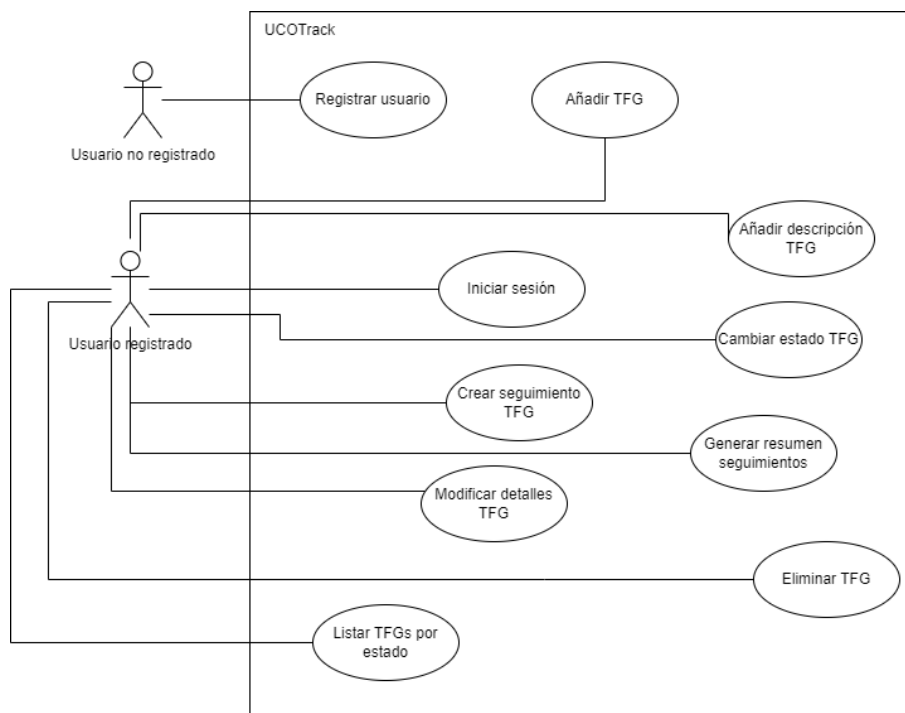


Figura 8.1: Diagrama de casos de uso

8.2.2. Actores del sistema

En el sistema presentado en este proyecto, se han identificado los siguientes actores:

- Usuario no registrado: Usuario no registrado en el sistema.
- Usuario registrado: Usuario registrado en el sistema. Este será el encargado del seguimiento y corrección del Trabajo Fin de Grado.

8.2.3. CU01 - Registrar usuario

Nombre	RegistrarUsuario
Identificador	CU-1
Descripción	El objetivo del caso de uso es que un usuario reciba una cuenta en el sistema.
Actor principal	Usuario no registrado
Actor secundario	Ninguno
Precondiciones	Ninguna
Flujo principal	<ol style="list-style-type: none">1. El caso de uso comienza cuando el sistema solicita los datos del registro al usuario.2. El usuario no registrado introduce los datos.3. El sistema comprueba que no existe un usuario registrado con esos credenciales.4. El sistema crea la cuenta.
Flujo alternativo	<ol style="list-style-type: none">1. Credenciales ya existentes - El flujo alternativo comienza después del paso 3, al comprobar que las credenciales ya están en uso, se da un mensaje de error y se termina el caso de uso.
Postcondición	Se ha creado una cuenta al usuario

8.2.4. CU02 - Iniciar sesión

Nombre	IniciarSesión
Identificador	CU-2
Descripción	El objetivo del caso de uso es que un usuario inicie sesión
Actor principal	Usuario registrado
Actor secundario	Ninguno
Precondiciones	No hay
Flujo principal	<ol style="list-style-type: none">1. El caso de uso comienza cuando el usuario accede al inicio de sesión.2. El usuario ingresa su nombre de usuario y contraseña.3. El sistema verifica las credenciales del usuario.4. Si las credenciales son válidas, el sistema inicia sesión.
Flujo alternativo	<ol style="list-style-type: none">1. Credenciales erróneas - El flujo alternativo comienza después del paso 3, al comprobar que las credenciales son correctas, se da un mensaje de error y se termina el caso de uso.
Postcondición	El usuario ha iniciado sesión en el sistema.

8.2.5. CU03 - Añadir TFG

Nombre	AñadirTFG
Identificador	CU-3
Descripción	Un usuario registrado crea un nuevo Trabajo de Fin de Grado (TFG).
Actor principal	Usuario registrado
Actor secundario	Ninguno
Precondiciones	El usuario registrado ha iniciado sesión en el sistema.
Flujo principal	<ol style="list-style-type: none">1. El caso de uso comienza cuando el usuario registrado accede a la función de creación de TFG.2. El sistema permite ingresar los detalles del nuevo TFG, como título, autor, curso académico.3. El usuario registrado rellena todo con la información requerida.4. El sistema guarda el nuevo TFG en la base de datos.
Postcondición	Se ha creado un nuevo TFG.

8.2.6. CU04 - Añadir descripción TFG

Nombre	AñadirDescripciónTFG
Identificador	CU-4
Descripción	Un usuario registrado añade información detallada a un Trabajo de Fin de Grado (TFG) existente, incluyendo título, alumno y datos de contacto.
Actor principal	Usuario registrado
Actor secundario	Ninguno
Precondiciones	El usuario registrado ha iniciado sesión en el sistema.
Flujo principal	<ol style="list-style-type: none">1. El caso de uso comienza cuando el usuario registrado selecciona un TFG existente al que desea añadir detalles.2. El sistema muestra para editar los detalles del TFG, incluyendo título, alumno y datos de contacto.3. El usuario registrado modifica los detalles según sea necesario.4. El sistema guarda los cambios realizados en la base de datos.
Postcondición	Los detalles del TFG han sido actualizados en la base de datos.

8.2.7. CU05 - Cambiar estado TFG

Nombre	CambiarEstadoTFG
Identificador	CU-5
Descripción	Un usuario registrado cambia el estado de un Trabajo de Fin de Grado (TFG) en el sistema, por ejemplo, de 'pendiente de petición' a 'petición aprobada'.
Actor principal	Usuario registrado
Actor secundario	Ninguno
Precondiciones	El usuario registrado ha iniciado sesión en el sistema.
Flujo principal	<ol style="list-style-type: none">1. El caso de uso comienza cuando el usuario registrado selecciona un TFG existente.2. El sistema muestra las opciones disponibles para cambiar el estado del TFG.3. El usuario registrado selecciona el nuevo estado deseado para el TFG.4. El sistema guarda el cambio de estado en la base de datos.
Postcondición	El estado del TFG ha sido actualizado en la base de datos.

8.2.8. CU06 - Crear seguimiento TFG

Nombre	CrearSeguimientoTFG
Identificador	CU-6
Descripción	Un usuario registrado crea un seguimiento para un Trabajo de Fin de Grado (TFG) específico, registrando fechas e información relevante sobre el progreso del trabajo.
Actor principal	Usuario registrado
Actor secundario	Ninguno
Precondiciones	El usuario registrado ha iniciado sesión en el sistema.
Flujo principal	<ol style="list-style-type: none">1. El caso de uso comienza cuando el usuario registrado selecciona añadir seguimiento a un TFG.2. El sistema proporciona una opción para crear un nuevo seguimiento para el TFG.3. El sistema solicita la información de seguimiento.4. El usuario registrado rellena la información relevante sobre el seguimiento, como fechas y detalles.5. El sistema guarda el seguimiento en la base de datos y lo asocia al TFG correspondiente.
Postcondición	El seguimiento ha sido creado y asociado al TFG en la base de datos.

8.2.9. CU07 - Generar resumen seguimientos

Nombre	GenerarResumenSeguimientos
Identificador	CU-7
Descripción	Un usuario registrado solicita al sistema que genere un documento resumen con todos los seguimientos registrados para un Trabajo de Fin de Grado (TFG) específico.
Actor principal	Usuario registrado
Actor secundario	Ninguno
Precondiciones	El usuario registrado ha iniciado sesión en el sistema.
Flujo principal	<ol style="list-style-type: none">1. El caso de uso comienza cuando el usuario registrado selecciona un TFG existente.2. El sistema proporciona una opción para generar un resumen de seguimientos para el TFG seleccionado.3. El usuario registrado confirma la solicitud.4. El sistema recopila la información de seguimiento asociada al TFG y genera un documento resumen.5. El sistema muestra el documento resumen al profesor para su visualización o descarga.
Postcondición	Se ha generado un documento resumen de seguimientos para el TFG seleccionado.

8.2.10. CU08 - Listar TFGs por estado

Nombre	ListarTFGsEstado
Identificador	CU-8
Descripción	Un usuario registrado lista los Trabajos de Fin de Grado (TFGs) disponibles en el sistema según su estado para una mejor organización.
Actor principal	Usuario registrado
Actor secundario	Ninguno
Precondiciones	El usuario registrado ha iniciado sesión en el sistema.
Flujo principal	<ol style="list-style-type: none"> 1. El caso de uso comienza cuando el usuario registrado accede a la función de listado de TFGs. 2. El sistema proporciona opciones para filtrar los TFGs por estado. 3. El usuario registrado selecciona el estado deseado para los TFGs. 4. El sistema muestra una lista de TFGs que coinciden con el estado seleccionado.
Postcondición	Se muestra una lista de TFGs según el estado seleccionado.

8.2.11. CU9 - Modificar detalles TFG

Nombre	ModificarDetallesTFG
Identificador	CU-9
Descripción	Un usuario registrado modifica los detalles de un Trabajo de Fin de Grado (TFG) existente en el sistema, como el título, el alumno asignado, etc.
Actor principal	Usuario registrado
Actor secundario	Ninguno
Precondiciones	El usuario registrado ha iniciado sesión en el sistema.
Flujo principal	<ol style="list-style-type: none">1. El caso de uso comienza cuando el usuario selecciona modificar TFG.2. El usuario registrado modifica los detalles del TFG según sea necesario.3. El sistema guarda los cambios realizados en la base de datos.
Postcondición	Los detalles del TFG han sido actualizados en la base de datos.

8.2.12. CU10 - Eliminar TFG

Nombre	Eliminar TFG
Identificador	CU-10
Descripción	Un usuario registrado elimina un Trabajo de Fin de Grado (TFG) del sistema.
Actor principal	Usuario registrado
Actor secundario	Ninguno
Precondiciones	El usuario registrado ha iniciado sesión en el sistema.
Flujo principal	<ol style="list-style-type: none">1. El caso de uso comienza cuando el usuario selecciona un TFG.2. El usuario registrado confirma la eliminación del TFG.3. El sistema elimina el TFG de la base de datos. <p>Flujo alternativo: Cancelar. El administrador decide cancelar la eliminación del TFG en cualquier momento durante el flujo principal. El sistema conserva el TFG en la base de datos.</p>
Postcondición	El TFG ha sido eliminado del sistema y ya no está disponible en la base de datos.

8.3. Matriz de trazabilidad

La matriz de trazabilidad (tabla 8.1) permite vincular los requisitos funcionales con los casos de uso.

Tabla 8.1: Matriz de trazabilidad

	CU-01	CU-02	CU-03	CU-04	CU-05	CU-06	CU-07	CU-08	CU-09	CU-10
RF01	X	X								
RF02			X							
RF03				X						
RF04					X					
RF05					X					
RF06						X				
RF07							X			
RF08								X		
RF09									X	
RF10										X

Capítulo 9

Análisis de la información

9.1. Descripción del modelo conceptual de datos

A continuación, se mostrarán las clases del sistema junto con sus relaciones y sus cardinalidades correspondientes (ver figura 9.1).

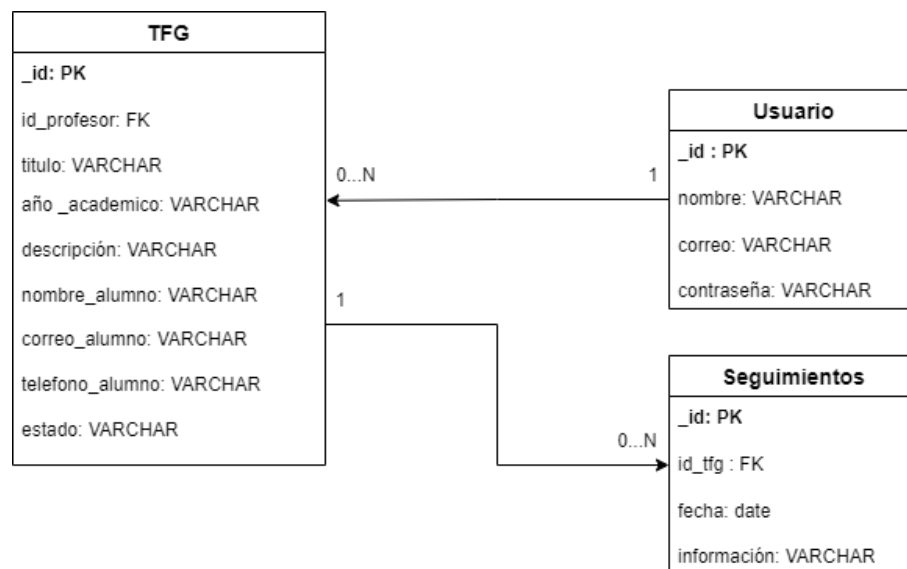


Figura 9.1: Diagrama de clases

9.2. Descripción de las clases

9.2.1. Clase TFG

Esta clase representa la información de un Trabajo de Fin de Grado (TFG) en el sistema. Se conforma por los siguientes atributos:

- Identificador (string): Identificador único del TFG en el sistema.
- Identificador profesor (string): Identificador del profesor asignado al TFG.
- Título (string): Título del TFG.
- Año académico (string): Año académico en el que se lleva a cabo el TFG.
- Descripción (string): Descripción del TFG.
- Nombre alumno (string): Nombre del alumno que está realizando el TFG.
- Correo alumno (string): Correo electrónico del alumno.
- Teléfono alumno (string): Número de teléfono del alumno.
- Estado (string): Estado del TFG.

9.2.2. Clase Usuario

Esta clase representa la información de un usuario registrado en el sistema. Se conforma por los siguientes atributos:

- Identificador (string): Identificador único del usuario registrado en el sistema.
- Nombre (string): Nombre del usuario con el que se registró en el sistema.
- Correo (string): Correo electrónico con el que el usuario inicia sesión en el sistema.
- Contraseña (string): Contraseña con la que el usuario inicia sesión en el sistema.



9.2.3. Clase Seguimientos

Esta clase representa la información de un seguimiento asociado a un Trabajo de Fin de Grado (TFG) en el sistema.

- Identificador (string): Identificador único del seguimiento en el sistema.
- Identificador del TFG (string): Identificador del TFG asociado.
- Fecha (date): Fecha en la que se registró el seguimiento.
- Información (string): Información del seguimiento. Incluye notas, observaciones o detalles relevantes sobre el progreso del TFG.

9.3. Diagrama de actividad de los casos de uso

9.3.1. Iniciar sesión

El diagrama de actividad de Iniciar sesión (figura 9.2) comienza accediendo a iniciar sesión, se ingresa el nombre de usuario y la contraseña y comprueba si las credenciales introducidas son válidas o no, si no fuesen correctas, volvería a solicitar nombre de usuario y contraseña, y si fuese correcto, inicia en el sistema.

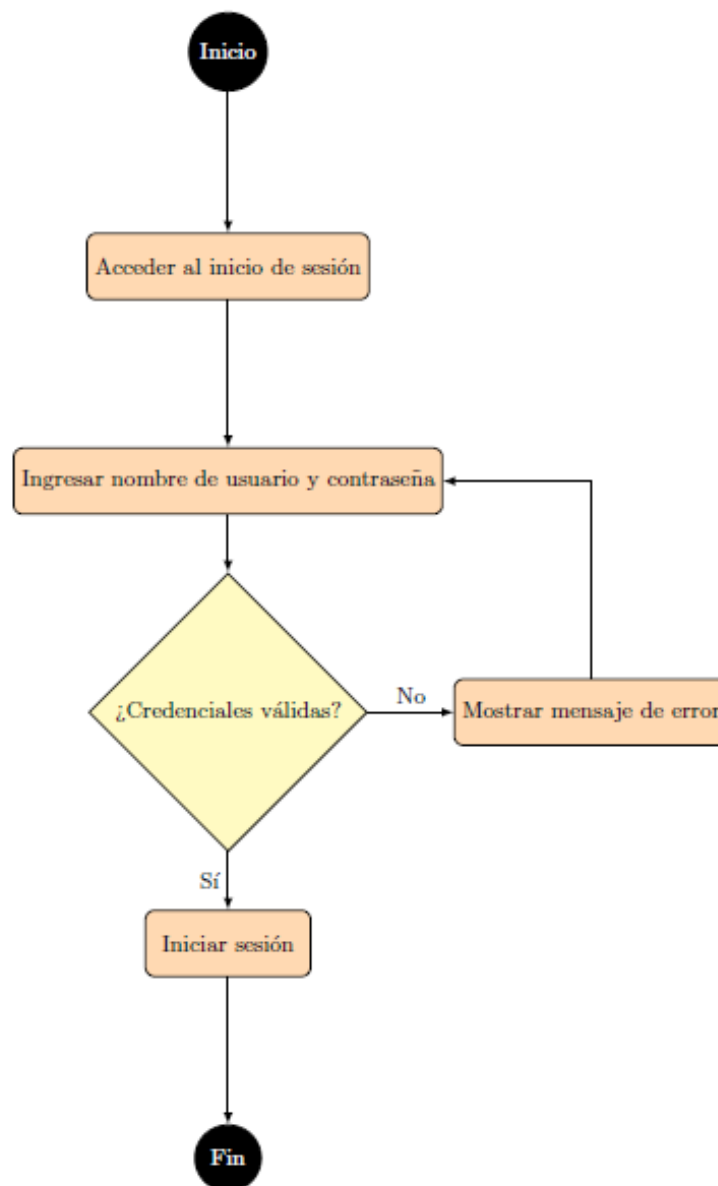


Figura 9.2: Diagrama de actividad: Iniciar sesión

9.3.2. Añadir TFG

El diagrama de actividad de Añadir TFG (figura 9.3) comienza accediendo a la creación de un TFG, se ingresan los detalles, se rellena la información solicitada, se crea el TFG y se guarda en la base de datos.

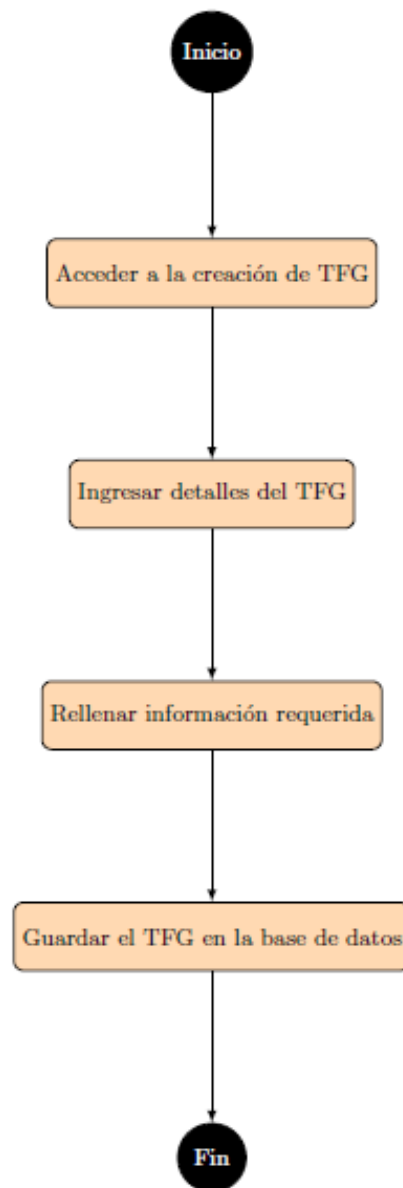


Figura 9.3: Diagrama de actividad: Añadir TFG

9.3.3. Eliminar TFG

El diagrama de actividad de Eliminar TFG (figura 9.4) comienza seleccionando el TFG, se pregunta si se desea eliminar, en caso afirmativo se elimina de la base de datos y en caso contrario, se vuelve al inicio.

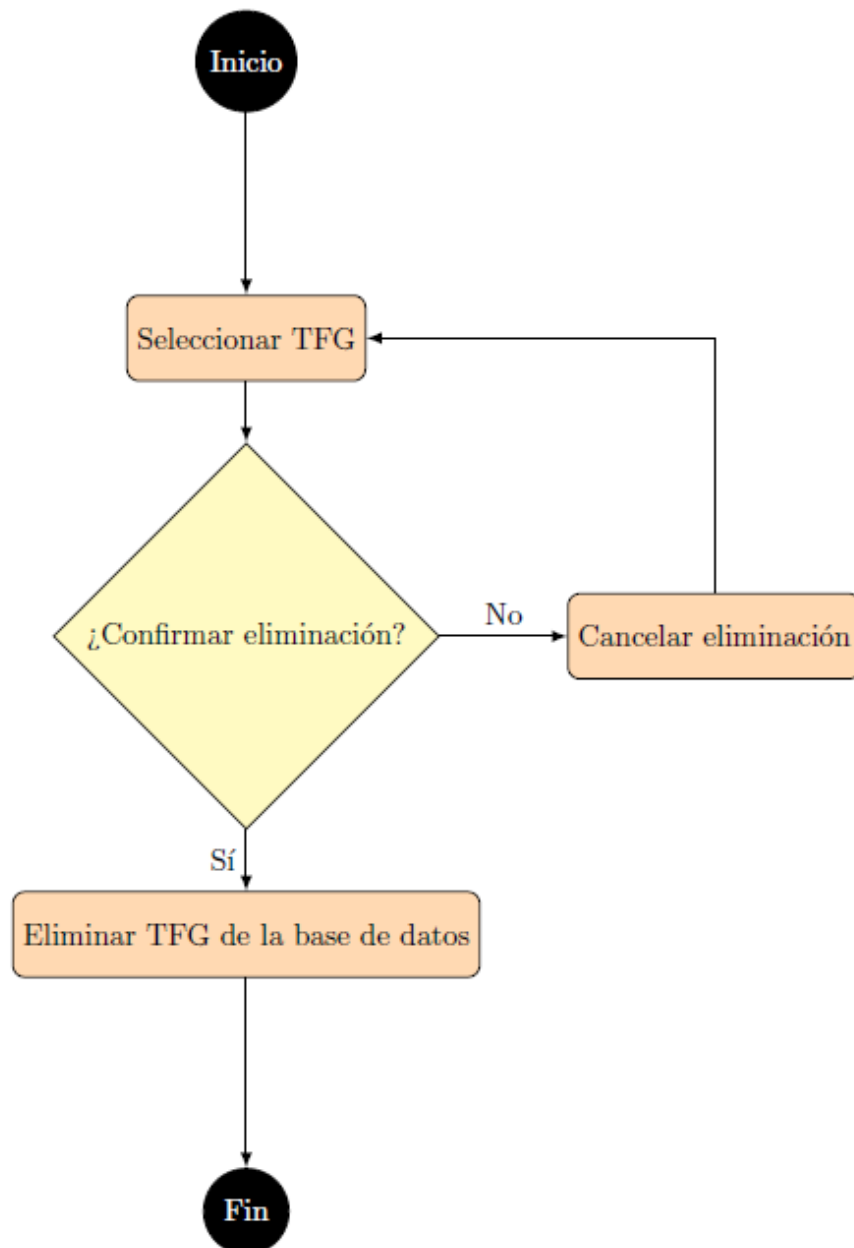


Figura 9.4: Diagrama de actividad: Eliminar TFG

9.3.4. Modificar TFG

El diagrama de actividad de Modificar TFG (figura 9.5) comienza seleccionando la opción de modificar TFG, se hacen los cambios oportunos y se guardan los cambios en la base de datos.

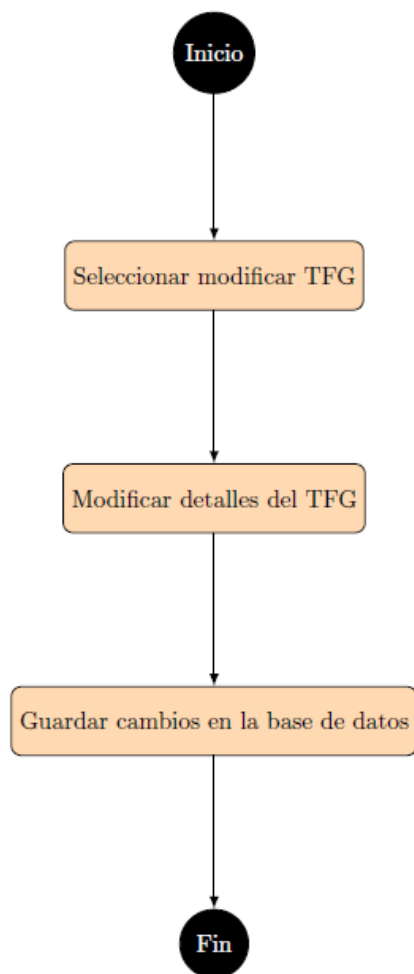


Figura 9.5: Diagrama de actividad: Modificar TFG

9.3.5. Crear seguimiento TFG

El diagrama de actividad de Crear seguimiento TFG (figura 9.6) comienza seleccionando añadir seguimiento, se solicita la información, se rellena y se guarda en la base de datos.

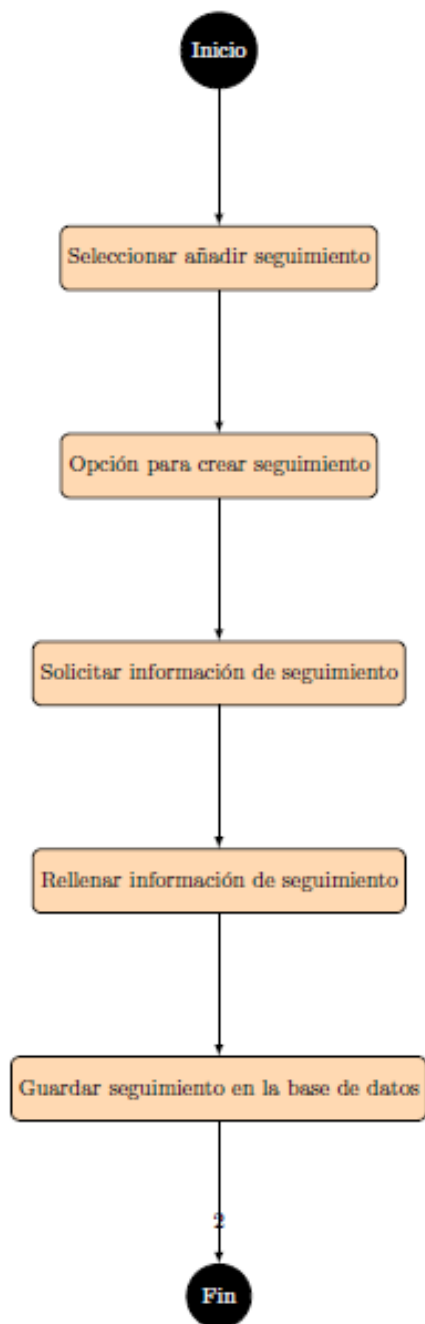


Figura 9.6: Diagrama de actividad: Crear seguimiento TFG

Capítulo 10

Diseño

Una vez terminada la etapa de análisis, nos centramos en la etapa de diseño. Esta etapa se enfocará en realizar una definición detallada de la arquitectura y el funcionamiento del sistema de software que se va a desarrollar. Este nuevo modelo está relacionado con la etapa anterior; el modelo de diseño estará construido basándonos en el modelo de análisis, añadiendo detalles de implementación del sistema.

10.1. Diagramas de interacción

Los diagramas utilizados para describir en detalle y modelar los determinados escenarios de los casos de uso son los llamados diagramas de interacción. Dentro de los mismos nos encontramos tanto los diagramas de secuencia como los diagramas de colaboración o comunicación. Ambos tipos de diagramas tratan la misma información, pero cada uno hace énfasis en un aspecto particular en cuanto a la forma de mostrarla.

10.1.1. Diagramas de secuencia

El propósito de los diagramas de secuencia es describir el funcionamiento dinámico del sistema y destacar la sucesión de los mensajes que intercambian los objetos. En resumen, los diagramas de secuencia se presentan como un grupo de objetos que interactúan a lo largo del tiempo. En nuestro caso, realizaremos los diagramas de secuencia que sean más representativos del comportamiento de nuestra aplicación.

10.1.1.1. Diagrama de secuencia: Añadir TFG

El diagrama de secuencia de Añadir TFG (figura 10.1) consiste en que el usuario registrado introduce todos los datos necesarios para crear un TFG (título, año

académico, nombre del alumno...). Comprueba si los campos son válidos; si es correcto, se crea el TFG, en caso contrario, sale mensaje de error.

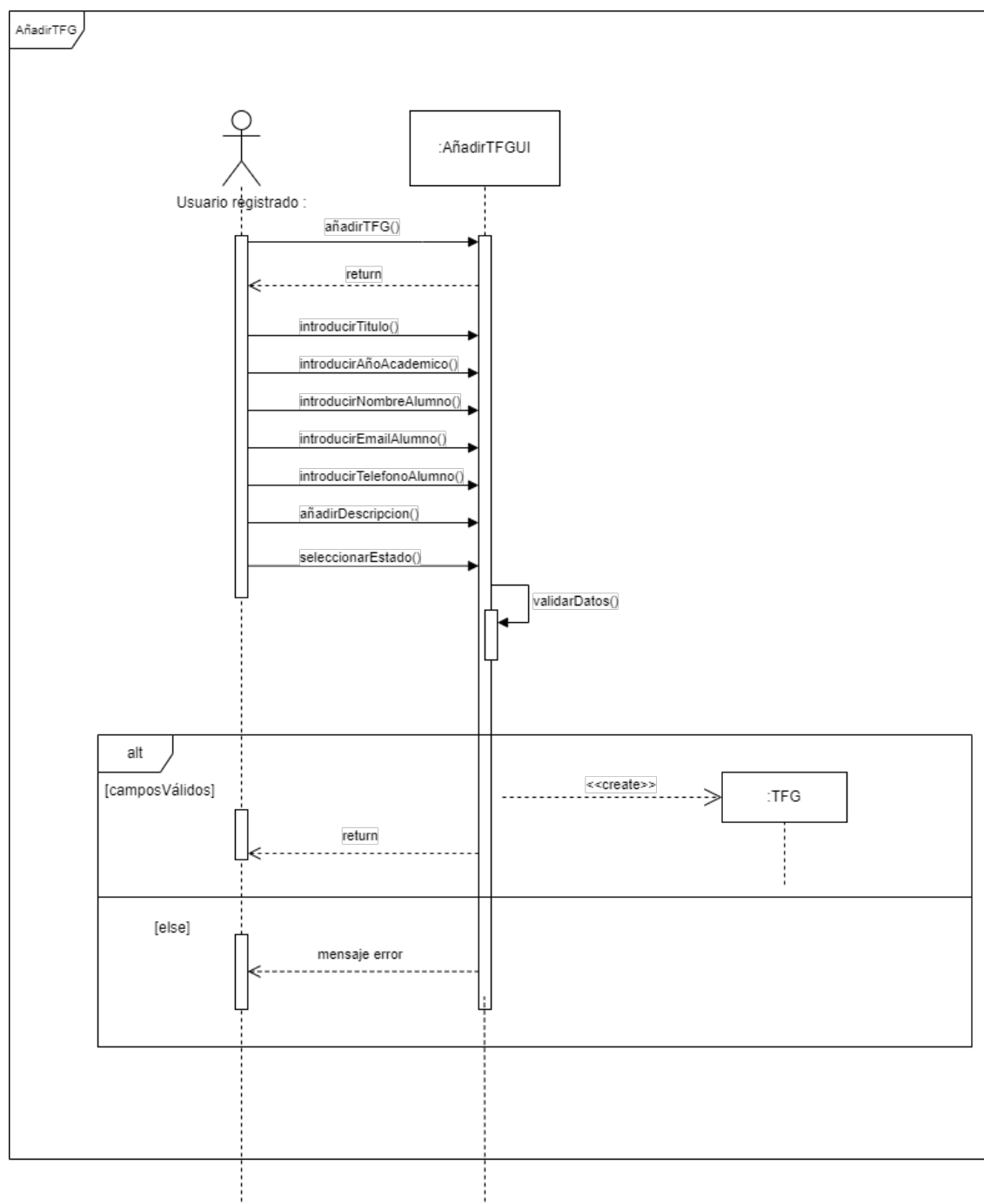


Figura 10.1: Diagrama de secuencia: Añadir TFG

10.1.1.2. Diagrama de secuencia: Listar TFG

El diagrama de secuencia de Listar TFG (figura 10.2) consiste en que el usuario solicita listar los TFGs, se obtienen todos los datos de la base de datos, y se devuelve todo el TFG con sus datos.

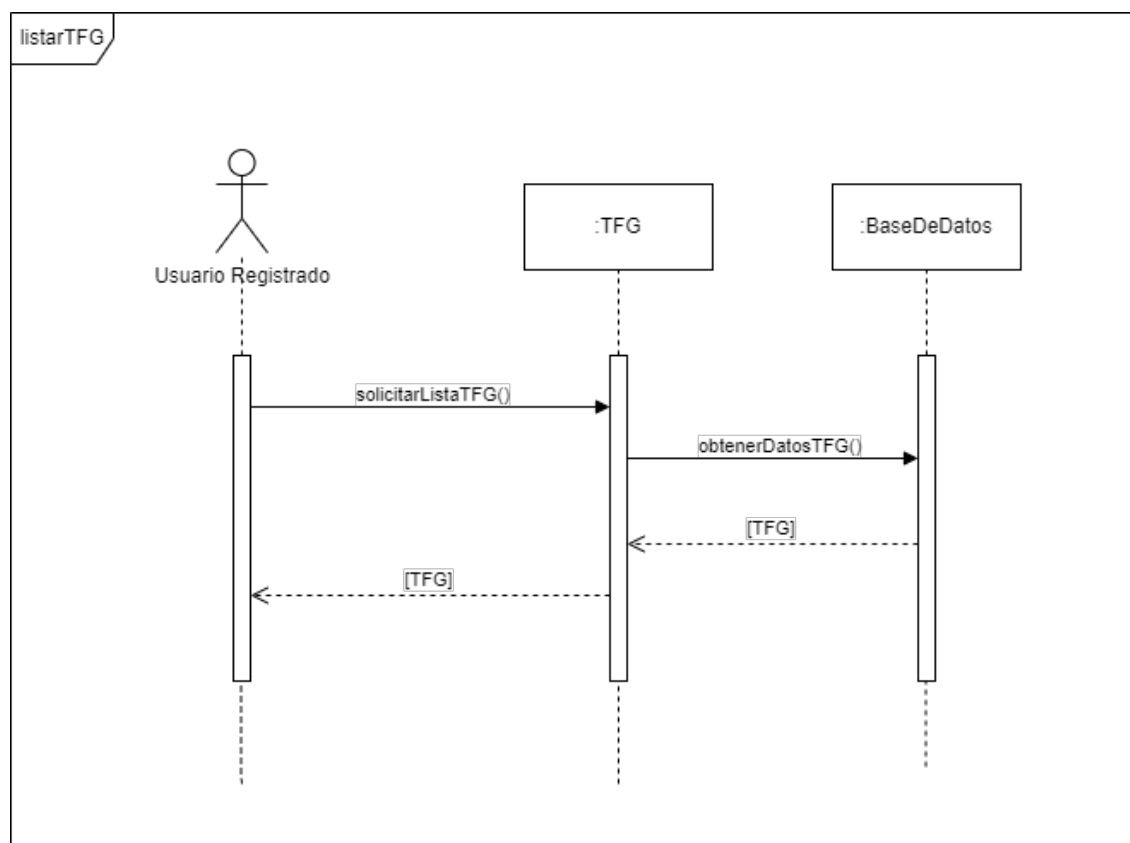


Figura 10.2: Diagrama de secuencia: Listar TFG

10.2. Diseño de la interfaz

El diseño de la interfaz de usuario es un aspecto vital que debemos tener en cuenta para el desarrollo de nuestra aplicación. La interfaz no solo se centrará en la apariencia visual, sino que con este diseño se pretenderá mejorar la experiencia de usuario. Es decir, el objetivo será aumentar la usabilidad, facilitando la interacción entre el usuario y el software, productividad, eficiencia y reducción de errores.

10.2.1. Maquetas

Para este apartado se ha decidido la realización de mockups o maquetas. Nuestro objetivo con estas maquetas será mostrar un modelo a medio nivel para proporcionar una representación visual clara y detallada de cómo se verá y funcionará la interfaz. Por lo tanto, en estos mockups representaremos las situaciones más importantes de nuestra aplicación como el inicio de sesión, creación de TFGs, listado de TFGs...

10.2.1.1. Maqueta de la vista de inicio de sesión

En el centro de la figura 10.3, se presenta un recuadro que contiene un formulario que solicita un correo electrónico y una contraseña. Debajo de estos campos de entrada, se encuentra un botón con la etiqueta 'Iniciar Sesión'. Justo debajo del formulario de inicio de sesión, aparece un botón adicional etiquetado como 'Registrarse'

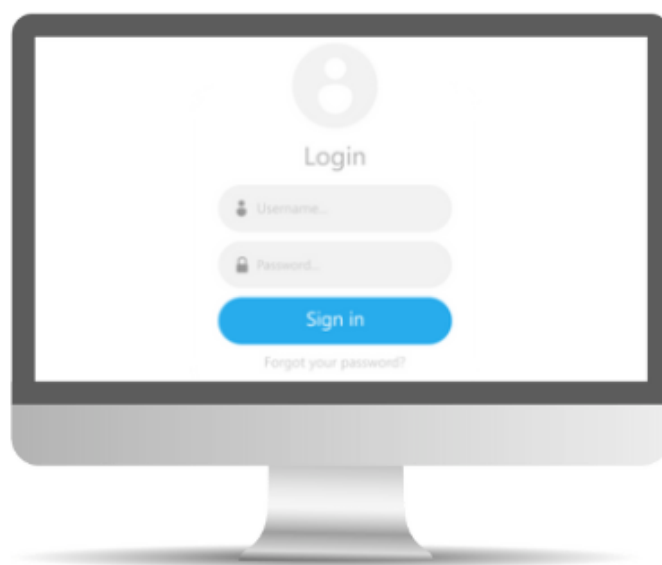


Figura 10.3: MockUp Inicio de sesión

10.2.1.2. Maqueta de la vista de la página principal

Tal y como se ve en la figura 10.4 en el centro de la pantalla aparecen las opciones 'Crear TFG' y 'Listar TFG'. Y en la parte superior de la pantalla, aparece el logo y para cerrar sesión.



Figura 10.4: MockUp Página principal.

10.2.1.3. Maqueta de la vista de la creación de TFG

En el centro de la página se ve un formulario con los campos que aparecen en la figura 10.5 : Título TFG, Año académico, Descripción, Nombre alumno, Correo electrónico alumno, Teléfono alumno, Estado. En la parte superior de la pantalla, aparece el logo y para cerrar sesión.



Figura 10.5: MockUp Creación TFG

10.2.1.4. Maqueta de la vista de listado de TFG

Aquí aparece una lista con todos los TFGs creado, tal y como aparece en la figura 10.6. En la parte superior de la pantalla, aparece el logo y para cerrar sesión.



Figura 10.6: MockUp Listado TFG.

10.3. Arquitectura

En esta sección se diferenciará las distintas arquitecturas empleadas para el backend, el frontend y la base de datos de este proyecto.

10.3.1. Backend

El lenguaje utilizado para desarrollar el backend ha sido Nest.js. NestJS está escrito en TypeScript, lo que permite tener tipado estático y encontrar errores en tiempo de desarrollo. Esto mejora la productividad y reduce los errores de tipo en desarrollo. Además, está diseñado para ser altamente escalable, adecuado tanto para pequeñas aplicaciones como para grandes sistemas empresariales.

Domain Driven Design (DDD)

DDD o Domain Driven Design es un enfoque de diseño de software que sitúa el dominio del sistema en el centro del diseño. Aunque mi backend no sigue completamente los principios del DDD, sí que incorpora elementos clave para mejorar la modularidad y la claridad del diseño, como pueden ser los contextos. (figura 10.7)

- **Dominio:** DDD pone el centro en el dominio del sistema. El dominio es el ámbito para el cual se está desarrollando el software. En mi proyecto, La carpeta 'domain' está dentro de los módulos 'tfgs' y 'usuarios', que contienen los modelos y los repositorios. Esto demuestra que el dominio es una parte central del diseño.
- **Lenguaje Ubicuo:** En UCOTrack, el lenguaje ubicuo es fundamental para asegurar una comunicación clara y coherente entre los desarrolladores y los expertos en el dominio. Este garantiza que todos los involucrados en el proyecto compartan una comprensión común de los términos y conceptos clave, evitando malentendidos, por ejemplo, se han usado nombres muy característicos: 'TFG', 'Notas', 'Seguimientos'...
- **Repositorios:** Los repositorios son abstracciones o interfaces de los métodos que acceden y manipulan las entidades y agregados. Proporcionan métodos para buscar, almacenar y obtener objetos del modelo del dominio. En UCOTrack, hay varios repositorios, por ejemplo: el Repositorio de TFG se encarga de acceder y gestionar los datos relacionados con los Trabajos de Fin de Grado. Ofrece métodos como `findById`, `delete`... El Repositorio de Usuarios se ocupa de la gestión de los datos de los usuarios en el sistema. Incluye métodos como `findUserById`, `registerUser`...

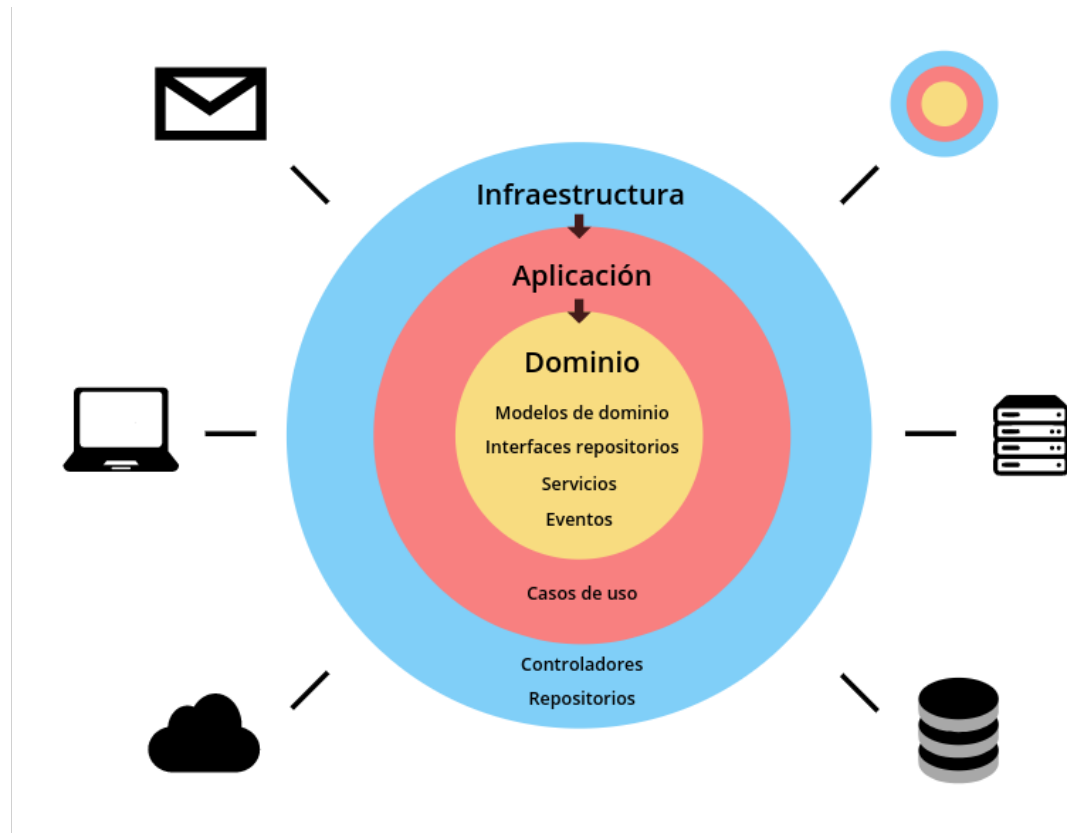


Figura 10.7: Domain Driven Design (DDD)

Arquitectura hexagonal

En UCOTrack, hemos adoptado la Arquitectura Hexagonal, también conocida como la arquitectura de puertos y adaptadores, para estructurar el diseño del sistema. Esta arquitectura, respaldada por el Diseño Guiado por el Dominio (DDD), divide la aplicación en tres capas principales, organizadas de la siguiente manera (de interna a externa, como se muestra en la figura 10.8):

- **Dominio.** Esta capa incluye todos los modelos del dominio relacionados con el seguimiento y gestión de los Trabajos de Fin de Grado (TFG). Aquí se encuentran los agregados raíz, donde hay entidades como TFG, Nota, Usuario... así como los objetos de valor relevantes. También se incluyen los eventos de dominio que representan cambios significativos en el estado del dominio (como por ejemplo, NotaAñadida o NotaEliminada) y excepciones de dominio que manejan errores específicos del contexto (por ejemplo, NotaNoEncontrada).
- **Aplicación.** Esta capa engloba los casos de uso del sistema. Aquí se manejan

las operaciones específicas del dominio, en nuestro proyecto son: la creación, modificación, eliminación de TFGs, usuarios...

- **Infraestructura.** La capa de infraestructura contiene las implementaciones de los servicios definidos en las capas inferiores, así como toda la interacción con el exterior (controladores, acceso a bases de datos, etc.). En UCOTrack, esta lógica se encuentra en la carpeta `infrastructure` de cada módulo, con archivos como `tfgController.ts` y `tfgRouter.ts`.

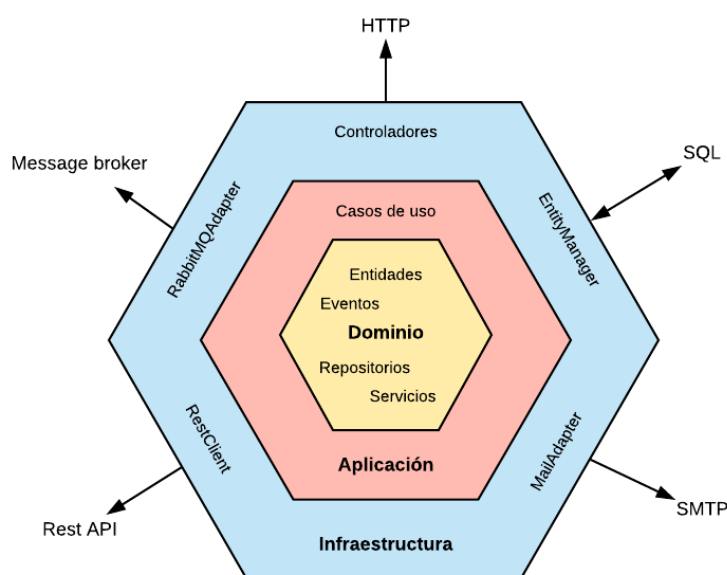


Figura 10.8: Arquitectura hexagonal

10.3.2. Frontend

El frontend ha sido desarrollado utilizando Next.js, un framework basado en React escrito en TypeScript. Esto proporciona tipado estático, lo que permite detectar errores durante el desarrollo y mejora la calidad del código. Next.js es conocido por su sistema de rutas basado en archivos, simplificando la creación y organización de rutas en la aplicación. La razón por la que he elegido Next.js es debido a que proporciona una base sólida para construir aplicaciones escalables y eficientes.

JWT (JSON Web Token)

Para autenticarse en la aplicación de este proyecto, un usuario debe iniciar de sesión. La validación de los usuarios se realizará mediante la comprobación de un JWT (JSON Web Token) [12] en las comunicaciones Front-Back. El flujo seguido en UCOTrack es el siguiente:

- El usuario introduce email y contraseña para iniciar sesión.
- El frontend envía estos datos al backend, el cual valida el inicio de sesión. En caso de que sea válido, devuelve un JWT.
- El frontend recoge y guarda ese JWT y lo enviará en todas las sucesivas peticiones que haga el backend.
- Ante una petición del frontend al backend, este recibirá el JWT que deberá validar previo a hacer la petición.

Client-Side Rendering (CSR)

El Client-Side Rendering (CSR) (Renderizado del lado del cliente) es una técnica de desarrollo web en la cual la mayor parte del procesamiento y renderizado de la interfaz de usuario ocurre en el navegador del cliente, en lugar de en el servidor. Mejora la experiencia de usuario porque los tiempos de carga son asíncronos a la visualización de la página; se visualiza sin cargar completamente, pero el usuario ya tiene la funcionalidad previa.

10.3.3. Base de datos

A la hora de almacenar los datos de los TFGs, se hará uso de una base de datos relacional. Para ello, usaremos MySQL [9]. El diagrama E-R de nuestra base de datos sería el de la figura 10.9

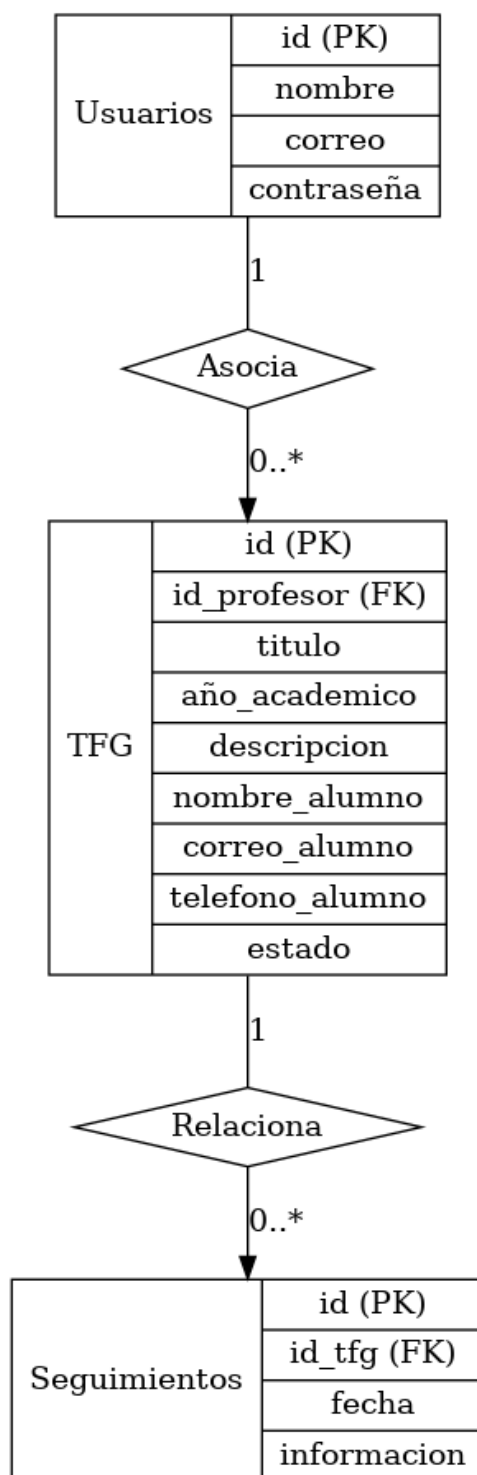


Figura 10.9: Diagrama E-R



Nuestra base de datos cuenta con tres entidades principales: Usuarios, TFG (Trabajos de Fin de Grado), y Seguimientos. Estas entidades están interrelacionadas para facilitar la gestión y el seguimiento de los TFG.

- Usuarios representa a los usuarios de UCOTrack, que en este caso son los profesores encargados de supervisar los TFG. Los atributos de esta entidad incluyen: id, nombre, correo y contraseña.
- TFG representa los Trabajos de Fin de Grado. Cada TFG está asociado a un profesor (usuario) que lo supervisa. Los atributos de esta entidad incluyen: id, id-profesor, título, año-académico, descripción, nombre-alumno, correo-alumno, teléfono-alumno y estado.
- Seguimientos representa los seguimientos que se realizan para cada TFG. Cada seguimiento está vinculado a un único TFG. Los atributos de esta entidad incluyen: id, id-tfg, fecha e información (información detallada del seguimiento).

Las relaciones entre las entidades son las siguientes: existe una relación de uno a muchos entre Usuarios y TFG, donde un usuario (profesor) puede supervisar varios TFGs, pero cada uno de estos está asociado a un único usuario. Esto se modela mediante el atributo id-profesor en la entidad TFG. Asimismo, existe una relación de uno a muchos entre TFG y Seguimientos, donde un TFG puede tener múltiples seguimientos asociados, pero cada seguimiento está vinculado a uno único, lo cual se modela a través del atributo id-tfg en la entidad Seguimientos.

Capítulo 11

Pruebas

En cualquier sistema software son necesarias una serie de pruebas que validen y aseguren la calidad, confiabilidad y el correcto funcionamiento del sistema. En este capítulo nos enfocaremos en las diferentes pruebas realizadas con el objetivo de detectar los errores en el software y probar si este cumple con los requisitos previamente establecidos.

11.1. Pruebas de caja blanca

Las pruebas de caja blanca son pruebas estructurales que se centran en la estructura interna del sistema. En otras palabras, en estas pruebas se evalúa la estructura interna del código fuente y se evalúa cómo se ejecutan los diferentes caminos lógicos dentro del programa. En nuestro caso, este tipo de pruebas se han ido realizando a lo largo del proceso de implementación. Esto es debido a que durante todo el proceso se ha ido comprobando tanto la calidad como el funcionamiento correcto del código.

11.2. Pruebas de caja negra

A diferencia de las pruebas de caja blanca, en las pruebas de caja negra se prueba el software sin conocer la estructura interna del código ni los detalles de implementación. El objetivo es probar las funcionalidades del sistema desde la perspectiva del usuario, por lo que se evaluará si se producen los resultados esperados ante diferentes entradas y condiciones. Las pruebas realizadas han sido las siguientes:

11.2.1. CP1: RegistrarNuevoUsuario

Caso de Prueba	CP1: RegistrarNuevoUsuario
Objetivo de la prueba	Verificar el registro de un nuevo usuario en el sistema con datos válidos.
Resultado esperado	El usuario debe ser registrado correctamente y redirigido a la página de inicio de sesión.
Resultado obtenido	Correcto
Problemas encontrados	Ninguno
Solución	-

11.2.2. CP2: IniciarSesión

Caso de Prueba	CP2: IniciarSesión
Objetivo de la prueba	Verificar el inicio de sesión de un usuario existente en el sistema con credenciales válidas.
Resultado esperado	Si todo es correcto, el usuario será redirigido a la página principal de la aplicación, y, en caso contrario, se mostrará que la comprobación de los datos introducidos es errónea.
Resultado obtenido	Correcto
Problemas encontrados	Ninguno
Solución	-

11.2.3. CP3: CrearTFG

Caso de Prueba	CP3: CrearTFG
Objetivo de la prueba	Verificar que un usuario pueda crear un nuevo TFG con datos válidos.
Resultado esperado	El TFG debe ser creado correctamente y aparecer en la lista de TFGs del usuario.
Resultado obtenido	Correcto
Problemas encontrados	Ninguno
Solución	-

11.2.4. CP4: EditarTFG

Caso de Prueba	CP4: EditarTFG
Objetivo de la prueba	Verificar que un usuario pueda editar un TFG existente con datos válidos.
Resultado esperado	Los cambios realizados en el TFG deben guardarse correctamente y reflejarse en la lista de TFGs del usuario.
Resultado obtenido	Correcto
Problemas encontrados	Ninguno
Solución	-

11.2.5. CP5: EliminarTFG

Caso de Prueba	CP5: EliminarTFG
Objetivo de la prueba	Verificar que un usuario pueda eliminar un TFG existente.
Resultado esperado	El TFG debe ser eliminado correctamente y ya no debe aparecer en la lista de TFGs del usuario.
Resultado obtenido	Correcto
Problemas encontrados	Ninguno
Solución	-

11.2.6. CP6: FiltrarTFGs

Caso de Prueba	CP6: FiltrarTFGs
Objetivo de la prueba	Verificar que un usuario pueda filtrar la lista de TFGs por estado.
Resultado esperado	La lista de TFGs debe actualizarse correctamente según los filtros aplicados.
Resultado obtenido	Correcto
Problemas encontrados	Ninguno
Solución	-

11.2.7. CP7: IniciarSesionCredencialesIncorrectas

Caso de Prueba	CP7: IniciarSesionCredencialesIncorrectas
Objetivo de la prueba	Verificar que el sistema maneje correctamente el intento de inicio de sesión con credenciales incorrectas.
Resultado esperado	Se debe mostrar un mensaje de error indicando que las credenciales son incorrectas.
Resultado obtenido	Correcto
Problemas encontrados	Ninguno
Solución	-

11.2.8. CP8: RegistrarUsuarioConDatosInvalidos

Caso de Prueba	CP8: RegistrarUsuarioConDatosInvalidos
Objetivo de la prueba	Verificar que el sistema maneje correctamente el intento de registro con datos inválidos, como correos electrónicos mal formateados
Resultado esperado	El sistema debe mostrar mensajes de error indicando los problemas con los datos ingresados, y el usuario no debe ser registrado.
Resultado obtenido	Correcto
Problemas encontrados	Ninguno
Solución	-

11.2.9. CP9: CerrarSesión

Caso de Prueba	CP9: CerrarSesión
Objetivo de la prueba	Verificar que un usuario pueda cerrar sesión correctamente desde cualquier página de la aplicación.
Resultado esperado	El usuario debe ser redirigido a la página de inicio y su sesión debe finalizarse correctamente.
Resultado obtenido	Correcto
Problemas encontrados	Ninguno
Solución	-

11.3. Problemas encontrados

- En el proyecto, se identificó un problema en el que, tras iniciar sesión, un usuario podía crear y editar TFGs, pero al recargar la página, todos los TFGs



creados por todos los usuarios eran visibles, en lugar de solo los TFGs pertenecientes al usuario autenticado. Este problema se debía a que la aplicación no estaba restringiendo correctamente el acceso a los TFGs basándose en el usuario que los había creado. Para resolver este problema, se implementaron las siguientes medidas:

1. Restricción de Acceso a los TFGs: Se configuró el backend para que cada usuario solo pudiera acceder a los TFGs que él mismo había creado. Esto se logró modificando las consultas en el servidor para filtrar los TFGs según el identificador del usuario autenticado. Ahora, la API de recuperación de TFGs ahora verifica el identificador del usuario autenticado antes de devolver los resultados.
 2. Manejo del Token de Autenticación: El token de autenticación del usuario se guarda y se utiliza para mantener la sesión activa entre recargas de la página sin necesidad de volver a iniciar sesión. Cuando el usuario cierra sesión, el token se elimina del almacenamiento local para asegurar que el usuario no tenga acceso a los datos protegidos después de cerrar sesión.
- Durante el desarrollo de la funcionalidad para añadir notas a un TFG, se identificó un problema relacionado con el esquema de la base de datos. Inicialmente, al diseñar la base de datos, no se consideró la inclusión de un campo para almacenar los seguimientos asociados a cada TFG. Como resultado, la aplicación no podía registrar ni gestionar las notas de forma adecuada, lo que impedía que los usuarios añadieran o visualizaran notas vinculadas a los TFGs. Para abordar este problema, se añadió un nuevo campo en la tabla correspondiente a los TFGs para almacenar las notas o seguimientos. Este campo fue diseñado para asociar múltiples notas a cada TFG. Además, se actualizaron las operaciones del backend para gestionar el nuevo campo. Esto incluyó la modificación de las rutas de la API para permitir la adición y eliminación de notas asociadas a los TFGs.
 - Otro problema que aún no tiene solución es que, tras registrarse en UCO-Track, los usuarios no son redirigidos automáticamente a la página principal. La solución adoptada ha sido mostrar un mensaje de confirmación de registro y permitir que el usuario ingrese sus datos para iniciar sesión por sí mismo.

Capítulo 12

Conclusiones y Futuras Mejoras

En este capítulo final, realizaremos una evaluación global del proyecto, destacando los objetivos alcanzados, las lecciones aprendidas y las perspectivas de mejora y evolución.

12.1. Objetivos conseguidos

En general, podemos decir que el objetivo principal ha sido completado con éxito, dado que el Trabajo Fin de Grado consistía en el diseño y desarrollo de una aplicación web que permita el seguimiento de TFGs. Por otro lado, veremos de manera más específica cada uno de los objetivos alcanzados para tener una visión más detallada de lo logrado con este proyecto:

- Acceder al sistema mediante usuario y contraseña.
- Crear TFGs y asignarlos a cursos académicos.
- Añadir información detallada a cada TFG, incluyendo el título, el alumno y sus datos de contacto.
- Permitir la asignación y el cambio de estados a los TFGs (pendiente de petición, petición aprobada, desarrollo, entregado, presentado, ...).
- Crear seguimientos para cada TFG, registrando fechas y proporcionando un espacio para información relevante (qué se ha hecho, que se ha corregido, que se ha hablado...).
- Generar un documento con todos los seguimientos de un TFG a modo de resumen



- Listar los TFGs por tipo/estado para una mejor organización.
- Diseñar una interfaz intuitiva y amigable para que los profesores puedan gestionar TFGs.

12.2. Objetivos no conseguidos

Durante el desarrollo del proyecto, uno de los objetivos propuestos fue enlazar el TFG con la petición de tema una vez aprobada. Sin embargo, tras la evaluación de las necesidades y prioridades del proyecto, se determinó que esta funcionalidad no era esencial para la fase actual. La prioridad se centró en implementar y estabilizar otras funcionalidades críticas que tenían un impacto más inmediato en la experiencia del usuario y en la operativa básica del sistema.

En cambio, la incorporación de un sistema de estados para los TFG se consideró una mejora valiosa para la gestión del proceso. Este sistema de estados permite categorizar los TFG en diferentes etapas, tales como: 'Pendiente de Petición', 'Petición Aprobada', 'Presentado'... Esta funcionalidad facilita el seguimiento y la gestión del progreso de cada TFG, proporcionando una visión clara del estado actual del proceso.

La decisión de posponer la integración entre el TFG y la petición de tema se alineó con el enfoque y las necesidades actuales del proyecto, y esta característica puede ser considerada para futuras fases de desarrollo según las necesidades emergentes y las prioridades del sistema.

12.3. Aprendizaje

Al finalizar un proyecto, siempre es valioso realizar una retrospectiva y extraer las lecciones aprendidas. A lo largo del desarrollo de **UCOTrack**, he tenido la oportunidad de adquirir y perfeccionar una serie de conocimientos y habilidades que han enriquecido mi experiencia. Este proceso me ha permitido explorar y familiarizarme con diversos aspectos técnicos y conceptuales, contribuyendo significativamente a mi crecimiento como profesional en el campo de la ingeniería informática. Algunos de los aprendizajes clave han sido:

12.3.1. Next.js

Aunque ya tenía experiencia previa con React, el trabajar con Next.js me ha proporcionado un entendimiento más profundo sobre la creación de aplicaciones web con renderizado del lado del servidor. Este framework me ha enseñado a optimizar el rendimiento de la aplicación y mejorar la experiencia del usuario, además de facilitar la navegación y la organización del código mediante su enfoque basado en páginas.

12.3.2. NestJS

Este proyecto ha sido mi primera incursión en NestJS, un framework que me ha permitido estructurar y desarrollar la API de manera eficiente. NestJS me ha ayudado a comprender mejor los patrones de diseño y las mejores prácticas en el desarrollo de aplicaciones backend en TypeScript.

12.3.3. TypeScript

Trabajar con TypeScript ha sido una experiencia enriquecedora que ha mejorado la calidad de mi código. Gracias a su tipado estático, he podido detectar errores en tiempo de desarrollo y escribir código más robusto y predecible. Esta herramienta ha sido clave para gestionar la complejidad de la aplicación, asegurando que los distintos componentes del sistema interactúen de manera segura y eficiente.

12.3.4. Estructuración de Proyectos

La organización del código ha sido fundamental para el éxito de este proyecto. He aprendido a estructurar un proyecto complejo como **UCOTrack** utilizando conceptos como módulos y componentes, tanto en el frontend con Next.js como en el backend con NestJS. Esta organización ha mejorado la modularidad y mantenibilidad del código, facilitando la colaboración y futuras expansiones del proyecto.

En conclusión, el desarrollo de **UCOTrack** ha sido una experiencia de aprendizaje integral que me ha permitido profundizar en tecnologías clave, mejorar mis habilidades de programación y adquirir una visión más clara sobre cómo gestionar proyectos de software desde su concepción hasta su implementación.

12.4. Futuras Mejoras

Aunque el presente proyecto ha sido completado con éxito y ha logrado un nivel satisfactorio en términos de funcionalidad y finalización, siempre existe la posibilidad



de ir más allá. Un proyecto puede expandirse tanto como lo permita la creatividad y la ambición. Por ello, hay varias ideas adicionales que podrían implementarse en el futuro para enriquecer aún más la aplicación, tales como:

- Permitir que los TFG se etiqueten o clasifiquen por áreas de investigación, departamentos o temas específicos para facilitar la búsqueda y el filtrado.
- Permitir la integración con calendarios (Google Calendar, Outlook) para que los usuarios puedan ver fechas importantes como plazos de entrega o reuniones con alumnos.
- Proveer estadísticas y gráficos que muestren el progreso de los TFG, como número de TFG completados, en progreso...

Bibliografía

- [1] Martin Dougiamas. Moodle., 2002. URL <https://moodle.uco.es/moodlemap/>. Accedido: 10.04.2024.
- [2] Trello Inc. Trello., 2011. URL <https://trello.com/es>. Accedido: 10.04.2024.
- [3] Next.js. URL <https://nextjs.org/>. Accedido: 12.04.2024.
- [4] React. URL <https://reactjs.org/>. Accedido: 12.04.2024.
- [5] Vercel. URL <https://vercel.com/>. Accedido: 12.04.2024.
- [6] Nestjs. URL <https://nestjs.com/>. Accedido: 12.04.2024.
- [7] Overleaf. URL <https://www.overleaf.com/>. Accedido: 12.04.2024.
- [8] Microsoft. Visual studio code., 2011. URL <https://code.visualstudio.com/>. Accedido: 10.04.2024.
- [9] Mysql. URL <https://www.mysql.com/>. Accedido: 12.04.2024.
- [10] Docker. URL <https://www.docker.com/>. Accedido: 31.07.2024.
- [11] Draw.io. URL <https://app.diagrams.net/>. Accedido: 12.04.2024.
- [12] Jwt. URL <https://jwt.io/>. Accedido: 31.07.2024.

Anexo A

Manual de Usuario

En este anexo se explicará el funcionamiento de la aplicación web para el seguimiento de TFGs. En esta sección se busca brindar a los usuarios una experiencia completa y detallada para el uso efectivo de la aplicación, asegurando que dispongan de toda la información necesaria para resolver sus dudas y maximizar su productividad.

A.1. Requisitos previos

Antes de proceder con la instalación de la aplicación, es necesario cumplir ciertos requisitos previos. A continuación, se detalla el software necesario y cómo instalarlo.

A.1.1. Instalación de Node.js

Para poder ejecutar la aplicación, se requiere tener instalado Node.js y npm (Node Package Manager). Puedes instalarlos ejecutando el siguiente comando en la terminal:

```
sudo apt install nodejs npm
```

A.1.2. Instalación de Prisma

Prisma es una herramienta de ORM (Object-Relational Mapping) que facilita la interacción con bases de datos SQL mediante una API moderna y segura. Para instalar Prisma, ejecuta el siguiente comando en la terminal dentro del proyecto:

```
npm install @prisma/client
```

A.2. Instalación de la aplicación

A.2.1. Levantar la base de datos

El primer paso para la instalación es levantar la base de datos utilizando Docker. Ejecuta el siguiente comando en la raíz del proyecto:

```
docker-compose up
```

A.2.2. Clonación del repositorio

A continuación, se debe clonar el repositorio del proyecto desde GitHub. Ejecuta el siguiente comando:

```
https://github.com/irenelopez30/ucotrack.git
```

```
git clone https://github.com/irenelopez30/ucotrack.git
```

Esto descargará una carpeta llamada UCOTrack, la cual contiene todo el código del proyecto.

Para descargar todas las dependencias del proyecto, tanto en la carpeta backend como en el frontend debes hacer un

```
npm install
```

A.3. Configuración del entorno

A.3.1. Variables de entorno

Es posible que la aplicación requiera algunas variables de entorno para funcionar correctamente. Asegúrate de que el archivo `.env` esté correctamente configurado en la raíz del proyecto. Si es necesario, contacta al administrador del sistema o revisa la documentación interna del proyecto para obtener los valores necesarios.

A.4. Ejecución de la aplicación

Una vez que hayas completado los pasos anteriores, ya estarás listo para ejecutar la aplicación.



A.4.1. Arranque del frontend

Para arrancar el frontend, navega a la carpeta **frontend** y ejecuta el siguiente comando:

```
npm run dev
```

A.4.2. Arranque del backend

Del mismo modo, para arrancar el backend, dirígete a la carpeta **backend** y ejecuta el siguiente comando:

```
npm run start:dev
```

Una vez que ambos servidores estén en funcionamiento, podrás acceder a la aplicación a través de un navegador web, ingresando la siguiente dirección:

<http://localhost:3000/>

A.5. Descripción de la Interfaz

En este apartado, se proporcionará una explicación detallada de la interfaz de usuario de la aplicación **UCOTrack**. Es fundamental comprender cómo interactuar con la interfaz y aprovechar al máximo las diferentes funcionalidades y herramientas disponibles. El objetivo de esta sección es familiarizarse con la disposición de los elementos, los controles de navegación y las distintas secciones de la interfaz.

A.5.1. Inicio de Sesión

Al acceder por primera vez a la aplicación, lo primero que se visualizará será la vista de la página principal (Figura A.1). Si pulsamos en “Comienza ahora.” “Iniciar sesión” pasamos a la página de inicio de sesión (Figura A.2). En esta pantalla, se solicita al usuario ingresar tanto un correo electrónico como una contraseña. También se ofrece una opción “Registrarse”, que, al seleccionarse, le mandará a otra página para rellenar el cuestionario y registrarse (Figura A.3). Si las credenciales son correctas, se redirige al usuario a la página principal de la aplicación. En caso contrario, se notificará que el correo o la contraseña introducidos no son correctos.



Figura A.1: Página principal

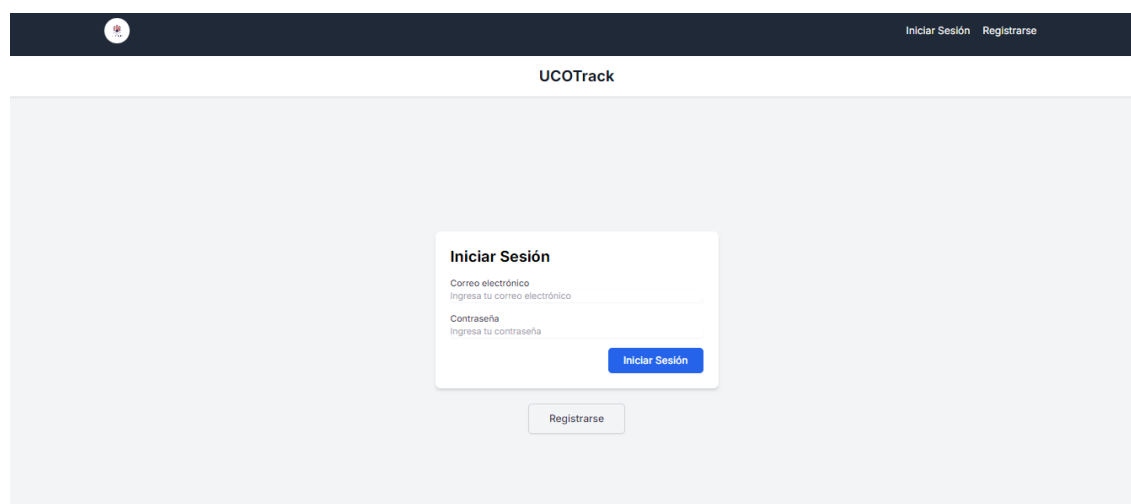


Figura A.2: Página de inicio de sesión

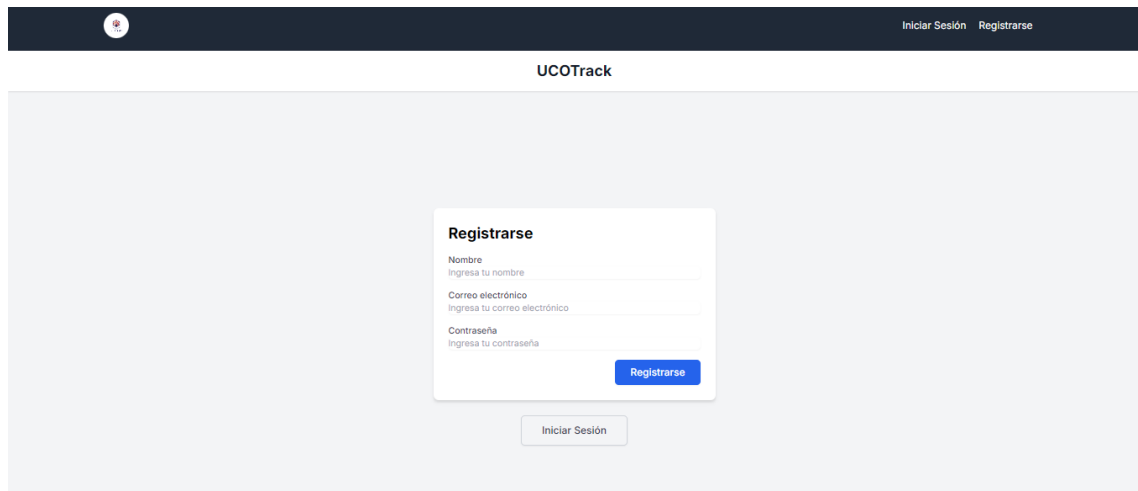


Figura A.3: Página de Registro

A.5.2. Página de Inicio

Cuando el usuario inicia sesión correctamente, es redirigido a la página de inicio de la aplicación (Figura A.4). Esta será la vista donde el usuario podrá acceder a todas las funcionalidades del sistema. A la izquierda, se encuentran varias opciones de navegación, que permiten al usuario gestionar diferentes aspectos del sistema como “Crear TFG”, “Listar TFGs” y “Cerrar Sesión”. La barra de navegación superior, visible en todas las vistas, también proporciona acceso rápido a todas las secciones de la aplicación, mejorando la navegabilidad general.

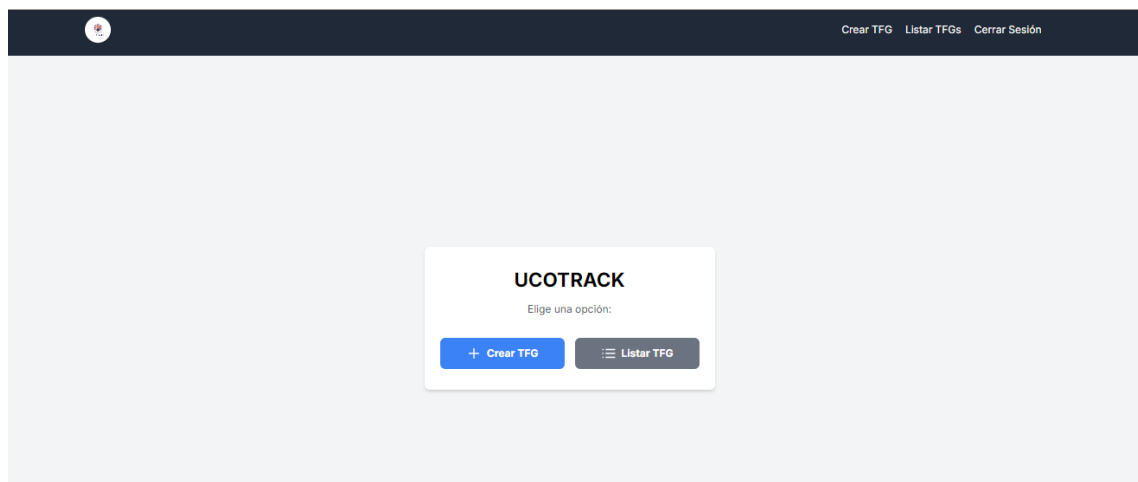


Figura A.4: Página de inicio

A.5.3. Creación de TFG

Al seleccionar la opción “Crear TFG”, se presenta al usuario un formulario para la creación de un nuevo Trabajo de Fin de Grado (TFG) (Figura A.5). El formulario incluye campos para ingresar el nombre del proyecto, el año académico, el nombre del alumno, su correo electrónico, y una descripción del proyecto. El usuario también puede asignar el estado actual del TFG (e.g., “Pendiente de petición”, “En desarrollo”). Una vez completados todos los campos, el usuario debe hacer clic en el botón “Crear” para añadir el TFG al sistema.

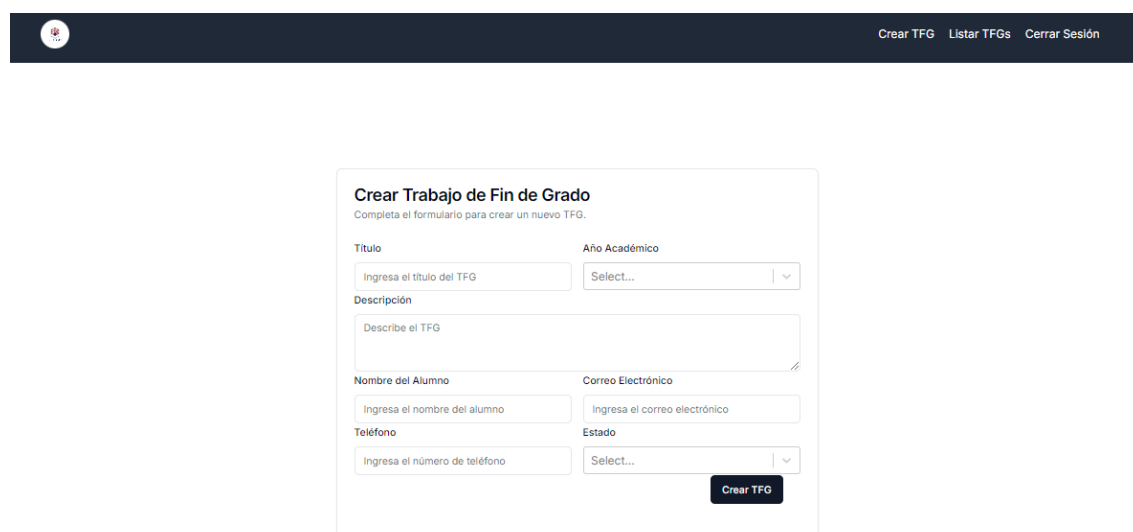


Figura A.5: Formulario de creación de TFG

A.5.4. Listado de TFGs

Después de crear un TFG, el usuario puede visualizar una lista de todos los TFGs disponibles en el sistema, seleccionando la opción “Listar TFGs” (Figura A.6). Esta lista muestra cada TFG con su título, el año académico, una descripción, y el estado actual. Además, se proporcionan opciones para editar o eliminar cada TFG.

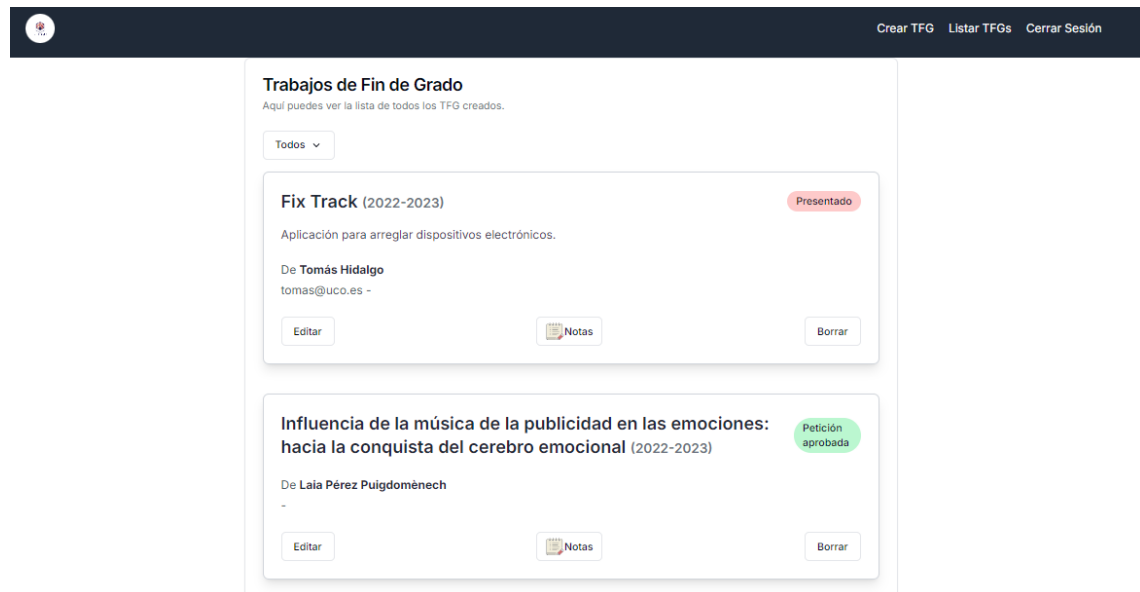


Figura A.6: Listado de TFGs

A.5.5. Confirmación de Eliminación de TFG

Cuando un usuario decide eliminar un TFG, se le presenta una ventana emergente de confirmación (Figura A.7). En esta ventana, se le pregunta al usuario si está seguro de que desea eliminar el TFG seleccionado, ofreciendo las opciones “Eliminar” o “Cancelar” para confirmar o rechazar la acción, respectivamente.

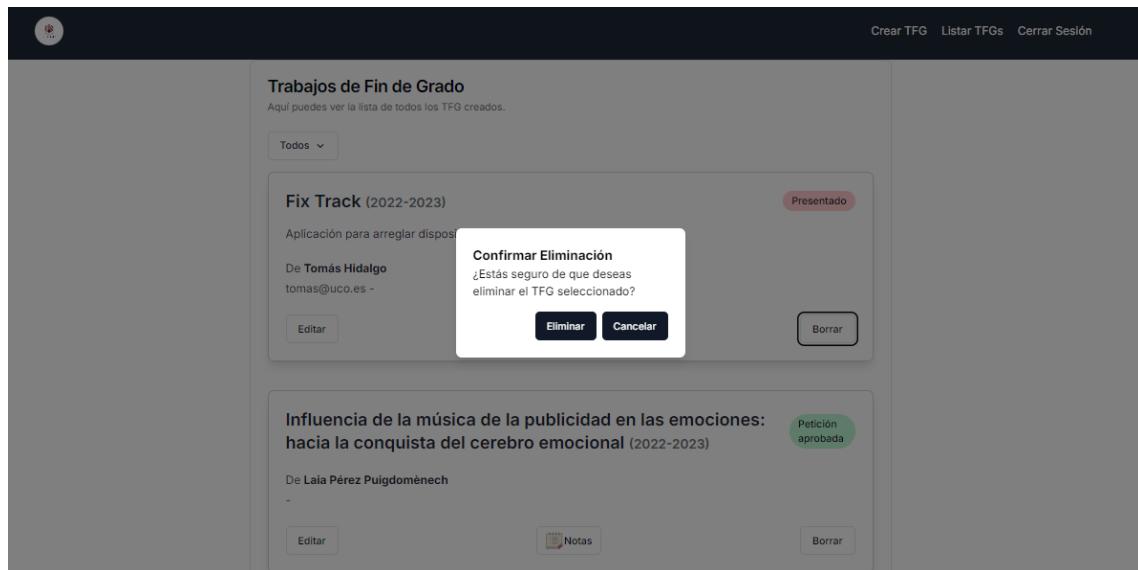


Figura A.7: Ventana de confirmación de eliminación de TFG

A.5.6. Filtrado de TFGs

En la vista de listado de TFGs, el usuario tiene la opción de filtrar los TFGs según su estado utilizando un menú desplegable (Figura A.8). Las opciones de filtrado incluyen “Todos”, “Pendiente de petición”, “Petición aprobada”, “En desarrollo”, “Entregado” y “Presentado”. Esta funcionalidad permite al usuario enfocarse en los TFGs que son más relevantes para sus necesidades actuales.

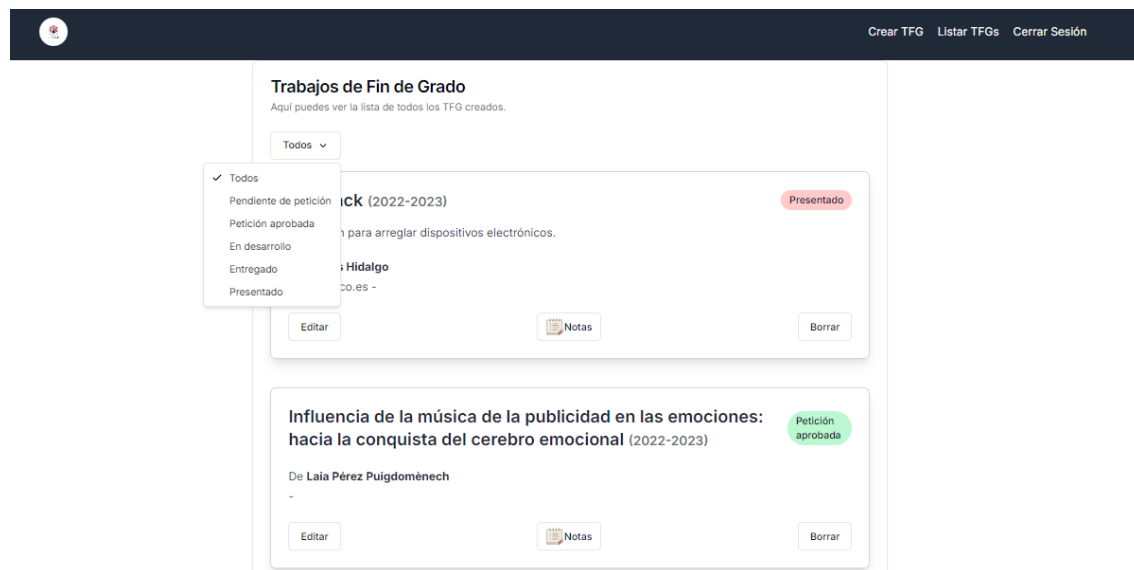


Figura A.8: Menú desplegable de filtrado de TFGs

A.5.7. Añadir Notas a un TFG

Al visualizar un TFG específico, el usuario tiene la opción de agregar notas relacionadas con el progreso del proyecto (Figura A.9). Se muestra un campo de texto donde se puede ingresar la nota, y un botón para agregarla. Las notas existentes se listan debajo con la opción de eliminarlas si es necesario.

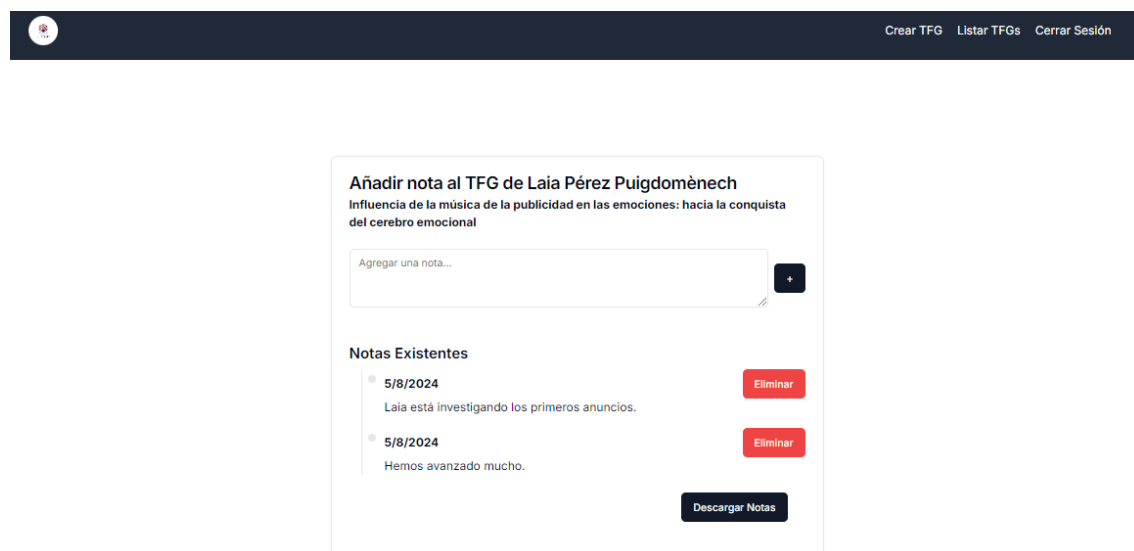


Figura A.9: Pantalla para añadir notas a un TFG

A.5.8. Descarga de Notas de TFG

En la misma pantalla donde se añaden notas, también se ofrece la opción de descargar todas las notas existentes en un formato compatible para su posterior revisión. Este botón, ubicado en la parte inferior de la lista de notas, permite al usuario guardar las notas de progreso localmente en su dispositivo.

A.5.9. Edición de TFG

Además de la creación y eliminación, la aplicación permite la edición de TFGs existentes (Figura A.10). Al seleccionar el botón “Editar” junto a un TFG en la lista, el usuario es llevado a una página donde puede modificar la información previamente ingresada, como el nombre del proyecto, la descripción, y el estado actual. Una vez realizados los cambios, se debe guardar la actualización para que se reflejen en la base de datos.

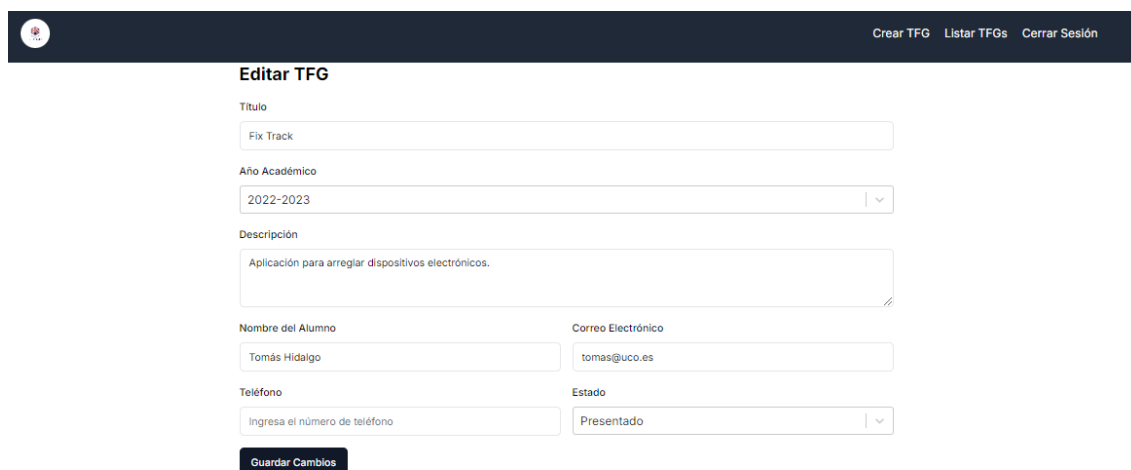


Figura A.10: Página de edición de un TFG

A.5.10. Cierre de Sesión

Finalmente, el usuario puede cerrar su sesión actual seleccionando el ícono correspondiente en la barra de navegación superior. Esto garantiza que su cuenta esté protegida después de haber terminado de usar la aplicación.

Anexo B

Manual de Código

Este Manual de Código pretende servir de guía para entender el código de la aplicación UCOTrack. Está diseñado para desarrolladores y mantenedores del proyecto para ayudarles a comprender el código fuente, las tecnologías utilizadas y cualquier requisito o dependencia relevante

El código del proyecto al completo se encuentra en el siguiente repositorio:

<https://github.com/irenelopez30/ucotrack.git>

B.1. Descripción del Proyecto

UCOTrack es una aplicación web destinada al seguimiento de Trabajos de Fin de Grado (TFGs). La aplicación permite a los profesores supervisar los TFGs y gestionar el progreso de los mismos a través de una interfaz web.

B.2. Tecnologías y lenguajes utilizados

Backend

- Lenguaje: TypeScript
- Framework: NestJS
- Base de Datos: MySQL
- ORM: Prisma
- Contenedor: Docker (para la base de datos)



Frontend

- Lenguaje: TypeScript
- Framework: Next.js
- Renderizado: Client-Side Rendering (CSR)

Autenticación

- Método: JWT (JSON Web Token)

B.3. Requisitos y dependencias

Requisitos

- Debemos instalar node.js y npm

```
sudo apt install nodejs npm
```

Dependencias del proyecto

- Backend
 - @nestjs/core - Núcleo del framework NestJS.
 - @nestjs/common - Módulos comunes de NestJS.
 - @nestjs/typeorm - Integración con TypeORM.
 - @prisma/client - Cliente de Prisma para la interacción con la base de datos.
 - typeorm - ORM para la gestión de bases de datos.
 - mysql2 - Conector de MySQL para Node.js.
- Frontend
 - next - Framework para React que permite el renderizado del lado del servidor y del cliente.
 - react - Biblioteca para construir interfaces de usuario.
 - react-dom - Biblioteca para manejar el DOM en React.
 - typescript - Lenguaje de programación con tipado estático.
- Autenticación
 - jsonwebtoken - Biblioteca para trabajar con JWT.



Configuraciones Adicionales

- Docker: La base de datos MySQL se ejecuta en un contenedor Docker. Debemos tener Docker y Docker Compose instalados.
- Variables de Entorno: Se debe configurar el archivo .env con las variables necesarias para el correcto funcionamiento del proyecto. En UCOTrack, tendremos la cadena de conexión para la base de datos MySQL

```
DATABASE_URL="mysql://irene:ucotrack@localhost:3306/UCOTrack"
```

- mysql:// - El protocolo de conexión, en este caso, MySQL.
- irene - El nombre de usuario para conectarse a la base de datos.
- ucotrack - La contraseña del usuario para la autenticación.
- localhost - La dirección del servidor de la base de datos.
- 3306 - El puerto en el que el servidor de MySQL está escuchando.
- UCOTrack - El nombre de la base de datos a la que se está conectando.

B.4. Estructura del Código

- Backend
 - Carpeta src: Contiene la mayoría del código fuente del backend.
 - domain/ - Modelos del dominio y repositorios.
 - application/ - Casos de uso del sistema.
 - infrastructure/ - Implementaciones concretas y servicios externos.
 - Carpeta config/ - Archivos de configuración para NestJS y Pri

```
ucotrack-backend/
```

```
database/  
  init_sql/  
    init.sql
```

```
dist/  
node_modules/  
prisma/  
  schema.prisma
```



```
src/  
  tfgs/  
    aplicacion/  
      createTfg.ts  
      deleteTfg.ts  
      getTfg.ts  
      getTfgs.ts  
      index.ts  
      modifyTfg.ts  
  
    domain/  
      index.ts  
      nota.ts  
      tfg.ts  
      tfgRepository.ts  
  
    infrastructure/  
      tfgController.ts  
      tfgRouter.ts  
  
  usuarios/  
    aplicacion/  
      createUsuario.ts  
      deleteUsuario.ts  
      getUsuario.ts  
      getUsuarios.ts  
      index.ts  
      loginUser.ts  
  
    domain/  
      Email.ts  
      index.ts  
      usuario.ts  
      usuarioRepository.ts  
  
    infrastructure/  
      usuarioController.ts  
      usuarioRouter.ts
```



```
main.ts
```

```
.env  
.eslintrc.js  
.prettierrc  
docker-compose.yml  
nest-cli.json  
package-lock.json  
package.json
```

■ Frontend

- Carpeta pages/ - Rutas y páginas de la aplicación.
- Carpeta components/ - Componentes reutilizables de React.
- Carpeta styles/ - Archivos de estilos CSS y de diseño.
- Carpeta public/ - Archivos estáticos accesibles públicamente.

```
ucotrack-fronted/
```

```
.next/  
node_modules/  
public/  
  
src/  
  app/  
    api/  
      auth/  
        route.js  
  
    inicio/  
      page.tsx  
  
    providers/  
      AuthProvider.js  
  
    registro/  
      page.tsx  
      sesion/  
        page.tsx
```




```
tfg/  
  crear/  
    page.tsx  
  editar/  
    [id]/  
      page.tsx  
  listar/  
    page.tsx  
  seguimientos/  
    [id]/  
      page.tsx  
usuario/  
  page.tsx/  
  
layout.tsx  
  
global.css  
  
page.tsx  
  
components.tsx/  
  component/  
    crear.tsx  
    editar.tsx  
    inicio.tsx  
    listar.tsx  
    registro.tsx  
    seguimiento.tsx  
    sesion.tsx  
    usuario.tsx  
  Footer/  
    Footer.tsx  
  Header/  
    Header.tsx  
  ProtectedRoutes/  
    ProtectedRoutes.tsx  
  ui  
    button.tsx  
    card.tsx  
    dropdown-menu.tsx
```



```
input-tsx  
label.tsx  
select.tsx  
textarea.tsx
```

```
lib/  
  utils.js
```

```
.eslintrc.json  
components.json  
next-env.d.ts  
next.config.mjs  
postcss.config.mjs  
tailwind.config.ts  
tsconfig.json  
package-lock.json  
package.json
```