



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA
Mención en Ingeniería del Software

**Ampliación y mejora de una aplicación web
para la gestión de TFGs en la Escuela de
Ingeniería Informática de Valladolid**

Alumno: Diego Bombín Sanz

Tutor: Yania Crespo González-Carvajal

...

Resumen

Este Trabajo de Fin de Grado (TFG) ha dado continuidad a una iniciativa previa de la Escuela de Ingeniería Informática de la Universidad de Valladolid, enfocada en la digitalización y unificación de la gestión de los TFGs. El objetivo principal de este proyecto es ampliar las funcionalidades ya existentes, incorporando un sistema integral para la oferta de TFGs por parte de profesores y empresas, así como para la solicitud de los mismos por parte del alumnado. Además, se buscó mejorar la eficiencia y usabilidad de la aplicación mediante un análisis exhaustivo del trabajo anterior y la mejora de su documentación. Se ha continuado con la arquitectura y tecnologías utilizadas para el desarrollo de la versión inicial de este sistema. Se cuenta con una plataforma web formada por dos subproyectos, el backend, desarrollado con Spring boot, y el frontend, desarrollado con React. Se ha realizado una planificación y un seguimiento del proyecto basada en un modelo de proceso híbrido, por fases y ágil. Se planificaron y ejecutaron una fase Inicial y una fase de Ingeniería Inversa que permitieron comprender el proyecto desarrollado previamente. Se ha trabajado con stakeholders y usuarios reales de la aplicación recogiendo sus necesidades de nuevas funcionalidades y de mejora de las existentes. Adicionalmente, se ha mejorado la configuración y despliegue para los entornos de desarrollo y de producción.

Abstract

This Final Degree Project (FDP) builds upon a prior initiative by the School of Informatics Engineering at the University of Valladolid, which focused on the digitalisation and unification of FDP management. The primary objective of this project was to extend existing functionalities, by incorporating a comprehensive system for the offering of FDPs by academic staff and companies, and for their application by students. Furthermore, efforts were made to enhance the application's efficiency and usability through a comprehensive analysis of the preceding work and the refinement of its documentation. The architectural design and technologies employed for the system's initial iteration have been retained. The system encompasses a web platform comprising two distinct sub-projects: the backend, developed using Spring Boot, and the frontend, built with React. Project planning and oversight were conducted based on a hybrid, phased, and agile process model. An Initial Phase and a Reverse Engineering Phase were both planned and executed, enabling a comprehensive understanding of the pre-existing project. Direct engagement was fostered with both stakeholders and actual application users to ascertain their requirements for novel functionalities and improvements to existing features. Moreover, the configuration and deployment procedures for both development and production environments have been further optimised.

Índice general

Resumen	III
Abstract	V
Lista de figuras	XI
Lista de tablas	XIII
1. Introducción	1
1.1. Contexto	1
1.2. Motivación	2
1.3. Objetivos	2
1.4. Estructura de la memoria	3
2. Requisitos y Planificación	5
2.1. Proceso de desarrollo basado en un modelo híbrido	5
2.1.1. Sprint	6
2.1.2. Ingeniería Inversa	6
2.2. Stakeholders	7
2.3. Backlog inicial	8
2.4. Análisis de riesgos	9
2.5. Calendarización	12

2.6. Presupuesto	12
3. Análisis	15
3.1. Modelo de Procesos	15
3.2. Modelado del dominio	19
3.3. Casos de Uso	19
4. Tecnologías utilizadas	33
4.1. Tecnologías de desarrollo	33
4.1.1. Tecnologías utilizadas para el frontend	33
4.1.2. Tecnologías utilizadas para el backend	34
4.1.3. Tecnologías utilizadas para otros fines	34
4.2. Tecnologías de gestión	34
4.2.1. Clockify	34
4.3. Tecnologías de documentación	35
5. Diseño	37
5.1. Arquitectura Backend	37
5.1.1. Arquitectura del sistema	38
5.1.2. Identificación y autenticación	38
5.1.3. Propuestas en convenio	39
5.2. Arquitectura Frontend	43
5.2.1. Estructura del proyecto	43
5.2.2. Listado de componentes nuevos	43
5.2.3. Listado de componente modificados	44
6. Implementación y pruebas	45
6.1. Pruebas de la API con Postman	45
6.1.1. Autenticación	46

6.1.2. Propuestas	46
6.1.3. Solicitud de Propuestas	46
6.1.4. Autopropuestas	47
6.2. Implementación	47
6.2.1. Base de datos y consultas	47
6.2.2. Implementación sobre la aplicación web	47
6.2.3. Modificación externa de listas de direcciones de correo	48
7. Seguimiento del proyecto	51
7.1. Fase Inicial	51
7.2. Fase Ingeniería Inversa	51
7.3. Fase de Nuevos Requisitos	53
7.3.1. Sprint 1	53
7.3.2. Sprint 2	54
7.3.3. Sprint 3	54
7.3.4. Sprint 4	55
7.4. Fase de Mejoras	56
7.4.1. Sprint 5	56
7.4.2. Sprint 6	57
7.5. Cierre del Proyecto	58
7.6. Resumen de la ejecución del proyecto	59
7.6.1. Calendarización final	59
7.6.2. Gestión de Riesgos	59
7.6.3. Costes	61
8. Conclusiones	63
8.1. Líneas de trabajo futuras	63

Bibliografía	68
A. Manuales	69
A.1. Parámetros de configuración dinámica y segura	69
A.1.1. Modificación de las listas de usuarios con roles especiales	69
A.2. Lanzamiento del sistema en entornos diferenciados (<i>Backend</i>)	70
A.2.1. Estructura de configuración	70
A.2.2. Lanzamiento en entorno de desarrollo	70
A.2.3. Lanzamiento en entorno de producción	71
A.2.4. Borrado de datos tras las pruebas	72
A.3. Lanzamiento del sistema en entornos diferenciados (<i>Webapp</i>)	72
A.3.1. Lanzamiento en Desarrollo	73
A.3.2. Lanzamiento en Producción	73
A.4. Manual de usuario	73
A.4.1. Menú principal	73
A.4.2. Propuestas	75
A.4.3. Mis solicitudes	78
A.4.4. Mis autopropuestas	79
A.4.5. Tribunales	80
B. Resumen de enlaces adicionales	83

Lista de Figuras

3.1. Fragmento del proceso centrado en la creación de propuesta de TFG	16
3.2. Fragmento del proceso centrado en la solicitud de propuesta ofertada	17
3.3. Fragmento del proceso centrado en la autopropuesta y selección de estudiante en TFG en convenio	18
3.4. Modelo del dominio	21
3.5. Casos de Uso	22
3.6. Diagrama de estados de las propuestas según el TFG anterior	23
3.7. Diagrama de estados de las propuestas	23
5.1. Diagrama de la arquitectura de paquetes en el backend	39
5.2. Diagrama de despliegue local, sin el uso del CAS	40
5.3. Flujo principal para la creación de una propuesta de TFG en convenio	41
5.4. Flujo detallado de las transacciones del backend	42
5.5. Arquitectura MVVM tomada de Modelo-vista-modelo [9]	43
6.1. Diagrama de la Base de Datos Previo	48
6.2. Diagrama de la base de datos final	49
A.1. Menú principal como estudiante	74
A.2. Menú principal como Tutor UVa	74
A.3. Menú principal como Coordinador de Título	75

A.4. Menú principal como Delegado de empresas 75

A.5. Listado personal de propuestas publicadas por Tutor UVa 76

A.6. Listado de propuestas publicadas 76

A.7. Creación de propuesta en convenio 77

A.8. Vista de la información de una propuesta con rol de Coordinador de Título . 78

A.9. Vista de más detalles de propuesta de convenio 78

A.10.Vista de la información de una propuesta con rol de Delegado de empresas . . 79

A.11.Mis solicitudes 79

A.12.Mis autopropuestas 80

A.13.Listado de tribunales 81

A.14.Editar tribunal 81

Lista de Tablas

2.1. Backlog inicial del proyecto 9

2.2. Riesgos del proyecto 10

2.3. Exposición a los riesgos 10

2.4. Medidas de contingencia y mitigación 11

2.5. Calendarización del proyecto 12

2.6. Presupuesto simulado 13

3.1. CU01: Solicitar propuesta de TFG 20

3.2. CU02: Aceptar solicitud propuesta de TFG 24

3.3. CU03: Crear Propuesta de TFG 25

3.4. CU04: Crear Propuesta de TFG en convenio 26

3.5. CU05: Solicitud de adjudicación de TFG (modificación) 27

3.6. CU06: Completar información del TFG en convenio 28

3.7. CU07: Consultar autopropuestas 29

3.8. CU08: Autoproponerse 29

3.9. CU09: Consultar solicitudes 30

3.10. CU10: Seleccionar Estudiante 31

7.1. Fase Inicial 52

7.2. Fase Ingeniería Inversa 52

7.3. Sprint 1 53

7.4. Sprint 2 54

7.5. Sprint 3 55

7.6. Sprint 4 56

7.7. Sprint 5 57

7.8. Sprint 6 58

7.9. Cierre del proyecto 59

7.10. Resumen de la calendarización final 60

7.12. Presupuesto Real 61

7.11. Calendarización final del proyecto 62

Capítulo 1

Introducción

1.1. Contexto

El presente Trabajo de Fin de Grado (TFG) es la continuación del trabajo realizado anteriormente por otros estudiante el pasado curso [2]. En él se buscaba unificar y digitalizar los procesos referentes a la gestión de los TFGs de la Escuela de Ingeniería Informática de la Universidad de Valladolid (UVa).

A lo largo de dicho proyecto, se lograron notables avances tanto en el área de las adjudicaciones de TFGs, como en la gestión de tribunales por el Comité de Título, siendo posible sentar las bases para lo que sería una gestión digital de los TFGs. No obstante, la envergadura que implicó su desarrollo y la cantidad de aspectos que podrían ser abordados a largo de un proceso mayor en relación con los TFGs de la Escuela, resultó en la propuesta de realizar un segundo TFG, ampliando y mejorando lo ya realizado.

El presente trabajo tiene como finalidad dar continuidad a esa línea de desarrollo, explorando las funcionalidades que se dejaron fuera del alcance en el TFG anterior y abordando las mejoras necesarias que permitan incrementar la eficiencia y usabilidad del sistema en cuestión.

Para ello, este TFG se apoyará en la normativa vigente, incluyendo el Reglamento sobre la elaboración y evaluación del Trabajo de Fin de Grado de la Universidad de Valladolid [14] y la Normativa de Evaluación del Trabajo de Fin de Grado de E.T.S de Ingeniería Informática [13]. Así como en la colaboración con el Servicio de Tecnologías de la Información y las Comunicaciones de la Universidad (S.T.I.C.), con los Técnicos de la Escuela, el actual Coordinador de la Titulación y con el actual Subdirector de Relaciones con Empresas de nuestra Escuela que participa como delegado de los TFGs en convenio con empresas.

1.2. Motivación

El trabajo previo supuso un comienzo en la digitalización y unificación en la gestión de los TFGs en la Escuela. Sin embargo, quedaron aspectos importantes sin implementar como el hecho de ofertar un TFG, ya sea por un profesor o por una empresa, o la solicitud de dichas ofertas por parte del alumnado, permitiendo reducir el tiempo dedicado tanto a la gestión como a la tramitación y mejorando así la experiencia de usuario.

Actualmente la Escuela cuenta con una sencilla aplicación en la que los profesores pueden ofertar TFGs y los estudiantes pueden consultar los mismos. La motivación en este caso es integrarla en una misma aplicación, abordando de paso su modernización y permitiendo que los estudiantes puedan solicitar el TFG ofertado a través de la misma aplicación y, a la vez, que cuando el profesor asigne el TFG ofertado a un estudiante, se presente automáticamente al Comité de Título para su adjudicación/asignación y posterior publicación.

En lo que respecta a mi persona, el proyecto me ofrece la oportunidad de profundizar en varias áreas clave del desarrollo y gestión de software, poniendo en práctica tanto mis conocimientos adquiridos en la carrera como los autodidactas, por lo que personalmente es un desafío que me obliga a mejorar mis habilidades y equilibrarme para futuros desafíos.

1.3. Objetivos

Para un desarrollo exitoso es importante fijarse unos objetivos claros con los que asentar las bases del proyecto. Sin embargo, al tratarse de un proyecto basado en un modelo de proceso híbrido, que incorpora un enfoque ágil, estos objetivos pueden variar a lo largo del desarrollo del proyecto.

El objetivo general para la realización de este TFG es, en grandes rasgos, la continuación del TFG anterior. Debido a que en la realización de un único TFG no fue posible desarrollar un proyecto de tal envergadura, hubo ciertas funcionalidades que se quedaron sin implementar, convirtiéndose en los objetivos específicos de este TFG.

Dichos objetivos son los mostrados a continuación:

- Incorporar nuevas funcionalidades en relación a la oferta y solicitud de TFGs.
- Realizar un análisis del proyecto previo y llevar a cabo modificaciones que mejoren tanto la usabilidad como el rendimiento de la aplicación.
- Mejorar la documentación existente. Debido a que se trata de un proyecto de grandes dimensiones, la documentación debe ser completa y detallada.

1.4. Estructura de la memoria

Este documento se estructura de la siguiente forma:

Capítulo 2 Requisitos y planificación: Describe todo el proceso previo a la implementación, siendo la parte donde se agrupa todo lo que se realizará y la forma en que se llevará a cabo.

Capítulo 3 Análisis: Recoge el análisis sobre el trabajo realizado en el proyecto anterior, para la inclusión de las nuevas funcionalidades.

Capítulo 4 Tecnologías utilizadas: Describe aquellas tecnologías utilizadas en ambos proyectos, a modo de contextualizar el porqué de usarlas.

Capítulo 5 Diseño: Recopila todas aquellas decisiones llevadas a cabo en ambas aplicaciones, *frontend* y *backend*, para el desarrollo de las funcionalidades.

Capítulo 6 Implementación y pruebas: Describe cómo se han realizado tanto el desarrollo durante el proyecto, cómo la forma que se ha utilizado para probar las funcionalidades incluidas.

Capítulo 7 Seguimiento del proyecto: Se expone el seguimiento detallado del proyecto, mostrando el avance de cada fase conforme a la planificación.

Capítulo 8 Conclusiones: Reflexiona sobre los resultados del proyecto y el aprendizajes conseguido con la realización del mismo, presentando las conclusiones finales.

Anexo A Manuales: Recopila la documentación técnica, incluyendo manuales de instalación, configuración, mantenimiento y uso.

Anexo B Resumen de enlaces adicionales: Se incluyen enlaces de interés sobre el proyecto, tales como el repositorio de código o el acceso a la documentación de la API.

Capítulo 2

Requisitos y Planificación

En este capítulo se estudiarán los riesgos que pueden surgir a lo largo del proyecto y se describe un plan de proyecto para el desarrollo del mismo que tenga en cuenta tanto los riesgos como la realidad del estudiante.

2.1. Proceso de desarrollo basado en un modelo híbrido

Durante el desarrollo del TFG anterior se utilizó un proceso basado en la combinación de dos modelos de proceso de desarrollo que fueron un proceso en cascada, en las primeras fases del proyecto, y un proceso ágil basado en el marco de trabajo Scrum [3], para las fases posteriores al diseño correspondientes a la implementación y desarrollo de funcionalidades.

Sin embargo, en el desarrollo del presente TFG se ha optado por un modelo híbrido, combinando un enfoque iterativo para las fases iniciales (Fase Inicial y Fase de Ingeniería Inversa) con un enfoque ágil basado en el marco Scrum para las Fases de Nuevos Requisitos y para la Fase de Mejoras. El enfoque iterativo para las dos primeras fases sustituye al proceso en cascada debido a que el proyecto no empieza desde cero, sino que es una continuación. El modelo de proceso en cascada suele aplicarse en un desarrollo completamente nuevo con fases rígidas y secuenciales.

El proceso iterativo se utilizará durante las primeras fase del proyecto en las que se analizará lo entregado en el TFG anterior, comprobando y validando que no surgen errores que dificulten la inclusión de nuevas funcionalidades o mejoras a lo largo del nuevo proyecto. Además, se utilizará durante la Fase de Ingeniería Inversa, durante la cual se debe obtener información para poder llevar a cabo las siguientes fases.

Por otro lado, una vez finalizado el momento de revisión y análisis de la información, se llevarán a cabo dos etapas diferenciadas, las cuales se realizarán mediante un proceso de desarrollo ágil basado en Scrum. Mediante estos procesos se procederá con la implementación

y el desarrollo de las nuevas funcionalidades en una primera fase, y con modificaciones y mejoras, en la última fase.

Al igual que en el TFG de partida, el equipo estará compuesto únicamente por el estudiante y la tutora, manteniendo así las concesiones [6] y adaptaciones [2] realizadas sobre el modelo general propuesto por dicho marco de trabajo.

Con respecto a las adaptaciones mencionadas, dejan la asignación de roles de la siguiente manera. El estudiante toma varios papeles que según el marco Scrum deberían estar separados en varios individuos. EL rol de Product Owner (PO) será la tutora, quien representará los intereses de los stakeholders. El rol de Equipo de Desarrollo (DT) corresponderá al estudiante, siendo el único integrante del equipo de desarrollo. Por último, y para no sobrecargar a la tutora, el rol de Scrum Master (SM) será el estudiante.

2.1.1. Sprint

Un sprint [10] es un periodo breve en el que el equipo trabaja para completar una cantidad de trabajo establecida, siendo una herramienta importante para organizar procesos, eliminando cuellos de botella y evitando futuros problemas.

En un proyecto basado en Scrum se debe definir la duración del mismo y la cantidad de horas de trabajo que representa. El estudiante realizará, de manera simultánea a este proyecto, las prácticas en empresa, con una jornada de 8 horas. Por este motivo, la dedicación de Lunes a Sábado se pretende que sea de una media de 3 horas trabajo diario, reduciendo así la carga en el fin de semana. De esta forma tenemos un trabajo semanal de 18 horas de media.

Como es habitual en la estimación de la duración de un sprint, y de la misma manera que se realizó en el TFG anterior, su duración suele estar comprendida entre las 30 y las 40 horas. Por tanto, si se dedican 18 horas semanales, resulta ideal considerar que la duración de los sprints será de 2 semanas.

2.1.2. Ingeniería Inversa

Durante la segunda fase del TFG y aplicando metodología ágil basada en Scrum se lleva a cabo un proceso de ingeniería inversa sobre el TFG previo a este.

¿Por qué aplicar ingeniería inversa [5]? Al tratarse de una fase en la que vamos a trabajar sobre el producto resultante de otro TFG para realizar mejoras e incorporar nuevas funcionalidades sobre él, este tipo de proceso nos permite comprender el diseño, la arquitectura y el funcionamiento interno. Descomponiendo la aplicación y obteniendo la información necesaria, de forma detallada, se podrán llevar a cabo las modificaciones necesarias, mejorando así tanto la eficiencia, como la experiencia del usuario.

La ingeniería inversa cuenta con una serie de fases, las cuales llevaremos a cabo de manera personalizada:

- **Recopilación de información:** Al contar tanto con la memoria como con el código fuente, la recopilación de la información es más sencilla, facilitando el análisis posterior.
- **Análisis Estático:** Esta fase se centra en examinar el código sin ejecutarlo, ayudando a comprender la estructura del programa y permitiendo que a la hora de realizar las mejoras, el trabajo sea más sencillo.
- **Análisis Dinámico:** Ejecutando el código en un entorno controlado, se puede observar el comportamiento en tiempo real, entendiendo como responde el software.
- **Reconstrucción:** Finalmente, una vez analizado todo aquello que se ha obtenido en las fases previas, se documentan o reconstruyen aquello que sea necesario.

Con estas fases terminadas y con todo el conocimiento acerca del producto a mejorar, se puede comenzar con las dos fases de implementación, la de añadir nueva funcionalidad y la de mejoras a funcionalidad existente.

2.2. Stakeholders

Los stakeholders de un proyecto son aquellas personas, organizaciones o empresas que están interesadas y/o afectadas por dicho proyecto. Para este proyecto mantendremos los que se tuvieron en cuenta en el TFG anterior e incluiremos nuevos, quedando la lista de la siguiente manera:

1. **Estudiantes:** Comprende a todos los estudiantes, tanto matriculados como sin matricular, que tienen intención de comenzar su proyecto TFG. Dichos estudiantes tienen interés ya sea, en poder encontrar con mayor facilidad toda la información relativa a la asignatura del TFG, como en visualizar o solicitar los temas disponibles presentados tanto por empresas como por los profesores de forma ágil y sencilla.
2. **Tutores:** Este grupo recoge a aquellos profesores que ofertan o tutorizan proyectos a los estudiantes en el contexto del TFG, teniendo como principal interés en una plataforma actualizada que les permita realizar todas aquellas acciones que debe realizar un profesor con respecto a un TFG de manera eficiente.
3. **Coordinador de Título:** Esta persona es la encargada de la gestión de los TFG. Su principal interés es la automatización de todas aquellas acciones que hasta la fecha las realiza de forma manual.
4. **Subdirector de Relaciones con las Empresas:** Este rol es aquel que se encarga de la publicación de las ofertas de TFG con convenios en las empresas asociadas con la Escuela de Ingeniería Informática. Hasta la fecha, dichas ofertas se realizan de manera manual, por lo que una automatización de estos procesos es su principal interés.
5. **Técnicos de la Escuela:** Serán los encargados del mantenimiento de la aplicación una vez completado el TFG.

6. **STIC**:personal de STIC de la UVa a cargo del SSO (Single Sign On). Todas las aplicaciones que se desplieguen en la UVa y requieran identificarse deben seguir la política establecida de “una persona-una identidad” y conectarse con el SSO de la UVa. El personal a cargo de esto debe aceptar los datos que se soliciten desde esta aplicación, autorizar la máquina en la que se despliegue esta aplicación y suministrar en Sandbox (SSO en pruebas)

2.3. Backlog inicial

Durante la fase de ingeniería inversa se llevan a cabo una serie de entrevistas con los stakeholders. En estas entrevistas se buscaba obtener información relevante de posibles mejoras o nuevas funcionalidades que añadir durante las siguientes fases del proyecto. Dichas entrevistas fueron realizadas al actual Coordinador del Comité de Título y al Subdirector de Relaciones con las Empresas, para la obtención de información acerca de las acciones que llevan a cabo ambos roles y cómo interactúan con los demás stakeholders.

Para la clasificación y división de toda la información recopilada en las entrevistas se decide optar por la definición de épicas. Una épica [4] es una historia de usuario que por su gran tamaño, el equipo descompone en historias con un tamaño más adecuado para ser gestionada con los principios y técnicas ágiles: estimación y seguimiento cercano (normalmente diario). En el caso del desarrollo de software, las épicas suelen tener un formato identificable como es el siguiente: se establece lo principal, y por tanto lo primero, quién es el interesado de la acción correspondiente a la épica, seguido de qué es lo que quiere dicho rol y por último, para qué lo quiere.

A la hora de definir las épicas hay que tener en cuenta quiénes de los stakeholders definidos anteriormente serán los usuarios de la aplicación. De entre los interesados, los usuarios de la aplicación serán los siguientes: los estudiantes, los profesores de la universidad que tutorizan los TFG, y que denotaremos por “tutores UVa”, el Coordinador de Título quien podrá ser representado en la aplicación por otros miembros del Comité de Título manteniendo el mismo rol. De igual manera que el Coordinador de Título y considerándose generalidad, el Subdirector de Relaciones con Empresas, que será denominado como “Delegado de Empresas”, podrá ser representado por otros miembros de la universidad en quienes delegue, manteniendo el mismo rol.

EP1 - Como *tutor UVa*, quiero ofertar TFGs para que los estudiantes puedan consultarlas y valorar diferentes opciones para la realización de su TFG.

EP2 - Como *tutor UVa*, quiero poder asignar un TFG de los ofertados a un estudiante, cerrar dicha oferta y comunicar la asignación al comité de título para que el estudiante pueda realizar el TFG con la aprobación del Comité de Título.

EP3 - Como *tutor UVa*, quiero autopropornerme para tutorizar un TFG ofertado en convenio con una empresa, para facilitar una posible asignación de dicho TFG a un estudiante con el tutor académico necesario.

- EP4** - Como *coordinador de título*, quiero poder asignar a un estudiante un TFG ofertado en convenio bajo la tutorización de un tutor UVa y la de un tutor de dicha empresa para cumplir la normativa de TFG.
- EP5** - Como *estudiante* quiero poder solicitar un TFG propuesto tanto por una empresa como por un tutorUVa para llevar a cabo la asignatura.
- EP6** - Como *delegado de empresa* quiero dar a conocer los TFGs propuestos por las empresas y seleccionar a un estudiante para realizar alguno de estos TFGs cuando no se ha encontrado tutor UVa, para que el Comité de título colabore con la búsqueda de tutor UVa y el estudiante pueda realizar su TFG en convenio con la empresa.
- EP7** - Como *coordinador de título* quiero poder visualizar aquellos TFGs en los cuales los profesores se han autopropuesto para, una vez seleccionado un estudiante en uno de esos TFGs, poder asignárselo a un tutor UVa.

A partir de las Épicas definidas podemos establecer un backlog inicial en la Tabla 2.1 para la planificación y seguimiento del proyecto.

ID	Epica	Nombre
B01	EP01	Creación de oferta de TFG
B02	EP02	Modificación del CU creación de adjudicación
B03	EP03	Autopropuesta para tutorizar TFG
B04	EP04	Creación de adjudicación de TFG en convenio
B05	EP05	Creación de solicitud de TFG
B06	EP06	Creación de oferta de TFG en convenio
B07	EP06	Selección de estudiante para TFG en convenio
B08	EP07	Visualizar autopropuestas de tutorización de TFG

Tabla 2.1: Backlog inicial del proyecto

2.4. Análisis de riesgos

En la Tabla 2.2 se presentan los riesgos identificados para el desarrollo del TFG. Asimismo, con el fin de evaluar la necesidad de medidas preventivas y de contención, se presentará en la Tabla 2.3 para identificar aquellos que requieren de una especial atención de los riesgos [1]. Con un enfoque especial en aquellos con exposición media o alta. Con la anterior información, en la Tabla 2.4 se describen los planes de mitigación y contingencia adoptados luego de la identificación de los riesgos. Entendiendo por plan de mitigación, un conjunto de acciones que busca la disminución de la probabilidad de ocurrencia de un riesgo y como plan de contingencia la acciones que se realizan si, y solo si, el riesgo sucede.

ID	Nombre	Descripción
R01	Inadecuada gestión de requisitos	La variabilidad de los requisitos puede provocar una gestión inadecuada implicando una falta de claridad o suponer la modificación del trabajo realizado.
R02	Comunicación y colaboración	Podrían surgir problemas imprevistos durante el proyecto, como situaciones en las que no pueda comunicarme con mi tutora, impidiendo continuar con mi trabajo de la forma planificada.
R03	Gestión de recursos	La ineficiencia en la planificación y asignación de recursos puede provocar retrasos, una sobrecarga laboral masiva y agotamiento del equipo.
R04	Dependencias externas	La dependencia de aquellos implicados en el proyecto de manera externa, como los responsables del protocolo CAS, en el que se basa en SSO de la UVa, para la identificación de usuarios, o como los involucrados, como el Delegado de Empresas, pueden restar fluidez en el desarrollo.
R05	Enfermedad	Durante el desarrollo del proyecto puedo caer enfermo, afectando tanto a mi disponibilidad como a mi productividad.
R06	Trabajo externo	De manera simultánea a la realización del proyecto, me encontraré realizando las prácticas, lo que podría afectar en mi productividad o el tiempo disponible.

Tabla 2.2: Riesgos del proyecto

ID	Probabilidad	Impacto	Exposición
R01	Alta	Baja	Medio
R02	Baja	Alta	Medio
R03	Medio	Medio	Alta
R04	Alta	Medio	Alta
R05	Baja	Baja	Baja
R06	Medio	Alta	Medio

Tabla 2.3: Exposición a los riesgos

ID	Plan de Contingencia	Plan de Mitigación
R01	Verificar los requisitos con el equipo y reintroducir prioridades si el cambio es significativo.	Monitorizar desde el principio un proceso de gestión de requisitos bien definido con control periódico.
R02	Establecer un canal de comunicación sustituto en caso de surgir problemas imprevistos.	Añadir reuniones periódicas y asegurar herramientas de colaboración accesibles para ambos.
R03	Redistribuir las tareas o los recursos si surgen retrasos críticos en la planificación.	Hacer uso del colchón de horas de sprints anteriores para rectificar el cálculo erróneo de planificación.
R04	Identificar soluciones alternativas o reducir la necesidad de secundarios si las dependencias externas fallan.	Mantener comunicación constante con los implicados externos.
R05	Riesgo ajeno al control que se maneja.	Recuperación de las horas perdidas. Planificar de nuevo o retrasar la fecha de finalización.
R06	Reorganizar horarios, estableciendo uno que permita compatibilizar ambas cosas, sin generar conflictos entre sí.	Establecer tareas prioritarias o poder optar a retrasar la fecha de finalización del proyecto.

Tabla 2.4: Medidas de contingencia y mitigación

2.5. Calendarización

El inicio del plan del proyecto se realiza con la calendarización inicial, según toda la información que se presentará y que se ha abordado a lo largo de esta sección. La Tabla 2.5 muestra la programación del trabajo, que comienza el 4 de febrero de 2025 con fecha de finalización a fecha 9 de junio de 2025.

Hito	Inicio	Fin	Comentario
Fase Inicial	04/02/2025	17/02/2025	
Reunión Semanal	Se realizan cada martes.		Se realizaron reuniones semanales para la planificación del proyecto, definición del trabajo previo, contextualización del TFG previo (tecnologías).
Fase Ingeniería Inversa	18/02/2025	03/03/2025	Análisis del TFG previo, ingeniería inversa de procesos. Definición de objetivos.
Fase Nuevos Requisitos	03/03/2025	14/04/2025	
Sprint 1	04/03/2025	17/03/2025	Desarrollo de la funcionalidad 1 (Ofertas).
Sprint 2	18/03/2025	31/03/2025	Desarrollo de la funcionalidad 2 (Solicitudes).
Sprint 3	01/04/2025	14/04/2025	Bugs/Retrospectiva y agregar algo pendiente.
Fase de Mejoras	15/04/2025	26/05/2025	
Sprint 4	15/04/2025	28/04/2025	Mejora 1 (por definir).
Sprint 5	29/04/2025	12/05/2025	Mejora 2 (por definir).
Sprint 6	13/05/2025	26/05/2025	Sprints Review, Retrospectiva y terminar si queda algo pendiente.
Cierre Proyecto	27/05/2025	09/06/2025	Este último periodo se enfocará en realizar retrospectiva y la documentación final de lo que esté pendiente.

Tabla 2.5: Calendarización del proyecto

2.6. Presupuesto

Para proceder con el cálculo del presupuesto simulado, se ha de tener en cuenta los factores a analizar como son, las herramientas utilizadas en desarrollo y despliegue o las licencias de las aplicaciones. Acerca de las licencias de las herramientas, en el ámbito académico son gratuitas. Con respecto al cálculo del coste humano tomamos como referencia el salario me-

dio de un desarrollador Junior en España que se sitúa en los 2000€/mes (12 pagas) [8] en jornada completa. El sueldo no solo recoge la función de desarrollador, si no que también recoge acciones como la planificación y la gestión del proyecto. En el caso que nos corresponde y continuando con lo mencionado en la Sección 2.1.1 estaremos al 45,0 % (18 horas semanales) de una jornada completa. Añadiendo sobre este coste el 30 % de Seguridad Social y considerando que la duración del proyecto es de 300 horas, calculamos el coste personal (CP) mediante la siguiente formula.

$$CP = \frac{2000 \frac{\text{€}}{\text{mes}} * 0,45 * 300h * 1,3}{68 \frac{h}{\text{mes}}} = 5161,76\text{€} \quad (2.1)$$

Además del coste personal hay que tener en cuenta el coste de materiales (CH), para lo cual, tenemos en cuenta nuestro propio equipo, que es el utilizado para el desarrollo del proyecto. El equipo utilizado es un portátil HP Notebook equipado con un procesador Intel Core i5-5200U, 8GB de memoria RAM y 250GB de almacenamiento, corre un Ubuntu 22.04 y fue adquirido en 2022 por 700€. Utilizando las tablas de amortización de la Agencia Tributaria [12], se establece como coeficiente de amortización lineal máximo del 26 % para bienes de este tipo con un periodo máximo de amortización de 10 años.

$$CH = \frac{700 * 0,26 \frac{\text{amor}}{\text{año}} * 4,19\text{meses}}{12 \frac{\text{mes}}{\text{año}}} = 63,54\text{€} \quad (2.2)$$

Juntando todo lo calculado previamente podemos desglosar el presupuesto en la siguiente tabla, suponiendo un coste total simulado de 5 225.31€.

Concepto	Importe (€)
Personal	5 161.76
Software	0
Hardware	63.54
Total	5 225.31

Tabla 2.6: Presupuesto simulado

Capítulo 3

Análisis

Antes de que empiece el trabajo de desarrollo, es necesario realizar una fase de Análisis del problema con el fin de determinar exactamente lo que pertenece al dominio del problema y cuáles son las necesidades que se pretenden cubrir a partir de lo que ya se tiene del TFG previo. Este análisis es el resultado tanto de la Fase de Ingeniería Inversa como de las primeras tareas que se realizan tanto en la Fase de Nuevos Requisitos como en la Fase de Mejoras. En este Capítulo se documenta el resultado conjunto.

Dos de las actividades que forman parte de la primera etapa en esta fase de análisis han sido realizadas a la vez: modelado de dominios y modelado de procesos.

3.1. Modelo de Procesos

Además de los procesos representados en los diagramas UML del TFG previo, en la gestión de TFG hay más procesos a automatizar o simplificar. Estos comprenden la creación y publicación de propuestas de TFG, tanto de Tutores UVa, como las propuestas de convenio de las empresas, y todo lo que eso requiere. Para la validación y definición de dichos procesos se ha llevado a cabo una reunión con el Coordinador de Título y el Delegado de Empresas, con la que se ha obtenido toda la información necesaria para llevarlo a cabo. En las Figuras 3.1, 3.2 y 3.3 se muestran los diagramas resultantes de analizar la información obtenida de esa entrevista, en los que se pueden apreciar los nuevos flujos que siguen los roles dentro de la aplicación.

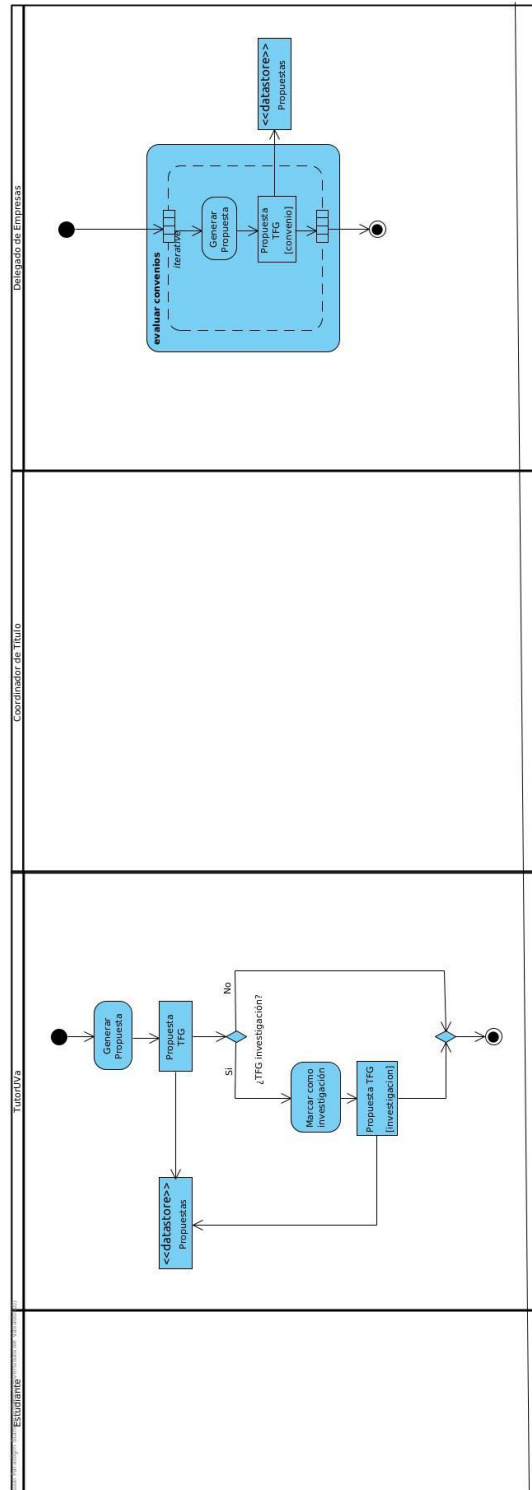


Figura 3.1: Fragmento del proceso centrado en la creación de propuesta de TFG

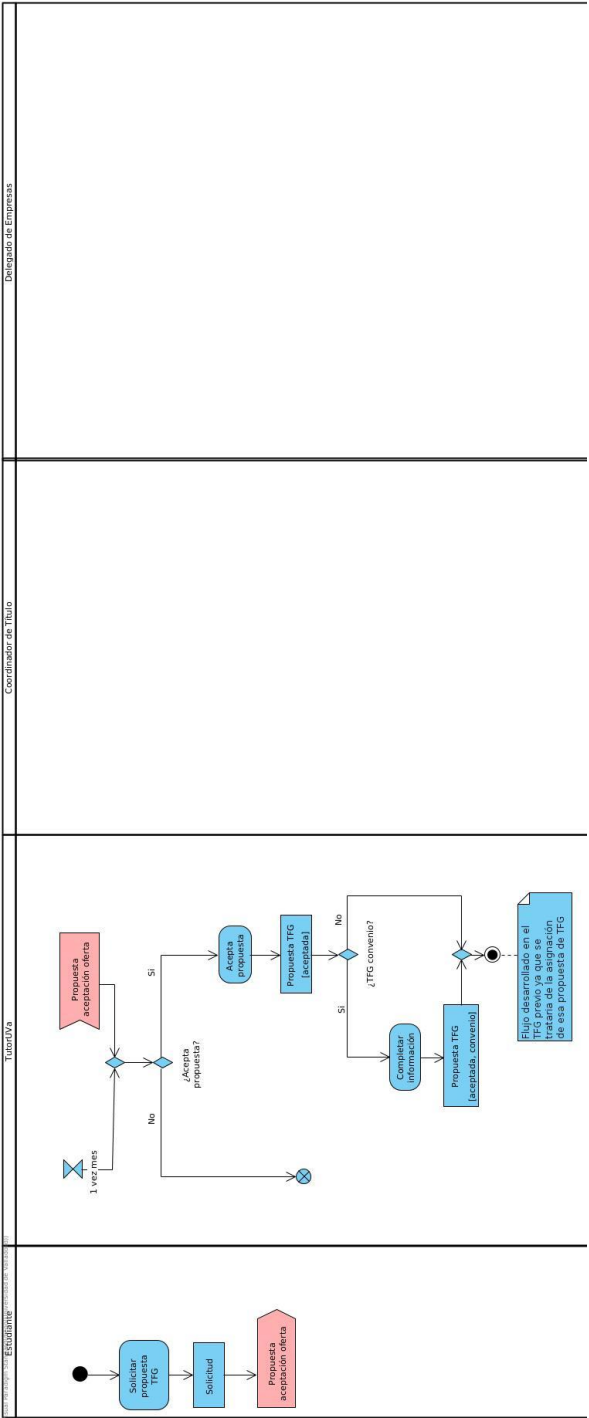


Figura 3.2: Fragmento del proceso centrado en la solicitud de propuesta ofertada

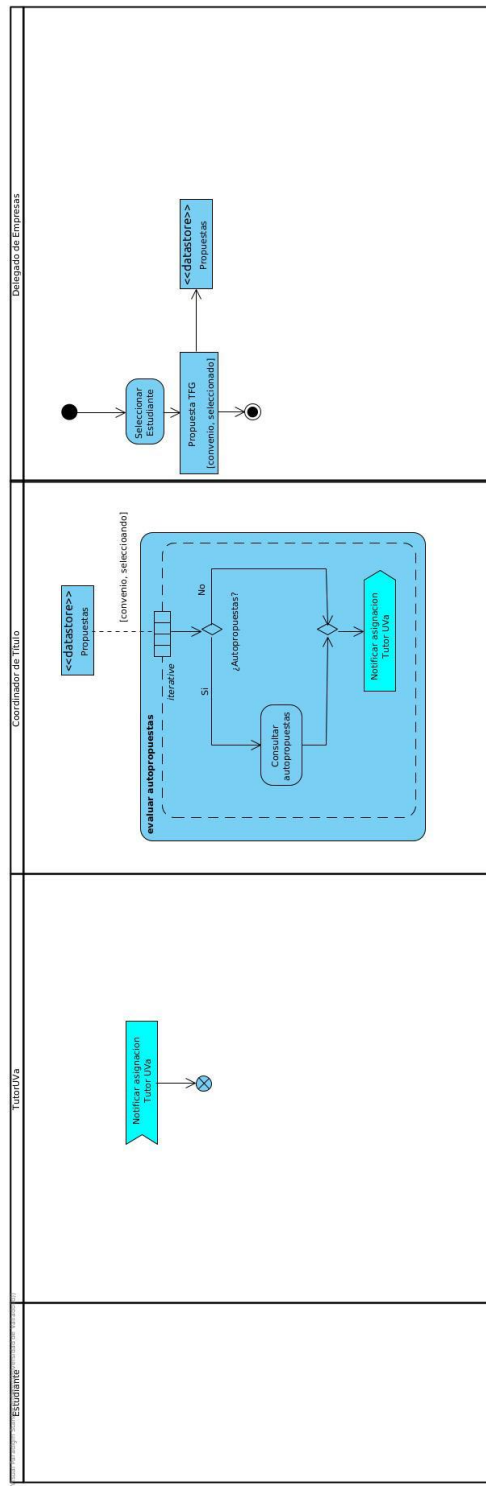


Figura 3.3: Fragmento del proceso centrado en la autopropuesta y selección de estudiante en TFG en convenio

3.2. Modelado del dominio

La modelización del dominio parte de un punto bastante avanzado como es toda la parte previa del TFG anterior. Para esta fase del problema había que llevar a cabo una serie de cambios sobre ese modelo ya que se incorporaba información nueva.

Entre las propuestas de TFG podemos diferenciar dos tipos: la propuesta de un tutor de la universidad, que ya actúa como tutor académico y, el otro tipo de propuesta, es la de TFG en convenio con las empresas, en la que este, cuenta con un tutor externo dentro de la empresa que oferta dicho TFG.

Para hacer una diferenciación clara del nuevo análisis sobre el modelo de dominio del TFG previo se han representado los cambios con un color diferente. En amarillo las nuevas entidades, en azul las existentes y en verde las modificadas 3.4.

3.3. Casos de Uso

En la Figura 3.5 se muestran los principales casos de uso del sistema, tanto los que había inicialmente del TFG anterior, en azul, los nuevos, en amarillo y aquellos que se han visto modificados en verde. También remarcar que se han añadido casos de uso que se desarrollaron pero no quedaron representados en el diagrama.

Además, en las Tablas de la 3.1 a la 3.10 corresponden a las descripciones detalladas de cada uno de los casos de uso expuestos en el diagrama. Esta descripción se centra en el flujo más principal, incluyendo también flujos alternativos.

En los siguientes casos de uso se tratan algunas de las transiciones de estados permitidas dentro de las entidades. De igual manera a los otros diagramas, en la Figura 3.7 se pueden apreciar los nuevos estados añadidos con respecto a las Propuestas de TFG, con la continuación que tendrían con las Adjudicaciones (Figura 3.6, viéndose así la línea de vida.

CU01	Solicitar propuesta de TFG
Actor	Estudiante
Descripción	El actor solicita en la plataforma que se le asigne un TFG propuesto.
Precondiciones	
1	El actor está identificado en el sistema.
2	El actor tiene el rol de Estudiante.
3	El estudiante no posee otros TFG en los estados asignado, adjudicado, pendiente de defensa o finalizado.
Flujo Principal	
1	El actor inicia la solicitud de asignación de TFG.
2	SISTEMA muestra las propuestas de TFG disponibles.
3	El actor selecciona el TFG a solicitar.
4	SISTEMA solicita confirmación para publicar la solicitud de propuesta.
5	El actor confirma la solicitud de la propuesta.
6	SISTEMA publica la solicitud.
El caso de uso termina.	
Flujos Alternativos	
5.1	Si el actor no confirma, el caso de uso se cancela y queda sin efecto.
*	El actor puede cancelar el caso de uso en cualquiera de sus pasos dejándolo sin efecto.

Tabla 3.1: CU01: Solicitar propuesta de TFG

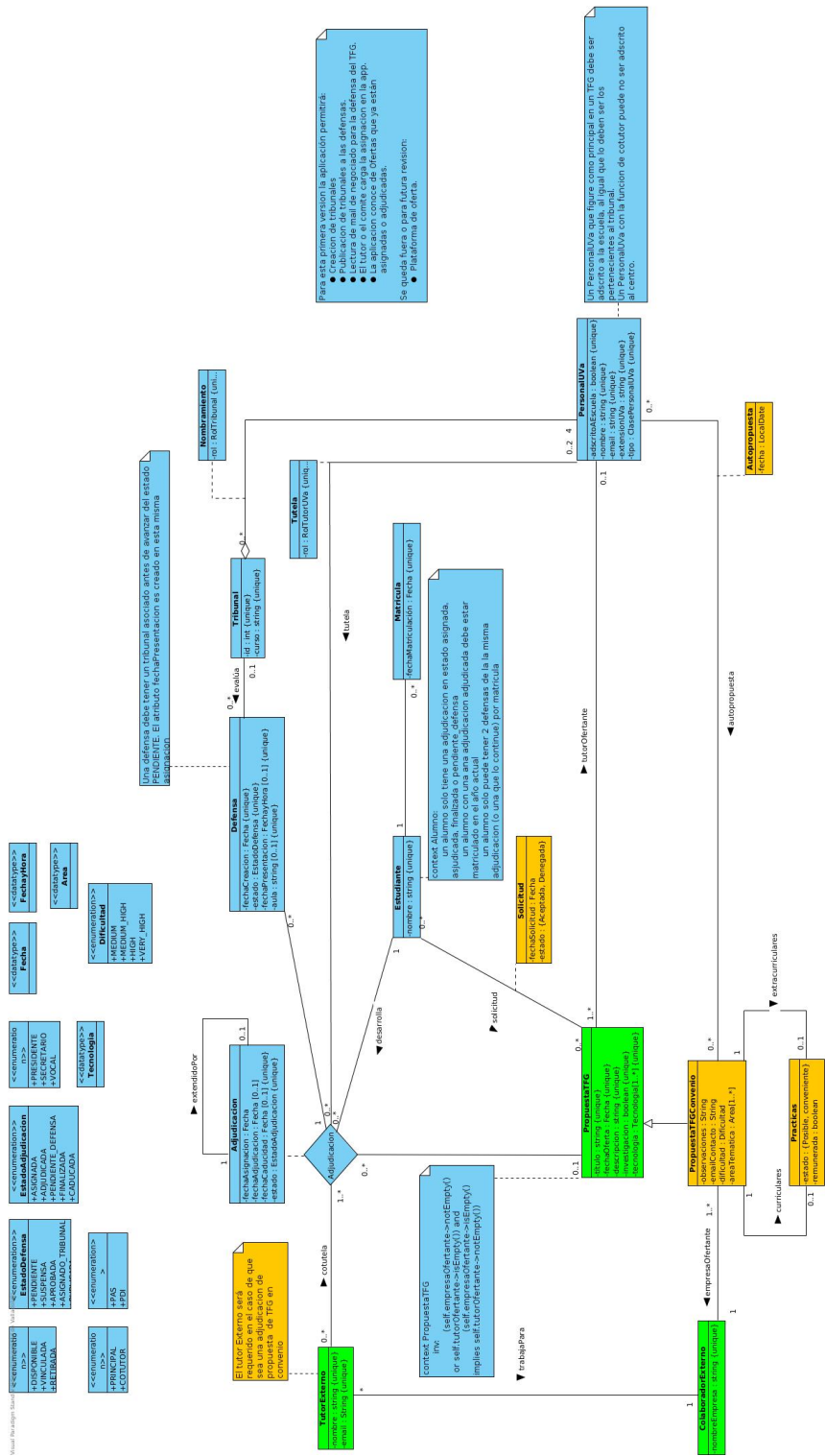


Figura 3.4: Modelo del dominio

3.3. CASOS DE USO

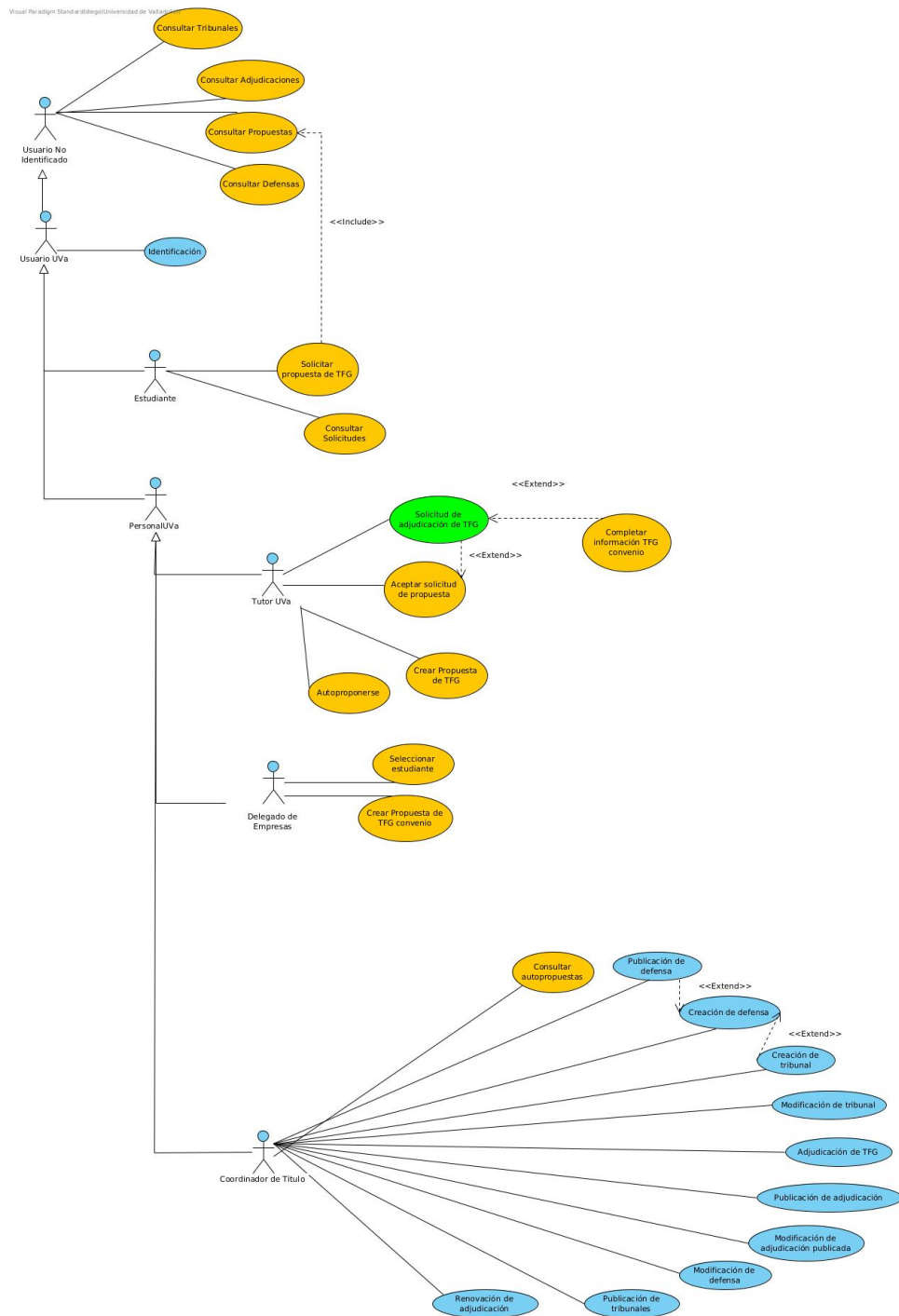


Figura 3.5: Casos de Uso

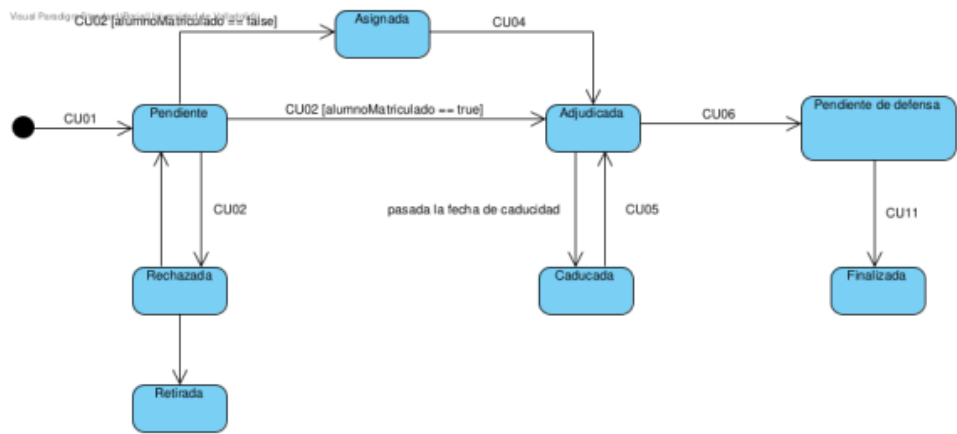


Figura 3.6: Diagrama de estados de las propuestas según el TFG anterior

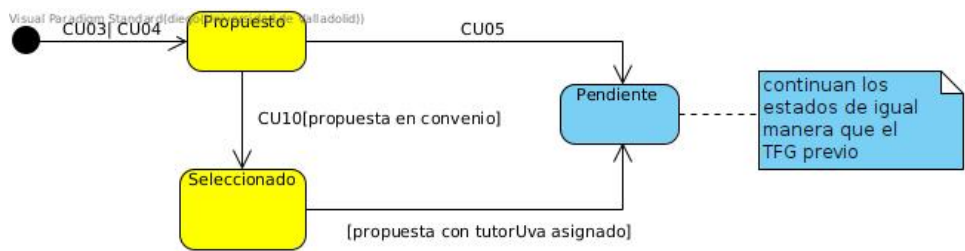


Figura 3.7: Diagrama de estados de las propuestas

CU02	Aceptar solicitud propuesta de TFG
Actor	Tutor UVa
Descripción	Ante una solicitud de un estudiante, el actor toma la decisión de aceptar o no la tutorización del TFG solicitado.
Precondiciones	
1	El actor está identificado en el sistema.
2	El actor tiene el rol de Tutor UVa.
Flujo Principal	
1	SISTEMA muestra la solicitud de propuesta de TFG.
2	SISTEMA solicita confirmación para publicar la solicitud de propuesta.
3	El actor confirma la aceptación de la solicitud de propuesta.
4	SISTEMA deja de hacer visible la propuesta.
5	SISTEMA continúa en el caso de uso 05.
El caso de uso termina.	
Flujos Alternativos	
3.1	Si el actor no confirma, se notifica al estudiante, el caso de uso se cancela y queda sin efecto.
*	El actor puede cancelar el caso de uso en cualquiera de sus pasos dejándolo sin efecto.

Tabla 3.2: CU02: Aceptar solicitud propuesta de TFG

CU03	Crear Propuesta de TFG
Actor	Tutor UVa
Descripción	El actor quiere publicar una propuesta de TFG para tutorizar.
Precondiciones	
1	El actor está identificado en el sistema.
Flujo Principal	
1	El actor solicita la creación de una nueva propuesta de TFG.
2	SISTEMA solicita al actor el título y la descripción.
3	El actor introduce el título y la descripción.
4	SISTEMA solicita al actor confirmación de si se trata de un TFG de investigación.
5	El actor introduce la confirmación de que no se trata de un TFG de investigación.
6	SISTEMA muestra el título, la descripción y la confirmación del TFG de investigación y solicita confirmación para publicar la propuesta.
7	El actor confirma la publicación de la propuesta.
8	SISTEMA publica la propuesta.
El caso de uso termina.	
Flujos Alternativos	
5.1	Si el actor confirma que se trata de un TFG de investigación, SISTEMA añade la confirmación y continúa en el paso 7.
7.1	Si el actor no confirma, el caso de uso se cancela y queda sin efecto.
*	El actor puede cancelar el caso de uso en cualquiera de sus pasos dejándolo sin efecto.

Tabla 3.3: CU03: Crear Propuesta de TFG

CU04	Crear Propuesta de TFG en convenio
Actor	Delegado de Empresas
Descripción	El actor quiere dar a conocer las ofertas de TFG que proponen las empresas.
Precondiciones	
1	El actor está identificado en el sistema.
2	El actor tiene el rol de Delegado de Empresas.
Flujo Principal	
1	El actor solicita la creación de una nueva propuesta de TFG en convenio.
2	SISTEMA solicita al actor el título y la descripción.
3	El actor introduce el título y la descripción.
4	SISTEMA solicita al actor las observaciones, el email de contacto, las tecnologías a utilizar, la dificultad y las áreas temáticas.
5	SISTEMA comprueba que la información introducida es sintácticamente correcta.
6	SISTEMA comprueba que las tecnologías y las áreas temáticas existen en el sistema.
7	SISTEMA muestra la información introducida previamente y solicita confirmación para publicar la propuesta.
8	El actor confirma la publicación de la propuesta.
9	SISTEMA publica la propuesta.
El caso de uso termina.	
Flujos Alternativos	
7.1	Si las áreas temáticas no están en el sistema, se añaden y se continúa el caso de uso en el paso 7.
7.2	Si las tecnologías no están en el sistema, se añaden y se continúa el caso de uso en el paso 7.
8.1	Si el actor no confirma, el caso de uso se cancela y queda sin efecto.
*	El actor puede cancelar el caso de uso en cualquiera de sus pasos dejándolo sin efecto.

Tabla 3.4: CU04: Crear Propuesta de TFG en convenio

CU05	Solicitud de adjudicación de TFG (modificación)
Actor	Tutor UVa
Descripción	El actor crea una nueva propuesta de tutoría de TFG.
Precondiciones	
1	El actor está identificado en el sistema.
2	El actor tiene el rol de Tutor UVa.
Flujo Principal	
1	El actor solicita la creación de una nueva adjudicación de TFG.
2	SISTEMA comprueba si se trata de la continuación del CU02 y solicita al actor el título y la descripción.
3	El actor introduce el título y la descripción.
4	SISTEMA solicita la información del estudiantes.
5	El actor introduce el estudiante.
6	SISTEMA solicita confirmación de si se trata de un TFG en convenio.
7	El actor confirma que se trata de un TFG de convenio.(El caso de uso continua en el CU06)
8	SISTEMA muestra la información aportada y solicita confirmación al actor para enviar la solicitud.
9	El actor confirma el envío de la solicitud.
10	SISTEMA almacena la solicitud.
El caso de uso termina.	
Flujos Alternativos	
2.1	Si el flujo proviene del CU02, el SISTEMA obtiene los datos directamente de la propuesta.
4.1	Si hay estudiante seleccionado, el sistema solicita no lo solicita.
5.1	Si el actor no confirma, el caso de uso se cancela y queda sin efecto.
*	El actor puede cancelar el caso de uso en cualquiera de sus pasos dejándolo sin efecto.

Tabla 3.5: CU05: Solicitud de adjudicación de TFG (modificación)

CU06	Completar información del TFG en convenio
Actor	Tutor UVa
Descripción	El actor completa la información adicional necesaria en un TFG en convenio.
Precondiciones	
1	El actor está identificado en el sistema.
2	El actor tiene el rol de Tutor UVa.
3	La propuesta de TFG corresponde con una en convenio.
Flujo Principal	
1	El actor solicita complementar la información acerca del TFG en convenio.
2	SISTEMA solicita el nombre del tutor externo y su email de contacto.
3	El actor introduce el nombre del tutor externo y su correspondiente email de contacto.
4	SISTEMA solicita confirmación para publicar la solicitud de adjudicación.
5	El actor confirma la solicitud de adjudicación.
6	SISTEMA publica la solicitud.
El caso de uso termina.	
Flujos Alternativos	
5.1	Si el actor no confirma, el caso de uso se cancela y queda sin efecto.
*	El actor puede cancelar el caso de uso en cualquiera de sus pasos dejándolo sin efecto.

Tabla 3.6: CU06: Completar información del TFG en convenio

CU07	Consultar autopropuestas
Actor	Coordinador de Título
Descripción	El actor consulta las autopropuestas sobre un TFG en convenio.
Precondiciones	
1	El actor está identificado en el sistema.
2	El actor tiene el rol de Coordinador de Título.
3	La propuesta de TFG corresponde con una en convenio.
Flujo Principal	
1	El actor solicita consultar las autopropuestas en un TFG en convenio.
2	SISTEMA muestra todas las autopropuestas de los tutores UVa.
El caso de uso termina.	
Flujos Alternativos	

Tabla 3.7: CU07: Consultar autopropuestas

CU08	Autoproponerse
Actor	Tutor UVa
Descripción	El actor se autopropone para tutorizar un TFG en convenio.
Precondiciones	
1	El actor está identificado en el sistema.
2	El actor tiene el rol de Tutor UVa.
3	La propuesta de TFG corresponde con una en convenio.
Flujo Principal	
1	El actor selecciona un TFG en convenio.
2	SISTEMA muestra la información acerca de la propuesta de TFG en convenio.
3	El actor solicita autoproponerse.
4	SISTEMA solicita confirmación para publicar la autopropuesta.
5	El actor confirma la autopropuesta.
6	SISTEMA publica la autopropuesta.
El caso de uso termina.	
Flujos Alternativos	
5.1	Si el actor no confirma, el caso de uso se cancela y queda sin efecto.
*	El actor puede cancelar el caso de uso en cualquiera de sus pasos dejándolo sin efecto.

Tabla 3.8: CU08: Autoproponerse

CU09	Consultar solicitudes
Actor	Estudiante
Descripción	El actor consulta solicitudes de propuesta de TFG enviadas.
Precondiciones	
1	El actor está identificado en el sistema.
2	El actor tiene el rol de Estudiante.
Flujo Principal	
1	El actor solicita consultar las solicitudes de propuestas de TFG.
2	SISTEMA muestra sus solicitudes con su estado.
El caso de uso termina.	
Flujos Alternativos	

Tabla 3.9: CU09: Consultar solicitudes

CU10	Seleccionar estudiante
Actor	Delegado de Empresas
Descripción	El actor solicita seleccionar a un Estudiante para un TFG en convenio.
Precondiciones	
1	El actor está identificado en el sistema.
2	El actor tiene el rol de Delegado de Empresas.
3	La propuesta de TFG corresponde con una en convenio.
4	La propuesta de no tiene un estudiante ya seleccionado.
5	El estudiante no posee otros TFG en los estados asignado, adjudicado, pendiente de defensa o finalizado.
Flujo Principal	
1	El actor selecciona un TFG en convenio.
2	SISTEMA muestra la información acerca de la propuesta de TFG en convenio.
3	El actor solicita seleccionar un estudiante.
4	SISTEMA solicita la información del estudiante.
5	El actor introduce la información del estudiante.
6	SISTEMA solicita confirmación para modificar la propuesta.
7	El actor introduce confirmación.
8	SISTEMA publica la propuesta pero dejando de ser visible para otros estudiantes.
El caso de uso termina.	
Flujos Alternativos	
7.1	Si el actor no confirma, el caso de uso se cancela y queda sin efecto.
*	El actor puede cancelar el caso de uso en cualquiera de sus pasos dejándolo sin efecto.

Tabla 3.10: CU10: Seleccionar Estudiante

Capítulo 4

Tecnologías utilizadas

En este Capítulo se presenta un resumen de las tecnologías utilizadas tanto para la gestión del proyecto, como para el desarrollo del mismo.

A modo de continuación del proyecto anterior, las tecnologías utilizadas que se mantienen, solamente se enumeran y quedan descritas en el TFG previo [2]. En los casos en los que se ha añadido el uso de alguna nueva tecnología, se describe brevemente la misma y para qué se utiliza en el proyecto.

4.1. Tecnologías de desarrollo

4.1.1. Tecnologías utilizadas para el frontend

En el caso de la aplicación web se ha mantenido la arquitectura y se decide mantener las tecnologías utilizadas. Estas tecnologías son:

- **React:** Se mantiene por ser una biblioteca consolidada además de adecuada para desarrollar interfaces de usuario dinámicas. Otra de las razones de mantenerlo es que su cambio supondría una refactorización de la aplicación web, impidiendo completar los objetivos principales.
- **Vite:** Aunque se mantiene como parte del entorno configurado en el TFG anterior, en este no se ha utilizado activamente durante el desarrollo. Sin embargo, no se ha eliminado, dado que no ha presentado conflictos ni impedimentos técnicos.
- **Yarn:** En el proyecto anterior se utilizaba *Yarn* como gestor de paquetes. En esta ocasión, se ha decidido utilizar *NPM* en su lugar, debido a la mayor familiaridad con este gestor y su integración más directa con el entorno de desarrollo utilizado.

- **TypeScript**: Se mantiene como lenguaje base para el frontend, ya que al estar ya presente en el proyecto anterior, su continuidad resulta natural y conveniente.

4.1.2. Tecnologías utilizadas para el backend

De igual manera que en el frontend, se utilizan la misma arquitectura y las mismas tecnologías para el desarrollo de la aplicación backend que las utilizadas en el TFG anterior.

Swagger

Se decide añadir al stack tecnológico el uso de Swagger [11]. Es un conjunto de herramientas y una especificación (OpenAPI Specification) cuyo objetivo es facilitar el trabajo con APIs RESTful, desde su diseño hasta su consumo. Permite crear una descripción estandarizada de la estructura y las funcionalidades de la API y, a través de herramientas asociadas como Swagger UI, generar documentación interactiva y visual a partir de dicha especificación. En el desarrollo se ha decidido utilizar con el fin de dejar de manera ordenada y clara los endpoints del proyecto y sus respectivos parámetros e información que utilizan cada uno de ellos.

4.1.3. Tecnologías utilizadas para otros fines

Al igual que en el proyecto anterior, se han utilizado otra serie de tecnologías aparte de las utilizadas en el propio desarrollo, como los IDE utilizados para cada una de las aplicaciones, o el sistema de control de versiones. Estas tecnologías mencionadas son las siguientes:

- **Postman**. Para llevar a cabo las pruebas sobre la aplicación backend como se especifica en el Capítulo 6
- **IntelliJ**. Es el IDE utilizado para el desarrollo del backend.
- **Visual Studio Code**. Es el IDE utilizado para el desarrollo del frontend.
- **Docker**. Herramienta utilizada para empaquetar las diferentes partes de proyecto, permitiendo además dividir el contenido en función de si se encuentra en desarrollo o en producción como se especifica en el manual del Anexo A.2.

4.2. Tecnologías de gestión

4.2.1. Clockify

Herramienta utilizada para llevar una gestión del tiempo a la hora de realizar las tareas premarcadas, permitiendo tener una diferenciación por sprints mediante las TAGs y poder

visualizar tanto las horas como el volumen de trabajo por sprints.

4.3. Tecnologías de documentación

Estas, al igual que en las secciones anteriores, se mantienen las mismas utilizadas por el TFG anterior. Siendo estas las siguientes:

- **LaTeX y Overleaf:** Redacción y maquetación del documento
- **Visual Paradigm:** Modelado y diseño con UML

Capítulo 5

Diseño

En esta sección nos centramos más en detalle en aquellas decisiones de diseño tomadas para llevar a cabo el proyecto, tanto en la aplicación web como en el servidor *backend*. No podemos perder de vista que este TFG se trata de una continuación y mejora de un TFG anterior.

Al igual que en el proyecto previo, este capítulo se divide en dos secciones independientes para las dos aplicaciones: la parte *frontend* y la parte *backend*.

5.1. Arquitectura Backend

A continuación, se describen aquellas decisiones de diseño tomadas respecto al servidor (*backend*).

El servidor consiste en una API RESTful, como se indica el proyecto anterior, en la cual se centra en desacoplar el cliente-servidor. Además de mantener y preservar las características fundamentales de una API REST, también hemos mantenido la arquitectura de capas implementada.

Durante el desarrollo del proyecto se han incluido una serie de endpoints nuevos con los que completar las funcionalidades, estos endpoints son los siguientes:

- */autoproposal*: Este recoge aquellas funcionalidades acerca de las autopropuestas de los tutores UVa.
- */proposalRequest*: En este endpoint se agrupan aquellas acciones que se permiten acerca de las solicitudes de las propuestas de un TFG (solicitar, aceptar, denegar, etc).
- Los complementarios acerca de una propuesta en convenio: Estos recogen la información adicional que conforman una propuesta en convenio.

Al igual que se han creado diferentes endpoints durante el desarrollo del proyecto, también se han modificado algunos en función de cómo variaba el contenido de la aplicación. Los endpoints modificados que se han visto afectados son los siguientes:

- */proposal*: De igual forma que en el TFG anterior recoge la funcionalidad acerca de las propuestas, aunque ahora incluye también las acciones referidas a las propuestas de TFGs en convenio con empresas.
- */adjudication*: Al modificarse las propuestas, de igual forma se ven afectadas las adjudicaciones ya que ambas entidades comparten una gran cantidad de información.
- */court*: Una de las modificaciones sobre este endpoint fue la variación de miembros suplentes de los tribunales para las defensas de los TFGs, lo que afectaba a la mayoría de acciones.

A modo complementario se tomó la decisión de hacer una documentación mediante la librería de Swagger 4.1.2. De esta manera se recogía en un mismo punto todos aquellos endpoints que conforman la API, además de la información relevante para su uso. En el siguiente enlace se puede acceder a dicha documentación.

En la arquitectura general del backend, basada en capas, se han añadido algunos paquetes que se muestran en la Figura 5.1. Al igual que se utilizó en el capítulo 3 de Análisis este diagrama cuenta con un código de colores para visualizar en este caso los paquetes que se mantienen de color azul, los añadidos en color amarillo y los paquetes en los que se ha introducido alguna modificación en color verde.

5.1.1. Arquitectura del sistema

De manera similar al punto en el que se continuó el proyecto, este disponía de un diagrama de despliegue para poner en funcionamiento los diferentes módulos de la aplicación. El diagrama de despliegue en producción se mantiene sin modificaciones. Pero en este proyecto se ha aportado una solución local que permite probar la aplicación desarrollada desacoplándose del CAS de la UVa como se muestra en el diagrama de despliegue de la Figura 5.2.

5.1.2. Identificación y autenticación

El sistema de identificación utilizado en el proyecto es el Inicio de Sesión Único (SSO, por las siglas en inglés), basado en el Servicio de Identificación Central (CAS) [2][sección 5.4.1] de la Universidad de Valladolid. Debido a esto, únicamente se permite la autenticación desde la máquina virtual proporcionada y donde reside el servidor, por lo que se tomó la decisión de saltarse el paso de la creación y validación de un ticket por parte del CAS durante el proceso de desarrollo, haciendo que la información del inicio de sesión sea mockeado desde la aplicación web.

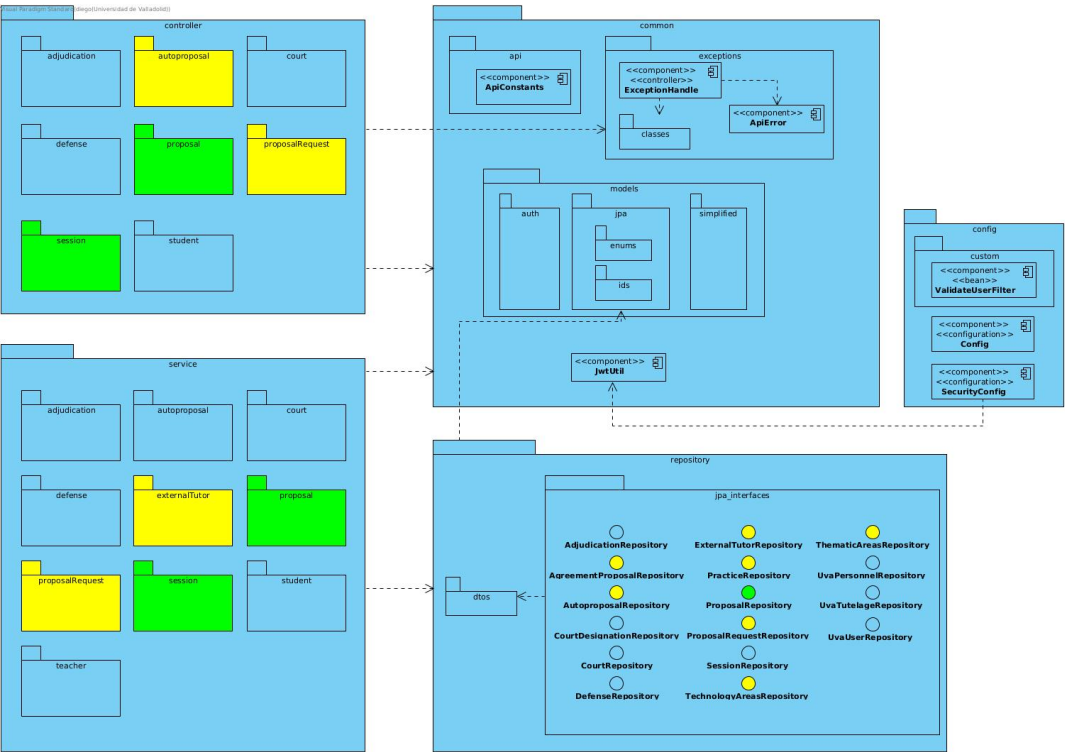


Figura 5.1: Diagrama de la arquitectura de paquetes en el backend

Además del uso del sistema de autenticación de la Universidad (CAS), el proyecto cuenta con rutas bloqueadas en función del rol que desempeña la acción. De igual manera que con la autenticación estos bloqueos se desactivaron durante el procesos de desarrollo para facilitar las pruebas de las acciones.

5.1.3. Propuestas en convenio

Una de las nuevas funcionalidades incluidas en el TFG corresponden a la creación de propuestas en convenio con empresas. Este cambio supone aumentar la complejidad con todo aquello que ya existía en relación con las propuestas. Para mantener una complejidad óptima se tomaron diversas decisiones que se expresan a continuación.

Diseño del flujo de creación de propuestas de convenio en la aplicación

Como se menciona previamente, el sistema permite al Delegado de empresas crear propuestas en convenio con empresas de tal manera que son una forma nueva de propuesta.

En la Figura 5.3, se especifica cómo es la interacción del usuario con la aplicación frontend

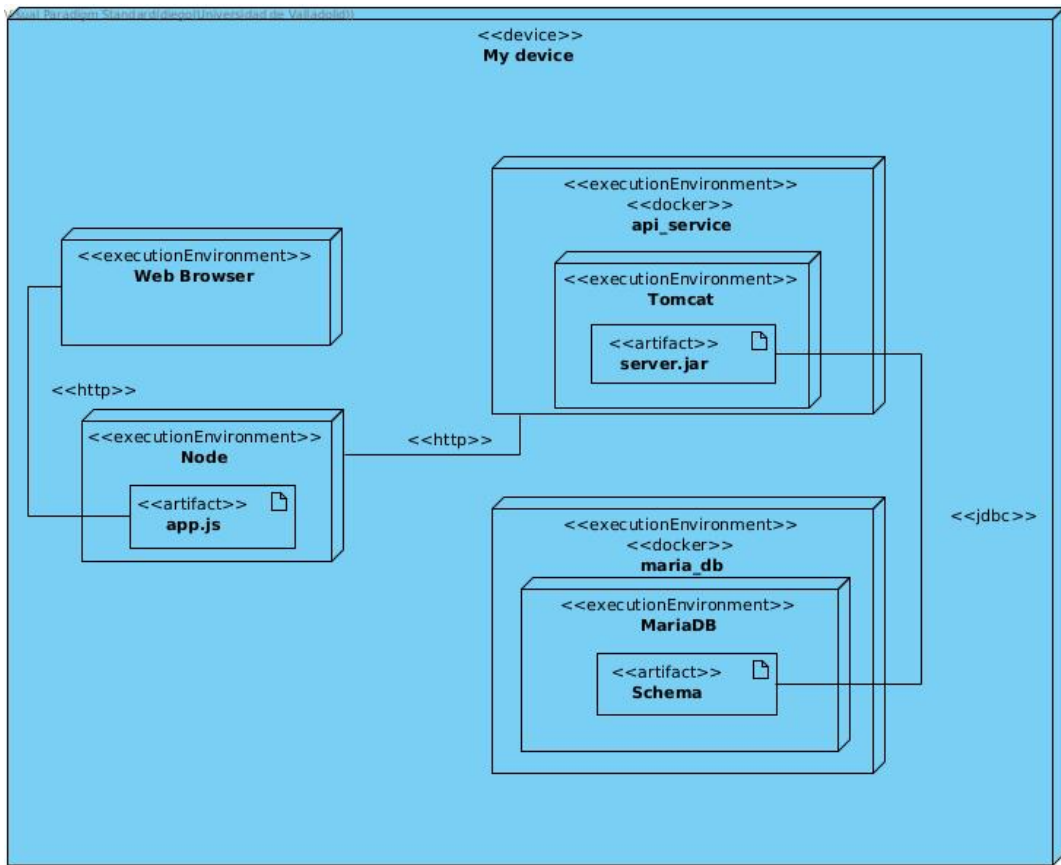


Figura 5.2: Diagrama de despliegue local, sin el uso del CAS

y una simplificación del backend en una sola entidad que recibe las peticiones, a la hora de llevar a cabo la acción de crear una propuesta de TFG en convenio.

1. El usuario, previamente identificado y con el rol de Delegado de empresas, procede a iniciar el proceso, lo que le lleva a una vista nueva que contiene todos aquellos campos que deberá rellenar para crear la propuesta.
2. La aplicación web validará que todos los campos sean correcto y enviará la información recogida a la aplicación backend, la cual lo verificará de nuevo la información antes de guardar en persistencia la propuesta en convenio.
3. Una vez verificada la información enviada desde la aplicación web, el backend creará la propuesta, guardándola en persistencia y haciéndola publica para mostrarla en la aplicación web.
4. Una vez creada y publicada por el backend la aplicación web mostrará la lista de propuestas incluyendo la creada previamente.

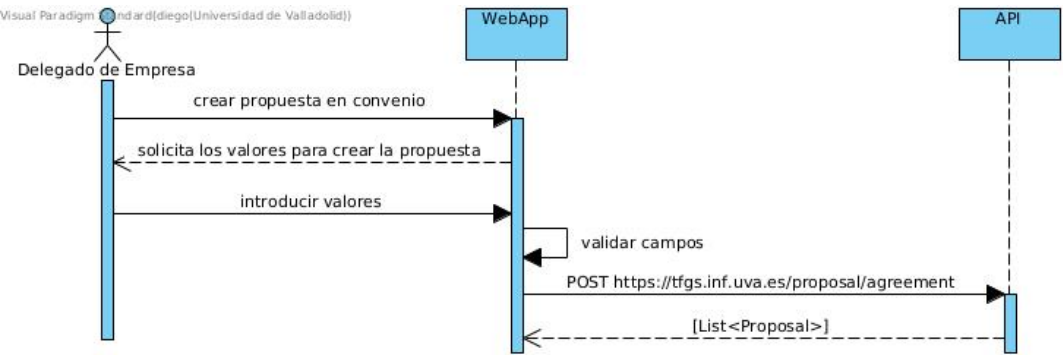


Figura 5.3: Flujo principal para la creación de una propuesta de TFG en convenio

De manera más detallada, en la aplicación backend, se llevan acabo una serie de comprobaciones y acciones diferentes, ya que se trata de un flujo más complejo que el de una propuesta normal, como se puede apreciar en la Figura 5.4

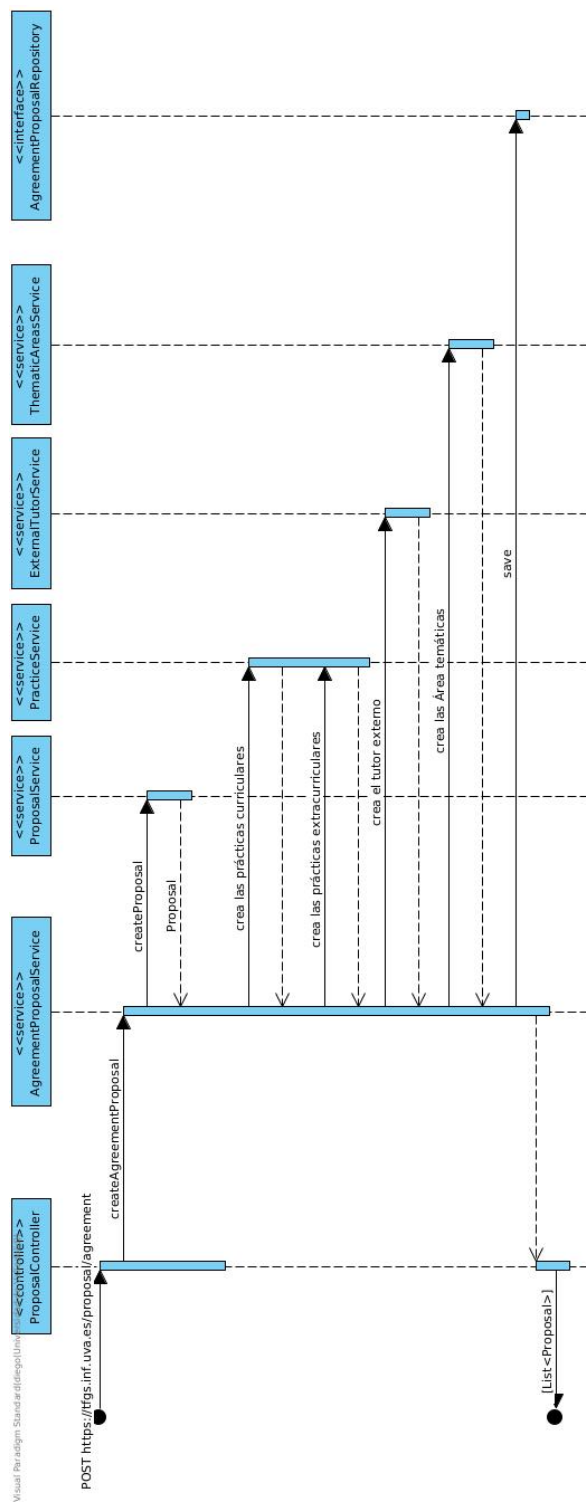


Figura 5.4: Flujo detallado de las transacciones del backend

5.2. Arquitectura Frontend

Para esta segunda sección vamos a tratar aquellas decisiones relativas a la aplicación web que se han tomado a lo largo del proyecto.

Partiendo de la premisa establecida anteriormente, donde se reconoció la flexibilidad de React en cuanto a la estructura de proyectos y su énfasis en la reutilización de componentes, se optó por un modelo organizativo. La elección recayó en una estructura por tipos de elementos de React, valorando su adaptabilidad para la reutilización de funcionalidades frente a una organización por casos de uso. En consecuencia, y en línea con lo adelantado, esta estructura, fundamentada en los elementos definidos de React, es la que posibilita establecer la relación identificada entre la arquitectura de nuestra aplicación y el patrón arquitectónico MVVM.

5.2.1. Estructura del proyecto

A modo de mantener la continuidad con el TFG anterior, se mantiene la estructura y la arquitectura previa, únicamente se añaden los nuevos directorios para ampliar las funcionalidades y se modifica el contenido permitiendo mejorar las acciones.

La arquitectura utilizada en el proyecto anterior, y que mantenemos en este, es la denominada MVVM, es cual es un patrón de arquitectura de software creado con el objetivo de separar la lógica de aplicación y de negocio de la interfaz de la aplicación (UI) [9], como se puede apreciar en la Figura 5.5.

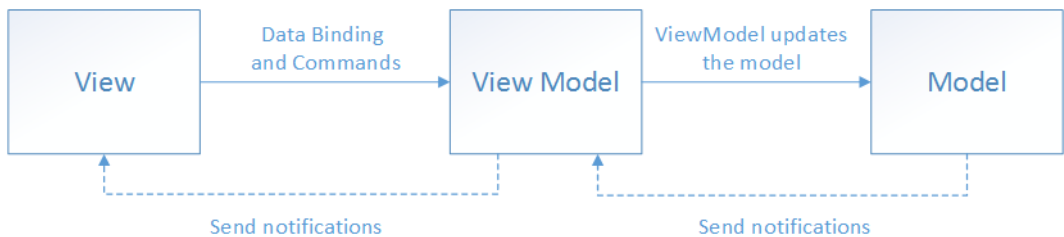


Figura 5.5: Arquitectura MVVM tomada de Modelo-vista-modelo [9]

5.2.2. Listado de componentes nuevos

Con las nuevas mejoras incorporadas en la aplicación web se debían incorporar y modificar algunos de los componentes ya existentes. Los componentes nuevos son los mostrados a continuación:

proposalRequest : este componente agrupa aquellas funcionalidades y vistas relacionadas con las solicitudes de las propuestas de TFG. Este recoge tanto la vista personal del estudiante, como la funcionalidad utilizada sobre el listado de propuestas para aceptar o denegar dichas solicitudes.

autoproposal : Al igual que el componente anterior, recoge la vista utilizada, en este caso por los tutores UVa, para ver sus autopropuestas y la lista por cada propuesta en la que el Coordinador de Título puede visualizarlas.

5.2.3. Listado de componente modificados

La gran mayoría de las modificaciones y nuevas funcionalidades requieren de cambios sobre componentes existentes. Estas modificaciones en algunos de ellos han supuesto una adaptación completa, como en el caso de los tribunales. En otros solo han sido pequeñas modificaciones o incluir nuevos campos como se muestra a continuación.

proposal : Este componente recoge las principales modificaciones al incluir el tipo nuevo de propuesta de TFG. Esto ha supuesto la creación de vistas propias y la adaptación de las existentes para poder mostrar toda la información en función del tipo de propuesta.

course : La adaptación de este componente ha supuesto la creación uno nuevo debido al cambio de funcionalidad sugerido por el Coordinador de Título.

adjudication : Las modificaciones que ha sufrido este componente son simples incorporaciones de filtros sobre la vista que contiene los listados y modificación a recargas asíncronas para reducir el volumen de carga en cada acción realizada.

Capítulo 6

Implementación y pruebas

La fase de pruebas es la etapa decisiva del desarrollo de software que le permite confirmar la funcionalidad, confiabilidad y seguridad del sistema tras su implementación. En este proyecto, se siguió el enfoque de caja negra, que consiste en verificar el comportamiento del sistema sin un conocimiento preliminar de su implementación.

Para una mayor claridad a la hora de realizar futuras pruebas y tener un mayor conocimiento de los endpoints proporcionados por el servidor backend se dispone de un archivo swagger con el que visualizar los endpoints de manera más clara.

Para lograrlo, se implementó una metodología de testeo principal basada en Postman, con el cual se logra testear la API REST descrita implementando pruebas básicas para asegurarse de que los distintos endpoints respondan correctamente ante diferentes peticiones.

A continuación, se detallarán las pruebas realizadas en las microimplementaciones del proyecto desarrollado.

6.1. Pruebas de la API con Postman

Con el fin de asegurarse de que se lleve a cabo una comunicación adecuada entre el frontend y el backend, se llevaron a cabo pruebas de API a través de Postman. Cada uno de los endpoints utilizados en la API REST fue testeado en términos de:

- Respuestas esperadas: GET, POST, PUT y DELETE requests enviadas resultando en obtener los mismos datos.
- Códigos de estado HTTP: la API devuelve los códigos correctos y adecuados 200 OK, 201 Created, 400 Bad Request, 404 Not Found, 500 Internal Server Error, etc.
- Formato de respuesta y estructura: la API devuelve los datos en formato JSON correctamente formateados.

6.1.1. Autenticación

Uno de los puntos fundamentales de la API es su división de funcionalidad en función del rol del interesado. Sin embargo, durante el proceso de desarrollo no es viable el uso del sistema de autenticación de la UVa para cada una de las consultas. Para poder desarrollar las funcionalidades utilizando el token de sesión, desarrollé un endpoint con el que saltarse el paso de autenticación con el CAS de la UVa.

En el siguiente enlace aparecen una serie de llamadas a la API con las que validar tanto el ticket del CAS como para autenticarse y obtener el token con el que hacer las peticiones.

NOTA *En la llamada obtener token si no se le pasa ningún parámetro el rol de la persona a autenticarse será Estudiante. Para modificar el email o el nombre habrá que hacerlo en el código del propio método.*

6.1.2. Propuestas

Como piedra angular de este segundo TFG, continuación del anterior, se centra en las funcionalidades acerca de las propuestas de TFG. Para probar las funcionalidades de los endpoints de las propuestas se optó por varias llamadas desde Postman con diferentes parámetros en el cuerpo de la llamada.

El siguiente enlace recoge los endpoints con las acciones principales, que en este caso son, crear las diferentes propuestas, obtener las propuestas y eliminar una propuesta. Esta última es una funcionalidad implementada para facilitar el desarrollo del proyecto.

- Obtener propuestas (GET): obtiene tanto la lista como una propuesta por id. En caso de la lista, filtra por email si la llamada contiene token de autenticación.
- Crear propuestas (POST): crea tanto propuestas normales como propuestas de convenio. Además hay una diferenciación en crearla con estudiante asociado o no. Incluye también una para probar la creación de múltiples propuestas en convenio, simulando el uso de creación mediante una plantilla Excel.

6.1.3. Solicitud de Propuestas

Una de las nuevas funcionalidades de este proyecto era la inclusión de acciones asociadas a uno de los nuevos stakeholders (Capítulo 2.2). El estudiante puede solicitar una propuesta de TFG publicada.

En el siguiente enlace incluyen tanto la obtención de las solicitudes, con diferentes filtros, como la creación y la aceptación o denegación de las solicitudes.

6.1.4. Autopropuestas

Con el objetivo de facilitar la búsqueda de tutores al Coordinador de la titulación cuando se tienen propuestas de TFGs en convenio, se optó por incorporar la funcionalidad necesaria para que un profesor de la universidad pueda autoproponerse para tutorizar un TFG en convenio propuesto.

En el siguiente enlace aparecen una serie de llamadas a la API con las que validar las funcionalidades respectivas a las autopropuestas, ya sea crearlas como consultarlas.

6.2. Implementación

Durante el periodo de desarrollo e implementación se llevan a cabo una serie de cambios en el propio diseño del proyecto previo.

Los cambios implementados a lo largo del proyecto pueden verse representados de manera más sencilla en el diagrama del Modelo de dominio (Figura 3.4). Sin embargo, las modificaciones sobre el análisis han requerido de cambios mayores en la implementación.

6.2.1. Base de datos y consultas

Una de las partes más afectadas por los cambios realizados en el modelo del dominio ha sido el diseño de la base de datos. Al requerir de nuevas entidades y cambios en los atributos, se ha requerido de un rediseño.

Como se puede apreciar en la Figura 6.1, que corresponde a la disposición de las tablas de la base de datos resultante del proyecto anterior, es bastante simple con respecto a las Propuestas ya que fue una de las secciones que quedó sin implementar.

Con el desarrollo de esta nueva funcionalidad de las propuestas, tanto normales como en convenio con empresas, el diseño debía verse modificado. El resultado tras incluir las entidades para las funcionalidades incluidas durante el desarrollo de este TFG, es el mostrado en la Figura 6.2. Al igual que en otros diagramas mostrados a lo largo de esta memoria, cuenta con la misma estructura de colores: en amarillo aparecen las tablas añadidas, en verde las modificadas y en azul las mantenidas.

6.2.2. Implementación sobre la aplicación web

Tras un análisis de usabilidad sobre la aplicación web, los resultados obtenidos sugerían un cambio en la forma de mostrar la información. En el TFG anterior todas aquellas listas de información (propuestas, adjudicaciones, ...) se mostraban filtradas por el usuario autenticado, no pudiendo ver el resto de información una vez autenticado.

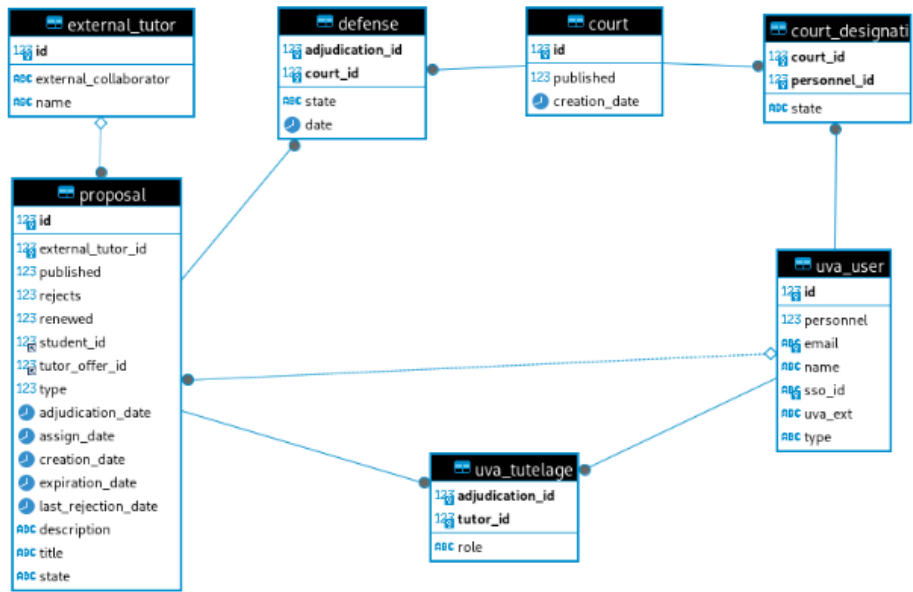


Figura 6.1: Diagrama de la Base de Datos Previo

Para eliminar esta restricción, el cambio que se planteó fue el de tener una sección de información personal en la que diferenciar aquella información global de la individual. Este cambio permite a los usuarios autenticados poder ver la información global y de manera separada en otra vista su información.

A modo de mejorar la experiencia del usuario sobre la aplicación web, se han incorporado componentes de filtrado en las tablas que listan las propuestas y las adjudicaciones, permitiendo seleccionar por estado de las entidades. Además de los componentes de filtro, estas mismas vistas incluyen la opción de ordenar las entidades alfabéticamente, facilitando el uso de las tablas dentro de la aplicación.

6.2.3. Modificación externa de listas de direcciones de correo

En el sistema desarrollado, se ha optado por externalizar ciertas listas sensibles (en este caso las direcciones del *Comité de Título* y del *Subdirector de Relaciones con las Empresas*) fuera del código fuente, con el objetivo de permitir su modificación sin necesidad de recompilar el proyecto completo.

Estas listas están definidas en el archivo de configuración `application.yml`, el cual es cargado externamente desde el entorno del contenedor Docker, y por tanto puede ser modificado directamente en producción o desarrollo bajo supervisión técnica. Esta aproximación permite mantener una configuración flexible y segura, reduciendo la rigidez del binario empaquetado (`.jar`).

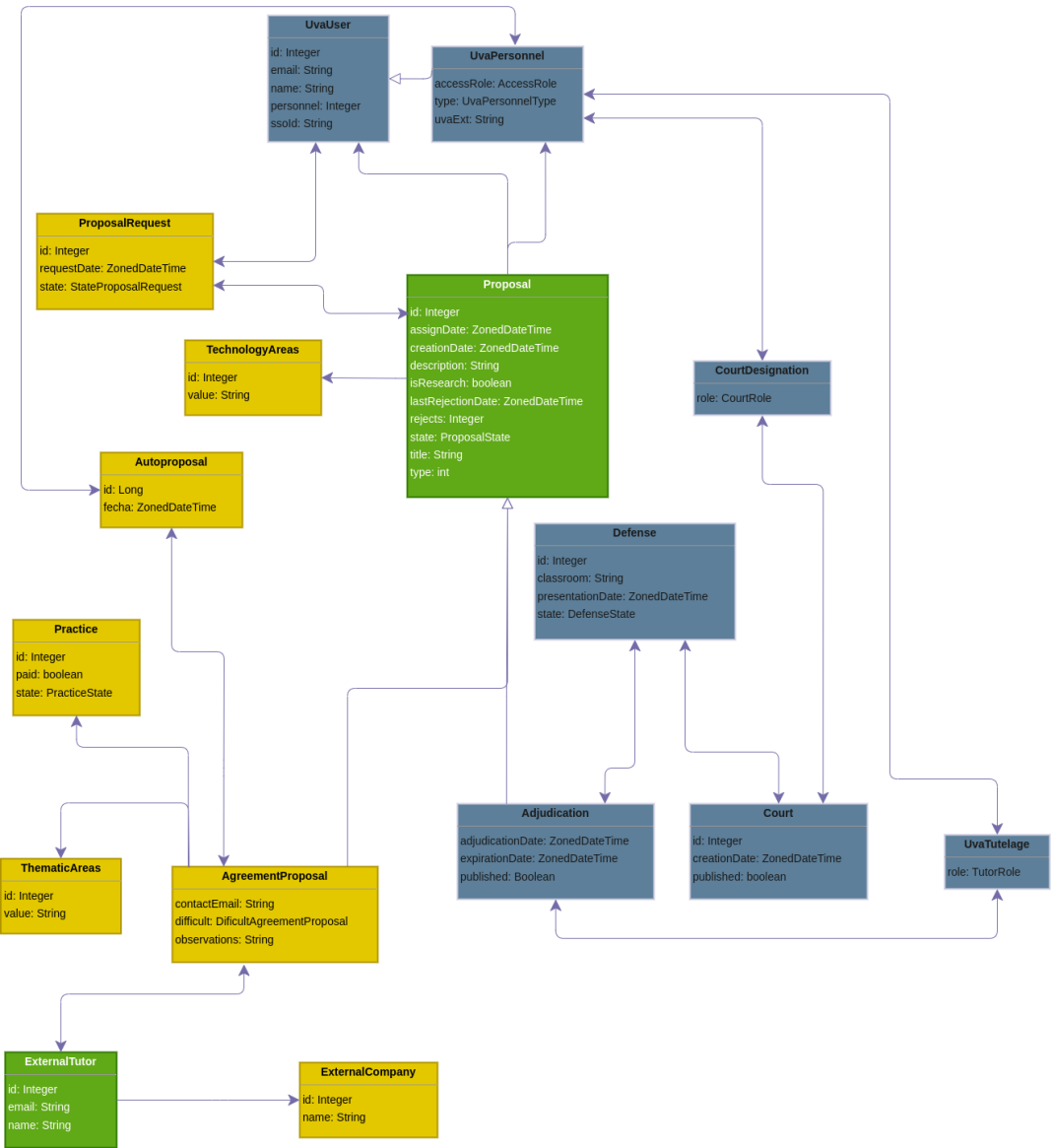


Figura 6.2: Diagrama de la base de datos final

Capítulo 7

Seguimiento del proyecto

El seguimiento del proyecto se aborda siguiendo las fases definidas en la sección 2.5 el proyecto se divide en varias fases, siendo la primera de ellas la Fase Inicial. Esta Fase Inicial es en la que se decidió contextualizar el TFG previo para poder desarrollar con mayor información este proyecto.

7.1. Fase Inicial

Esta fase se desarrolla aplicando un modelo tradicional, iterativo. A lo largo de toda esta fase previa, se han realizado reuniones entre el alumno y la tutora, además de reuniones entre la tutora, el Coordinador de Título y el Subdirector de Relaciones con las Empresas (ver definición de stakeholders, apartado 2.2). En dicha reunión se ha obtenido información adicional con la que poder llevar a cabo, en las fases posteriores, tanto el análisis como la implementación de las nuevas funcionalidades. Debido a la falta de trabajo durante el comienzo de esta fase, se ha podido realizar de manera más temprana todas las tareas. Esto ha supuesto la finalización de manera anticipada permitiendo comenzar la siguiente fase antes de lo planificado como se muestra en la Tabla 7.1.

7.2. Fase Ingeniería Inversa

Al igual que en la fase previa, esta se desarrolla aplicando un modelo iterativo. Durante el desarrollo de esta fase, en la que se pretendía analizar el problema en el punto en el que se dejó el TFG previo, surge un riesgo no planteado en la planificación inicial (Tabla 2.2). Este riesgo fue la pérdida del sistema operativo en el dispositivo utilizado en la realización de la asignatura. Aunque el riesgo no estuviera contemplado en la planificación mencionada, y para evitar la pérdida de trabajo, se utiliza en el desarrollo del TFG un sistema de versiones Git [7]. El plan que se ha seguido para que la planificación no se vea afectada en gran medida

7.2. FASE INGENIERÍA INVERSA

Fase Inicial	
Fecha de inicio	4 febrero 2025
Fecha de fin	12 febrero 2025
Metas	- 01: Trabajo de contextualización del TFG previo - 02: Definición de la metodología de trabajo - 03: Análisis de riesgos - 04: Calendarización
Metas alcanzadas	Todas
Tiempo invertido	22 horas

Tabla 7.1: Fase Inicial

ha sido la instalación desde 0 del mismo sistema operativo. Como plan de contingencia que se ha llevado a cabo después de solucionar el riesgo, ha sido la creación de un archivo para el despliegue de un entorno de desarrollo.

Además de la resolución del conflicto surgido, se analizó el problema en el punto que se dejó en el TFG anterior junto con la información obtenida en la reunión de la fase previa. Este análisis ha dado como resultado tanto los diagramas de procesos, como de casos de uso, diagrama de estados y de dominio.

Como se puede apreciar en la Tabla 7.2 las metas cumplidas fueron la mayoría de las planteadas, teniendo que corregir pequeñas cosas en el siguiente sprint.

Fase Ingeniería Inversa	
Fecha de inicio	12 febrero 2025
Fecha de fin	4 marzo 2025
Metas	- 01: Elicitación y análisis de requisitos - 02: Modelado de los procesos - 03: Modelado del dominio - 04: Casos de uso: definición y descripción - 05: Documentación acerca del seguimiento de la Fase Inicial
Metas alcanzadas	01,03,05
Tiempo invertido	28 horas

Tabla 7.2: Fase Ingeniería Inversa

7.3. Fase de Nuevos Requisitos

7.3.1. Sprint 1

Para este sprint se planificó, según la calendarización inicial 2.5 que comenzaría el desarrollo de nuevas funcionalidades. La funcionalidad acordada para desarrollar fue la correspondiente con el caso de uso CU04.

Desde el comienzo del sprint nos encontramos con uno de los problemas principales hasta el momento, y es el hecho de que no es posible trabajar en local con la configuración actual del proyecto. Añadir también, que la documentación aportada en la memoria del TFG previo únicamente hace referencia al manual de despliegue sobre la maquina virtual. Debido a este problema, el objetivo de crear nueva funcionalidad se retrasa, pasando a ser el principal objetivo poder trabajar desde la maquina del estudiante de manera sencilla. El riesgo, contemplado en el Análisis de riesgos iniciales (Tabla 2.2) como *Dependencias Externas (R04)*, no se tuvo en cuenta en un primer momento ya que en alguna de las ramas del repositorio sí estaban configuradas para un trabajo en local. Sin embargo, estas no contenían la funcionalidad final del proyecto anterior.

El plan de actuación ante la aparición de este riesgo ha sido el siguiente, evaluar cada una de los fallos que iba generando hasta encontrar la manera de hacer una configuración nueva que permitiera tener ambos entornos de desarrollo de manera independiente. Como plan de mitigación, tanto para el desarrollo del proyecto, como para futuros TFGs, se comienza una documentación con la que poder tener un conocimiento acerca de las configuraciones de los entornos.

En la Tabla 7.3 se puede observar que finalmente se pudieron conseguir las metas acerca del modelado y análisis de los procesos, empleando un total de 31 horas de trabajo en el sprint. La meta 03 correspondiente a la implementación de la CU04 no se pudo conseguir, debido a lo explicado anteriormente, quedando pendiente para el siguiente sprint.

Sprint 1	
Fecha de inicio	4 marzo 2025
Fecha de fin	17 marzo 2025
Metas	- 01: Modelado de los procesos - 02: Casos de uso: definición y descripción - 03: CU04: Crear Propuesta de TFG en convenio. 5 Puntos de Historia - 04: Documentación acerca del seguimiento de la Fase de Ingeniería Inversa
Metas alcanzadas	01,02,04
Tiempo invertido	31 horas

Tabla 7.3: Sprint 1

7.3.2. Sprint 2

Para este segundo sprint se planificó, según la calendarización inicial 2.5 que se continuaría con la implementación de nuevas funcionalidades. Debido a la situación del sprint anterior, las funcionalidades pasan a ser previstas para este sprint, es decir, el CU04.

Se comienza el desarrollo por el servidor backend para poder proporcionar una estructura sólida al frontend. El CU04 consiste, según se especifica en la Tabla 3.4, en la funcionalidad acerca de la creación de ofertas de TFG en convenio. Para su implementación, la primera de las tareas era modificar la estructura de roles del sistema ya que entraba en juego un nuevo stakeholder, el Delegado de Empresa. Al finalizar la funcionalidad principal tanto en el backend como en el frontend, me he dado cuenta de que el encargado de llevar a cabo esta acción, hasta el momento lo hacía mediante una plantilla de Word que las empresas rellenaban con la información pertinente. Por tanto, se ha tomado la decisión de incluir la posibilidad de generar las ofertas de TFG en convenio a partir de una plantilla de Excel con la que facilitar la acción.

En la Tabla 7.4 se puede observar que finalmente se pudieron conseguir las metas acerca de la implementación del CU04, empleando un total de 32 horas de trabajo en el sprint. Además, durante el sprint se ha trabajado en la memoria y en la generación de una batería de pruebas en en Postman para los endpoints (ver Capítulo 6).

Sprint 2	
Fecha de inicio	18 marzo 2025
Fecha de fin	31 marzo 2025
Metas	- 01:CU04: Crear Propuesta de TFG en convenio. 5 Puntos de Historia - 02: Documentación acerca del seguimiento del sprint anteraquellasior
Metas alcanzadas	01, 02
Tiempo invertido	32 horas

Tabla 7.4: Sprint 2

7.3.3. Sprint 3

La planificación para el tercer sprint, según la calendarización inicial 2.5 consiste en continuar con la implementación de nuevas funcionalidades. Las funcionalidades estipuladas para este sprint corresponden con los casos de uso CU01, CU02 y CU09, los cuales están relacionados entre sí.

Al igual que en el desarrollo del sprint anterior, se comienza por el backend para poder proporcionar una estructura sólida con la que contar. Este sprint se centra en aquellas funcionalidades más relacionadas con la asociación entre el estudiante y el tutor como son la

solicitud de una propuesta y la aceptación o denegación de dicha propuesta.

Uno de los principales problemas a la hora de enfrentarse a estos casos de uso era las relaciones circulares que se podían producir a la hora de relacionar las entidades. Gracias al seguimiento exhaustivo del análisis del problema 3.4 y al uso de etiquetas adecuadas he conseguido no tener relaciones circulares entre las entidades del dominio.

En la Tabla 7.5 se puede observar que finalmente se pudieron conseguir todas las metas planteadas, empleando un total de 29 horas de trabajo en el sprint.

Sprint 3	
Fecha de inicio	1 abril 2025
Fecha de fin	14 abril 2025
Metas	- 01:CU01: Solicitar propuesta de TFG. 2 Puntos de Historia - 02: CU02: Aceptar solicitud de TFG. 1 Punto de Historia - 03: CU09: Consultar solicitudes. 1 Punto de Historia - 04: Finalización del capítulo de Analisis - 05: Documentación acerca del seguimiento del sprint anterior
Metas alcanzadas	TODAS
Tiempo invertido	29 horas

Tabla 7.5: Sprint 3

7.3.4. Sprint 4

La planificación para el cuarto sprint, según la calendarización inicial 2.5 debería ser el primer sprint de la Fase de Mejoras, sin embargo, debido al retraso ocasionado por los problemas de configuración de los entornos durante el sprint 1, se ha decidido hacer un cambio en dicha calendarización.

El sprint 4 se convierte en el último sprint de la Fase de Nuevos Requisitos, dejando unicamente dos sprints en la Fase de Mejoras. Este cambio permite poder implementar todos aquellos nuevos requisitos planteados durante la fase Inicial en el Product Backlog 2.1. Los casos de uso CU07, CU08 y CU10 pasan al sprint backlog.

Uno de los principales problemas a la hora de implementar los casos de uso CU07 y CU08 en el backend fue la aparición de relaciones circulares entre las entidades, produciéndose errores en el mapeo de las clases.

En la Tabla 7.6 se puede observar que finalmente se pudieron conseguir todas las metas planteadas, empleando un total de 18 horas de trabajo en el sprint, esta facilidad me permitió

dedicarme a ampliar otras partes de memoria, como la realización de una documentación acerca de los endpoints del *backend*.

Sprint 4	
Fecha de inicio	15 abril 2025
Fecha de fin	28 abril 2025
Metas	- 01: CU07: Consultar autopropuestas. 1 Punto de Historia - 02: CU08: Autoproponerse. 2 Punto de Historia - 03: CU10: Seleccionar estudiante. 1 Punto de Historia - 04: Trabajo sobre el capítulo de Diseño e Implementación y pruebas - 05: Documentación acerca del seguimiento del sprint anterior
Metas alcanzadas	TODAS
Tiempo invertido	18 horas

Tabla 7.6: Sprint 4

7.4. Fase de Mejoras

7.4.1. Sprint 5

Este sprint se tenía planificado, según la calendarización inicial 2.5, que sería el segundo sprint de la Fase de mejoras. Sin embargo, y de igual forma que le sucedió al sprint anterior, ahora este sería el primer sprint para el desarrollo de mejoras sobre el proyecto anterior. Además de verificar las mejoras que se fueron realizando durante los sprints anteriores, para que todo funcionara de manera correcta, en este sprint se planteaba la idea de mejorar algo que ya estuviera en el producto final del proyecto anterior.

En el proyecto anterior, el ciclo de una propuesta de TFG comenzaba cuando el profesor la creaba para enviarla a aprobación por el comité de título a la misma vez que indicaba el estudiante para realizar la adjudicación/asignación del TFG.

En el nuevo ciclo de vida de una propuesta de TFG (que se ha abordado en la fase anterior), el profesor crea una propuesta de TFG para ofertarla. Los estudiantes la solicitan. El profesor acepta/deniega las solicitudes. Cuando ya tiene una solicitud aceptada, el ciclo continúa enviando la propuesta con la asignación de estudiante al comité de título para su aprobación y adjudicación/asignación. Esto mismo puede suceder también en relación con la nueva incorporación a la aplicación de los TFGs en convenio.

En este sprint se centró la atención en la modificación principal que era la de continuar el flujo del ciclo de vida de una propuesta de TFG tras la creación de la propuesta.

Además, con el fin de clarificar y facilitar el uso tanto de los desarrolladores como de los técnicos, se ha trabajado en una diferenciación de los entornos, de desarrollo y de producción, de manera automatizada gracias al uso de contenedores docker.

En la Tabla 7.7 se puede observar que finalmente se pudieron conseguir todas las metas planteadas, empleando un total de 30 horas de trabajo en el sprint.

Sprint 5	
Fecha de inicio	29 abril 2025
Fecha de fin	12 mayo 2025
Metas	- 01: CU05: Solicitud de adjudicación de TFG. 5 Punto de Historia - 02: Trabajo de división y documentación de entornos - 03: Trabajo de despliegue sobre la máquina virtual - 04: Documentación sobre bloques de la memoria - 05: Documentación acerca seguimiento de la Fase Inicial
Metas alcanzadas	TODAS
Tiempo invertido	30 horas

Tabla 7.7: Sprint 5

7.4.2. Sprint 6

La planificación para el sexto sprint, según la calendarización inicial 2.5 corresponde con el último sprint utilizado para el trabajo de desarrollo de las aplicaciones en la Fase de Mejoras. Durante el sprint se han llevado a cabo 2 nuevas reuniones con dos de los stakeholders 2.2 para revisar las funcionalidades desarrolladas y poder realizar algunas mejoras que no estaban contempladas desde el primer momento. Además de las mejoras planteadas por los usuarios finales de la aplicación, se ha llevado a cabo la mejora que estaba estipulada para este sprint. Dicha mejora está relacionada con la sección de Tribunales, en la cual se habían detectado problemas de usabilidad para el Coordinador de la titulación.

Las mejoras sugeridas por los entrevistados suponen la gran mayoría de cambios durante el sprint. Las propuestas por el Coordinador de Título son aquellas relacionadas con la inclusión de filtros o la posibilidad de editar los Tribunales, siendo esta la modificación correspondiente al sprint. Las propuestas de mejora sugeridas por el Delegado de empresas son aquellas que tienen que ver con las propuestas en convenio, como la posibilidad de eliminar una propuesta o que el tutor externo sea introducido por el comité de título, debido a que esa información no cuenta con ella el Delegado cuando crea la propuesta.

La refactorización sobre los Tribunales ha supuesto la renovación completa del componente permitiendo editar o eliminar cada tribunal y además de incluir varios suplentes a cada tribunal.

Durante los momentos finales del sprint surgen problemas relacionados con cambio o modificaciones que se han ido realizando a lo largo del proyecto sin comprobar el flujo completo. Por tanto, antes de la finalización del sprint se decide hacer un guion de flujos y comprobar que todo queda correctamente funcionando.

En la Tabla 7.8 se puede observar que finalmente se pudieron conseguir todas las metas planteadas, además de las establecidas en las reuniones con los interesados, empleando un total de 34 horas de trabajo en el sprint, siendo así el último sprint destinado a desarrollo de código.

Sprint 6	
Fecha de inicio	13 mayo 2025
Fecha de fin	26 mayo 2025
Metas	- 01: Modificaciones de los Tribunales - 02: Modificaciones propuestas por el Coordinador de Título - 03: Modificaciones propuestas por el Delegado de Empresas - 04: Documentación acerca del seguimiento del sprint anterior
Metas alcanzadas	TODAS
Tiempo invertido	34 horas

Tabla 7.8: Sprint 6

7.5. Cierre del Proyecto

Tras la finalización de todos los sprints definidos en la calendarización 2.5, esta define la última fase del proyecto en la que terminar de completar la memoria y llevar a cabo todas aquellas comprobaciones necesarias para dar por finalizado el proyecto.

Para comenzar esta fase, tras la reunión semanal con la tutora, se han hecho una serie de flujos en los que comprobar que todas la acciones e implementaciones llevadas a cabo se completan de manera exitosa. Durante la ejecución de dichos flujos surgen algunos errores pequeños, los cuales se solucionan de manera sencilla. Tras completar todas estas comprobaciones, el siguiente paso es dejar el código operativo corriendo en la maquina virtual.

De manera simultánea durante todo el desarrollo, como se ha ido especificando en las tareas de los sprints, se ha ido realizando un trabajo sobre la memoria. Sin embargo, antes

de poder entregarlo todo, queda terminar alguna sección que ha de completarse una vez finalizado el desarrollo del software. Estas secciones de la memoria han sido las que se han terminado en esta fase, a la vez de realizar varias revisiones de lo que ya estaba terminado.

Como se puede apreciar en la Tabla 7.9, estas han sido las acciones llevadas a cabo durante esta última fase de la realización del TFG y que ha requerido de 68 horas de trabajo.

Cierre del proyecto	
Fecha de inicio	27 mayo 2025
Fecha de fin	11 de junio 2025
Metas	- 01: Finalización memoria - 02: Complementar documentación - 03: Dejar en funcionamiento operativo el proyecto sobre la maquina virtual
Metas alcanzadas	TODAS
Tiempo invertido	68 horas

Tabla 7.9: Cierre del proyecto

7.6. Resumen de la ejecución del proyecto

Para finalizar, realizo una comparación entre lo que se analizó en la Sección 2.5 y la realidad tras el desarrollo del proyecto. Para esta comparación, se analiza la calendarización, el presupuesto y los riesgos que han ido surgiendo durante la ejecución.

7.6.1. Calendarización final

De igual forma que se muestra en la Tabla 2.5, el proyecto se desarrolla basándose en un modelo híbrido, contando con dos primeras fases de análisis inicial e ingeniería inversa, y seis sprints de desarrollo.

Tras el final del proyecto, el resumen de la calendarización se muestra en la Tabla 7.10 y queda más desarrollado en la Tabla 7.11. El plan se llevó a cabo según lo planeado hasta el primer sprint, momento en el que los entornos no permitían desarrollar las nuevas funcionalidades. Esto provocó que el primer sprint se dedicara a otra cosa que no era los especificado. La causa del bloqueo se trata a continuación.

7.6.2. Gestión de Riesgos

Al abordar el desarrollo de este proyecto, se realizó una identificación inicial de los riesgos que podían afectar su transcurso, los cuales fueron detallados en la Tabla 2.2. La gestión

	Horas
Fase Inicial	22
Fase Ingeniería Inversa	28
Sprint 1	31
Sprint 2	32
Sprint 3	29
Sprint 4	18
Sprint 5	30
Sprint 6	34
Cierre del Proyecto	72
Horas totales	296

Tabla 7.10: Resumen de la calendarización final

dinámica de estos elementos fue clave para la consecución de los objetivos planteados.

Afortunadamente, una gran mayoría de los riesgos inicialmente identificados no llegó a materializarse, o lo hizo de forma controlada, gracias a una planificación y ejecución cuidadosas. El riesgo de *Comunicación y Colaboración (R02)*, por ejemplo, fue mitigado eficazmente. El establecimiento de reuniones regulares y claras, permitió abordar cualquier duda o necesidad de adaptación de manera proactiva, asegurando un avance constante sin bloqueos significativos. De manera similar, la *Gestión de Recursos (R03)* no presentó dificultades importantes; la planificación detallada de tareas y una distribución realista de la carga de trabajo evitaron sobrecargas y retrasos asociados a una asignación ineficiente. Asimismo, el *Trabajo Externo (R06)* pudo ser gestionado de manera que no supuso un impedimento para el desarrollo del proyecto ya que se redujeron las horas de dedicación a las prácticas en empresa, distribuyendo de mejor manera el tiempo, permitiendo minimizar su impacto.

No obstante, el ciclo de vida de un proyecto rara vez transcurre sin imprevistos y algunos de los riesgos contemplados sí llegaron a manifestarse, poniendo a prueba la capacidad de adaptación y resolución. El riesgo de *Inadecuada Gestión de Requisitos (R01)* se manifestó durante el desarrollo. Inicialmente, se había planteado una estructura de sprints que combinaba fases de añadir nuevas funcionalidades con una de completado antes de finalizar la fase. Sin embargo, la evolución y la especificación más detallada de las funcionalidades a implementar llevaron a redefinir los objetivos, dedicando finalmente tres sprints completos, cada uno a una funcionalidad concreta. Esta adaptación, si bien implicó un ajuste en la planificación original, se gestionó sin generar trabajo extra significativo gracias a la flexibilidad adoptada y la claridad en la obtención de los requisitos en cada iteración.

Por otro lado, el riesgo relacionado con las *Dependencias Externas (R04)* se evidenció de forma concreta en los problemas con los entornos de desarrollo y despliegue. Estas dificultades, que requirieron la implicación por parte del estudiante en todo el primer sprint, ocasionaron un retraso de un sprint en la planificación general del proyecto hasta que se lograron solucionar las incidencias y se obtuvo la disponibilidad y estabilidad necesarias.

7.6.3. Costes

En esta sección, se procede a analizar la variación entre el presupuesto estimado inicialmente en la Sección 2.6 y los costes finales que el proyecto habría generado si se hubiera ejecutado en un escenario semi real. Para llevar a cabo esta comparativa, se mantendrá la metodología de cálculo previamente establecida, ajustando únicamente aquellos parámetros que dependen directamente de la duración real del desarrollo.

Concretamente, los valores que requieren ser actualizados corresponden a las horas de dedicación del personal implicado en el proyecto y al periodo de amortización de las herramientas de hardware utilizadas. Las fórmulas con los datos revisados y los resultados derivados de esta actualización pueden consultarse en las Ecuaciones 7.1 y 7.2.

$$CP = \frac{2000 \frac{\text{€}}{\text{mes}} * 0,45 * 296h * 1,3}{68 \frac{h}{\text{mes}}} = 5092,94\text{€} \quad (7.1)$$

$$CH = \frac{700 * 0,26 \frac{\text{amor}}{\text{año}} * 4,26\text{meses}}{12 \frac{\text{mes}}{\text{año}}} = 64,61\text{€} \quad (7.2)$$

Comparando la Tabla 2.6 y la Tabla 7.12 nos encontramos con una reducción de los costes con respecto a lo presupuestado de 49.76€, o lo que supone un 0.95 % del presupuesto. original.

Concepto	Importe (€)
Personal	5 092.94
Software	0
Hardware	64.61
Total	5 157.55

Tabla 7.12: Presupuesto Real

7.6. RESUMEN DE LA EJECUCIÓN DEL PROYECTO

	Inicio	Fin	Comentario	Horas
Fase Inicial	04/02/2025	17/02/2025		22
Reunión Semanal	Se realizan cada martes.		Se realizaron reuniones semanales para la planificación del proyecto, definición del trabajo previo, contextualización del TFG previo (tecnologías).	
Fase Ingeniería Inversa	18/02/2025	03/03/2025	Análisis del TFG previo, ingeniería inversa de procesos. Definición de objetivos.	28
Fase Nuevos Requisitos	03/03/2025	14/04/2025		
Sprint 1	04/03/2025	17/03/2025	Trabajo para poner en marcha entornos.	31
Sprint 2	18/03/2025	31/03/2025	Desarrollo de las propuestas de convenio.	32
Sprint 3	01/04/2025	14/04/2025	Desarrollo de las solicitudes de propuestas.	29
Sprint 4	15/04/2025	28/04/2025	Desarrollo de las autopropuestas	18
Fase de Mejoras	15/04/2025	26/05/2025	Se realizaron reuniones con el Coordinador de Título, el Delegado de empresas y la tutora para dar comentarios de mejoras.	4
Sprint 5	29/04/2025	12/05/2025	Mejora para unificar los flujos.	30
Sprint 6	13/05/2025	26/05/2025	Mejora propuestas en las reuniones.	34
Cierre Proyecto	27/05/2025	11/06/2025	Adicionalmente, se dedicó tiempo a ir probando con los usuarios finales, detectar errores en uso de la aplicación y mejoras, y corregir los errores y abordar algunas mejoras	72
Horas totales				296

Tabla 7.11: Calendarización final del proyecto

Capítulo 8

Conclusiones

El desarrollo de este Trabajo de Fin de Grado me ha permitido alcanzar los objetivos que me propuse, representando un avance importante en la creación de una plataforma digital para la gestión de los TFGs de la Escuela. Si bien el alcance total de la digitalización de este proceso era ambicioso y superaba lo que un único TFG puede abarcar, este proyecto consiguió sentar una base sólida, demostrando la viabilidad de la idea y preparando el camino para futuras ampliaciones y mejoras.

Más allá de aplicar y afianzar mis conocimientos técnicos, el mayor aprendizaje durante la realización del TFG ha sido sobre la gestión de un proyecto de tal envergadura. Fue un reto interesante tomar un trabajo ya existente, comprenderlo a fondo y, lo más importante, ser capaz de añadir nuevas funcionalidades que son fruto de mi propio esfuerzo. Esta capacidad de dominar un contexto desconocido y generar algo nuevo ha sido fundamental para mi crecimiento profesional, a pesar de los momentos de agotamiento que surgieron. En retrospectiva, este TFG ha cumplido con mis expectativas, permitiéndome profundizar en áreas donde tenía menos experiencia y prepararme de forma práctica para lo que venga en mi carrera profesional.

8.1. Líneas de trabajo futuras

Durante el desarrollo de este TFG se cumplieron algunas de las líneas de trabajo futuras que se dejaron reflejadas en el TFG anterior [2]. Sin embargo, no ha sido posible realizar todas de las que se propusieron. A continuación se muestran las líneas de trabajo futuro, tanto las que ya estaban propuestas que no se han abordado en este, como las nuevas que se sugieren a partir de este TFG.

Líneas que quedaron pendientes del TFG anterior y que no se han abordado:

- Dispositivos portátiles: Si bien esta aplicación se ha conceptualizado como una aplica-

ción web usada principalmente en escritorio podría resultar interesante, sobre todo para los estudiantes, que la app se comportara mejor en este tipo de dispositivos. Esto puede abrir la posibilidad de portar la app a SSR (Server-side Rendering) y adaptaciones de la app a pantallas pequeñas y verticales.

- DevOps: Implementar un ciclo de vida DevOps aumentaría en gran medida la velocidad de testeo y automatizaría el despliegue de las aplicaciones.

Las propuestas a partir de la realización de este TFG:

- Usabilidad: Dado que la elección del TFG puede ser una decisión abrumadora desde el punto de vista de los estudiantes, incorporar un *agente de recomendación* que permita al estudiante filtrar entre los TFG propuestos y encontrar aquellos que son más acordes a sus gustos o predisposiciones, podría ser interesante y de utilidad.
- Usabilidad: Ya que en la aplicación se realizan acciones importantes dentro de la gestión de los TFGs, sería recomendable la opción de notificación durante los flujos de trabajo dentro de la aplicación así como mediante correos electrónicos.
- Optimización: Sería interesante llevar a cabo una refactorización de la arquitectura del backend y, apoyándose de la contenerización, convertirla en un servidor basado en micro-servicios. Esto permitiría la inclusión de nuevas funcionalidades de una manera más sencilla sin restringir el funcionamiento de la aplicación.
- Optimización y Usabilidad: Debido a que el volumen de datos puede llegar a ser elevado, sería recomendable el uso de elementos visuales para que el usuario perciba que los datos se están cargando y además que estos datos no se traigan de manera general en todas las vistas, si no que se hagan de manera optimizada para cada necesidad.

Algunas puntualizaciones importantes que deben tenerse en cuenta cuando se vaya a poner en uso la aplicación se comentan a continuación.

Será necesario pedir a todos los profesores de la Escuela que se identifiquen en la aplicación para poder tener sus usuarios. Es necesario tener la información de los profesores en nuestro sistema para poder asignarles tutorizaciones académicas por parte del comité de título. También es preferible tener su información ya incorporada al sistema antes de crear tribunales. Es posible crear tribunales solamente indicando nombre y dirección de correo electrónico, pero si luego están mal los emails (que son identificadores únicos en el sistema), el efecto sería como si hubiéramos creado un profesor nuevo.

Deberá informarse bien al profesorado de que, antes de aceptar una solicitud a una propuesta de TFG el profesor debe hablar por email y/o personalmente con el estudiante cuya solicitud quiere aceptar. Esto es importante para asegurarnos de que, al aceptar su solicitud, se está respetando la decisión final del estudiante sobre otras posibles solicitudes que haya realizado. Esto es muy relevante porque, para mantener la consistencia y no generar bloqueos en el sistema, cuando un profesor acepta la solicitud de un estudiante a un TFG que ha propuesto, el sistema deniega automáticamente sus otras solicitudes de TFGs y las solicitudes de otros estudiantes sobre el TFG en cuestión.

Para que el Coordinador de la titulación, o en quien delegue ese rol, pueda asignar un profesor a una tutorización académica de un TFG en convenio con empresas es importante que el profesor se haya identificado previamente en la aplicación (primer punto comentado antes).

Opcionalmente, el profesorado puede crear una propuesta de TFG, indicando al mismo tiempo el estudiante que lo realizará. Antes de crear la propuesta con el nombre y el email del estudiante, el profesor debe preguntar al estudiante si tiene otros TFGs solicitados porque en ese caso el estudiante deberá borrar las solicitudes que haya realizado a otros TFGs no vaya a ser que otro profesor se la acepte. Además, al crear la propuesta con estudiante, el profesor debe prestar mucha atención a la corrección del email. El email identifica unívocamente a los estudiantes y podría estarse generando un nuevo estudiante en el sistema.

Se recomienda al Coordinador de la titulación no borrar tribunales de TFG. Esto puede tener efectos muy perversos tal como está planteado actualmente el sistema. Es preferible modificar el tribunal o crear un tribunal nuevo con los mismos profesores o con casi todos los miembros iguales.

Podría darse el caso de que lleguen dos propuestas al comité de título en estado pendiente con el mismo estudiante. En este caso, el Comité debería consultar sobre todo con el estudiante, pero también informar a los tutores que han presentado las propuestas a aprobación. Ante dos opciones posibles, la preferencia del estudiante debe prevalecer. El coordinador, o en quien delegue, puede borrar una y asignar la otra.

Bibliografía

- [1] Tom Addison. Controlling software project risks. https://people.dsv.su.se/~joco2917/ft_gateway.cfm97.pdf, 2002. Accedido: 12/02/2025.
- [2] Borja Arnáez Pérez. Gestión de tfgs en la escuela de ingeniería informática de valladolid. TFG disponible en <https://uvadoc.uva.es/handle/10324/71329>, 2024. Capítulo 2.1.1, Página 6.
- [3] Atlassian. What is scrum and how to get started. <https://www.atlassian.com/agile/scrum>, 2023. Accedido: 12/02/2025.
- [4] Scrum Manager BoK. Epic. <https://www.scrummanager.com/bok/index.php?title=Epic>, 2023. Actualizado: 18/12/2023.
- [5] Digital.ai. Exploración de la ingeniería inversa: beneficios, uso indebido y el papel del endurecimiento de aplicaciones. <https://digital.ai/es/catalyst-blog/exploring-reverse-engineering-benefits-misuse-and-the-role-of-application-hardening/>, 2023. Última actualización: 20/03/2024. Accedido: 07/02/2025.
- [6] Michael Gant. Scrum and the solo dev. <https://medium.com/@jmgant.cleareyeconsulting/scrum-and-the-solo-dev-fb8e810ed42b>, 2019. Última actualización: 16/04/2019. Accedido: 06/02/2025.
- [7] GitLab. Gitlab. <https://gitlab.inf.uva.es>, 2025.
- [8] Glassdoor. Sueldos para el puesto de desarrollador de software en españa. https://www.glassdoor.es/Sueldos/desarrollador-de-software-sueldo-SRCH_K00,25.htm, 2025. Con una experiencia de 1-3 años. Última actualización: 05/02/2025. Accedido: 09/02/2025.
- [9] Microsoft Learn. Modelo-vista-modelo de vista (mvvm). <https://learn.microsoft.com/es-es/dotnet/architecture/maui/mvvm>, 2025. Última actualización: 10/09/2024. Accedido: 06/05/2025.
- [10] Max Rehkopf. ¿qué son los sprints en la gestión de proyectos? <https://www.atlassian.com/es/agile/scrum/sprints>, 2025. Accedido: 09/02/2025.
- [11] Smartbear. Swagger. https://swagger.io/docs/specification/v3_0/about/, 2025.

- [12] Agencia Tributaria. Tabla de amortización simplificada. https://sede.agenciatributaria.gob.es/Sede/ayuda/manuales-videos-folletos/manuales-practicos/folletos-actividades-economicas/3-impuesto-sobre-renta-personas-fisicas/3_5-estimacion-directa-simplificada/3_5_4-tabla-amortizacion-simplificada.html, 2025. Accedido: 09/02/2025.
- [13] Universidad de Valladolid. Normativa de evaluación del trabajo fin de grado de e.t.s. de ingeniería informática., 2023. Última actualización: 27/04/2023. Accedido: 10/11/2023.
- [14] Universidad de Valladolid. Reglamento sobre la elaboración y evaluación del trabajo fin de grado, 2023. Última actualización: 27/04/2023. Accedido: 10/11/2023.

Apéndice A

Manuales

A.1. Parámetros de configuración dinámica y segura

A.1.1. Modificación de las listas de usuarios con roles especiales

Debido a la política de una persona-una identidad, el sistema de identificación de la universidad (SSO) no permite diferenciar si un usuario tiene actualmente ostenta un cargo determinado. Algunos usuarios de la aplicación juegan un rol especial como coordinador de la titulación, o en quien delegue, o subdirector de relaciones con empresas, o en quien delegue. Se ha optado por una configuración dinámica del sistema basada en listas de correo electrónico. De esta forma pueden asociarse varios correos electrónicos a cada rol.

Para modificar las direcciones de correo, se debe editar directamente el archivo `secure-application` en el servidor host, utilizando un editor de texto (por ejemplo, `nano`, `vim`, o mediante una herramienta de CI/CD segura). Posteriormente, se requiere reiniciar el contenedor Docker correspondiente para que los cambios surtan efecto, ya que las propiedades se cargan en tiempo de arranque de la aplicación.

```
docker-compose -f compose.yml down
docker-compose -f compose.yml up
```

Seguridad

El archivo se monta en el contenedor en modo `read-only (:ro)`, de modo que la modificación de sus valores sólo puede realizarse desde el host y por usuarios autorizados, lo que reduce la posibilidad de modificaciones accidentales o malintencionadas en producción.

Ubicación y estructura del archivo

El archivo se encuentra montado en el contenedor mediante un volumen Docker, en modo solo lectura, desde una ubicación en el host. La ruta típica es:

- **En el host:** `./config/secure-application.yml`
- **Dentro del contenedor:** `/config/application.yml`

La estructura de las listas en dicho archivo se muestra al final del apartado A.2.3.

A.2. Lanzamiento del sistema en entornos diferenciados (*Backend*)

El proyecto ha sido diseñado para funcionar tanto en modo **desarrollo local** como en modo **producción**, mediante el uso de múltiples archivos *dockercompose* y perfiles de Spring Boot.

A.2.1. Estructura de configuración

- *application.yml*: configuración común para todos los entornos (base de datos, listas de correos, etc.).
- *applicationlocal.yml*: configuración específica del entorno de desarrollo (puerto HTTP, SSL desactivado).
- *applicationprod.yml*: configuración específica del entorno de producción (puerto HTTPS, certificado SSL, etc.).

La activación del perfil adecuado se realiza a través de la variable de entorno:

```
SPRING_PROFILES_ACTIVE=prod
```

A.2.2. Lanzamiento en entorno de desarrollo

Para ejecutar el sistema en desarrollo local, se debe utilizar el archivo *compose_des.yml*. Este archivo levanta los contenedores necesarios (base de datos y backend) sin características de seguridad adicionales como HTTPS.

```
docker-compose -f compose_des.yml up -build
```

El perfil activo en este caso es `local`, y debe definirse como variable de entorno en el servicio correspondiente o por defecto en el `application.yaml`:

```
SPRING_PROFILES_ACTIVE=local
```

A.2.3. Lanzamiento en entorno de producción

Para ejecutar el sistema en producción, se utiliza el archivo `compose.yml`. Este archivo incluye configuración de red segura (puerto 8443 con HTTPS), y carga de configuración externa mediante volúmenes montados con archivos protegidos.

```
sudo docker-compose -f compose.yml up -build
```

La definición del volumen incluirá el archivo de configuración como se indica:

volumes:

- `./config/secure-application.yaml:/config/application.yaml:ro`

El perfil activo será `prod`, y deberá definirse también como variable de entorno:

```
SPRING_PROFILES_ACTIVE=prod
```

Notas adicionales

Es fundamental que el archivo de configuración externo contenga las variables que deben ser dinámicamente cargadas por Spring Boot, como:

app:

```
webapp-url: https://tfgs.inf.uva.es
local-state: false
```

domain:

```
coordinators:
  - usuario1@uva.es
  - usuario2@uva.es
```

```
business:
  - usuario3@uva.es
  - usuario4@uva.es
```

Estas serán leídas automáticamente al inicio del contenedor, integrándose sin necesidad de recompilar el artefacto `.jar`.

A.2.4. Borrado de datos tras las pruebas

Este proyecto para poder ser usado con el sistema de identificación del CAS de la UVa, debe ser probado desde la máquina virtual antes de poder usarse de manera normal. Por ello, en esta sección del manual se especifica cómo eliminar todos aquellos valores guardados por la aplicación durante las pruebas y cómo volverlo a dejar listo para funcionar.

Lo primero de todo es eliminar los contenidos y para ello, paramos las imágenes que están corriendo en la máquina virtual:

```
sudo docker-compose -f ~/git/infuva_tfg_manager_service/compose.yaml down
```

Una vez paradas las imágenes, se procede con el borrado de la información. Para ello eliminamos el volumen de datos y la imagen asociada a ese volumen:

```
sudo docker container prune
```

```
sudo docker image prune
```

```
sudo docker system prune -a -volumes
```

Cuando ya está todo eliminado, para volver a ponerlo en marcha únicamente quedaría seguir los pasos especificados en el apartado A.2.3 del Manual.

A.3. Lanzamiento del sistema en entornos diferenciados (*Webapp*)

Para gestionar las diferencias entre los entornos de desarrollo y producción, se utilizarán archivos de configuración como `.env` y `.env.production`. Estos archivos permiten cambiar las configuraciones de la aplicación dependiendo del entorno en que se ejecute.

- **Desarrollo:** Se utiliza el archivo `.env` que contiene las configuraciones para el entorno local.
- **Producción:** Se utiliza el archivo `.env.production` que contiene las configuraciones específicas para el entorno de producción.

En estos archivos se definirán variables clave, como las URLs de la API y los valores del entorno a desplegar.

Estos valores están definidos de la siguiente manera:

A continuación, se explican los pasos necesarios para ejecutar el frontend en los entornos de desarrollo y producción.

```
VITE_APP_ENV = local | production  
VITE_API_URL = [la url del entorno]
```

Fragmento de código A.1: Contenidos archivo `.env` o `.env.production`

A.3.1. Lanzamiento en Desarrollo

Para ejecutar la aplicación en el entorno de desarrollo, se deben seguir los siguientes pasos:

1. Asegurarse de tener el archivo `.env` correctamente configurado para el entorno de desarrollo (local), como se muestra en la sección previa en el Fragmento de código A.1.
2. Ejecutar el siguiente comando para iniciar la aplicación:

```
npm start
```

A.3.2. Lanzamiento en Producción

Para ejecutar la aplicación en producción, se deben seguir los siguientes pasos:

1. Asegurarse de tener el archivo `.env.production` correctamente configurado para producción (production), como se muestra en la sección previa en el Fragmento de código A.1.
2. Ejecutar el siguiente comando para construir la aplicación para producción:

```
npm run build
```

Esto generará una carpeta `build/` con los archivos optimizados para producción.

A.4. Manual de usuario

En esta sección se repasarán las diferentes pantallas añadidas o modificadas a la aplicación y una explicación de uso de cada una de estas.

A.4.1. Menú principal

La pantalla inicial de la aplicación se muestra en las Figuras de la A.1 a la A.4. En estas pantallas se aprecia una modificación en relación con la versión anterior. Sobre el botón de inicio de sesión, una vez autenticado el usuario, se ven sus diferentes opciones en función de su rol en la aplicación. Estas opciones se corresponden con aquellas acciones personales de cada persona registrada.



Figura A.1: Menú principal como estudiante



Figura A.2: Menú principal como Tutor UVa



Figura A.3: Menú principal como Coordinador de Título



Figura A.4: Menú principal como Delegado de empresas

A.4.2. Propuestas

Desde esta pantalla A.5 los tutores UVa pueden consultar sus propuestas de TFG. En el caso del Delegado de empresas, puede ver únicamente las publicadas por él. Además, para

diferenciar las propuestas públicas en la vista principal A.6, cuentan con una notación y un filtro. Otra incorporación, tanto en esta vista como en la de Adjudicaciones, es la opción de ordenar alfabéticamente las tablas por Título o Empresa haciendo click sobre la cabecera de la tabla.

INFUVA - TFG

Adjudicaciones

Defensas

Propuestas

Tribunales

Yania Crespo

Mis Propuestas

Crear

Filtrar por

Todas

TÍTULO	TUTOR	ESTUDIANTE	TIPO	EMPRESA
> prueba de realidad	Yania Crespo			
> segunda oposición	Yania Crespo	NEREA LAZARINA VILLUELAS		

Rows per page: 15

1-2 of 2

<

<

>

>

Borja Arriola Pérez y Diego Bombón Sanz - 2023-2025

Figura A.5: Listado personal de propuestas publicadas por Tutor UVa

INFUVA - TFG

Adjudicaciones

Defensas

Propuestas

Tribunales

Iniciar sesión

Propuestas Publicadas

Filtrar por

Todas

TÍTULO	TUTOR	TIPO	EMPRESA
> prueba de realidad	Yania Crespo		
> oposición de realidad		Convenio	Empresa A

Rows per page: 15 + 1-2 of 2 |< < > >|

Borja Arriola Pérez y Diego Bombón Sanz - 2023-2025

Figura A.6: Listado de propuestas publicadas

Crear propuesta en convenio

Siendo la persona autenticada en la aplicación la definida con el rol de Delegado de empresas, este puede crear propuestas en convenio como se ve en la siguiente pantalla A.7. En esta deberá añadir toda la información de la que dispone en ese momento que es obligatoria. En caso de no disponer de alguno de los campos deberán ser los no obligatorios. Estos campos los completaría el Comité de Título.

NUEVA - TFG

AsignacionesDefensasPropuestasTribunalesCalificaciones

Nueva propuesta

* el campo es obligatorio

Título*

Descripción *

Observaciones *

Difficultad *Areas temáticas *

Es de Investigación

Technologies *

Información de la empresa

Nombre de la entidad *Email de contacto de la empresa *

Tutor externo

Email del tutor externoNombre del tutor externo

Prácticas curriculares

¿Prácticas curriculares remuneradas?Estado de las prácticas curriculares

Prácticas extra curriculares

¿Prácticas extra curriculares remuneradas?Estado de las prácticas extra curriculares

Estudiante

NombreE-mail

CANCELAR

CREAR

Diego Andrés Pérez y Diego Borrán Soto - 2023-2026

Figura A.7: Creación de propuesta en convenio

Más información sobre las propuestas

Desde cualquiera de las vistas en las que se listan las propuestas, y desplegando la información asociada a una propuesta, podemos visualizar en función del rol diferentes secciones y acciones. Con el rol de Coordinador de la titulación, o en quien delegue, se pueden apreciar las autopropuestas de los profesores sobre las propuestas en convenio A.8. Esto quiere decir que son profesores que se han autopropuesto para ser el tutor UVa necesario, aparte del tutor de empresa. También con este rol, está disponible el botón con el que completar la información que falta para asignar o adjudicar dicha propuesta. Además, al incluir las propuestas de convenio, esto ha hecho que se incluyan los campos que estas contienen en la vista de los detalles A.9.

Con el rol de Delegado de empresas la vista es algo diferente, pudiendo ver las solicitudes al igual que un tutor UVa sobre su TFG A.10 o el botón para seleccionar un estudiante para realizar el TFG propuesto.

77

A.4.3. Mis solicitudes

La única vista personal que tienen los estudiantes es la de Mis solicitudes A.11, en la cual se pueden ver las solicitudes que ha realizado el estudiante y el estado en el que se encuentran.

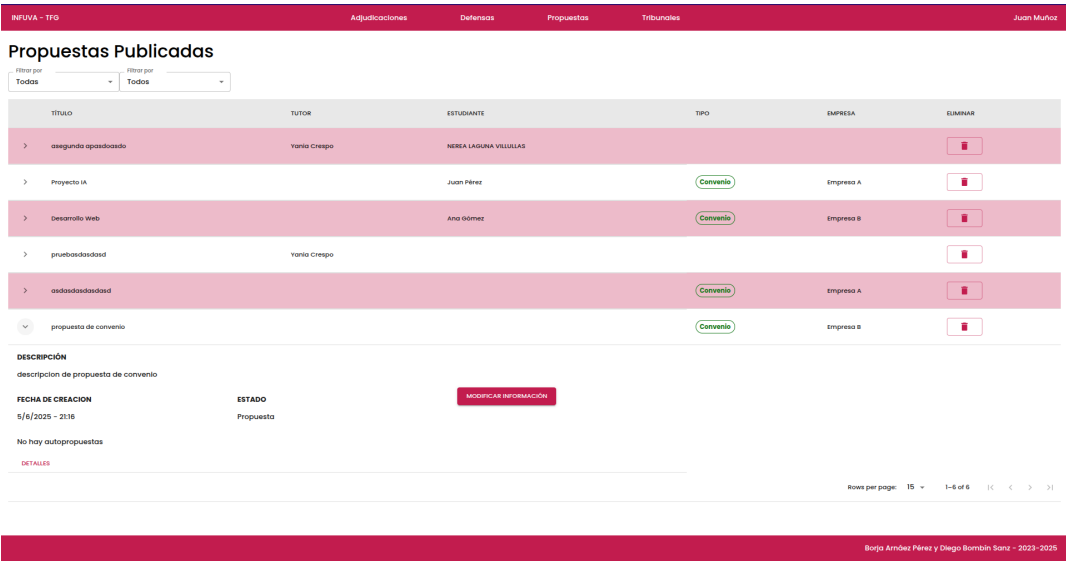


Figura A.8: Vista de la información de una propuesta con rol de Coordinador de Título

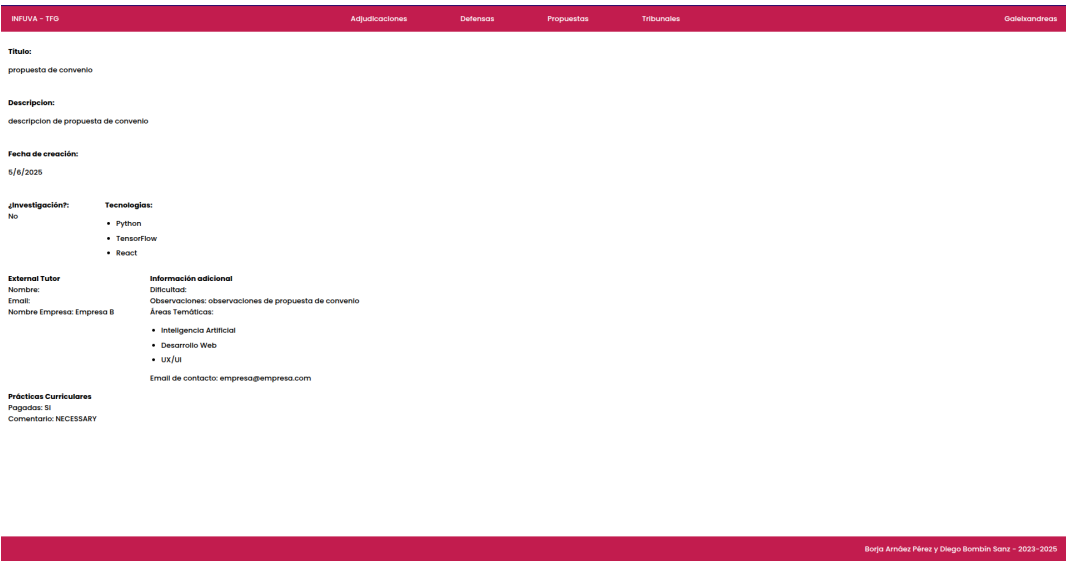


Figura A.9: Vista de más detalles de propuesta de convenio

RFUVA - TFG

AdjudicacionesDefensasPropuestasTribunales

Guillem Andreu

Propuestas Publicadas

Crear Propuesta Convenio

Filtrar por

Todas

TÍTULO	TUTOR	ESTUDIANTE	TIPO	EMPRESA
> pruebasdadad	Yanis Cerezo			
> asdadadadadad			Convenio	Empresa A
✓ propuesta de convenio			Convenio	Empresa B

DESCRIPCIÓN

descripcion de propuesta de convenio

FECHA DE CREACION

5/6/2025 - 2116

ESTADO

Propuesta

SELECCIONAR ESTUDIANTE

DETALLES

Rows per page: 15

1-3 of 3

<<<>>>

Figura A.10: Vista de la información de una propuesta con rol de Delegado de empresas

RFUVA - TFG

AdjudicacionesDefensasPropuestasTribunales

Miguel Cubino

Mis Solicitudes

TÍTULO	TUTOR	FECHA SOLICITUD	ESTADO
pruebasdadad	yancrespiguas	5/6/2025 - 2100	PENDING

Rows per page: 15

1-1 of 1

<<<>>>

Figura A.11: Mis solicitudes

A.4.4. Mis autopropuestas

Otra de las vista personal que tienen los tutores UVa es la de Mis Autopropuestas A.12, en la cual se pueden ver los TFGs en convenio en los cuales el tutor identificado en la aplicación

se ha propuesto para tutorizar.

RPUVA - TRS

Adjudicaciones

Defensas

Propuestas

Tribunales

Yanis Crespo

Mis Auto-Propuestas

TÍTULO	EMPRESA	FECHA
oasdasdasdasdad	Empresa A	5/6/2025 - 20:51

Rows per page:

15

1-1 of 1

Borja Arriola Pérez y Diego Bombón Sanz - 2023-2025

Figura A.12: Mis autopropuestas

A.4.5. Tribunales

En la pantalla A.13 de los tribunales podemos apreciar el listado de tribunales publicados. Para poder crearlos, editarlos o eliminarlos solamente puede hacerlo el usuario con rol de Coordinador de Título. Las pantallas A.14 tanto de edición como de creación cuentan con la modificación para que haya varios suplentes.

Borja Arnedo Pérez y Diego Bombín Sanz - 2023-2025

Borja Arnedo Pérez y Diego Bombín Sanz - 2023-2026

Apéndice B

Resumen de enlaces adicionales

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

- Repositorios:

Frontend : https://gitlab.inf.uva.es/tfgs/gestiontfgseiinfva/infuva_tfg_manager_webapp/

Backend : https://gitlab.inf.uva.es/tfgs/gestiontfgseiinfva/infuva_tfg_manager_service/

Documentación : https://gitlab.inf.uva.es/tfgs/gestiontfgseiinfva/tfg_documentacion/

- URL del sistema desplegado: <https://tfgs.inf.uva.es>

- URL para ver la documentación de la API: <https://tfgs.inf.uva.es:8443/swagger-ui/index.html>