



UNIVERSIDAD DEL BÍO-BÍO, CHILE

FACULTAD DE CIENCIAS EMPRESARIALES

Departamento de Sistemas de Información

AcTitUBB: SOFTWARE DE COORDINACION DE ACTIVIDADES DE TITULACION DE CARRERAS INFORMATICAS FACE-UBB.

PROYECTO DE TÍTULO
INGENIERÍA DE EJECUCIÓN EN COMPUTACIÓN E INFORMÁTICA
DANIEL ABRAHAM AGUAYO MUÑOZ
DIRIGIDA POR
LUIS CABRERA CROT

2025

Resumen

Resumen

El presente proyecto de titulación aborda la problemática de la gestión manual y desorganizada del proceso de titulación en las carreras de informática de la Facultad de Ciencias Empresariales (FACE) de la Universidad del Bío-Bío. Actualmente, la coordinación de propuestas, asignación de profesores guías e informantes, seguimiento de avances, programación de reuniones y control de entregas se realiza mediante correos electrónicos y hojas de cálculo, lo que genera pérdida de información, dificultades en la trazabilidad de los procesos, deficiencias en la coordinación entre los actores involucrados y problemas en el cumplimiento de plazos académicos.

Como solución a esta problemática, se desarrolló AcTitUBB, un sistema web full-stack que centraliza y automatiza la gestión integral del proceso de titulación. El sistema fue implementado utilizando tecnologías modernas: Angular 18 para el frontend con arquitectura de componentes standalone, Node.js con Express para el backend siguiendo el patrón MVC, MySQL 8.0 como gestor de base de datos relacional, y Docker para la contenedorización y despliegue del sistema completo.

La aplicación integra módulos diferenciados por roles (estudiante, profesor y administrador) que abarcan funcionalidades clave del proceso académico: gestión completa del ciclo de vida de propuestas (desde borrador hasta aprobación automática que genera el proyecto de titulación), sistema flexible de asignación de profesores en cinco roles específicos (guía, co-guía, informante, revisor y de sala), seguimiento del estado del proyecto mediante catorce estados configurables que reflejan el avance real, gestión de hitos y cronogramas personalizados con validaciones automáticas de fechas límite, sistema de calendario inteligente con algoritmo de matching de disponibilidades horarias entre estudiantes y profesores, notificaciones automáticas por correo electrónico ante eventos críticos, y dashboards personalizados con métricas en tiempo real según el rol del usuario.

La implementación del sistema incluye características técnicas avanzadas como autenticación mediante JSON Web Tokens (JWT) con sistema de blacklist para invalidación segura de sesiones, gestión robusta de archivos con validaciones de tipo y tamaño mediante middleware Multer, manejo de transacciones en base de datos para garantizar la integridad referencial, sistema de auditoría completo mediante tablas de historial que registran todas las acciones realizadas, diseño *responsive* que garantiza usabilidad en dispositivos móviles y de escritorio, y arquitectura modular que facilita el mantenimiento y la escalabilidad del sistema.

Como resultado del proyecto, se espera reducir los tiempos de gestión administrativa en un 60–70 % mediante la automatización de procesos repetitivos, mejorar la comunicación entre estudiantes y profesores mediante un canal centralizado con retroalimentación estructurada, eliminar la pérdida de información crítica gracias a la persistencia centralizada de documentos y registros, optimizar el control de plazos académicos mediante alertas automáticas configurables, y proporcionar a la coordinación académica visibilidad completa y en tiempo real del estado de todos los proyectos de titulación activos mediante reportes y métricas integrados.

Este proyecto contribuye a la modernización digital de la Universidad del Bío-Bío mediante la aplicación de metodologías ágiles de desarrollo de software, patrones de diseño consolidados en la industria, y buenas prácticas de ingeniería de software aplicadas a la solución de problemas académicos reales. El sistema establece un precedente para la transformación digital de otros procesos administrativos de la institución, demostrando el valor de la automatización y centralización en la gestión de procesos académicos complejos que involucran múltiples actores con roles diferenciados.

Palabras Clave — Sistema de gestión académica, proceso de titulación, aplicación web full-stack, Angular, Node.js, automatización de procesos, coordinación académica, Universidad del Bío-Bío.

Agradecimientos

Deseo expresar mi más sincero agradecimiento a todas las personas que, de diversas maneras, me acompañaron, apoyaron y motivaron durante el proceso de desarrollo de este proyecto. Este logro es más que un resultado personal; es la suma del cariño, la paciencia y la guía que recibí a lo largo de mi vida.

En primer lugar, quiero agradecer profundamente a mis profesores y académicos, quienes con dedicación, compromiso y profesionalismo aportaron de manera fundamental a mi formación. Cada orientación, cada corrección y cada palabra de apoyo contribuyeron al desarrollo de este trabajo. Gracias por ser un ejemplo de disciplina, conocimiento y vocación.

A mi familia, quiero expresarles toda mi gratitud. Su apoyo constante, su comprensión en los momentos de cansancio y su firme creencia en mis capacidades fueron pilares esenciales para seguir adelante. Sin ustedes, este camino habría sido mucho más difícil.

Quiero dedicar unas palabras especialmente sentidas a la memoria de mi madre. Aunque su ausencia ha marcado profundamente mi vida, su amor perdura en cada uno de mis recuerdos y en las enseñanzas que alcanzó a entregarme en tan poco tiempo. Gracias, mamá, por tu cariño infinito y por haber sembrado en mí la semilla de la fortaleza y la sensibilidad. A pesar de que no pudiste acompañarme físicamente en este camino, tu recuerdo ha sido siempre una fuente de inspiración y consuelo. Este logro es también para ti, con el profundo deseo de que, donde estés, puedas sentir el orgullo que siempre quise darte.

A mis amigos, gracias por su apoyo constante, por las conversaciones que me devolvieron la energía cuando más la necesitaba, por los momentos de descanso y por cada palabra de ánimo. Su amistad ha sido un pilar invaluable durante esta etapa.

A todos ustedes, gracias por haber sido parte esencial de mi camino, por acompañarme en los momentos difíciles y por celebrar conmigo cada logro. Este trabajo no solo refleja mi esfuerzo, sino también el de todas las personas que han formado parte de mi vida.

Índice General

1. Introducción	1
1.1. Presentación del área y su problemática	1
1.2. Propuesta de Solución	6
1.3. Análisis de los Principales Trabajos Realizados en el área de la propuesta	7
1.4. Justificación de la Propuesta	8
1.5. Objetivos del proyecto	9
1.6. Composición del Informe	9
2. Proyecto	11
2.1. Objetivos del Software	11
2.2. Metodología de Desarrollo	12
2.3. Tecnologías Utilizadas	14
2.4. Estándares de Documentación	14
2.5. Técnicas y notaciones	15
2.6. Herramientas, framework, lenguaje usados en el desarrollo del proyecto	15
2.7. Factibilidad del Proyecto	16
3. Especificación de Requerimientos - Producto de Software	24
3.1. Límites	24
3.2. Restricciones Técnicas	25
3.3. Requerimientos Funcionales	26
3.4. Requerimientos No Funcionales	29
3.5. Interfaces Externas de Entrada	31
3.6. Interfaces Externas de Salida	32
4. Análisis - Casos de Uso	34
4.1. Historias de Usuario	34
4.2. Actores de casos de uso	38
4.3. Diagramas y Especificación de casos de uso	38
4.4. Matriz de Trazabilidad de Requisitos	46

5. Diseño	48
5.1. Diseño de Servicios Web (API REST)	48
5.2. Diseño de Datos	49
5.3. Diseño e interfaz de navegación	52
6. Desarrollo del Trabajo	56
6.1. Diseño de Arquitectura	56
6.2. Estructura del Código	59
6.3. Desarrollo del Software	61
6.4. Despliegue y Operación	62
6.5. Conclusión del Capítulo	63
7. Implantación y Puesta en Marcha	65
7.1. Plan de Capacitación	65
7.2. Estrategia de Implantación y Puesta en Marcha	66
8. Conclusiones	69
Referencias	71
A. Definiciones y abreviaciones del Negocio	74
A.1. Definiciones y abreviaciones del Negocio	74
B. Pruebas de Aceptación	76
C. Recopilación de Información	77
C.1. Técnicas Aplicadas	77
C.2. Reuniones con Profesor Guía	78
C.3. Observación Participante	84
C.4. Revisión de Documentación	91
C.5. Consolidación de Resultados	95
C.6. Conclusiones	96
D. Diccionario de datos	98
D.1. Tablas Principales	98
D.2. Diagrama de Relaciones	105
D.3. Convenciones y Estándares	105
E. Aspectos de gestión de proyectos	106
E.1. Carta Gantt con línea base y desviaciones	106
E.2. Riesgos de Alto Nivel (Amenazas), Impacto y Estrategia	108
E.3. Estimación CU (Use Case Points - UCP)	108
E.4. Resumen de Esfuerzo Real	111
E.5. Retrospectiva Proyecto	111

E.6. Análisis Éxito / Fracaso del Proyecto	112
E.7. Riesgos que se Concretaron y Efectos / Consecuencias	112
E.8. Análisis de Ajuste: Planificación vs Esfuerzo Real vs Estimación UCP	112
E.9. Comparación y Análisis de Resultados	113
E.10. Conclusiones	113
E.11. Iteraciones en el desarrollo	113

Índice de Figuras

1.1. Asignatura Proyecto Final IECI (Universidad del Bío-Bío, 2025d)	2
1.2. Asignaturas de Título ICINF (Universidad del Bío-Bío, 2025c)	2
1.3. Datos de alumnos Ieci Titulados en 2024, sede concepcion FACE, (Dirección General de Análisis Institucional, Universidad del Bío-Bío, 2025a)	2
1.4. Datos de alumnos INCIF titulados 2024 sede concepción FACE,(Dirección General de Análisis Institucional, Universidad del Bío-Bío, 2025a)	3
1.5. docentes 2025 depto. de sistemas de información FACE., (Dirección General de Análisis Institucional, Universidad del Bío-Bío, 2025b)	3
2.1. Esquema de la metodología iterativa aplicada al proyecto (Fuente: PNGWing, 2025)	13
2.2. Fórmula del Valor Actual Neto (VAN) para evaluación de proyectos	22
4.1. Diagrama de Casos de Uso: Visión General del Sistema	38
4.2. Diagrama de Casos de Uso: Gestión de Propuestas	42
4.3. Diagrama de Casos de Uso: Gestión de Proyectos	44
5.1. Diagrama de la estructura académica y gestión de usuarios.	50
5.2. Diagrama del ciclo de vida: De Propuesta a Proyecto y asignaciones.	51
5.3. Diagrama del subsistema de seguimiento, hitos y reuniones.	52
5.4. Escudo UBB	53
5.5. Colores Principales	53
5.6. Escala de Grises	53
5.7. Colores de Estado	53
5.8. Sistema de Identidad Visual y Paleta Cromática	53
5.9. Gradiente Azul	53
5.10. Gradiente Gris	53
5.11. Gradiente Rojo	53
5.12. Gradientes corporativos aplicados en la interfaz	53
5.13. Vista de Login	54
5.14. Pag principal Estudiante	54
5.15. Pag principal de Profesores: Gestión de asignaciones y revisiones	55
5.16. Módulo de Calendario	55
5.17. Formulario de Propuesta	55

5.18. Interfaces de gestión operativa del sistema	55
6.1. Diagrama de la Arquitectura del Sistema AcTitUBB	57

Índice de Tablas

1.1. Comparativa de herramientas de gestión de proyectos existentes	8
2.1. Características del proyecto	12
2.2. Tecnologías y herramientas utilizadas en el proyecto	14
2.3. Lenguajes, frameworks y herramientas utilizadas	16
2.4. Especificación de software requerido para el desarrollo del proyecto	17
2.5. Especificación de hardware utilizado en el desarrollo del proyecto	18
2.6. Especificación de hardware para servidores de desarrollo y explotación	18
2.7. Licencias y costo del software utilizado en el proyecto	20
2.8. Costo de hosting y dominio para el proyecto (NIC Chile, 2025)	20
2.9. Cálculo estimado del costo de desarrollo y soporte.	20
2.10. Proyección de flujos de caja para evaluación económica del proyecto (valores en CLP)	21
2.11. Términos de la fórmula de VAN	22
2.12. Cálculo del VAN	22
3.1. Especificación de Requerimientos Funcionales del sistema AcTitUBB	26
3.2. Especificación de interfaces externas de entrada del sistema	32
3.3. Especificación de interfaces externas de salida del sistema	32
4.1. Historias de Usuario – Rol Estudiante	35
4.2. Historias de Usuario – Rol Profesor	37
4.3. Historias de Usuario – Rol Admin	37
4.4. Actores del Sistema AcTitUBB	38
4.5. Matriz de Trazabilidad RF vs CU	46
6.1. Mapeo de puertos para servicios en producción	58
6.2. Configuración de URLs y puertos en el servidor de producción	63
C.1. Resumen de técnicas de recopilación aplicadas	77
C.2. Priorización de requisitos con MoSCoW	80
C.3. Problemas identificados en el proceso de propuestas	85
C.4. Resumen de problemas observados	90

C.5. Documentos institucionales revisados	91
C.6. Requisitos derivados de documentación oficial (Reglamento 2025)	93
C.7. Requisitos derivados de documentación oficial	94
C.8. Muestra de requisitos funcionales y técnicas de respaldo	95
C.9. Requisitos no funcionales validados	96
C.10. Métricas de mejora esperadas	96
E.1. Cronograma de Ejecución Iterativa (Marzo - Noviembre)	107
E.2. Riesgos, impacto, estrategia y estado	108
E.3. Desglose de horas reales invertidas	111

Capítulo 1

Introducción

La gestión del proceso de titulación en las carreras de informática de la Facultad de Ciencias Empresariales (FACE) de la Universidad del Bío-Bío presenta desafíos persistentes asociados al uso de correos electrónicos y hojas de cálculo como medios principales de coordinación. Esta fragmentación dificulta la trazabilidad de hitos, la comunicación entre actores y el cumplimiento de plazos académicos. En respuesta, se desarrolla AcTitUBB, una plataforma web centralizada que automatiza y ordena el ciclo de titulación mediante la gestión de propuestas, asignación de profesores por roles, seguimiento de proyectos con hitos y cronogramas, programación de reuniones con compatibilidad de disponibilidad y notificaciones automáticas. El proyecto adopta una arquitectura full-stack basada en Angular, Node.js/Express, MySQL y Docker, incorpora autenticación segura y registra auditorías completas de las acciones. Con ello se busca reducir tiempos administrativos, mejorar la coordinación y aportar a la modernización institucional mediante la digitalización integral del proceso.

1.1. Presentación del área y su problemática

La Universidad del Bío-Bío (UBB), institución estatal heredera de la tradición de la Universidad Técnica del Estado (1947) y el IPROCH, se consolida hoy como un referente regional en docencia e investigación. El proyecto se enmarca en la Facultad de Ciencias Empresariales (FACE) ([Universidad del Bío-Bío, 2025f](#)), específicamente en el Departamento de Sistemas de Información, responsable de las carreras de Ingeniería de Ejecución en Computación e Informática (IECI) e Ingeniería Civil en Informática (ICINF).

La presente iniciativa busca implementar un piloto tecnológico para optimizar la gestión de la actividad de titulación, proceso crítico supervisado por dicho departamento, garantizando un seguimiento eficiente y moderno para docentes y estudiantes.

1.1.1. Presentación de los procesos del ámbito del proyecto en la Empresa

La actividad de titulación es una asignatura fundamental dentro del plan de estudios de los programas académicos profesionales, cuyo objetivo principal es permitir que los estudiantes

apliquen los conocimientos adquiridos a lo largo de su carrera en un contexto práctico, desarrollando un proyecto que resuelva una problemática real en el área de estudio correspondiente. Este proyecto les da la oportunidad de demostrar las competencias generales y profesionales que el estudiante ha adquirido durante su formación y que se establecen en su perfil. (Face, 2025)

En la carrera de Ingeniería de Ejecución en Computación e Informática (IECI por sus siglas en español), la actividad de titulación se lleva a cabo en el semestre 8 Como se muestra en la figura 1.1 (Universidad del Bío-Bío, 2025b), mientras que en la carrera de Ingeniería Civil en Informática (ICINF por sus siglas en español), la misma actividad está programada para el semestre 10, como se muestra en la figura 1.2 (Universidad del Bío-Bío, 2025a). Estas actividades son supervisadas por un comité de profesores del Departamento de Sistemas de información, quienes se encargan de guiar a los estudiantes en sus proyectos, asegurando que cumplan con los objetivos académicos y profesionales definidos por la universidad.

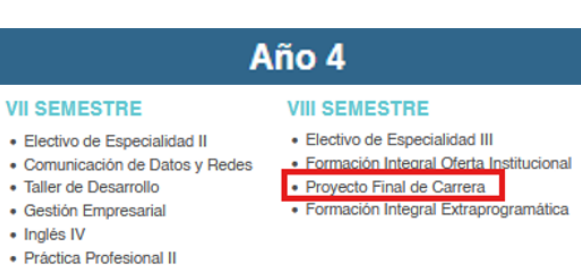


Figura 1.1: Asignatura Proyecto Final IECI (Universidad del Bío-Bío, 2025d)

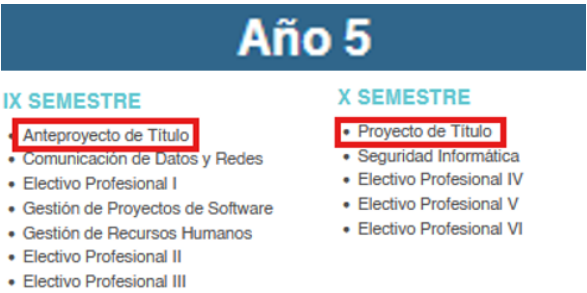


Figura 1.2: Asignaturas de Título ICINF (Universidad del Bío-Bío, 2025c)

Los profesores que guían o dirigen las tesis de los alumnos de informática en la sede Concepción pertenecen al Departamento de Sistemas, especializado en áreas como desarrollo de software, bases de datos, inteligencia artificial, redes y ciberseguridad. Estos docentes juegan un papel crucial en el desarrollo del proyecto de titulación, no solo ofreciendo su experiencia académica, sino también brindando orientación sobre las tendencias actuales de la industria y las mejores prácticas del sector. Tal como se detalla en las tablas 1.3 y 1.4, al año 2024, 105 alumnos se titularon en ambas carreras, es decir, al año lo cual nos permite decir que al menos existen 52 proyectos al año, dirigidos por los 17 profesores del departamento como se muestra en la figura 1.5.

Año	Código	Carrera	Sede	Campus	Titulados (as)	Nº Mujeres	% Mujeres	Nº Hombres	% Hombres
2024	29037	INGENIERÍA DE EJECUCIÓN EN COMPUTACIÓN E INFORMÁTICA	Concepción	Concepción	50	3	6.0 %	47	94.0 %

Figura 1.3: Datos de alumnos Ieci Titulados en 2024, sede concepcion FACE, (Dirección General de Análisis Institucional, Universidad del Bío-Bío, 2025a)

Año	Código	Carrera	Sede	Campus	Titulados (as)	Nº Mujeres	% Mujeres	Nº Hombres	% Hombres
2024	29027	INGENIERÍA CIVIL EN INFORMÁTICA	Concepción	Concepción	55	3	5.5 %	52	94.5 %

Figura 1.4: Datos de alumnos INCIF titulados 2024 sede concepción FACE, (Dirección General de Análisis Institucional, Universidad del Bío-Bío, 2025a)

De los 17 profesores disponibles para guiar proyectos de titulación, no todos ellos pueden asumir la misma cantidad de estudiantes debido a la carga administrativa que tienen en otros roles dentro de la universidad, como Decanatura, Jefatura de Carrera, Director de Departamento u otro cargo. Por esta razón, es esencial un proceso de distribución equitativa que garantice que todos los alumnos reciban una orientación adecuada y personalizada durante el desarrollo de su proyecto final.

Para comprender cómo se realizan los procesos hoy en día en el área de seguimiento de actividades de titulación, es fundamental describir tanto los pasos involucrados como los actores que participan en cada fase del proceso.

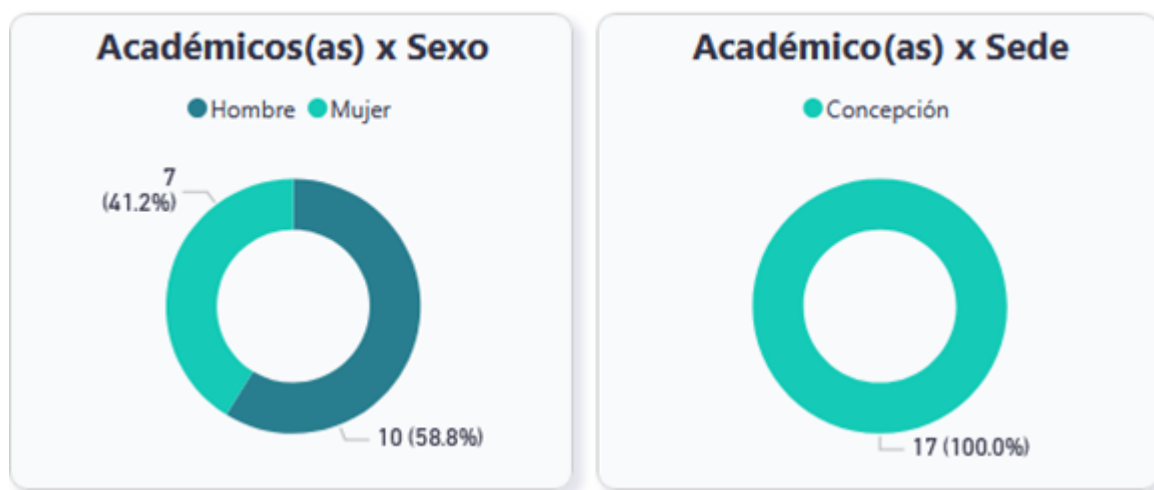


Figura 1.5: docentes 2025 depto. de sistemas de información FACE., (Dirección General de Análisis Institucional, Universidad del Bío-Bío, 2025b)

- **Inicio de clases:** Se informa sobre la charla informativa y el formato para la presentación de propuestas de Proyecto de titulación (PT).
- **Asignación:** Etapa previa a la formulación del documento, donde se realiza la asignación oficial del tema a desarrollar. En esta fase, el estudiante acuerda la temática a desarrollar o selecciona una de los temas disponibles, antes de proceder a la escritura formal de la propuesta.
- **Recepción de Propuestas de Proyectos:** Las propuestas deben enviarse a través de la plataforma institucional en formato PDF con un nombre de archivo específico. No se aceptan entregas por correo electrónico.

- **Revisión de propuestas:** El encargado del curso solicita a una comisión de profesores que evalúen la propuesta, con un plazo de revisión de 3 a 5 días.
- **Resultados de Evaluación de Propuestas:** Se informan los resultados por correo electrónico. En caso de que las propuestas requieran ser replanteadas, los alumnos deben re-enviarlas modificadas.
- **Inicio del desarrollo del proyecto:** Se da comienzo a la ejecución del proyecto de titulación.
- **Evaluación de avances:** Durante la ejecución del proyecto el profesor guía revisa el progreso realizado por el estudiante. En esta instancia se califica el grado de cumplimiento de los objetivos planificados hasta la fecha.
- **Entrega del Informe Final:** El informe debe estar firmado o acompañado de un email que confirme la conformidad del profesor guía. Luego se entrega al profesor informante para su evaluación.
- **Revisión de Informes y Software:** Las semanas de revisión están divididas en períodos específicos para la revisión del informe y del software entregado.
- **Defensa:** Una vez revisado el informe y el producto software, el alumno junto a los profesores guía e informante coordinan la fecha y sala. Las notas se formalizan en un acta enviada a la Jefatura de Carrera. En dicha acta es obligatorio consignar tanto la nota de avance como la calificación del informe final, validadas por ambos profesores.
- **Cierre de Actas y Calificaciones:** Se finaliza el proceso con la entrega oficial de calificaciones y cierre de actas.
- **Segunda Fecha para Entrega (con costo):** Opción disponible para estudiantes que necesiten más tiempo, pagando la Cuota Básica de Matrícula. debido a que la entrega se realiza desfasada, la defensa se programa para el semestre siguiente.

1.1.2. Descripción del Problema

El proceso actual de seguimiento de tesis, aunque está estructurado y con fechas bien definidas, presenta varios desafíos y limitaciones que afectan tanto a los estudiantes como a los profesores. A continuación, se describen los problemas y áreas que pueden mejorarse:

1. Falta de centralización del seguimiento:

Actualmente, el seguimiento del estado de las tesis depende en gran medida de la comunicación directa entre los estudiantes y los profesores guía, a menudo a través de correos electrónicos o reuniones presenciales. Esta falta de un sistema centralizado hace que el proceso sea menos transparente y puede generar confusión o retrasos en la comunicación. Para los profesores, el manejo de múltiples tesis sin un sistema centralizado puede ser una carga administrativa significativa, lo que puede llevar a cometer errores en el seguimiento de los proyectos o incluso que a algunos alumnos no se les haga un seguimiento adecuado, resultando en la pérdida de información y en retrasos en el proceso de evaluación. Esto afecta tanto a los docentes como a los estudiantes, quienes podrían no recibir el apoyo necesario a tiempo.

2. Dependencia de correo electrónico para notificaciones:

El sistema actual basa su comunicación crítica con los estudiantes y profesores únicamente en correos electrónicos para notificarles sobre la recepción de documentos, revisiones y resultados de evaluaciones. Este enfoque presenta varios problemas específicos. Primero, existe el riesgo de que los correos electrónicos se pierdan en las bandejas de entrada de los estudiantes y/o profesores, especialmente si tienen una alta cantidad de mensajes o utilizan filtros que podrían redirigir correos importantes a carpetas no revisadas regularmente.

Además, los estudiantes y/o profesores podrían no revisar sus correos electrónicos con suficiente frecuencia, lo que podría causar retrasos en la respuesta o en la entrega de documentos necesarios para el avance de la tesis. También, los correos electrónicos de notificación pueden mezclarse con otros mensajes no relacionados, haciendo que los estudiantes y profesores pasen por alto comunicaciones importantes o confundan las notificaciones críticas con mensajes menos urgentes.

Otro problema es que, en algunos casos, los correos electrónicos pueden no ser entregados correctamente debido a problemas con los servidores de correo, filtros de spam, o errores en la dirección de correo electrónico proporcionada.

3. Visibilidad limitada del proceso de revisión:

Los estudiantes no tienen visibilidad en línea sobre el estado de la revisión de sus informes o propuestas. Esto significa que, después de enviar un documento, deben esperar pasivamente una respuesta del profesor guía o profesores informantes, sin saber en qué etapa se encuentra la revisión.

La falta de claridad sobre el estado de revisión de propuestas o informes genera incertidumbre y ansiedad en los estudiantes, afectando su capacidad para planificar los siguientes pasos en sus proyectos. Esta falta de visibilidad dificulta la gestión eficiente del tiempo y puede retrasar el avance académico. Además, los profesores y jefes de carrera, al ser la única fuente de información sobre el progreso, pueden sentirse sobrecargados, ya que deben proporcionar actualizaciones de manera constante a varios estudiantes, lo que aumenta su carga administrativa.

4. Carga administrativa para los profesores:

Los profesores guías deben gestionar múltiples tesis, realizar revisiones y mantener una comunicación constante con cada estudiante de manera individual. Sin un sistema automatizado, esta carga administrativa puede volverse considerable, aumentando el riesgo de errores o retrasos en el proceso de revisión.

La falta de automatización no solo incrementa el tiempo y esfuerzo dedicados a cada tarea, sino que también dificulta el seguimiento eficiente del progreso de los estudiantes, lo que puede afectar la calidad y la rapidez del proceso de retroalimentación.

Los problemas identificados subrayan la necesidad de un sistema centralizado y automatizado para el seguimiento de proyectos de título que permita una mejor gestión del proceso tanto para estudiantes como para profesores. La implementación de dicho sistema facilitaría la comunicación, mejoraría la visibilidad del estado de los proyectos, y ofrecería mayor flexibilidad en la gestión de plazos. Al abordar estos problemas, el sistema propuesto no sólo optimizará el proceso de

titulación, sino que también reducirá la carga administrativa para los profesores y aumentará la satisfacción y el éxito de los estudiantes.

1.2. Propuesta de Solución

Cómo apoyo para los problemas identificados en el proceso actual de seguimiento de proyecto, se propone desarrollar un sistema centralizado de gestión de proyectos de título que permita a los estudiantes y profesores visualizar el estado de estos proyectos, sistematizar las notificaciones, facilitar la comunicación interna y permitir un seguimiento eficaz por parte de los distintos actores involucrados en este proceso. Esta propuesta mejorará la transparencia, reducirá la carga administrativa, proporcionará flexibilidad en la gestión de plazos y hará más eficiente el proceso de evaluación, beneficiando a todos los actores involucrados en el proceso de titulación.

El sistema está orientado a facilitar y transparentar el desarrollo de los proyectos de título, tanto para los estudiantes como para los profesores, abarcando desde la presentación de la propuesta inicial hasta la entrega del informe final. Permitirá a los estudiantes seguir su progreso desde la confirmación de la recepción por parte del profesor, hasta la revisión y aprobación final. Cada fase del proceso estará documentada, lo que brindará a estudiantes y profesores visibilidad del estado actual de cada tarea y sus respectivas responsabilidades. Esto no solo mejorará la comunicación entre los actores involucrados, sino que también agilizará el proceso de seguimiento, asegurando que todos los requisitos sean cumplidos en tiempo y forma.

Características del proyecto:

- 1. Centralización de la Información:** Todos los datos relevantes de los proyectos de título (estado de revisiones, comentarios de los profesores, fechas de entrega, etc.) estarán accesibles desde una única plataforma, lo que facilita la consulta y gestión por parte de estudiantes y profesores.
- 2. Visualización del estado del proyecto:** Los estudiantes podrán visualizar el estado actualizado de su proyecto en cada fase, desde la presentación de la propuesta hasta la aprobación final. Los profesores podrán revisar el progreso y gestionar sus tareas de revisión de manera organizada.
- 3. Sistematización de Notificaciones:** El sistema enviará notificaciones automáticas dentro de la aplicación y por correo, tanto a estudiantes como a profesores cuando se realicen cambios en el estado de un proyecto, como la recepción de documentos, la revisión completada o la solicitud de modificaciones.
- 4. Facilidad en la Comunicación:** Incluye una herramienta interna de mensajería que permitirá una comunicación fluida entre los estudiantes y sus profesores guías. Si bien medios externos como el correo electrónico constituyen una evidencia válida con fecha y hora, su uso aislado puede dispersar el seguimiento. Esta herramienta centraliza las interacciones

para evitar la confusión o pérdida de información que ocurre al manejar múltiples canales

- 5. **Documentación de cada Fase:** Cada etapa del proceso, desde la recepción de la propuesta hasta la revisión y aprobación del informe final, quedará registrada. Esto permitirá una trazabilidad completa, facilitando la transparencia y asegurando que se cumplan todos los pasos necesarios.
- 6. **Gestión de Plazos y Tareas:** Profesores y estudiantes podrán visualizar fechas límite y tareas pendientes, permitiendo una mejor planificación y cumplimiento de los plazos de entrega. El sistema permitirá configurar recordatorios automáticos.
- 7. **Historial de Revisión:** El sistema almacenará un registro detallado de las versiones de los archivos subidos, así como de los comentarios de texto realizados, permitiendo que estudiantes y profesores recuperen entregas anteriores y visualicen la evolución del documento.
- 8. **Escalabilidad:** El proyecto será escalable, lo que permitirá agregar más funcionalidades o ajustarlo para su implementación en otras áreas académicas o instituciones educativas.

1.3. Análisis de los Principales Trabajos Realizados en el área de la propuesta

Se investigó en el buscador Google el día **25 de junio de 2025** con el propósito de identificar otras soluciones de software actualmente disponibles que faciliten la gestión de proyectos de titulación y en qué se diferencian del software propuesto para la Universidad del Bío-Bío. A continuación, se describen las soluciones encontradas:

Herramienta	Funciones y características	Costos (usuario/mes)	Diferencia con la propuesta
Trello	Organización mediante tableros, listas y tarjetas. Colaboración en tiempo real. Ideal para tareas simples.	Gratis / \$5 USD	El software propuesto incluye notificaciones académicas, gestión de plazos y visibilidad de tesis, aspectos no centrales en Trello.
Asana	Gestión de proyectos con vistas de lista, cronograma y tablero. Informes de progreso.	Gratis / \$10,99 USD	Asana es genérico. La propuesta se personaliza para la titulación (documentación automática y trazabilidad).
Notion	Gestión de notas, wikis y bases de datos. Muy personalizable con múltiples vistas.	Gratis / \$8 USD	Notion requiere configuración manual. El sistema propuesto automatiza procesos definidos como validaciones.

Herramienta	Funciones y características	Costos (usuario/mes)	Diferencia con la propuesta
Monday.com	Automatización de tareas, muy visual y con columnas personalizadas.	Gratis (2 us.) / \$9 USD	Pensado para industria general. La propuesta se adapta al flujo académico rígido y roles universitarios.
ClickUp	Gestión completa (Gantt, docs, tiempo). Personalización avanzada.	Gratis / \$7 USD	ClickUp es generalista y puede resultar abrumador. Este sistema ofrece control específico por fases.

Fuente: Elaboración propia, basada en información oficial de las plataformas [Atlassian Trello](#), [Asana Inc.](#), [Notion Labs Inc.](#), [monday.com Ltd.](#) y [ClickUp](#).

Tabla 1.1: Comparativa de herramientas de gestión de proyectos existentes

1.4. Justificación de la Propuesta

El desarrollo de un sistema centralizado que permita el seguimiento de proyectos de título responde directamente a los desafíos actuales que enfrentan tanto estudiantes como profesores en la Facultad de Ciencias Empresariales de la Universidad del Bío-Bío. La falta de visibilidad sobre el estado de las revisiones, el seguimiento de los avances y la comunicación entre los actores genera incertidumbre, retrasos y una carga administrativa significativa para los profesores guías. Estos problemas impactan negativamente en la planificación y el progreso de los proyectos de titulación, afectando tanto el rendimiento académico de los estudiantes como la eficiencia del proceso docente.

Aunque existen herramientas como Google Classroom Moodle y plataformas de gestión de proyectos como Trello, estas no han sido diseñadas específicamente para cubrir el proceso de titulación. Si bien permiten una organización general y comunicación básica, no abordan de manera eficiente aspectos clave como el seguimiento detallado de entregas, la gestión de múltiples revisiones ni el envío de notificaciones automáticas ajustadas a los plazos establecidos por la institución.

El sistema propuesto busca cubrir estas necesidades particulares mediante una solución personalizada, diseñada para la Universidad del Bío-Bío. A diferencia de herramientas genéricas, ofrecerá:

- **Seguimiento automatizado y detallado** de cada fase del proyecto, desde la propuesta inicial hasta la defensa final.
- **Visibilidad global para el encargado del Curso**, permitiéndole monitorear mediante un panel de control el estado de todos los estudiantes y el cumplimiento de los distintos profesores, facilitando la detección temprana de retrasos.
- **Notificaciones específicas** relacionadas con el estado de las entregas y revisiones, adaptadas a los plazos establecidos por la carrera.
- **Gestión centralizada de avances y revisiones**, permitiendo a los profesores realizar un seguimiento eficiente sin recurrir a herramientas externas que requieren procesos manuales adicionales.

Al centralizar la información y automatizar la comunicación, se disminuirá la carga administrativa para los profesores, se reducirán los errores humanos, y se facilitará una retroalimentación oportuna. Para los estudiantes, el sistema permitirá una mejor planificación de sus actividades, reduciendo la incertidumbre y aumentando la transparencia del proceso.

Este sistema contribuirá a un proceso de titulación más eficiente, organizado y claro, beneficiando a todos los actores involucrados. Además, presenta un potencial de escalabilidad y adaptación para su uso en otras facultades o instituciones de educación superior que enfrenten problemáticas similares.

1.5. Objetivos del proyecto

1.5.1. Objetivo General

Desarrollar una solución centralizada que permita gestionar la asignación, el seguimiento y el control de los proyectos de titulación de las carreras de informática de la Facultad de Ciencias Empresariales de la Universidad del Bío-Bío.

1.5.2. Objetivos Específicos

1. **Analizar** las necesidades específicas de los estudiantes y profesores de la FACE -Universidad del Bío-Bío en cuanto al seguimiento, revisión y comunicación de proyectos de titulación de las carreras de informática mediante el estudio del flujo de trabajo actual y la identificación de puntos críticos en el proceso.
2. **Diseñar** un sistema de software adaptado a los requerimientos institucionales de la Universidad del Bío-Bío, incluyendo funcionalidades como notificaciones automáticas y seguimiento
3. **Implementar** el sistema de gestión de proyectos de titulación en la Universidad del Bío-Bío, asegurando su correcta integración con las plataformas académicas existentes.

1.6. Composición del Informe

El presente documento contiene las especificaciones técnicas correspondientes al desarrollo del proyecto de titulación de la carrera Ingeniería de Ejecución en Informática, titulado:

“AcTitUBB:Software de coordinacion de actividades de titulacion de carreras Informaticas FACE-UBB”

El documento se organiza en 8 capítulos o secciones. El primer capítulo incluye la presentación del área de estudio, la institución involucrada, la problemática observada y una propuesta de solución. Además, se incorporan antecedentes de trabajos relacionados y la justificación del proyecto.

El segundo capítulo aborda los aspectos específicos del proyecto: se presentan los objetivos generales y específicos, la metodología de desarrollo seleccionada, las actividades necesarias para

alcanzar los objetivos, los estándares de documentación utilizados, las técnicas aplicadas y las herramientas tecnológicas empleadas.

El tercer capítulo se centra en la definición de requerimientos, detallando las necesidades funcionales y no funcionales del sistema, así como las reglas de negocio y restricciones operativas identificadas para el correcto funcionamiento de la plataforma AcTitUBB.

El cuarto capítulo presenta el análisis del sistema, modelando el comportamiento de la solución a través de diagramas de casos de uso y la especificación detallada de las interacciones entre los distintos actores (estudiantes, profesores y administradores) y el software.

El quinto capítulo expone el diseño técnico de la solución, describiendo la arquitectura de software, el modelo de datos relacional, la definición de los servicios web (API REST) y el diseño de las interfaces de usuario bajo los estándares visuales de la institución.

El sexto capítulo documenta el proceso de desarrollo, profundizando en la implementación de los módulos, la estructura del código fuente y la integración de las tecnologías seleccionadas tanto para el frontend como para el backend.

El séptimo capítulo aborda la puesta en marcha y pruebas, detallando el ambiente de despliegue, la configuración de la infraestructura mediante contenedores y la validación funcional del sistema.

Finalmente, el octavo capítulo presenta las conclusiones del proyecto, realizando un contraste entre los resultados obtenidos y los objetivos planteados inicialmente, además de identificar limitaciones y proponer líneas de trabajo futuro para la evolución del sistema.

Capítulo 2

Proyecto

El presente capítulo describe los aspectos metodológicos, técnicos y de factibilidad del proyecto de desarrollo del sistema AcTitUBB. Se presentan los objetivos del software, la metodología de desarrollo seleccionada, las tecnologías y herramientas utilizadas, los estándares de documentación aplicados y el análisis de factibilidad técnica, operativa y económica que sustenta la viabilidad del proyecto.

2.1. Objetivos del Software

el propósito principal de este trabajo es desarrollar un sistema web centralizado de gestión y seguimiento de proyectos de título para las carreras de informática de la Facultad de Ciencias Empresariales de la Universidad del Bío-Bío. Esta plataforma facilitará la interacción entre estudiantes, profesores y administradores, automatizando el control de plazos, revisiones y notificaciones para mejorar la eficiencia y transparencia del proceso de titulación.

2.1.1. Objetivos Específicos

- Implementar un módulo de gestión de propuestas que permita el envío, revisión y aprobación de proyectos de título, incluyendo la asignación automática de profesores guías e informantes según disponibilidad y afinidad temática, con capacidad de reasignación manual por parte del administrador ante eventualidades.
- Desarrollar un sistema de seguimiento de hitos y plazos, definidos tanto a nivel global por el administrador como de manera personalizada por los profesores, que permita a los estudiantes visualizar el estado actual de cada proyecto, registrando avances, entregas y revisiones en tiempo real.
- Crear un sistema de notificaciones automáticas que alerte a los usuarios sobre fechas límite, cambios de estado, solicitudes de revisión y otras acciones relevantes del proceso de titulación.
- Diseñar una interfaz de usuario intuitiva y accesible que facilite la adopción del sistema por parte de todos los actores involucrados, minimizando la curva de aprendizaje.

- Aplicar una metodología de desarrollo iterativa e incremental que permita la validación continua de funcionalidades mediante retroalimentación de usuarios, asegurando que el producto final satisfaga las necesidades reales de la institución.

2.2. Metodología de Desarrollo

Tras analizar las características del proyecto, se realiza una evaluación sistemática considerando diversos factores que determinan el enfoque metodológico más adecuado. Se cuenta con experiencia previa en el desarrollo de sistemas de gestión de información y aplicaciones web, lo que reduce significativamente el riesgo técnico. La complejidad del problema se clasifica como moderada, ya que si bien el proceso de seguimiento de proyectos de título involucra múltiples actores (estudiantes, profesores guías, informantes y jefaturas) y diversas etapas (propuesta, desarrollo, revisión, defensa), los flujos de trabajo están claramente definidos por el reglamento institucional ([Face, 2025](#)).

El tamaño del problema se considera mediano, dado que el sistema operará en un entorno controlado dentro de las carreras de informática de la Facultad de Ciencias Empresariales, con aproximadamente 52 proyectos anuales y 17 académicos involucrados ([Dirección General de Análisis Institucional, Universidad del Bío-Bío, 2025a,b](#)).

La Tabla 2.1 caracteriza el proyecto a desarrollar, en distintos aspectos:

Ítem	Nivel	Descripción
Experiencia en el problema	Medio	Se cuenta con conocimiento general del proceso académico de titulación.
Tamaño del problema	Medio	Involucra varios procesos como gestión de plazos, revisión de tesis y notificaciones, además de interacción con distintos usuarios (profesores, alumnos).
Complejidad del problema	Medio	Los procesos están bien definidos, pero requieren entendimiento de flujos académicos y validaciones, lo cual implica cierto grado de análisis.
Tamaño del software	Medio-Alto	Se deben construir varias funcionalidades: gestión de plazos, revisión de tesis, generación de notificaciones, entre otras.
Complejidad del software	Moderada	Las funcionalidades no requieren cálculos complejos ni uso de hardware/software específico.
Experiencia en el software	Alta	Se cuenta con alta experiencia técnica en el desarrollo de sistemas de gestión académica similares.
Modularidad	Alta	El software puede desarrollarse de forma iterativa y modular, permitiendo dividir su funcionalidad en partes independientes que evolucionan por ciclos.

Tabla 2.1: Características del proyecto

Tras evaluar estos factores, se determinó que el riesgo total del proyecto es bajo, lo que ofrece

flexibilidad para seleccionar una metodología de desarrollo adecuada. Se ha optado por una **metodología iterativa e incremental** inspirada en principios ágiles, que permite el desarrollo por ciclos de corta duración. Cada iteración contempla las fases de análisis, diseño, implementación, pruebas y validación de un conjunto acotado de funcionalidades, ciclo que se ilustra en la Figura 2.1.

Este enfoque presenta las siguientes ventajas para el proyecto:

- **Retroalimentación temprana:** Al entregar versiones funcionales del sistema en cada iteración, los usuarios finales pueden probar las funcionalidades implementadas y proporcionar observaciones que orienten el desarrollo subsecuente.
- **Gestión de cambios:** Permite incorporar modificaciones en los requerimientos de manera controlada, adaptándose a las necesidades reales que surgen durante el uso del sistema.
- **Reducción de riesgos:** Al validar continuamente cada módulo, se detectan y corrigen errores de forma temprana, evitando la acumulación de problemas hacia el final del proyecto.
- **Desarrollo incremental:** Facilita la priorización de funcionalidades críticas, asegurando que el sistema cuente con un núcleo funcional desde las primeras iteraciones.

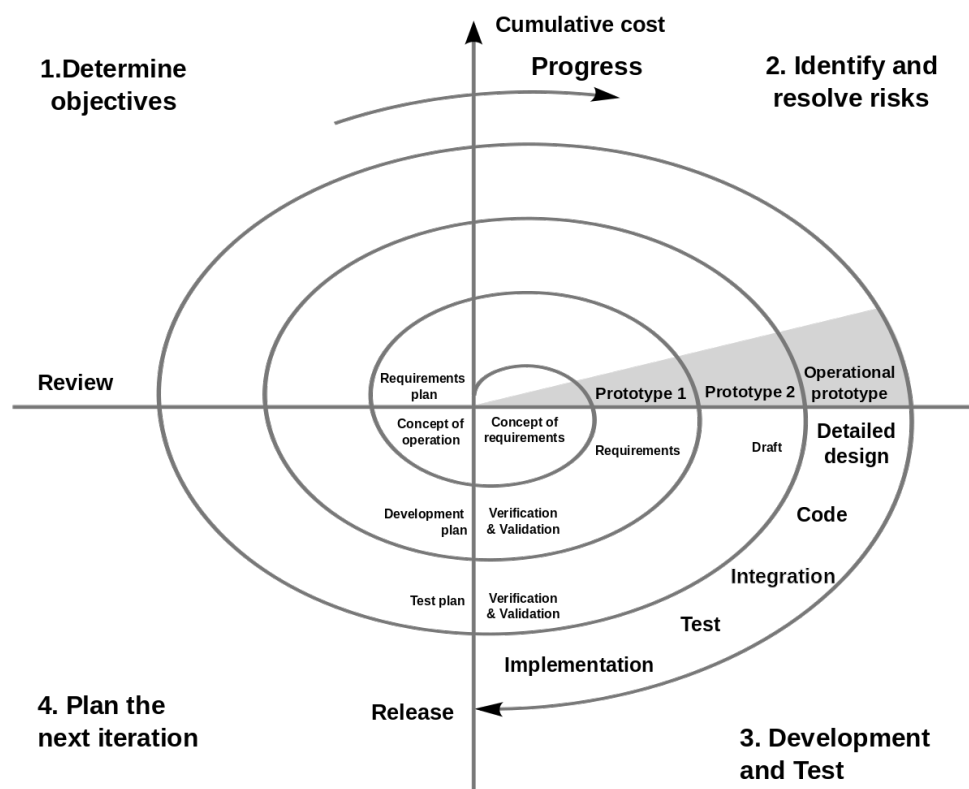


Figura 2.1: Esquema de la metodología iterativa aplicada al proyecto (Fuente: [PNGWing, 2025](#))

Dado que el sistema está destinado a ser utilizado por distintos perfiles de usuario, las pruebas de cada iteración se centraron en verificar la alineación de las funcionalidades con los flujos de trabajo y requerimientos específicos de cada rol. Este enfoque metodológico garantiza un desarrollo continuo y ordenado, asegurando que el producto final constituya una herramienta efectiva y adaptada a los procesos reales de seguimiento de proyectos de título en la Universidad del Bío-Bío.

2.3. Tecnologías Utilizadas

Categoría	Tecnologías / Herramientas
Frontend	Angular v20 con componentes standalone. SCSS para estilos.
Backend	Node.js v22.14. Express.js para manejo de rutas y controladores. Multer para manejo de subida de archivos. JWT para autenticación.
Base de datos	MySQL v8.0. MySQL Workbench para modelado y consultas.
Herramientas auxiliares	Postman para pruebas de endpoints REST. Git y GitHub para control de versiones. PM2 y Nginx para despliegue en servidor. Visual Studio Code como entorno de desarrollo. Docker para realizar pruebas en desarrollo.

Tabla 2.2: Tecnologías y herramientas utilizadas en el proyecto

2.4. Estándares de Documentación

El desarrollo del sistema AcTitUBB se rige por estándares internacionales de ingeniería de software que aseguran la calidad, trazabilidad y mantenibilidad del producto. Los principales estándares aplicados son:

2.4.1. IEEE Std 830-1998: Software Requirements Specification

Se adoptan los lineamientos establecidos por el estándar IEEE 830 (IEEE, 1998a) para la especificación de requisitos del software. Este estándar define la estructura y contenido de los documentos de requisitos, asegurando que sean:

- **Completos:** Todos los requisitos funcionales y no funcionales están documentados.
- **Consistentes:** No existen contradicciones entre los requisitos especificados.
- **Verificables:** Cada requisito puede ser validado mediante pruebas específicas.
- **Modificables:** La estructura permite actualizar requisitos sin afectar la coherencia del documento.

2.4.2. IEEE Std 829-1998: Software Test Documentation

Para la documentación de pruebas se utiliza como referencia el estándar IEEE 829 ([IEEE, 1998b](#)), que establece formatos para los siguientes documentos:

- **Plan de pruebas:** Define el alcance, enfoque, recursos y cronograma de las actividades de prueba.
- **Casos de prueba:** Especifican las condiciones de entrada, procedimientos de ejecución y resultados esperados para cada prueba.
- **Reportes de pruebas:** Documentan los resultados obtenidos, identificando defectos y no conformidades.

La aplicación de estos estándares asegura que el proceso de desarrollo del sistema AcTitUBB mantenga altos niveles de calidad, permitiendo la evaluación objetiva del cumplimiento de requisitos y facilitando futuras actividades de mantenimiento y evolución del software.

2.5. Técnicas y notaciones

Para el desarrollo y documentación del software se utilizan diversas técnicas y notaciones que permiten describir, analizar y diseñar el sistema de forma clara y estructurada. Entre las principales se encuentran:

- **Diagramas de Casos de Uso:** Utilizados para representar las funcionalidades del sistema desde el punto de vista de los usuarios (actores), facilitando la identificación de requisitos y el alcance del software.
- **Modelo relacional (MR):** Utilizado para diseñar la base de datos, definiendo entidades, atributos y relaciones, asegurando una estructura coherente y optimizada para el almacenamiento de datos.
- **Notación BPMN (Business Process Model and Notation):** Para representar procesos de negocio y flujos operativos, facilitando la comunicación entre los actores involucrados y el equipo de desarrollo.
- **Carta Gantt:** Para la planificación temporal de actividades del proyecto.

2.6. Herramientas, framework, lenguaje usados en el desarrollo del proyecto

A continuación, se detallan las herramientas, lenguajes y frameworks utilizados durante el desarrollo del sistema:

Categoría	Descripción
Lenguajes de programación	TypeScript: utilizado para el desarrollo del frontend con Angular (Microsoft, 2025a). JavaScript: utilizado en el backend con Node.js y Express.js (Node.js Foundation, 2025). SQL: para consultas y manipulación de la base de datos MySQL (Oracle Corporation, 2025a).

Frameworks y librerías	Angular v20: framework frontend basado en componentes standalone (Google LLC, 2025). Express.js: framework para la creación de rutas y controladores en el backend (OpenJS Foundation, 2025). Multer: librería utilizada para la carga de archivos en el servidor (Expressjs, 2025). JSON Web Token (JWT): utilizado para la autenticación de usuarios mediante tokens (Jones et al., 2015).
Herramientas de desarrollo	Visual Studio Code: entorno de desarrollo principal (Microsoft, 2025b). Postman: herramienta utilizada para probar y validar endpoints de la API REST (Postman Inc., 2025). Docker: utilizado para pruebas en entornos de desarrollo mediante contenedores (Docker Inc., 2025). Git y GitHub: control de versiones y gestión del repositorio del proyecto (Software Freedom Conservancy, 2025). MySQL Workbench: utilizado para modelado, diseño y ejecución de consultas sobre la base de datos (Oracle Corporation, 2025b).
Herramientas de despliegue	PM2: administrador de procesos para aplicaciones Node.js en entornos de producción (Keymetrics, 2025). Nginx: servidor proxy inverso utilizado en el despliegue del frontend (F5, Inc., 2025).

Tabla 2.3: Lenguajes, frameworks y herramientas utilizadas

2.7. Factibilidad del Proyecto

Para determinar la viabilidad del proyecto, se realizó un análisis que abarca aspectos técnicos, económicos y operativos. A continuación, se detallan los recursos necesarios y las condiciones para la ejecución exitosa.

2.7.1. Factibilidad Técnica

El desarrollo del sistema propuesto presenta una factibilidad técnica favorable, sustentada en los siguientes puntos:

Disponibilidad y Competencias del Equipo de Desarrollo

El proyecto será desarrollado de forma individual, contando con las competencias técnicas necesarias para su correcta ejecución. El desarrollador posee conocimientos y experiencia en tecnologías como **Node.js** para la programación del backend, **Angular** para el desarrollo del frontend, **MySQL** para la gestión de bases de datos, y **Docker** para la contenerización y despliegue de servicios. Además, maneja buenas prácticas de desarrollo, control de versiones y metodologías ágiles, lo que garantiza un proceso ordenado y eficiente.

Disponibilidad de Software de Desarrollo y de Explotación

El software requerido para el desarrollo del sistema está disponible mediante licencias gratuitas o de código abierto, lo cual facilita su acceso sin generar costos adicionales. Entre las herramientas a utilizar se encuentran: **Visual Studio Code**, **Node.js**, **Angular CLI**, **MySQL Workbench**, **Docker**, **Postman** y **Git**. Para la explotación del sistema, se contempla el uso de servidores compatibles con contenedores Docker, asegurando así su despliegue y escalabilidad sin necesidad de licencias propietarias.

Disponibilidad de Hardware de Desarrollo y de Explotación

Se dispone del hardware necesario para las labores de desarrollo, consistente en equipos de cómputo personales con especificaciones técnicas adecuadas, como procesadores multinúcleo, un mínimo de 16 GB de memoria RAM y almacenamiento sólido (SSD). Respecto al entorno de explotación, se prevé la utilización de servidores físicos o virtuales, ya sea en instalaciones propias o a través de servicios en la nube (como **VPS**, **AWS** o **Azure**), los cuales podrán ser adquiridos según las necesidades del sistema al momento de su implementación.

Especificación de Software Requerido

Tabla 2.4: Especificación de software requerido para el desarrollo del proyecto

Nombre del Software	Función	Observación
Visual Studio Code	Entorno de desarrollo (IDE)	Edición de código y gestión de proyectos
Node.js	Entorno de ejecución backend	Creación del servidor y la API REST (versión 22.14)
Angular CLI	Framework frontend	Generación y gestión de la aplicación cliente (versión LTS)
MySQL	Sistema gestor de bases de datos	Gestión de la base de datos relacional (versión 8.0)
MySQL Workbench	Herramienta de administración de bases de datos	Diseño, modelado y consulta de la base de datos
Docker	Contenerización	Creación de entornos aislados para desarrollo (versión 28.3.2)
Git	Control de versiones	Control y gestión del código fuente del proyecto (versión 2.50)

Especificación de Hardware de Desarrollo

Tabla 2.5: Especificación de hardware utilizado en el desarrollo del proyecto

Hardware	Descripción
Notebook Lenovo Thinkbook	Procesador AMD Ryzen 7 4700U con Radeon Graphics, 16 GB de RAM, SSD de 500 GB
Componentes adicionales	GPU integrada Radeon Graphics para tareas de desarrollo y pruebas

Especificación de Hardware para Servidor de Desarrollo y Explotación

Tabla 2.6: Especificación de hardware para servidores de desarrollo y explotación

Servidor	Descripción
Servidores propios	Procesador Intel Xeon E5-2683 v3 @ 2.00GHz, 64 GB de RAM, 250 GB NVMe SSD y 1 TB HDD en RAID 0
Capacidades	Alta capacidad de procesamiento y almacenamiento, adecuados para el despliegue de la plataforma completa

Conclusión de la Factibilidad Técnica

Con base en los antecedentes expuestos, se concluye que el desarrollo del sistema es técnicamente factible. Todo el equipamiento necesario para la programación, pruebas y despliegue del sistema existe o puede ser adquirido fácilmente.

El equipo de desarrollo cuenta con las competencias técnicas requeridas y tiene acceso al hardware necesario. Además, el software seleccionado es de libre acceso o de código abierto, lo cual facilita su obtención sin implicar costos adicionales.

Asimismo, las tecnologías empleadas como Node.js, Angular, MySQL y Docker cuentan con comunidades activas, documentación oficial y soporte constante, lo que asegura respaldo ante eventuales dificultades técnicas.

Por estos motivos, se valida la factibilidad técnica del proyecto.

2.7.2. Factibilidad Operativa

El proyecto presenta una factibilidad operativa favorable, sustentada en los siguientes aspectos:

Reconocimiento de la importancia por parte de los clientes

Las autoridades académicas y administrativas de la institución, como los jefes de carrera y coordinadores, reconocen la importancia de contar con un sistema de seguimiento de proyectos de título. Este software contribuirá a optimizar la gestión, control y trazabilidad de las propuestas, facilitando la toma de decisiones y el seguimiento de los procesos.

Reconocimiento de la importancia por parte de los usuarios

Los estudiantes y profesores, principales usuarios del sistema, comprenden los beneficios que aporta la plataforma, tales como la simplificación del envío y revisión de propuestas, la mejora en la comunicación y la transparencia en el proceso de aprobación. Esto genera interés y disposición a utilizar la herramienta.

Disponibilidad y competencias de los usuarios

Los usuarios finales cuentan con las competencias digitales básicas para interactuar con plataformas web. Además, el diseño del sistema priorizará la usabilidad y la interfaz amigable, asegurando que su uso no requiera conocimientos técnicos avanzados. En caso de ser necesario, se contempla la realización de capacitaciones breves o la entrega de manuales de usuario.

Conclusión de la Factibilidad Operativa

Considerando los aspectos mencionados, se concluye que el sistema es operativamente factible. Tanto las autoridades como los usuarios finales reconocen su utilidad, están dispuestos a adoptarlo y poseen las habilidades mínimas necesarias para su uso. Además, en caso de ser requerido, se implementarán acciones complementarias como capacitaciones o documentación de apoyo, asegurando así una correcta incorporación del sistema a los procesos institucionales.

2.7.3. Factibilidad Económica

El análisis económico tiene como objetivo determinar la rentabilidad y sostenibilidad financiera del proyecto. A continuación, se detallan los costos asociados al software, hardware y recursos humanos necesarios para el desarrollo, así como los costos operativos de hosting y dominio. Posteriormente, se presenta un análisis de ahorro proyectado y el cálculo de indicadores financieros como el Valor Actual Neto (VAN) para sustentar la viabilidad de la inversión.

Software	Licencia	Costo Licencia
Visual Studio Code	MIT License	\$0
Node.js	MIT License	\$0
Angular CLI	MIT License	\$0
MySQL	GPL v2	\$0
MySQL Workbench	GPL v2	\$0
Docker (uso personal/educativo)	Docker Subscription Service Agreement	\$0
Git	GPL v2	\$0

Tabla 2.7: Licencias y costo del software utilizado en el proyecto

Respecto al ítem de alojamiento (hosting), no se incurre en gastos operacionales directos debido a que la aplicación será desplegada en los servidores institucionales de la universidad. Esta infraestructura ya se encuentra disponible y operativa, por lo que el proyecto no requiere la contratación de servicios externos de nube.

Item	\$ mensual aprox. en CLP	\$ anual aprox. en CLP
Hosting	\$0	\$0
Dominio	-	\$10.000
Total	-	\$10.000

Tabla 2.8: Costo de hosting y dominio para el proyecto ([NIC Chile, 2025](#))

Recurso Humano	Cantidad de personal	Sueldo aprox. en CLP por mes	Sueldo aprox. en CLP por duración del proyecto (12 meses)
Desarrollador	1	\$950.000	\$11.400.000
Total	-	-	\$11.400.000

Tabla 2.9: Cálculo estimado del costo de desarrollo y soporte.

IV.3.1 Impacto en eficiencia y ahorro económico

La implementación de esta aplicación permitirá ahorrar aproximadamente 2 horas semanales por docente, al automatizar tareas como la recepción de propuestas, notificación de revisiones, seguimiento de plazos, generación de reportes y comunicación con estudiantes, que actualmente se realizan de forma manual mediante correos electrónicos y hojas de cálculo.

Dado que el departamento cuenta con 17 docentes, el ahorro semanal total es:

$$\text{Ahorro semanal} = 2 \text{ horas} \times 17 \text{ profesores} = 34 \text{ horas/semana}$$

Este proceso se realiza durante 6 semanas al año, por lo que el ahorro anual total será:

$$\text{Ahorro anual} = 34 \text{ horas/semana} \times 6 \text{ semanas} = 204 \text{ horas/año}$$

Considerando el valor estimado de la hora docente según el Portal de Transparencia de la Universidad del Bío-Bío ([Universidad del Bío-Bío, 2025e](#)), que tiene un valor promedio de \$12.500 CLP:

$$\begin{aligned} \text{Ahorro económico} &= 204 \text{ horas} \times \$12.500 \text{ CLP/hora} \\ &= \$2.550.000 \text{ CLP anuales} \end{aligned}$$

Flujo de caja

Para asegurar la viabilidad económica del proyecto, se empleará el indicador del Valor Actual Neto (VAN) como medida. Para ello, se realizará el cálculo del flujo de caja correspondiente a la inversión inicial, así como se proyectarán los flujos de caja para los primeros 5 años. Estos datos se presentan en detalle en la siguiente tabla:

Concepto	Año 0	Año 1	Año 2	Año 3	Año 4	Año 5
(+) Beneficios						
Ahorro por eficiencia administrativa	-	\$2.550.000	\$2.550.000	\$2.550.000	\$2.550.000	\$2.550.000
(-) Costos						
Costo de desarrollo	\$0	-	-	-	-	-
Dominio anual	-	\$10.000	\$10.000	\$10.000	\$10.000	\$10.000
Soporte y mantención	-	\$300.000	\$300.000	\$300.000	\$300.000	\$300.000
Flujo neto anual	\$0	\$2.240.000	\$2.240.000	\$2.240.000	\$2.240.000	\$2.240.000

Tabla 2.10: Proyección de flujos de caja para evaluación económica del proyecto (valores en CLP)

Cálculo del V.A.N

$$VAN = \sum_{t=1}^n \frac{F_t}{(1+k)^t} - I_0$$

Figura 2.2: Fórmula del Valor Actual Neto (VAN) para evaluación de proyectos

Donde cada uno de los términos se especifica en la Tabla:

Término	Significado
t	Intervalo de tiempo
n	Duración en años
I_0	Inversión inicial ($t = 0$)
K	Tasa de descuento
Vt	Flujos de caja obtenidos en el intervalo de tiempo t

Tabla 2.11: Términos de la fórmula de VAN

A continuación, se calculará el VAN con una tasa de descuento del 10.

Tabla 2.12: Cálculo del VAN

Año	Flujo de Caja
Año 0	$0/(1+0.10)^0 = 0$
Año 1	$2.240.000/(1+0.10)^1 = 2.036.364$
Año 2	$2.240.000/(1+0.10)^2 = 1.851.240$
Año 3	$2.240.000/(1+0.10)^3 = 1.682.036$
Año 4	$2.240.000/(1+0.10)^4 = 1.528.214$
Año 5	$2.240.000/(1+0.10)^5 = 1.389.285$

$$\begin{aligned} \text{VAN (10\%)} &= 0 + 2.036.364 + 1.851.240 + 1.682.036 + 1.528.214 + 1.389.285 \\ &= 8.487.139 \end{aligned}$$

Por lo tanto, el VAN (10%) es de \$8.487.139 CLP. En este caso, el VAN obtenido es positivo (\$8.487.139), lo que indica que el proyecto es viable desde una perspectiva económica.

Conclusión de Factibilidad

Gracias al análisis realizado en los puntos anteriores, se puede concluir que el proyecto es viable y factible en su totalidad. Desde el punto de vista técnico, el equipo de desarrollo cuenta con las competencias, experiencia y recursos necesarios para llevar a cabo el proyecto, utilizando herramientas y tecnologías modernas, accesibles y ampliamente respaldadas por sus comunidades. Además, se dispone del hardware y software adecuados tanto para la etapa de desarrollo como para la futura operación del sistema. En cuanto a la factibilidad operativa, los principales interesados autoridades, académicos y estudiantes reconocen la importancia y utilidad del sistema, mostrando disposición para su implementación y uso. Asimismo, se considera que los usuarios poseen las competencias digitales mínimas requeridas, lo que favorece la adopción y correcto funcionamiento de la plataforma en el entorno institucional. Por último, el análisis de la factibilidad económica demuestra que el proyecto genera un impacto positivo en términos de eficiencia y ahorro para la institución. La inversión inicial es recuperada gracias a la reducción de la carga administrativa, y el flujo de caja proyectado junto al Valor Actual Neto (VAN) positivo refuerzan la rentabilidad y conveniencia económica de la iniciativa. En conclusión, el proyecto es técnica, operativa y económicamente factible, justificando plenamente su desarrollo y puesta en marcha.

Capítulo 3

Especificación de Requerimientos - Producto de Software

El presente capítulo detalla los requerimientos del sistema AcTitUBB, estableciendo las bases para su diseño, desarrollo y validación. Su objetivo es garantizar una comprensión clara y unificada entre los stakeholders equipo técnico, clientes y usuarios finales sobre qué debe hacer el sistema, bajo qué restricciones y cómo interactuará con su entorno.

La estructura del capítulo sigue los lineamientos del estándar IEEE 830-1998 ([IEEE, 1998a](#)) para especificación de requerimientos de software, cubriendo los siguientes aspectos fundamentales:

- **Límites del sistema:** Define explícitamente las funcionalidades que el sistema no implementará, estableciendo expectativas realistas sobre su alcance.
- **Restricciones técnicas:** Especifica las limitaciones impuestas por la infraestructura tecnológica, estándares institucionales y restricciones de recursos.
- **Requerimientos funcionales:** Describe las acciones específicas que el sistema debe ejecutar en respuesta a entradas o situaciones determinadas.
- **Requerimientos no funcionales:** Establece atributos de calidad relacionados con rendimiento, seguridad, usabilidad y mantenibilidad según la norma ISO/IEC 25010:2011 ([ISO/IEC, 2011](#)).
- **Interfaces externas:** Define los flujos de entrada y salida de datos entre el sistema y sus actores o sistemas externos.

Cada requerimiento funcional se especifica siguiendo un formato estructurado que identifica claramente el actor involucrado, la acción a realizar, las condiciones o restricciones aplicables, el evento que dispara la funcionalidad y el resultado esperado tras su ejecución. Esta estructura facilita tanto la implementación técnica como la posterior validación mediante casos de prueba.

3.1. Límites

El sistema AcTitUBB opera dentro de un alcance definido que establece explícitamente las funcionalidades que no serán implementadas. Estos límites aseguran expectativas realistas entre

los stakeholders y facilitan la evaluación objetiva del cumplimiento del proyecto.

3.1.1. Limitaciones Funcionales

El sistema no permitirá:

- La aprobación automática de propuestas sin revisión y comentarios previos por parte de un profesor asignado. Toda propuesta debe pasar por un proceso de evaluación humana antes de cambiar a estado “Aprobada”.
- La modificación de propuestas una vez que hayan alcanzado estados finales (“Aprobada” o “Rechazada”). En estos casos, solo se permitirá acceso de lectura para consulta histórica.
- La creación de múltiples propuestas simultáneas por el mismo estudiante sin aprobación previa de al menos una de ellas. Esta restricción evita saturación del sistema de revisión.

3.1.2. Limitaciones de Seguridad y Acceso

El sistema no permitirá:

- El acceso a información de propuestas sin la debida autorización según el rol del usuario. Los estudiantes solo pueden visualizar sus propias propuestas, los profesores únicamente las que les han sido asignadas, y solo los encargados de curso tienen visibilidad completa del sistema.
- El registro de usuarios priorizará dominios institucionales autorizados (@alumnos.ubiobio.cl para estudiantes, @ubiobio.cl para académicos y personal administrativo) sin embargo, se permitirá el registro de Co-Guias externos con otros dominios de correo, previo flujo de validación o invitación.
- El acceso a funciones del sistema desde cuentas no confirmadas mediante el proceso de verificación por correo electrónico. La confirmación es obligatoria para garantizar la autenticidad de los usuarios.

3.1.3. Limitaciones Técnicas de Archivos

El sistema no permitirá:

- La carga de archivos en formatos distintos a PDF y DOCX, garantizando compatibilidad multiplataforma y seguridad en el almacenamiento documental.
- La carga de archivos que excedan el tamaño máximo configurado por defecto de 10 MB, protegiendo la capacidad de almacenamiento del servidor y optimizando los tiempos de transferencia.

3.2. Restricciones Técnicas

El desarrollo e implementación del sistema AcTitUBB está sujeto a las siguientes restricciones técnicas derivadas de decisiones arquitectónicas, infraestructura disponible y políticas institucionales:

- **Stack tecnológico:** El sistema está construido sobre Angular 18+ para el frontend, Node.js 20+ para el backend y MySQL 8.0+ como gestor de base de datos. Estas tecnologías son requisitos obligatorios para el correcto funcionamiento del sistema.
- **Despliegue:** La implementación utiliza Docker y Docker Compose para contenerización y orquestación de servicios, requiriendo estas tecnologías en los entornos de desarrollo, pruebas y producción.
- **Autenticación:** El sistema emplea JSON Web Tokens (JWT) con tiempo de expiración de 1 hora, requiriendo renovación periódica de tokens para mantener sesiones activas.
- **Almacenamiento:** Los archivos adjuntos se almacenan localmente en el sistema de archivos del servidor. La integración con servicios de almacenamiento en la nube (AWS S3, Google Cloud Storage) requeriría desarrollos adicionales no contemplados en esta versión.
- **Integración institucional:** El sistema no contempla integración directa con plataformas institucionales existentes (sistemas académicos, directorios LDAP, single sign-on) sin desarrollos posteriores específicos para cada servicio.
- **Notificaciones:** El envío de notificaciones por correo electrónico requiere configuración de un servidor SMTP institucional. Sin esta configuración, las notificaciones solo funcionarán dentro del sistema web.
- **Conectividad:** El sistema requiere conexión a internet para el funcionamiento completo de notificaciones por correo electrónico y confirmaciones de cuenta.

3.3. Requerimientos Funcionales

Los requerimientos funcionales especifican las acciones que el sistema debe ejecutar en respuesta a entradas específicas o en determinadas situaciones. Definen el comportamiento observable del sistema desde la perspectiva de los usuarios. La Tabla 3.1 detalla los diez requerimientos funcionales principales del sistema AcTitUBB, organizados según los procesos de gestión de propuestas de proyectos de título.

Tabla 3.1: Especificación de Requerimientos Funcionales del sistema AcTitUBB

ID	El sistema debe...
RF-01	Permitir el inicio de sesión mediante correo institucional válido y contraseña, generando un JSON Web Token (JWT) firmado con tiempo de expiración configurable. Este token deberá incluir el rol, el identificador de usuario y los permisos asociados.
RF-02	Bloquear automáticamente el acceso a rutas protegidas cuando el usuario no incluya token, incluya un token inválido o cuando este se encuentre expirado, registrando el intento fallido.
RF-03	Aplicar controles estrictos de autorización basados en rol (RBAC), negando el acceso a recursos no permitidos y registrando el evento para auditoría.

ID	El sistema debe...
RF-04	Permitir que el estudiante registre una propuesta inicial completando título, resumen y objetivos obligatorios. La propuesta quedará en estado “Borrador” hasta ser enviada para revisión.
RF-05	Permitir que el estudiante edite cualquier campo de su propuesta mientras esta se encuentre en estado “Borrador”. Al cambiar de estado, se registrará fecha, usuario y cambios.
RF-06	Permitir que el administrador apruebe o rechace propuestas enviadas, agregando observaciones obligatorias en caso de rechazo. El estado y las observaciones deberán reflejarse en el historial del estudiante.
RF-07	Listar propuestas filtrando por estado, autor, fecha o palabras clave, incorporando paginación en consultas grandes.
RF-08	Crear automáticamente un proyecto asociado cuando una propuesta sea aprobada, asignándole estado inicial “En Curso” y vinculándolo al estudiante propietario.
RF-09	Permitir que el estudiante visualice información consolidada de su proyecto: profesor guía, co-guía, informante, documentos, hitos y progreso general.
RF-10	Permitir que el administrador cambie el estado del proyecto a “Finalizado” o “Cancelado”, solo si se cumplen las reglas establecidas (por ejemplo, hitos completos o solicitud de retiro).
RF-11	Permitir al administrador asignar uno o más profesores a un proyecto, indicando rol (guía, co-guía o informante), registrando fecha y usuario responsable.
RF-12	Impedir que se duplique una asignación activa con el mismo par {proyecto, rol}, mostrando mensaje de error y registrando el intento.
RF-13	Permitir que el administrador reemplace o elimine asignaciones de profesores cuando estas estén incorrectas, manteniendo el registro histórico para auditoría.
RF-14	Permitir la creación de un cronograma por proyecto, indicando fechas estimadas para hitos principales.
RF-15	Permitir que el estudiante registre hitos con descripción y fecha planificada. Estos quedarán inicialmente en estado “Pendiente”.
RF-16	Permitir que el profesor actualice el estado de un hito a “En revisión” o “Aprobado”, dejando comentarios si corresponde.
RF-17	Listar hitos por proyecto, incluyendo filtros por estado, fechas y responsable.
RF-18	Permitir que cualquier usuario registre sus disponibilidades (fecha, hora inicio y hora fin), asociadas a su calendario personal.

ID	El sistema debe...
RF-19	Permitir que el estudiante solicite una reunión indicando profesor y proyecto, registrando motivo y fecha deseada.
RF-20	Procesar automáticamente la búsqueda de disponibilidad entre profesor y estudiante (matching), sugiriendo intervalos compatibles o próximos disponibles.
RF-21	Permitir que el profesor acepte o rechace la reunión sugerida, generando notificación inmediata al estudiante.
RF-22	Registrar historial completo de creación, reprogramación y cancelación de reuniones, incluyendo fecha, usuario responsable y estado.
RF-23	Permitir que el estudiante suba documentos asociados al proyecto (borradores, avances, versión final) en formatos PDF o DOCX, validando el tamaño máximo permitido.
RF-24	Registrar metadatos del documento: tipo, versión, fecha de subida, usuario que lo subió y relación con el proyecto.
RF-25	Permitir que el profesor revise documentos, deje comentarios y cambie estado a “Aprobado” o “Requiere corrección”.
RF-26	Permitir que los usuarios autorizados descarguen documentos aprobados o versiones anteriores, manteniendo control de historial.
RF-27	Generar notificaciones internas ante eventos críticos (propuesta aprobada, reunión aceptada, documento revisado).
RF-28	Enviar correos electrónicos según criticidad, incluyendo información del evento y enlace a la sección correspondiente.
RF-29	Permitir que todos los usuarios marquen las notificaciones como leídas, para mantener un panel actualizado.
RF-30	Mantener registros de auditoría para acciones críticas (asignaciones, revisiones, reuniones), incluyendo fecha, usuario y acción.
RF-31	Permitir filtros avanzados por estado en propuestas, proyectos, hitos y reuniones, combinables con búsqueda textual.
RF-32	Implementar paginación en todos los listados extensos para optimizar rendimiento y experiencia del usuario.

ID	El sistema debe...
RF-33	Mostrar y habilitar únicamente las opciones, acciones y campos pertinentes al rol del usuario autenticado. Especificaciones: (a) El sistema (rol id 4) visualiza y puede editar todos los campos y acciones del sistema (gestión completa de usuarios, estructura, contenidos y configuración); (b) El encargado de curso (rol id 3) puede gestionar usuarios y contenidos asignados pero no modificar contraseñas de otros administradores ni cambiar permisos de Super Admin; (c) El Profesor (rol id 2) accede a vistas donde puede ver y editar recursos relacionados con sus asignaciones (hitos, documentos y reuniones) sin poder acceder a la administración global; (d) El Estudiante (rol id 1) solo ve y modifica sus propias propuestas, archivos y solicitudes. Todos los cambios de visibilidad y habilitación deben ser implementados tanto en frontend como en backend (control de vistas y control de autorización en endpoints) y documentados.
RF-34	El sistema debe proporcionar mensajes aclaratorios y accesibles cuando una acción o campo esté oculto o deshabilitado por motivos de permisos, indicando el motivo (por ejemplo: “Acceso reservado a administradores”). Estos mensajes deben ser localizables y registrados en el historial de auditoría cuando la interfaz impida una operación por falta de permisos.

3.4. Requerimientos No Funcionales

Los requerimientos no funcionales especifican atributos de calidad del sistema que no están directamente relacionados con funcionalidades específicas, sino con características operacionales, de rendimiento, seguridad y mantenibilidad. Todos los requerimientos no funcionales se evalúan según los atributos de calidad definidos en la norma ISO/IEC 25010:2011 ([ISO/IEC, 2011](#)).

3.4.1. RNF 01: API REST e Interoperabilidad

Descripción:

El sistema expondrá una API REST completa con endpoints documentados para operaciones CRUD sobre propuestas, usuarios y proyectos. La API incluirá endpoints públicos de solo lectura para consultas estadísticas y endpoints protegidos con autenticación mediante **JSON Web Tokens (JWT)** para operaciones de escritura. Todos los endpoints seguirán estándares REST con códigos de respuesta HTTP apropiados.

Atributos de calidad asociados:

Adecuación Funcional: La API permite integración completa con sistemas externos manteniendo precisión de datos.

Eficiencia de Desempeño: Respuestas de API en menos de 2 segundos para consultas complejas.

Compatibilidad: Facilita interoperabilidad con sistemas institucionales futuros.

Usabilidad: Documentación completa de endpoints mejora experiencia de desarrolladores.

Fiabilidad: Manejo robusto de errores y validaciones en todos los endpoints.

Seguridad: Autenticación JWT y autorización por roles protegen acceso a datos.

Mantenibilidad: Estructura modular facilita extensión y mantenimiento.

3.4.2. RNF 02: Infraestructura y Despliegue

Descripción:

El sistema utilizará Docker para contenerización con docker-compose para orquestación de servicios (frontend Angular con Nginx, backend Node.js, base de datos MySQL). El código estará versionado en Git con integración continua para despliegue automático. El proyecto será open-source con documentación completa para instalación y configuración.

Atributos de calidad asociados:

Eficiencia de Desempeño: Nginx optimiza entrega de contenido estático y balance de carga.

Compatibilidad: Docker garantiza consistencia entre entornos de desarrollo y producción.

Usabilidad: Documentación detallada facilita instalación por nuevos usuarios.

Fiabilidad: Contenerización mejora disponibilidad y recuperación ante fallos.

Seguridad: Aislamiento de servicios y trazabilidad completa de cambios.

Mantenibilidad: Estructura modular y versionado facilitan mantenimiento evolutivo.

Portabilidad: Docker permite despliegue en múltiples plataformas sin modificaciones.

3.4.3. RNF 03: Rendimiento y Escalabilidad

Descripción:

El sistema debe soportar al menos 100 usuarios concurrentes con tiempos de respuesta menores a 3 segundos para operaciones estándar y 10 segundos para cargas de archivos. La base de datos debe manejar hasta 10,000 propuestas con consultas eficientes mediante indexación apropiada.

Atributos de calidad asociados:

Eficiencia de Desempeño: Tiempos de respuesta y capacidad de carga especificados se mantienen bajo estrés.

Usabilidad: Respuestas rápidas mejoran experiencia de usuario.

Fiabilidad: Sistema estable bajo carga normal y picos de uso.

Mantenibilidad: Monitoreo de rendimiento facilita detección de problemas.

3.4.4. RNF 04: Usabilidad y Accesibilidad de la Interfaz por Rol

Descripción:

La interfaz debe adaptarse según el rol del usuario, manteniendo principios de usabilidad y accesibilidad (WCAG 2.1 AA). Las vistas deberán cargar en menos de 1.5 segundos en condiciones

normales, y las interacciones relacionadas con permisos deberán devolver retroalimentación inmediata y comprensible. Se deberá garantizar compatibilidad con lectores de pantalla y navegación por teclado en las vistas críticas de gestión.

Atributos de calidad asociados:

Adecuación Funcional: La UI por rol asegura que las acciones visibles correspondan a endpoints y permisos implementados en el backend.

Eficiencia de Desempeño: Vistas críticas cargan en menos de 1.5 segundos; interacciones de permisos no bloquean la experiencia.

Compatibilidad: Interfaz compatible con navegadores modernos (Chrome, Edge, Firefox) y herramientas de accesibilidad.

Usabilidad: Diseño claro por rol, feedback contextual y navegación coherente por teclado.

Fiabilidad: Comportamiento consistente y manejo robusto de errores de permisos.

Seguridad: Control de visibilidad minimiza exposición de información sensible.

Mantenibilidad: Implementación modular facilita actualización de reglas de visibilidad.

Nota sobre atributos de calidad: Los atributos se evalúan según la norma ISO/IEC 25010:2011 ([ISO/IEC, 2011](#)), indicando en cada lista la contribución directa del requerimiento a la característica mencionada.

3.5. Interfaces Externas de Entrada

Las interfaces de entrada definen los datos que el sistema recibe desde fuentes externas, ya sea ingresados manualmente por usuarios o provenientes de otros sistemas. La Tabla 3.2 especifica el formato, tipo y restricciones de cada conjunto de datos de entrada.

ID	Nombre del ítem	Detalle de Datos
DE-01	Datos de registro de usuario	RUT_USUARIO (formato XX.XXX.XXX-X con dígito verificador), NOMBRE_COMPLETO (máximo 255 caracteres), EMAIL_INSTITUCIONAL (@alumnos.ubiobio.cl o @ubiobio.cl), CONTRASEÑA (mínimo 8 caracteres, alfanumérica).
DE-02	Datos de autenticación	EMAIL_INSTITUCIONAL (registrado previamente), CONTRASEÑA (texto plano encriptado con bcrypt), TOKEN_CONFIRMACION (UUID generado al registro).
DE-03	Datos de propuesta	TITULO_PROPOSTA (texto), DESCRIPCION_PROPOSTA (texto con formato Markdown opcional), ARCHIVO_PROPOSTA (PDF/-DOCX, máx 10MB), FECHA_ENVIO (timestamp), RUT_ESTUDIANTE (FK usuarios).
Continúa en la siguiente página...		

ID	Nombre del ítem	Detalle de Datos
DE-04	Datos de asignación	RUT_PROFESOR (validado), ID_PROPOSTA (validación estado "Pendiente"), FECHA_ASIGNACION (timestamp), USUARIO_ASIGNADOR (RUT admin).
DE-05	Datos de revisión	ID_PROPOSTA, RUT_PROFESOR, COMENTARIO_REVISION (texto obligatorio), CALIFICACION_ESTADO (aprobada/rechazada/correcciones), ARCHIVO_REVISION (PDF opcional).
DE-06	Datos de notificación	TIPO_NOTIFICACION (asignación, estado, comentario), USUARIOS_DESTINATARIOS (lista RUTs), CONTENIDO_MENSAJE (plantilla dinámica), METODO_ENVIO (email/sistema).
DE-07	Datos de gestión de archivos	ID_PROPOSTA, RUT_USUARIO, TIPO_DOCUMENTO (propuesta/revisión), ARCHIVO (validación MIME/extensión), CHECKSUM (SHA-256 integridad).
DE-08	Parámetros de configuración	TAMANO_MAXIMO_ARCHIVO (bytes), FORMATOS_PERMITIDOS (array extensiones), CONFIGURACION_EMAIL (credenciales SMTP), PARAMETROS_SISTEMA (JSON general).

Tabla 3.2: Especificación de interfaces externas de entrada del sistema

3.6. Interfaces Externas de Salida

Las interfaces de salida definen los datos que el sistema genera y entrega a usuarios o sistemas externos. La Tabla 3.3 especifica el formato y medio de entrega de cada tipo de información de salida.

Tabla 3.3: Especificación de interfaces externas de salida del sistema

Identificador	Nombre del ítem	Detalle de Datos	Medio de salida
IS-01	Confirmación de registro	NOMBRE_USUARIO, EMAIL_CONFIRMACION, ENLACE_ACTIVACION (válido 24 horas), INSTRUCCIONES_ACTIVACION.	Correo electrónico HTML

Continúa en la siguiente página

Tabla 3.3 – Continuación

Identificador	Nombre del ítem	Detalle de Datos	Medio de salida
IS-02	Notificación de asignación	TITULO_PROPUESTA, NOM-BRE_ESTUDIANTE, NOM-BRE_PROFESOR, FE-CHA_ASIGNACION, ACCIO-NES_REQUERIDAS, ENLA-CE_DIRECTO.	Correo electróni-co / Notificación en pantalla
IS-03	Estado de pro-puesta	ID_PROPUESTA, TITULO, ESTADO_ACTUAL, PRO-GRESO_REVISION, FE-CHA_ULTIMA_ACTUALIZACION, PROXIMOS_PASOS_SUGERIDOS.	Interfaz web / API REST (JSON)
IS-04	Retroalimentación de revisión	ID_PROPUESTA, COMEN-TARIOS_DETALLADOS, FE-CHA_REVISION, NOM-BRE_REVISOR, DECISION_FINAL, ARCHIVOS_ADJUNTOS (enlaces de descarga).	Interfaz web / Co-rreo electrónico
IS-05	Historial de audi-toría	ID_PROPUESTA, LOG_CAMBIOS (array cronológico), USUA-RIOS_RESPONSABLES, TIMES-TAMPS, ACCIONES_REALIZADAS, DATOS_MODIFICADOS.	Archivo PDF / CSV exportable
IS-06	Dashboard esta-dístico	RESUMEN_PROPUESTAS (con-tadores por estado), GRAFI-COS_EVOLUCION_TEMPORAL, METRICAS_RENDIMIENTO (tiem-pos promedio), ALERTAS_SISTEMA (propuestas atrasadas).	Interfaz web (ac-tualización tiem-po real)
IS-07	Respuestas API	DATOS_SOLICITADOS (payload JSON), METADATOS_PAGINACION (total, página, límite), CODI-GO_ESTADO_HTTP (200, 400, 401, 404, 500), MENSA-JES_ERROR_DESCRIPTIVOS.	JSON (API REST estándar)

Capítulo 4

Análisis - Casos de Uso

Este capítulo presenta el análisis funcional del sistema AcTitUBB mediante el enfoque de casos de uso, siguiendo la notación estándar del Lenguaje Unificado de Modelado (UML) ([Object Management Group, 2017](#)). El análisis incluye la identificación de actores del sistema, la especificación de historias de usuario desde perspectivas ágiles, y la descripción detallada de los casos de uso críticos para el negocio.

El capítulo se estructura en las siguientes secciones:

- **Historias de usuario:** Capturan las necesidades específicas desde la perspectiva de cada tipo de usuario.
- **Identificación de actores:** Define los roles que interactúan con el sistema.
- **Diagramas de casos de uso:** Modelan visualmente las interacciones principales.
- **Especificaciones detalladas:** Describen el flujo normal y alternativo de los casos de uso.

4.1. Historias de Usuario

Las historias de usuario describen las funcionalidades del sistema desde la perspectiva del usuario final.

4.1.1. Historias de Usuario - Estudiante

HU	Descripción y Criterios de Aceptación
HU-01	Descripción: Como estudiante de la UBB, quiero registrarme en el sistema con mi correo institucional para poder crear y gestionar mis propuestas de título.
	Criterios de aceptación: <ul style="list-style-type: none"> ■ Valida dominio @alumnos.ubiobio.cl. ■ Envía confirmación por correo electrónico. ■ Asigna rol de estudiante automáticamente.
HU-02	Descripción: Como estudiante autenticado, quiero crear una nueva propuesta de título para someter mi proyecto a revisión institucional.
	Criterios de aceptación: <ul style="list-style-type: none"> ■ Formulario con título, descripción y fecha de envío. ■ Carga opcional de archivos en formato PDF o DOCX. ■ Estado inicial “Pendiente de revisión”.
HU-03	Descripción: Como estudiante, quiero consultar el estado actual de mis propuestas para conocer el progreso del proceso de revisión.
	Criterios de aceptación: <ul style="list-style-type: none"> ■ Visualiza todas sus propuestas con estado actual. ■ Muestra la fecha de última actualización. ■ Incluye comentarios del profesor cuando existan.
HU-04	Descripción: Como estudiante, quiero recibir notificaciones automáticas para estar informado sobre cambios en mis propuestas.
	Criterios de aceptación: <ul style="list-style-type: none"> ■ Envía correos cuando se asigna un profesor. ■ Notifica cambios de estado.
HU-05	Descripción: Como estudiante, quiero descargar archivos de revisión para acceder a la retroalimentación detallada de mi profesor.
	Criterios de aceptación: <ul style="list-style-type: none"> ■ Habilita enlaces de descarga cuando existen archivos adjuntos.

Tabla 4.1: Historias de Usuario – Rol Estudiante

4.1.2. Historias de Usuario - Profesor

HU	Descripción y Criterios de Aceptación
HU-06	Descripción: (Rol Revisor) Como Profesor Revisor, quiero gestionar las propuestas asignadas para validar su viabilidad.
	Criterios de aceptación: <ul style="list-style-type: none"> ▪ Gestión: Lista filtrada de propuestas pendientes. ▪ Dictamen: Aprobar, Rechazar o Solicitar Correcciones.
HU-07	Descripción: (Rol Guía) Como Profesor Guía, quiero crear el cronograma de trabajo y monitorear el cumplimiento.
	Criterios de aceptación: <ul style="list-style-type: none"> ▪ Cronograma: Creación de hitos con fechas límite. ▪ Monitoreo: Indicadores de progreso y alertas.
HU-08	Descripción: (Rol Guía) Como Profesor Guía, quiero revisar las entregas de hitos y devolver archivos corregidos.
	Criterios de aceptación: <ul style="list-style-type: none"> ▪ Historial: Control de versiones de los documentos. ▪ Evaluación: Asignación de nota y estado.
HU-09	Descripción: (Agenda) Como Profesor, quiero gestionar mi disponibilidad y las solicitudes de reunión.
	Criterios de aceptación: <ul style="list-style-type: none"> ▪ Disponibilidad: Definición de bloques horarios. ▪ Solicitudes: Aceptar o rechazar peticiones.

HU	Descripción y Criterios... (continuación)
HU-10	Descripción: (Rol Informante) Como Profesor Informante, quiero evaluar el informe final dentro del plazo reglamentario.
	Criterios de aceptación: <ul style="list-style-type: none"> ■ Alerta: Indicador visual de días restantes. ■ Veredicto: Apto para defensa o No apto.
HU-11	Descripción: (Dashboard) Como Profesor, quiero un panel de control centralizado para visualizar tareas urgentes.
	Criterios de aceptación: <ul style="list-style-type: none"> ■ Métricas: Contadores de revisiones y reuniones.

Tabla 4.2: Historias de Usuario – Rol Profesor

4.1.3. Historias de Usuario - Administrador

HU	Descripción y Criterios de Aceptación
HU-15	Descripción: Como SuperAdmin, quiero gestionar cuentas de administrador y roles globales.
	Criterios de aceptación: <ul style="list-style-type: none"> ■ CRUD para administradores disponible. ■ Operaciones registradas en auditoría.
HU-16	Descripción: Como SuperAdmin, quiero configurar la estructura académica global y parámetros del sistema.
	Criterios de aceptación: <ul style="list-style-type: none"> ■ Panel de configuración con validaciones. ■ Cambios registrados en log.

Tabla 4.3: Historias de Usuario – Rol Admin

4.2. Actores de casos de uso

Actor	Cargo	Funciones	Conocimientos	Privilegios
Estudiante	Estudiante pregrado	Desarrollar proyecto de título	Básico	Restringido
Profesor	Académico UBB	Revisar y evaluar propuestas	Intermedio	Medio
Administrador	Jefe carrera	Supervisar flujo académico	Avanzado	Alto
Sistema Email	Servidor SMTP	Envío automático	N/A	N/A
SuperAdmin	Superadmin	Configuración global	Avanzado	Máximo

Tabla 4.4: Actores del Sistema AcTitUBB

4.3. Diagramas y Especificación de casos de uso

El sistema contempla 15 Casos de Uso (CU). La especificación se presenta dividida en tres módulos lógicos.

4.3.1. Visión General y Control de Acceso



Figura 4.1: Diagrama de Casos de Uso: Visión General del Sistema

CU_01: Registrar Usuario

Precondiciones: Email institucional válido (@alumnos.ubiobio.cl o @ubiobio.cl).

Actor	Aplicación
1.1) Accede a registro.	1) Despliega formulario (RUT, nombre, email, pass).
1.2) Completa campos.	3) Valida formatos. Si OK, llama a “CU_03”.
1.3) Confirma registro.	5) Asigna rol, crea cuenta “no confirmada” y envía email.
Flujos Alternativos	
1.2a) Email/datos inválidos.	3.2a) Muestra errores y mantiene datos válidos.
1.3a) Usuario ya existe.	5.3a) Informa existencia y ofrece recuperar pass.

Post condiciones: Usuario registrado “no confirmado”.

CU_02: Autenticar Usuario

Precondiciones: Usuario registrado y confirmado.

Actor	Aplicación
2.1) Accede a login.	1) Despliega formulario.
2.2) Ingresa credenciales.	3) Valida credenciales y estado.
2.3) Redirigido al dashboard.	5) Genera token JWT y redirige según rol.
Flujos Alternativos	
2.2a) Credenciales error.	3.2a) Muestra error genérico.
2.2b) Cuenta no confirmada.	3.2b) Solicita confirmar cuenta.

Post condiciones: Usuario autenticado con token JWT.

CU_03: Confirmar Cuenta

Precondiciones: Email con enlace válido recibido.

Actor	Aplicación
3.1) Clic en enlace email.	1) Valida token y expiración.
3.2) Confirma activación.	3) Actualiza a “confirmada”.
3.3) Redirigido a login.	5) Muestra éxito y llama a “CU_13”.
Flujos Alternativos	
3.1a) Enlace inválido/expirado.	1.1a) Muestra error y ofrece reenvío.

Post condiciones: Cuenta activada.

CU_10: Gestionar Usuarios**Precondiciones:** Admin autenticado.

Actor	Aplicación
10.1) Accede a gestión usuarios.	1) Lista usuarios con filtros.
10.2) Selecciona operación (CRUD).	3) Despliega formulario/confirmación.
10.3) Confirma operación.	5) Ejecuta acción y llama "CU_13".
Flujos Alternativos	
10.3a) Datos inválidos.	5.3a) Muestra errores.

CU_11: Ver Estadísticas**Precondiciones:** Admin en dashboard.

Actor	Aplicación
11.1) Accede a estadísticas.	1) Genera dashboard métricas.
11.2) Aplica filtros.	3) Regenera estadísticas.
11.3) Exporta reportes.	5) Genera archivo exportable.
Flujos Alternativos	
11.2a) Sin datos.	3.2a) Informa ausencia de datos.

CU_12: Configurar Sistema**Precondiciones:** Permisos de configuración.

Actor	Aplicación
12.1) Accede a configuración.	1) Muestra panel parámetros.
12.2) Modifica parámetros.	3) Valida y muestra previa.
12.3) Confirma cambios.	5) Aplica y audita cambios.
Flujos Alternativos	
12.2a) Valores inválidos.	3.2a) Rechaza y muestra error.

CU_13: Enviar Notificaciones**Precondiciones:** Evento gatillador detectado.

Actor	Aplicación
13.1) Detecta evento.	1) Identifica destinatarios.
13.2) Genera mensaje.	3) Aplica plantilla con datos.
13.3) Envía email.	5) Usa SMTP y registra log.
Flujos Alternativos	
13.3a) Fallo SMTP.	5.3a) Programa reintento.

CU_14: Gestionar Administradores

Precondiciones: SuperAdmin autenticado.

Actor	Aplicación
1.1) Accede a gestión admins.	1) Lista admins.
1.2) Crea/Edita admin.	3) Valida y notifica.
1.3) Desactiva cuenta.	5) Marca inactiva y audita.
Flujos Alternativos	
1.2a) Conflicto datos.	3.2a) Muestra error.
1.3a) Eliminar último SA.	5.3a) Impide operación.

CU_15: Configurar Estructura Académica

Precondiciones: SuperAdmin autenticado.

Actor	Aplicación
2.1) Accede a estructura.	1) Muestra depts/carreras.
2.2) Modifica relaciones.	3) Valida consistencia.
2.3) Confirma cambios.	5) Propaga cambios y audita.
Flujos Alternativos	
2.2a) Conflicto relaciones.	3.2a) Informa inconsistencia.

4.3.2. Módulo de Gestión de Propuestas

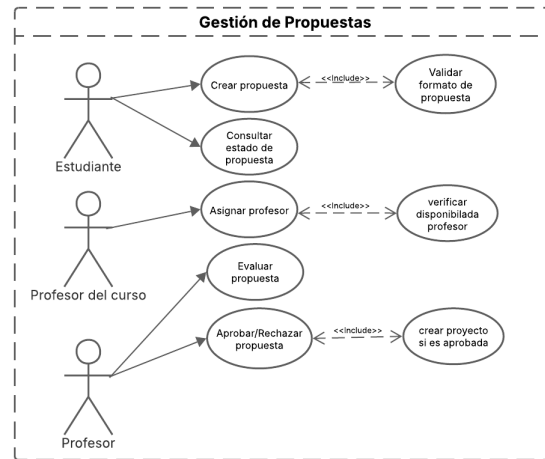


Figura 4.2: Diagrama de Casos de Uso: Gestión de Propuestas

CU_04: Crear Propuesta

Precondiciones: Estudiante registrado.

Actor	Aplicación
4.1) Accede a nueva propuesta.	1) Despliega formulario.
4.2) Completa y adjunta archivo.	3) Valida y llama "CU_06".
4.3) Confirma creación.	5) Guarda "Pendiente" y notifica.
Flujos Alternativos	
4.2a) Archivo inválido.	3.2a) Rechaza archivo.

CU_05: Consultar Estado Propuesta

Precondiciones: Usuario logueado.

Actor	Aplicación
5.1) Selecciona propuesta.	1) Filtra según rol.
5.2) Ve detalles/comentarios.	3) Muestra historial.
5.3) Descarga archivos.	5) Llama "CU_06".
Flujos Alternativos	
5.1a) Sin propuestas.	1.1a) Mensaje informativo.

CU_06: Gestionar Archivos**Precondiciones:** Permisos sobre archivo.

Actor	Aplicación
6.1) Selecciona operación.	1) Verifica permisos.
6.2) Ejecuta acción.	3) Valida archivo/acción.
6.3) Confirma completado.	5) Actualiza BD y audita.
Flujos Alternativos	
6.2a) Archivo inválido.	3.2a) Rechaza y muestra error.

CU_07: Asignar Profesor**Precondiciones:** Propuesta pendiente.

Actor	Aplicación
7.1) Selecciona propuesta.	1) Muestra info básica.
7.2) Clic Asignar Profesor.	3) Lista profesores disponibles.
7.3) Selecciona y confirma.	5) Registra y notifica (CU_13).
Flujos Alternativos	
7.3a) Prof. con sobrecarga.	5.3a) Advierte pero permite.

CU_08: Revisar Propuesta**Precondiciones:** Profesor asignado.

Actor	Aplicación
8.1) Selecciona propuesta.	1) Muestra estado actual.
8.2) Redacta comentarios.	3) Habilita editor.
8.3) Guarda revisión.	5) Llama CU_06 y notifica.
Flujos Alternativos	
8.3a) Error técnico.	5.3a) Guarda borrador.

CU_09: Aprobar/Rechazar Propuesta**Precondiciones:** Revisiones previas existen.

Actor	Aplicación
9.1) Selecciona Decisión.	1) Verifica comentarios previos.
9.2) Confirma decisión.	3) Advierte irreversibilidad.
9.3) Confirma final.	5) Actualiza estado y crea proyecto si aplica.
Flujos Alternativos	
9.1a) Sin comentarios.	1.1a) Redirige a CU_08.

4.3.3. Módulo de Gestión de Proyectos

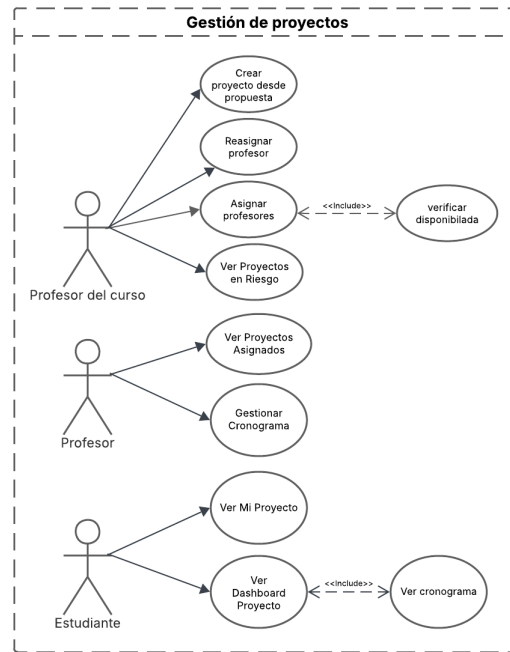


Figura 4.3: Diagrama de Casos de Uso: Gestión de Proyectos

CU_A04: Aprobar Propuesta y Crear Proyecto

Precondiciones: Admin, propuesta aprobada.

Actor	Aplicación
1.1) Revisa aprobación y confirma creación.	1) Crea proyecto, estado “esperando_asignacion”.
Flujos Alternativos	
1.1a) Error en migración de datos.	1.1a) Muestra alerta de inconsistencia y aborta creación.

CU_A05: Asignar Profesores a Proyecto

Precondiciones: Estado “esperando_asignacion”.

Actor	Aplicación
2.1) Asigna Guía, Co-Guía, Informante.	1) Valida duplicidad roles.
2.2) Confirma asignación.	3) Estado “en_desarrollo” y notifica.
Flujos Alternativos	
2.1a) Selecciona mismo prof. para 2 roles.	1.1a) Bloquea acción: “Un profesor no puede tener roles duplicados”.

CU_P09: Gestionar Cronograma**Precondiciones:** Profesor Guía.

Actor	Aplicación
3.1) Accede a Cronograma.	1) Muestra línea de tiempo.
3.2) Crea/Edita hitos y fechas.	3) Guarda y actualiza vista alumno.
Flujos Alternativos	
3.2a) Fecha fin menor a inicio.	3.2a) Muestra error de consistencia temporal.

CU_E07: Ver Dashboard Proyecto**Precondiciones:** Estudiante con proyecto.

Actor	Aplicación
4.1) Selecciona proyecto.	1) Despliega panel control.
4.2) Navega pestañas.	3) Filtra info y fechas.
Flujos Alternativos	
4.1a) Proyecto bloqueado/inactivo.	1.1a) Muestra vista de solo lectura.

CU_E08: Entregar Hito**Precondiciones:** Hito pendiente.

Actor	Aplicación
5.1) Sube archivo entrega.	1) Valida formato, peso y fecha vigente .
5.2) Confirma entrega.	3) Estado “entregado”, notifica Guía.
Flujos Alternativos	
5.1a) Formato no permitido.	1.1a) Rechaza archivo y solicita formato permitido.
5.1b) Plazo de entrega vencido.	1.1b) Bloquea la carga e informa: “El plazo ha expirado”.

CU_P05: Revisar Hito**Precondiciones:** Hito entregado.

Actor	Aplicación
6.1) Descarga y revisa archivo.	1) Habilita feedback.
6.2) Ingresa feedback y decide (Aprobar/Corregir).	3) Actualiza estado y notifica.
Flujos Alternativos	
6.2a) Falla conexión al guardar.	3.2a) Mantiene datos en local y sugiere reiniciar.

CU_E09: Reservar Reunión

Precondiciones: Horarios disponibles.

Actor	Aplicación
7.1) Selecciona prof. y hora.	1) Muestra horarios (CU-P06).
7.2) Define tipo y envía.	3) Bloquea horario, estado “pendiente”.
Flujos Alternativos	
7.2a) Horario tomado por otro usuario (concurencia).	3.2a) Informa “Bloque no disponible” y actualiza lista.

4.4. Matriz de Trazabilidad de Requisitos

La siguiente matriz presenta la correspondencia entre los Requisitos Funcionales (RF) definidos y los Casos de Uso (CU) diseñados, asegurando que todas las funcionalidades requeridas estén cubiertas por al menos un proceso del sistema.

Tabla 4.5: Matriz de Trazabilidad RF vs CU

ID RF	Requerimiento Funcional (Resumen)	Caso de Uso (CU) Asociado
RF-01	Inicio de sesión y generación de JWT	CU_02: Autenticar Usuario
RF-02	Bloqueo de acceso sin token válido	CU_02: Autenticar Usuario (Flujos Alt.)
RF-03	Control de autorización (RBAC)	CU_02, CU_10 y Transversal a todos
RF-04	Registro de propuesta inicial	CU_04: Crear Propuesta
RF-05	Edición de propuesta en borrador	CU_04: Crear Propuesta
RF-06	Aprobación/Rechazo por administrador	CU_09: Aprobar/Rechazar Propuesta
RF-07	Listado y filtrado de propuestas	CU_05: Consultar Estado Propuesta
RF-08	Creación automática de proyecto	CU_A04: Aprobar y Crear Proyecto

ID RF	Requerimiento (Continuación)	CU Asociado
RF-09	Visualización dashboard proyecto	CU_E07: Ver Dashboard Proyecto
RF-10	Finalización/Cancelación de proyecto	CU_E07: Ver Dashboard Proyecto (Admin)
RF-11	Asignación de profesores (Guía/Inf.)	CU_07 y CU_A05: Asignar Profesores
RF-12	Validación de no duplicidad de roles	CU_A05: Asignar Profesores
RF-13	Modificación de asignaciones	CU_A05: Asignar Profesores
RF-14	Creación de cronograma e hitos	CU_P09: Gestionar Cronograma
RF-15	Registro de hitos por estudiante	CU_E08: Entregar Hito / CU_P09
RF-16	Actualización estado hito (Profesor)	CU_P05: Revisar Hito
RF-17	Listado de hitos por proyecto	CU_E07 y CU_P09
RF-18	Gestión de disponibilidad horaria	CU_E09: Reservar Reunión (Config)
RF-19	Solicitud de reunión (Estudiante)	CU_E09: Reservar Reunión
RF-20	Matching de disponibilidad	CU_E09: Reservar Reunión
RF-21	Aceptación/Rechazo de reunión	CU_E09: Reservar Reunión
RF-22	Historial de reuniones	CU_E09: Reservar Reunión
RF-23	Subida de documentos (PDF/DOCX)	CU_06 y CU_E08: Entregar Hito
RF-24	Registro de metadatos documentos	CU_06: Gestionar Archivos
RF-25	Revisión y comentarios en documentos	CU_08 y CU_P05: Revisar Hito
RF-26	Descarga de versiones anteriores	CU_06: Gestionar Archivos
RF-27	Generación de notificaciones internas	CU_13: Enviar Notificaciones
RF-28	Envío de correos electrónicos	CU_13: Enviar Notificaciones
RF-29	Marcar notificaciones como leídas	CU_13 (Gestión de Panel)
RF-30	Registro de auditoría (Logs)	Transversal (Todos los CU con escritura)
RF-31	Filtros avanzados y búsqueda	CU_05 , CU_10 , CU_11
RF-32	Paginación de listados	Transversal (Todos los listados)
RF-33	Visibilidad según Rol (Frontend/Back)	CU_02 (Autenticación) y Transversal
RF-34	Mensajes de error de permisos	Transversal (Manejo de Excepciones)

Capítulo 5

Diseño

Este capítulo presenta el diseño técnico del Sistema AcTitUBB. El diseño prioriza la escalabilidad mediante una arquitectura desacoplada, la integridad de datos a través de un modelo relacional normalizado y la usabilidad siguiendo estándares institucionales.

La arquitectura del sistema se compone de:

- **Frontend:** Angular 18+ (SPA) con diseño responsive.
- **Backend:** Node.js/Express exponiendo una API RESTful.
- **Persistencia:** MySQL 8.0 con integridad referencial estricta.
- **Infraestructura:** Contenerización mediante Docker.

5.1. Diseño de Servicios Web (API REST)

El sistema implementa una arquitectura RESTful ([Higginbotham, 2022](#)) segura y escalable. La comunicación utiliza formato JSON y se protege mediante JSON Web Tokens([Jones et al., 2015](#)) y el Control de Acceso Basado en Roles.

Las Características Técnicas son:

- **Seguridad:** Autenticación JWT con rotación de 1 hora y Rate Limiting (100 req/min).
- **Estándares:** Uso semántico de verbos HTTP (GET, POST, PUT, PATCH, DELETE) y códigos de estado.
- **Documentación:** La especificación completa de los endpoints se encuentra en el **Anexo A**.

La API se estructura en cuatro módulos funcionales principales:

- (a) **Auth & Usuarios:** Gestión de registro, login, recuperación de contraseñas y perfiles. Valida correos institucionales (@ubiobio.cl).
- (b) **Gestión de Propuestas:** CRUD completo para el ciclo de vida de la propuesta (creación, revisión, aprobación).
- (c) **Gestión de Proyectos:** Control de hitos, cronogramas y avances una vez aprobada la propuesta.

- (d) **Administración:** Endpoints privilegiados para gestión global, reportes y configuración del sistema.

5.2. Diseño de Datos

El modelo de datos relacional (MySQL) está normalizado hasta la tercera forma normal (3FN) para garantizar la integridad y reducir la redundancia. Debido a la extensión del modelo (más de 30 tablas), la representación gráfica se ha dividido en tres diagramas temáticos para facilitar su comprensión y legibilidad. El detalle técnico completo de cada tabla (Diccionario de Datos) y los scripts de disparadores (*Triggers*) se detallan en el Anexo B.

5.2.1. Entidades Principales

El modelo se organiza en tres capas lógicas que se detallan a continuación mediante sus respectivos diagramas entidad-relación.

A. Estructura Académica y Usuarios (Capa de Acceso)

Esta capa gestiona los actores del sistema y su pertenencia a la estructura universitaria. Se centra en las **Entidades Nucleares** de usuarios y su jerarquía.

- **Usuarios:** Actores del sistema identificados por RUT.
- **Roles:** Define los permisos (Estudiante, Profesor, Admin).
- **Estructura:** Facultades, Departamentos y Carreras.

La Figura 5.1 ilustra cómo los usuarios (estudiantes y profesores) se relacionan con las unidades académicas.

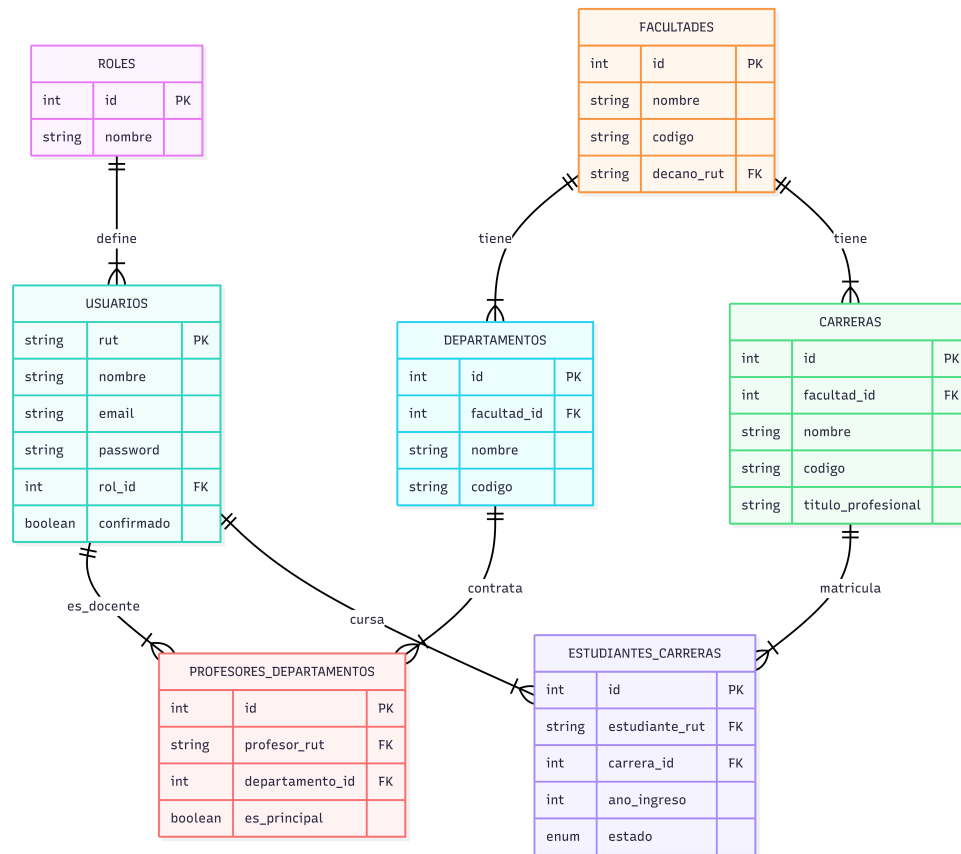


Figura 5.1: Diagrama de la estructura académica y gestión de usuarios.

B. Ciclo de Vida de Titulación (Capa “Core”)

Es el núcleo del negocio. Gestiona la transformación de una idea en un trabajo formal. Incluye tanto **Entidades Nucleares** como **Entidades de Control**:

- **Propuestas:** Solicitudes iniciales de titulación.
- **Proyectos:** Instancia activa derivada de una propuesta aprobada.
- **Roles y Estados:** Tablas paramétricas que controlan el flujo (ej. estados: *pendiente*, *en_revisión*, *aprobada*).
- **Asignaciones:** Tablas intermedias (N:M) que vinculan profesores con propuestas y proyectos.

En la Figura 5.2 se detalla la relación crítica entre la fase de Propuesta y la de Proyecto, incluyendo la asignación del equipo docente (Guía, Informante, etc.).

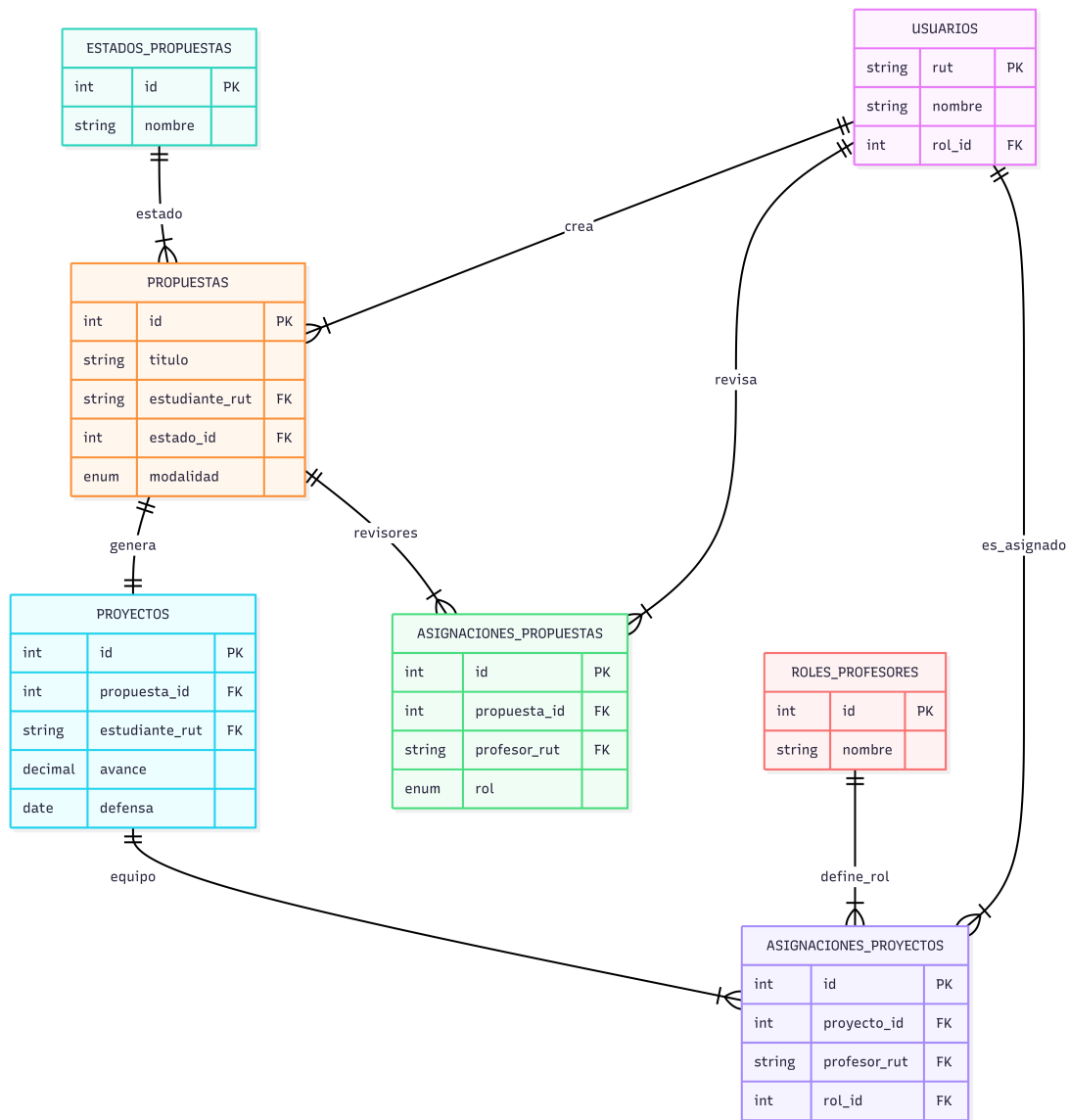


Figura 5.2: Diagrama del ciclo de vida: De Propuesta a Proyecto y asignaciones.

C. Seguimiento y Gestión (Capa Operativa)

Esta capa soporta la operatividad diaria del sistema una vez que el proyecto está en curso. Se compone de **Entidades Operativas**:

- **Avances**: Registro de reportes quincenales y entregas.
- **Cronogramas e Hitos**: Planificación temporal y entregables críticos.
- **Reuniones y Actas**: Coordinación de agenda académica y formalización de acuerdos.

La Figura 5.3 muestra el subsistema de operaciones, destacando cómo se registran las evi-

dencias del progreso del estudiante.

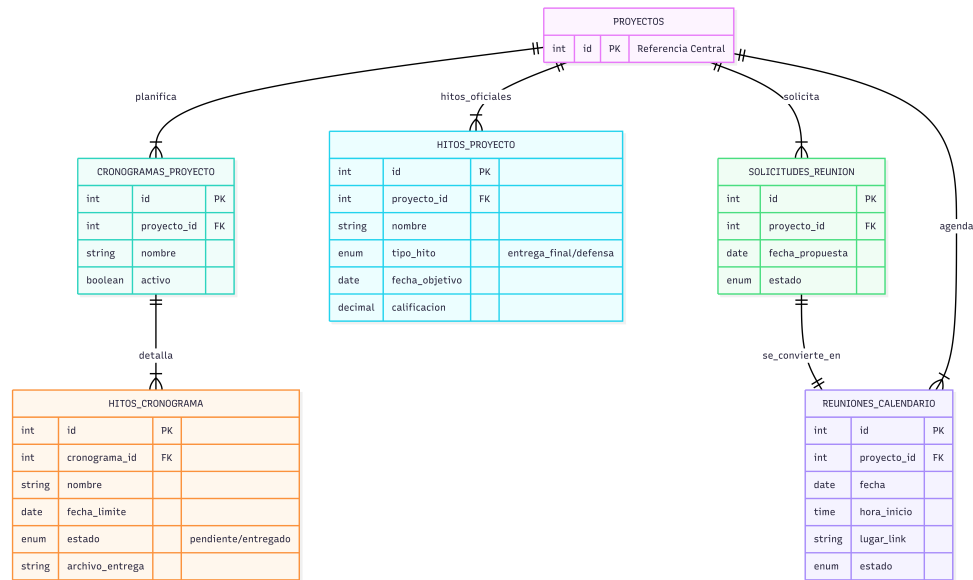


Figura 5.3: Diagrama del subsistema de seguimiento, hitos y reuniones.

5.2.2. Integridad y Reglas de Negocio

El sistema delega reglas críticas a la base de datos para asegurar consistencia:

- Integridad Referencial:** Uso estricto de Claves Foráneas con políticas **CASCADE** (para eliminar datos dependientes, como hitos de un proyecto borrado) o **RESTRICT** (para evitar borrar datos maestros en uso) para evitar registros huérfanos.
- Validaciones (Triggers):**
 - Validación de dominio de correo institucional.
 - Bloqueo de edición en propuestas con estado final (Aprobada/Rechazada).
 - Creación automática del registro de Proyecto al aprobarse una Propuesta.
- Optimización:** Índices estratégicos en campos de búsqueda frecuente (RUT, Estado, Fechas) para optimizar el rendimiento de consultas.

5.3. Diseño e interfaz de navegación

5.3.1. Guía de Estilos y Diseño de Interfaz

La guía de estilos asegura la consistencia visual y el alineamiento con la identidad institucional de la Universidad del Bío-Bío ([Universidad del Bío-Bío, 2025f](#)).

Identidad Visual y Sistema Cromático

El diseño integra el escudo institucional (Figura 5.4) y utiliza una paleta de colores corporativa. El Azul UBB (Figura 5.5) guía la navegación principal, mientras que la escala de grises (Figura 5.6) estructuran el contenido. Adicionalmente, se emplean colores de apoyo semánticos (Figura 5.7) para estados del sistema.



Figura 5.4: Escudo UBB

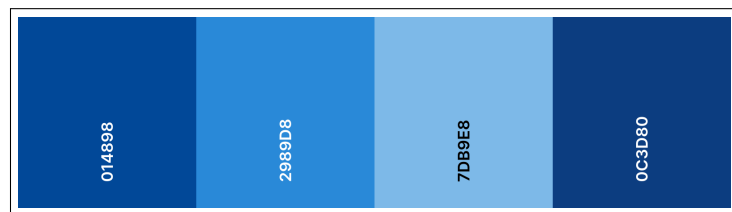


Figura 5.5: Colores Principales

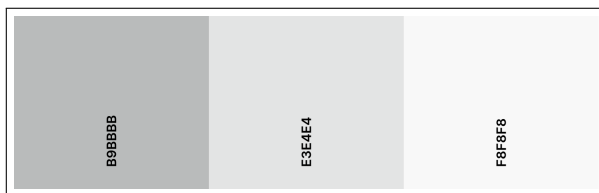


Figura 5.6: Escala de Grises

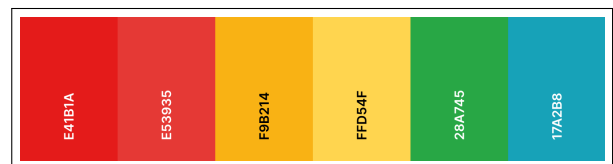


Figura 5.7: Colores de Estado

Figura 5.8: Sistema de Identidad Visual y Paleta Cromática

Gradientes Institucionales Se definen gradientes para encabezados y alertas, aportando profundidad a la interfaz sin perder la sobriedad académica.

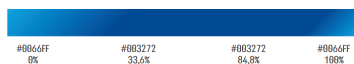


Figura 5.9: Gradiente Azul

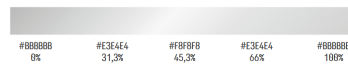


Figura 5.10: Gradiente Gris

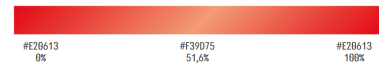


Figura 5.11: Gradiente Rojo

Figura 5.12: Gradientes corporativos aplicados en la interfaz

Tipografía y Componentes

Se utiliza la familia tipográfica poppins (Google, 2025) por su legibilidad en pantallas. Los componentes siguen los principios de *Material Design* con bordes redondeados (4px) y retroalimentación visual inmediata.

- **Botones:** Diferenciados por color según jerarquía (Azul: Primario, Verde: Éxito, Rojo: Peligro).
- **Tarjetas:** Contenedores blancos con sombra suave para agrupar información de propuestas y proyectos.
- **Badges:** Etiquetas de color (píldoras) para indicar estados (ej. “PENDIENTE” en amarillo, “APROBADA” en verde).
- **Accesibilidad:** Cumplimiento WCAG 2.1 AA con alto contraste y navegación por teclado.

5.3.2. Composición de Interfaces

Las siguientes capturas presentan la implementación final de las vistas críticas del sistema, demostrando la aplicación de la guía de estilos.

Acceso y Paneles Principales

La Figura 5.13 muestra el inicio de sesión unificado. Los paneles de control (Figuras 5.14 y 5.15) centralizan la información según el rol, mostrando estados de proyectos y tareas pendientes.

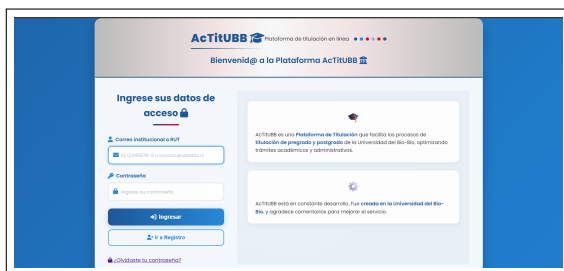


Figura 5.13: Vista de Login

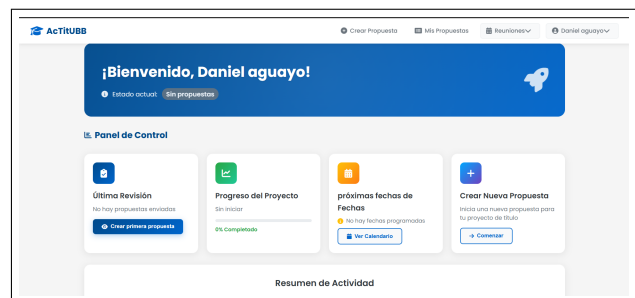


Figura 5.14: Pag principal Estudiante

Gestión Operativa

Las interfaces operativas permiten la coordinación diaria. La Figura 5.16 presenta el modulo del calendario y la Figura 5.17 el formulario de ingreso de una propuesta.

El diseño resultante logra una experiencia de usuario coherente, facilitando la adopción de la herramienta por parte de la comunidad académica.

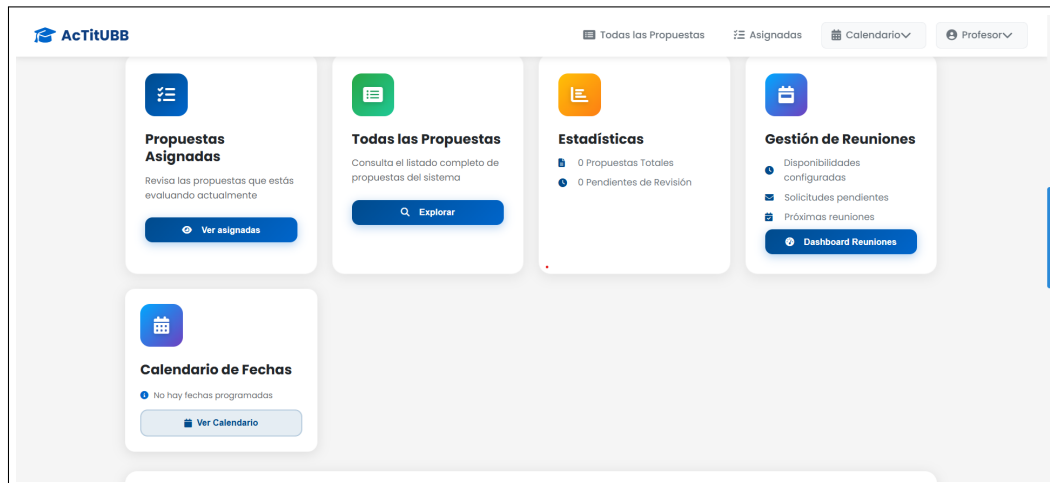


Figura 5.15: Pag principal de Profesores: Gestión de asignaciones y revisiones

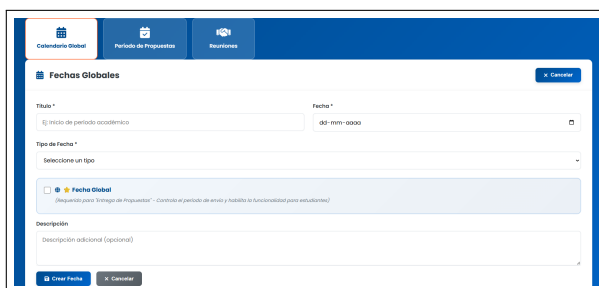


Figura 5.16: Módulo de Calendario



Figura 5.17: Formulario de Propuesta

Figura 5.18: Interfaces de gestión operativa del sistema

Capítulo 6

Desarrollo del Trabajo

Este capítulo documenta de forma exhaustiva el proceso de concepción, diseño, implementación, problemáticas afrontadas y validación del sistema AcTitUBB. Se describen las decisiones arquitectónicas, la estructura del código fuente, las prácticas aplicadas durante el desarrollo, los riesgos mitigados y los resultados obtenidos tras las pruebas y optimizaciones. El objetivo es dejar trazabilidad técnica clara que permita mantener, extender y auditar el sistema en fases futuras.

En términos de alcance, AcTitUBB es una plataforma web para la gestión del proceso de titulación en la Universidad del Bío-Bío, involucrando a tres perfiles principales: Administrador, Profesor y Estudiante. El sistema soporta: autenticación y autorización por roles, administración de usuarios y perfiles, gestión de propuestas y proyectos (incluyendo carga/descarga de documentos), agenda y calendario de hitos (fechas importantes), asignación de profesores, comunicación por correo para eventos críticos (por ejemplo, reseteo de credenciales) y un flujo seguro de cambio de contraseña. El desarrollo se realizó con Angular en el cliente, Node.js/Express en la API y MySQL como base de datos, todo contenedorizado con Docker para facilitar despliegue y operación.

6.1. Diseño de Arquitectura

La arquitectura se fundamenta en una separación por responsabilidades: frontend (cliente web Angular), backend (API REST Node.js/Express) y persistencia (MySQL 8). Todo orquestado mediante contenedores Docker y aislado en una red interna bridge para minimizar superficie de ataque. Se privilegió una arquitectura modular y escalable, con énfasis en seguridad de autenticación, integridad de datos y mantenibilidad.

6.1.1. Configuración Inicial del Entorno

La configuración inicial contempló:

- **Control de versiones:** Repositorio Git en rama principal `main`.
- **Lenguajes:** TypeScript (frontend), JavaScript ES Modules (backend), SQL (scripts DB).

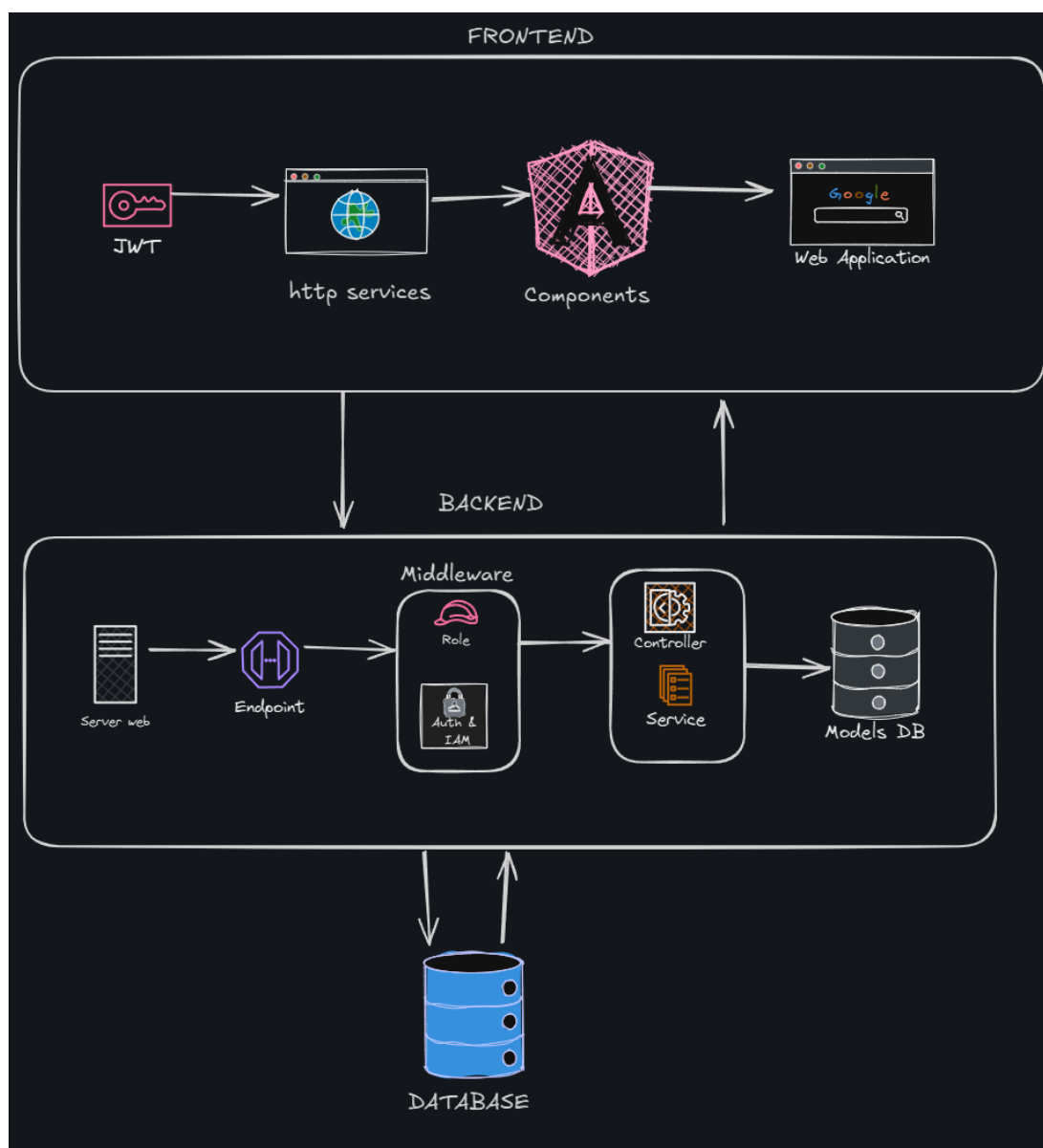


Figura 6.1: Diagrama de la Arquitectura del Sistema AcTitUBB

- **Base de datos:** MySQL 8.0 con inicialización mediante un script SQL montado en `/docker-entrypoint-initdb.d/`.
- **Gestión de dependencias:** npm para cada paquete (`backend/`, `frontend/`).
- **Contenedores:** Definición de servicios en `docker-compose.yml` (MySQL, Backend, Frontend).
- **Seguridad básica:** Variables de entorno en archivo `.env` (credenciales DB, secret JWT, credenciales SMTP).
- **Prácticas de autenticación:** Uso de JWT firmado y verificado en middleware.
- **Hasheo de contraseñas:** bcryptjs con costo (`salt rounds`) = 10 para consistencia rendimiento/seguridad.
- **Correo transaccional:** nodemailer para flujos de reseteo de contraseña.

6.1.2. Arquitectura de Implementación Final

La arquitectura final en producción se despliega sobre un servidor con IP pública 146.83.194.188 usando mapeo de puertos externos a internos mediante Docker:

Servicio	Puerto Externo	Puerto Interno
SSH	2240	22
HTTP (Frontend Nginx)	8090	80
HTTPS (Frontend Nginx)	8453	443
Flask (Reservado/Futuro)	2241	5000
Node.js Backend	2242	3000
MySQL	3306	3306

Tabla 6.1: Mapeo de puertos para servicios en producción

Patrones aplicados

- **Separación de capas:** Controladores, modelos, servicios, middlewares y rutas en el backend para desacoplar lógica.
- **DTO implícitos:** Respuestas normalizadas (estatus, mensaje, payload) para facilitar consumo en Angular.
- **Inmutabilidad lógica de credenciales:** Contraseñas nunca se retornan; se usa verificación con bcrypt.
- **Idempotencia en operaciones de reseteo:** La regeneración de contraseña temporal marca bandera `debe_cambiar_password` sin duplicar efectos secundarios.
- **Encapsulación:** Servicios como `email.service.js` centralizan interacción SMTP.

Docker Compose Se definieron tres servicios activos:

`mysql` con volumen persistente `mysql_data`; `backend` con build propio y healthcheck sobre ruta `/health`; `frontend` sirviendo contenido estático (Angular compilado) en puerto 80 re-mapeado al 8090 externo. La red `actitubb_network` aísla la comunicación interna (API `http://backend:3000`).

6.1.3. Modelo de Datos

El modelo de datos se basa en entidades coherentes con los procesos académicos y administrativos:

- **Usuario:** Credenciales, datos básicos y su asociación a un *Rol*; incluye bandera de seguridad `debe_cambiar_password`.
- **Rol:** Catálogo de permisos (Administrador, Profesor, Estudiante) consumido por el backend y el guard del frontend.
- **Proyecto:** Unidad central de trabajo académico, asociada a estudiantes y profesores.
- **Propuesta:** Documento(s) y metadatos de postulaciones; se persisten archivos en `uploads/`.
- **Avance:** Hitos o entregables intermedios asociados a un proyecto.
- **Calendario y Fecha Importante:** Gestión de eventos e hitos institucionales relevantes para el proceso.

Las relaciones específicas (uno-a-muchos y muchos-a-uno) se implementan en el nivel de modelos y consultas SQL, resguardando integridad y trazabilidad.

6.2. Estructura del Código

La estructura se diseñó para que cada área de responsabilidad fuera localizable y extensible.

6.2.1. Backend

```
backend/
  dockerfile
  package.json
  src/
    index.js           # Punto de entrada Express
    controllers/       # Lógica HTTP (admin, login, proyectos, etc.)
    db/               # Conexión y script inicial SQL
    middlewares/       # Verificación sesión, subida archivos, blacklist
    models/           # Abstracciones de acceso datos (usuarios, roles, proyectos)
    routes/           # Definición de endpoints organizados
    services/         # Servicios auxiliares (email, validación Rut, etc.)
    uploads/          # Almacenamiento documentos (propuestas)
    migrations/        # Scripts de modificación esquema (banderas seguridad)
```

Dependencias clave (extracto de `package.json`): `express`, `mysql2`, `jsonwebtoken`, `bcryptjs`, `multer`, `nodemailer`, `express-rate-limit`.

Responsabilidades destacadas

- **controllers:** Orquestan validaciones, llaman modelos y retornan respuesta.
- **models:** Encapsulan consultas SQL evitando SQL disperso.
- **middlewares:** Autenticación JWT, protección contra abuso (rate limiting), manejo uploads.

- **services:** Funciones reutilizables (correo, reglas negocio específicas).

Controladores y rutas principales (extracto real del repositorio):

- **login.controller** y **login.route:** Inicio de sesión, validación de credenciales, flujo de cambio de contraseña propia (PUT /users/cambiar-password).
- **admin.controller** y **admin.route:** Gestión de usuarios, reseteo de contraseñas con notificación por correo.
- **project.controller** y **project.route:** Operaciones sobre proyectos académicos.
- **propuesta.controller** y **propuesta.routes:** Carga, registro y gestión de propuestas (*uploads* con *multer*).
- **calendario.controller** y **calendario.route:** Administración de calendario y fechas importantes.
- **role.controller** y **role.route:** Catálogo de roles y asignaciones.
- **download.route:** Descarga de documentos resguardados en el servidor.
- **middlewares:** `verifySession` (JWT), `uploader` (archivos), `blacklist` (seguridad de rutas).
- **services:** `email.service` (Nodemailer), `RutVal.service`, `project.service`, `propuesta.service`.

6.2.2. Frontend

```
frontend/
  angular.json
  package.json
  src/
    main.ts                # Bootstrap aplicación Angular standalone
    app/                   # Configuración global, rutas, componente raíz
      app.routes.ts        # Rutas con lazy loading
      components/          # Componentes reutilizables (modales, etc.)
      pages/               # Pantallas: login, admin, profesor, estudiante, propuestas
        cambiar-password/ # Flujo forzado de cambio contraseña
    services/              # API client (HttpClient), auth helpers
    interceptors/          # Interceptor JWT
    styles.scss            # Estilos globales (temas, variables)
```

Dependencias clave: Angular 20 (core, router, forms, material), rxjs, jwt-decode, file-saver.

Patrones Angular

- **Standalone Components:** Evita sobrecarga de módulos y simplifica lazy loading.
- **Lazy Loading:** Rutas de secciones grandes cargan bajo demanda para optimizar rendimiento inicial.
- **Interceptor JWT:** Inserta token en encabezados Authorization y maneja expiraciones.
- **Servicios centralizados:** `api.ts` gestiona endpoints; promueve reutilización.

Rutas y vistas principales (extracto):

- `pages/login`: Acceso y validación de credenciales.
- `pages/register`: Registro (si aplica al despliegue).
- `pages/admin`: Gestión de usuarios, profesores, asignaciones y calendario.
- `pages/profesor` y `pages/estudiante`: Vistas específicas por perfil.
- `pages/propuestas`: Gestión documental.
- `pages/cambiar-password`: Flujo forzado de actualización de contraseña.
- `guards/auth.guard.ts`: Protección de rutas privadas.
- `interceptors/auth.interceptor.ts`: Inserción de JWT y manejo de respuestas 401.

6.3. Desarrollo del Software

El desarrollo se organizó en iteraciones cortas (enfoque incremental) priorizando entregas funcionales verificables. Se aplicó retroalimentación continua sobre problemas detectados en autenticación, seguridad y consistencia de datos.

6.3.1. Problemas Durante el Desarrollo

Principales desafíos y su mitigación:

- Contraseñas sin hasheo en reseteo**: El flujo inicial guardaba contraseñas temporales en texto plano generando error 401 al comparar con `bcrypt`. *Mitigación*: Unificación de estrategia de hasheo en creación y reseteo (salt consistente).
- Falta de bandera de seguridad**: No existía mecanismo para forzar cambio de contraseña tras reseteo. *Solución*: Migración agregando columna `debe_cambiar_password` y lógica de redirección frontend.
- Trazabilidad deficiente en correos**: Se requería correo transaccional fiable. *Acción*: Implementación de plantilla HTML accesible + manejo de errores silencioso registrado en log.
- Sin pruebas automatizadas iniciales**: Dificultad para regresión rápida. *Estrategia*: Se definieron casos de prueba manual estructurados + base para tests unitarios futuros (`api.spec.ts`).
- Acoplamiento potencial entre controlador y SQL**: Riesgo de duplicación. *Refactor*: Encapsular operaciones en `models` reduciendo repetición.
- Gestión de uploads**: Necesidad de persistir archivos propuestas docente/estudiante. *Implementación*: Uso de `multer` con restricciones de ruta aislada y montaje de volumen Docker.
- Riesgo de ataques de fuerza bruta**: Endpoint de login expuesto. *Mitigación*: Integración de `express-rate-limit` para limitar intentos.

6.3.2. Pruebas, Optimización y Métricas Finales

Estrategia de pruebas Se aplicó matriz de casos críticos:

- **Autenticación**: Login correcto, login erróneo, token expirado, acceso sin token.

- **Reseteo contraseñas:** Flujo completo (admin genera, correo llega, login temporal, cambio definitivo).
- **Integridad de datos:** Creación de proyectos, asignaciones y propuestas (consistencia relacional).
- **Uploads:** Tipos válidos (PDF/DOCX), rechazos por tamaño/forma.
- **Correo:** Entrega, renderizado en distintos webmails, verificación de codificación UTF-8.

Optimización

- **Cold Start Backend:** Uso de healthcheck para garantizar disponibilidad antes de servir frontend.
- **Carga inicial Angular:** Lazy loading reduce TTI (Time To Interactive) en módulos que el usuario aún no visita.
- **Reducción de lógica duplicada:** Consolidación de utilidades en servicios (validación rut, envío email) disminuyendo mantenimiento.

Métricas de Referencia Empírica

- **Latencia promedio login:** < 200 ms en entorno contenedorizado local.
- **Tiempo reseteo + correo:** 1,5 s – 3 s según respuesta SMTP.
- **Hasheo bcrypt (10 rounds):** 80 – 120 ms promedio por operación.
- **Uso de memoria backend:** < 120 MB en estado estable.

Seguridad validada

- Contraseñas nunca retornadas ni almacenadas en texto claro.
- Tokens JWT firmados con secreto en entorno (no hard-coded).
- Flujo obligatorio de cambio tras credencial temporal mitiga reutilización indebida.
- Aislamiento de red interna evita exposición directa de MySQL.

Limitaciones actuales

- Ausencia de pruebas automatizadas end-to-end integradas en CI.
- Falta de monitoreo activo (logs centralizados, métricas Prometheus).
- Política de rotación de contraseñas avanzada pendiente (historial y expiración).

6.4. Despliegue y Operación

Orquestación y red El despliegue se realiza con `docker-compose`, definiendo una red interna (bridge) que aísla MySQL y el backend del exterior, exponiendo únicamente el frontend y el backend según corresponda. Los volúmenes persisten datos de MySQL y documentos subidos.

Servidor de producción IP pública: 146.83.194.188. Los puertos externos se mapean a servicios internos:

Servicio	URL/Host externo	Puerto interno
Frontend (HTTP)	<code>http://146.83.194.188:8090</code>	80
Frontend (HTTPS)	<code>https://146.83.194.188:8453</code>	443
Backend (Node.js)	<code>http://146.83.194.188:2242</code>	3000
MySQL	<code>146.83.194.188:3306</code>	3306
SSH	<code>146.83.194.188:2240</code>	22

Tabla 6.2: Configuración de URLs y puertos en el servidor de producción

Nota: La terminación TLS (:8453) puede realizarse mediante un reverse proxy del servidor. El frontend consume la API interna como `http://backend:3000` dentro de la red Docker; externamente, los clientes pueden usar `http://146.83.194.188:2242` si se expone.

Variables de entorno (resumen)

- Backend: DB_HOST, DB_PORT, DB_NAME, DB_USER, DB_PASSWORD, JWT_SECRET, EMAIL_USER, EMAIL_PASS.
- Frontend: API_URL (apunta al backend en la red o al host externo según despliegue).

Comandos de despliegue (ejecución desde la raíz del proyecto):

```
docker compose build
docker compose up -d
docker compose ps
```

Operación y mantenimiento

- **Healthchecks:** Backend expone `/health` para validación de estado.
- **Backups:** Resguardar el volumen `mysql_data` periódicamente.
- **Logs:** Revisar logs de contenedores (`docker compose logs -f backend_actitubb`).
- **Seguridad:** Rotar JWT_SECRET y credenciales SMTP; mantener actualizado el stack de imágenes.

6.5. Conclusión del Capítulo

El proceso de desarrollo consolidó una base sólida, segura y mantenible para AcTitUBB. Se logró una arquitectura modular con separación clara de responsabilidades, endurecimiento de autenticación y trazabilidad del ciclo de vida de credenciales. Los problemas detectados (especialmente relacionados a seguridad de contraseñas y flujo de reseteo) se resolvieron mediante refactorizaciones quirúrgicas y ampliación de capacidades (correo transaccional, bandera `debe_cambiar_password`). La instrumentación básica (healthchecks, estructura de servicios) prepara el terreno para escalado futuro.

Como trabajo futuro se proyecta: ampliar suite de pruebas automatizadas, añadir monitoreo y alertas, incorporar métricas de uso e integrar el servicio Flask para analítica avanzada. Con

esto, la plataforma podrá evolucionar a un ecosistema más robusto sin comprometer los principios de seguridad y mantenibilidad establecidos.

Estado final: Arquitectura desplegable, flujo de autenticación seguro, módulos listos para extensión.

Capítulo 7

Implantación y Puesta en Marcha

Este capítulo presenta el plan integral para pasar de un entorno de desarrollo/validación a producción, asegurando continuidad operativa, adopción por parte de los usuarios y criterios de calidad objetivos. Incluye el plan de capacitación, la estrategia de despliegue basada en contenedores Docker, la gestión del cambio y los mecanismos de soporte y evaluación.

El alcance cubre los perfiles de *Administrador*, *Profesor* y *Estudiante*; la plataforma web **AcTi-tUBB** (Angular en el cliente, Node.js/Express en el backend, MySQL 8), y su operación en un servidor público con orquestación `docker-compose`.

7.1. Plan de Capacitación

La capacitación es un factor crítico para una adopción efectiva. Se estructura por objetivos, audiencias, materiales y modalidad de entrega, resguardando trazabilidad de asistencia y evaluación.

7.1.1. Objetivo del Plan de Capacitación

Objetivos específicos:

- **Adopción funcional:** Asegurar que cada rol pueda desempeñar sus tareas clave sin fricciones.
- **Autonomía operativa:** Reducir dependencia del equipo técnico para actividades rutinarias.
- **Seguridad y buenas prácticas:** Establecer pautas de uso seguro (gestión de contraseñas, manejo de documentos, sesiones).
- **Trazabilidad:** Registrar participación, resolver dudas frecuentes y medir aprendizaje.

Por rol

- **Administrador:** Gestión de usuarios, reseteo de contraseñas, asignaciones, calendario, monitoreo básico, lectura de logs y resolución de incidencias comunes.

- **Profesor:** Revisión y gestión de propuestas/proyectos, interacción con calendario, descarga de documentos, comunicación con estudiantes.
- **Estudiante:** Autenticación, carga de propuestas y documentos, seguimiento de estados, cambio de contraseña, lectura de notificaciones.

Entregables de capacitación

- Manuales PDF por rol (guía completa y guías rápidas de 1 página).
- Videos cortos (2–4 minutos) de tareas frecuentes.
- Banco de preguntas frecuentes (FAQ) y resolución de errores comunes.
- Registro de asistencia y evaluación breve por sesión (cuestionario de 5–10 ítems).

7.1.2. Fases del Plan de Entrenamiento

La estrategia se organiza en cuatro fases:

- (a) **Preparación** (Semana -2 a -1): levantamiento de necesidades por rol, calendarización, publicación de materiales, habilitación de ambientes de práctica.
- (b) **Formación** (Semana -1): sesiones sincrónicas por rol (remotas o presenciales), con demostraciones guiadas y ejercicios.
- (c) **Acompañamiento inicial** (Semana 0 a +1): soporte reforzado tipo "mesa de ayuda" durante el periodo de puesta en marcha.
- (d) **Evaluación y mejora** (Semana +2): análisis de métricas, encuesta de satisfacción, plan de ajustes.

KPIs de capacitación

- Cobertura de capacitación por rol (% usuarios capacitados).
- Puntaje promedio de evaluación (objetivo > 80 %).
- Tasa de incidencias por uso (tickets por 100 usuarios en Semana 0–1).

7.2. Estrategia de Implantación y Puesta en Marcha

La implantación se realiza con `docker-compose` en un servidor público, asegurando reproducibilidad, aislamiento por servicio y verificación de salud antes de apertura a usuarios.

7.2.1. Entorno y topología

- **Servidor producción:** IP pública 146.83.194.188.
- **Puertos externos → internos:** SSH (2240→22), HTTP (8090→80), HTTPS (8453→443), Flask reservado (2241→5000), Node.js (2242→3000), MySQL (3306→3306).
- **Red Docker:** bridge, con comunicación interna entre servicios (API como `http://backend:3000`).
- **Volúmenes:** Persistencia de datos MySQL (`mysql_data`) y `uploads/` de documentos.

7.2.2. Variables de entorno y secretos

- Backend: DB_HOST, DB_PORT, DB_NAME, DB_USER, DB_PASSWORD, JWT_SECRET, EMAIL_USER, EMAIL_PASS.
- Frontend: API_URL (interno o externo según topología).

7.2.3. Despliegue

Comandos de ejecución

Desde la raíz del proyecto

```
docker compose build
```

```
docker compose up -d
```

Validación

```
docker compose ps
```

Verificar healthcheck del backend (/health)

Verificaciones post-despliegue

- Frontend disponible en <http://146.83.194.188:8090> (y HTTPS si se configura proxy en 8453).
- Backend accesible en <http://146.83.194.188:2242> y saludable en `/health`.
- Carga/descarga de documentos funcional en rutas protegidas.
- Envío de correos (reseteo de contraseña) operativo.

7.2.4. Migración de datos y corte

- **Inicialización:** Script SQL montado en `/docker-entrypoint-initdb.d/` para poblar esquema/base.
- **Congelamiento de cambios** (T-24h): ventana de solo lectura si aplica migración desde sistema previo.
- **Corte** (T0): levantamiento de contenedores en producción y validación funcional guiada.
- **Apertura** (T+2h): comunicación a usuarios y monitoreo reforzado.

7.2.5. Plan de reversa

- Respaldo previo de `mysql_data` y `uploads/`.
- *Rollback* de imágenes a la etiqueta anterior.
- Restauración de respaldo en caso de inconsistencias.

7.2.6. Soporte y operación inicial

- **Mesa de ayuda:** canal dedicado Semana 0–1 para incidencias de adopción.
- **Monitoreo:** revisión de logs (`docker compose logs -f backend_actitubb`), métricas básicas (CPU/mem).

- **Seguridad:** rotación periódica de JWT_SECRET y credenciales SMTP; actualización de imágenes.

7.2.7. Gestión del cambio y comunicación

- Avisos institucionales con calendario y beneficios de la plataforma.
- Guías rápidas adjuntas y recordatorio del flujo de reseteo/cambio de contraseña.
- Recopilación de retroalimentación y mejoras priorizadas para el primer ciclo post-implantación.

Capítulo 8

Conclusiones

El desarrollo del sistema AcTitUBB permitió abordar de manera efectiva las limitaciones presentes en el proceso tradicional de gestión de titulación dentro de las carreras de informática de la Facultad de Ciencias Empresariales de la Universidad del Bío-Bío. A través de una solución web centralizada, se logró modernizar y estandarizar actividades críticas, como la gestión de propuestas, la asignación de roles académicos, el seguimiento de avances y la coordinación de reuniones, integrando funcionalidades automatizadas que mejoran significativamente la trazabilidad, la comunicación y la eficiencia operativa.

La plataforma implementada, construida con tecnologías modernas y soportada por una arquitectura modular, demostró ser técnicamente viable, operativamente adecuada y económicamente justificable. La incorporación de mecanismos robustos de autenticación, auditoría, manejo de archivos y control de estados asegura un funcionamiento confiable y alineado con los estándares institucionales y de calidad del software.

Asimismo, la evaluación metodológica y técnica realizada durante el proyecto evidenció que AcTitUBB no solo responde a una necesidad actual, sino que además constituye una base escalable para futuros mejoramientos e integraciones. Su adopción proyecta beneficios directos en la reducción de la carga administrativa, la disminución de errores humanos, el fortalecimiento de la coordinación académica y la mejora en la experiencia de estudiantes y profesores durante el proceso de titulación.

8.0.1. Trabajos Futuros

A partir de la base establecida por AcTitUBB, se identifican diversas líneas de trabajo futuro que permitirían potenciar aún más las capacidades del sistema:

En el corto plazo (3-6 meses), se recomienda implementar un sistema de notificaciones push y alertas por múltiples canales (correo, SMS, navegador), desarrollar una aplicación móvil progresiva (PWA) para mejorar la accesibilidad desde dispositivos móviles, incorporar autenticación de dos factores (2FA) para fortalecer la seguridad, y ampliar el dashboard con visualizaciones interactivas y métricas avanzadas de seguimiento. Además, se sugiere establecer un sistema automatizado de respaldos y recuperación ante desastres.

Para el mediano plazo (6-12 meses), resulta relevante desarrollar un repositorio institucional digital que permita almacenar, buscar y consultar proyectos finalizados, facilitando el acceso al conocimiento generado. La implementación de un sistema de mensajería interna con canales por proyecto y videollamadas integradas fortalecería la comunicación entre participantes. Asimismo, la integración bidireccional con el sistema académico institucional permitiría sincronizar datos de estudiantes, profesores y mallas curriculares. La incorporación de inteligencia artificial para el matching automático entre profesores y proyectos, considerando expertise, disponibilidad y carga académica, optimizaría el proceso de asignación. Por último, la implementación de rúbricas de evaluación estandarizadas y estructuradas contribuiría a la objetividad y consistencia en las evaluaciones.

En el largo plazo (12-24 meses), se plantea incorporar técnicas de análisis predictivo mediante machine learning para identificar proyectos en riesgo de retraso y proponer acciones correctivas tempranas. El desarrollo de aplicaciones móviles nativas para iOS y Android ampliaría significativamente el alcance y usabilidad del sistema. La implementación de un módulo completo de gestión de recursos físicos (laboratorios, equipamiento, salas) facilitaría la coordinación de espacios y materiales. La integración de las mallas curriculares completas permitiría validar automáticamente que los estudiantes cumplan con los requisitos académicos previos a la titulación. Finalmente, la migración hacia una arquitectura de microservicios mejoraría la escalabilidad, el mantenimiento y la capacidad de integración con otros sistemas universitarios.

Estas líneas de trabajo futuro no solo extenderían las funcionalidades actuales del sistema, sino que posicionarían a AcTitUBB como una plataforma integral de gestión académica, capaz de evolucionar junto con las necesidades institucionales y tecnológicas de la universidad.

En síntesis, el proyecto cumplió con los objetivos planteados, entregando una herramienta funcional, sólida y pertinente al contexto universitario, contribuyendo a la transformación digital de los procesos académicos y estableciendo las bases para un desarrollo continuo y sostenible en el tiempo.

Referencias

- Asana Inc. Asana - precios de planes personal, starter, advanced y enterprise. 2025. URL <https://asana.com/es/pricing>.
- Atlassian Trello. Trello - planes y precios. 2025. URL <https://trello.com/es/pricing>.
- ClickUp. Clickup - the everything app for work. 2025. URL <https://clickup.com/pricing>.
- Dirección General de Análisis Institucional, Universidad del Bío-Bío. Caracterización estudiantil - datos de titulación 2024. 2025a. URL <https://dgai.ubiobio.cl/modelos-de-gestion/caracterizacion-estudiantil/>.
- Dirección General de Análisis Institucional, Universidad del Bío-Bío. Personal institucional - departamento de sistemas de información. 2025b. URL <https://dgai.ubiobio.cl/modelos-de-gestion/personal-institucional/>.
- Docker Inc. Docker documentation. 2025. URL <https://docs.docker.com/>. Plataforma de contenedores.
- Expressjs. Multer: Node.js middleware for handling multipart/form-data. 2025. URL <https://github.com/expressjs/multer>. Librería npm.
- F5, Inc. Nginx: High performance load balancer, web server, reverse proxy. 2025. URL <https://www.nginx.com/>. Servidor Web.
- Face. Reglamento actividad de titulación carreras de la facultad de ciencias empresariales. 2025. URL https://moodleubb.ubiobio.cl/pluginfile.php/543485/mod_resource/content/1/Reglamento.titulaci%C3%B3n.IECI.2025%20%281%29.pdf.
- Google. Google fonts - poppins. 2025. URL <https://fonts.google.com/specimen/Poppins>.
- Google LLC. Angular - the modern web developer's platform. 2025. URL <https://angular.dev/>.
- James Higginbotham. *Principles of Web API Design: Delivering Value with APIs and Microservices*. Addison-Wesley Professional, Boston, MA, USA, 2022. ISBN 978-0137355631.

- IEEE. Ieee recommended practice for software requirements specifications. Inf. Téc. IEEE Std 830-1998, Institute of Electrical and Electronics Engineers, 1998a. Revision of IEEE Std 830-1993.
- IEEE. Ieee standard for software test documentation. Inf. Téc. IEEE Std 829-1998, Institute of Electrical and Electronics Engineers, 1998b. Revision of IEEE Std 829-1983.
- ISO/IEC. Iso/iec 25010:2011 systems and software engineering – systems and software quality requirements and evaluation (square) – system and software quality models. Inf. Téc. ISO/IEC 25010:2011, International Organization for Standardization, 2011.
- M. Jones, J. Bradley, y N. Sakimura. Json web token (jwt). RFC 7519, IETF, 2015. URL <https://tools.ietf.org/html/rfc7519>.
- Keymetrics. Pm2 - advanced, production process manager for node.js. 2025. URL <https://pm2.keymetrics.io/>. Gestor de procesos.
- Microsoft. Typescript: Javascript with syntax for types. 2025a. URL <https://www.typescriptlang.org/>. Accedido: 2025.
- Microsoft. Visual studio code - code editing. redefined. 2025b. URL <https://code.visualstudio.com/>. Herramienta de desarrollo.
- monday.com Ltd. monday.com - plataforma de trabajo colaborativo. 2025. URL <https://monday.com/lang/es/pricing/>.
- NIC Chile. Tarifas de dominios .cl. 2025. URL <https://www.nic.cl/dominios/tarifas.html>.
- Node.js Foundation. Node.js. 2025. URL <https://nodejs.org/>.
- Notion Labs Inc. Notion - planes y características. 2025. URL <https://www.notion.com/es/pricing>.
- Object Management Group. *OMG Unified Modeling Language (OMG UML), Version 2.5.1*. Object Management Group, Needham, MA, 2017. URL <https://www.omg.org/spec/UML/2.5.1>. Especificación Formal.
- OpenJS Foundation. Express - fast, unopinionated, minimalist web framework for node.js. 2025. URL <https://expressjs.com/>. Accedido: 2025.
- Oracle Corporation. Mysql :: Mysql 8.0 reference manual. 2025a. URL <https://dev.mysql.com/doc/refman/8.0/en/>.
- Oracle Corporation. *MySQL Workbench*. Oracle, 2025b. URL <https://www.mysql.com/products/workbench/>. Versión 8.0.
- PNGWing. Ilustración del modelo de desarrollo iterativo e incremental. 2025. URL <https://www.pngwing.com/es/free-png-mlrxn>. Imagen en formato PNG.

- Postman Inc. Postman api platform. 2025. URL <https://www.postman.com/>. Herramienta de pruebas.
- Software Freedom Conservancy. Git –local-branching-on-the-cheap. 2025. URL <https://git-scm.com/>. Sistema de control de versiones.
- Universidad del Bío-Bío. Admisión - ingeniería civil en informática concepción. 2025a. URL https://ubiobio.cl/admision/Todas_las_Carreras/17/Ingenieria_Civil_en_Informatica_Concepcion/.
- Universidad del Bío-Bío. Admisión - ingeniería de ejecución en computación e informática. 2025b. URL https://ubiobio.cl/admision/Todas_las_Carreras/21/Ingenieria_de_Ejecucion_en_Computacion_e_Informatica/.
- Universidad del Bío-Bío. Malla curricular - ingeniería civil en informática concepción. 2025c. URL https://ubiobio.cl/admision/Todas_las_Carreras/17/Ingenieria_Civil_en_Informatica_Concepcion/.
- Universidad del Bío-Bío. Malla curricular - ingeniería de ejecución en computación e informática. 2025d. URL https://ubiobio.cl/admision/Todas_las_Carreras/21/Ingenieria_de_Ejecucion_en_Computacion_e_Informatica/.
- Universidad del Bío-Bío. Portal de transparencia - remuneraciones. 2025e. URL <https://www.ubiobio.cl/transparencia/>.
- Universidad del Bío-Bío. Universidad del bío-bío, única universidad pública de las regiones del biobío y Ñuble. 2025f. URL <https://ubiobio.cl/>.

Apéndice A

Definiciones y abreviaciones del Negocio

A.1. Definiciones y abreviaciones del Negocio

Término/Sigla	Definición
UBB	Universidad del Bío-Bío. Institución estatal y pública de educación superior con presencia en las ciudades de Concepción y Chillán.
UTE	Universidad Técnica del Estado. Institución de educación superior antecesora de la UBB, fundada en 1947.
IPOCH	Instituto Profesional de Chillán. Entidad fusionada con la sede Concepción para conformar la actual Universidad del Bío-Bío.
FACE	Facultad de Ciencias Empresariales de la Universidad del Bío-Bío. Unidad académica donde se imparte la carrera de Ingeniería en Informática.
IECI	Ingeniería de Ejecución en Computación e Informática. Carrera profesional dictada por la UBB que contempla la actividad de titulación en el octavo semestre.
ICINF	Ingeniería Civil en Informática. Carrera profesional dictada por la UBB que contempla la actividad de titulación en el décimo semestre.
PT	Proyecto de Titulación. Documento y desarrollo que los estudiantes deben realizar para finalizar su proceso de titulación.
EC	Encargado de Curso. Encargado/a de coordinar y gestionar los procesos académicos de la carrera correspondiente.
Informe Final	Documento que recopila el desarrollo completo del proyecto de tesis, incluyendo resultados, metodología y conclusiones.

Continúa en la siguiente página...

Término/Sigla	Definición
Profesor Guía	Docente responsable de asesorar al estudiante en el desarrollo de su Proyecto de Tesis.
Profesor Informante	Docente que evalúa el Informe Final de Tesis desde una perspectiva crítica y externa al proceso de guía.
Defensa	Presentación oral del proyecto de título ante una comisión académica, como parte del proceso de evaluación final.

Apéndice B

Pruebas de Aceptación

Apéndice C

Recopilación de Información

En este anexo se detallan las metodologías y fuentes utilizadas para la obtención de los datos necesarios para el desarrollo del proyecto AcTitUBB, describiendo las instancias de participación y análisis documental realizadas.

C.1. Técnicas Aplicadas

C.1.1. Resumen de Técnicas

Técnica	Participantes	Objetivo
Reuniones con Profesor Guía	Profesor guía	Orientación, validación de requisitos y retroalimentación
Observación Participante	Autor (tesista)	Identificar problemáticas reales del proceso
Revisión Documental	Autor	Conocer normativas y procedimientos oficiales

Tabla C.1: Resumen de técnicas de recopilación aplicadas

C.2. Reuniones con Profesor Guía

C.2.1. Descripción General

Se realizaron reuniones periódicas con el profesor guía del proyecto de título con el objetivo de validar requisitos, recibir retroalimentación sobre el avance del desarrollo, y ajustar funcionalidades según las necesidades identificadas. Estas reuniones fueron fundamentales para orientar el proyecto y asegurar su alineación con las necesidades reales de la institución.

C.2.2. Reunión 1: Presentación Inicial del Proyecto

Participantes: Autor del proyecto, Profesor guía

Modalidad: Presencial - Oficina del profesor

Objetivo: Presentar la propuesta inicial del proyecto, definir alcance y obtener retroalimentación.

Temas Discutidos

- (a) Presentación de la problemática observada en el proceso de titulación actual
- (b) Definición del alcance del proyecto y sus límites
- (c) Identificación de actores principales del sistema
- (d) Discusión de tecnologías y arquitectura a utilizar
- (e) Establecimiento de calendario de reuniones periódicas
- (f) Definición de entregables y fechas clave

Acuerdos y Hallazgos

Problemática validada:

El profesor guía confirmó que el proceso actual de titulación presenta múltiples ineficiencias que afectan a todos los actores involucrados:

- Gestión manual y descentralizada de propuestas mediante correo electrónico
- Falta de trazabilidad en el estado de los proyectos
- Dificultad para hacer seguimiento de fechas límite y entregas
- Comunicación dispersa entre estudiantes, profesores y administración
- Búsqueda manual de profesores disponibles para asignación
- Información duplicada en múltiples planillas Excel

Alcance definido:

- Sistema web para gestión integral del proceso de titulación
- Módulos principales: gestión de propuestas, asignación de profesores, seguimiento de proyectos, repositorio de documentos
- Roles del sistema: Estudiante, Profesor, Jefe de Carrera, Super Administrador
- Prioridad en funcionalidades core de gestión y seguimiento sobre funcionalidades avanzadas
- Fase 1: Funcionalidades esenciales / Fase 2: Funcionalidades complementarias

Stack tecnológico aprobado:

- **Frontend:** Angular (versión 18+) con TypeScript
- **Backend:** Node.js con Express.js
- **Base de datos:** MySQL 8.0
- **Contenedorización:** Docker y Docker Compose
- **Control de versiones:** Git

Calendario de reuniones acordado: Reuniones quincenales para revisión de avances, validación de funcionalidades implementadas, y resolución de dudas técnicas o de alcance.

Firma de acuerdos:

C.2.3. Reunión 2: Validación de Requisitos Funcionales

Participantes: Autor del proyecto, Profesor guía

Modalidad: Presencial - Oficina del profesor

Objetivo: Presentar y validar requisitos funcionales identificados tras el análisis inicial (Iteración 0).

Requisitos Presentados

Se presentó un documento con 38 requisitos funcionales identificados, organizados por módulo:

- (a) **Módulo de Gestión de Usuarios** (8 requisitos)
 - Registro y autenticación de usuarios
 - Gestión de perfiles por rol (Estudiante, Profesor, Admin, Super Admin)
 - Recuperación de contraseña
 - Gestión de datos personales
- (b) **Módulo de Propuestas** (10 requisitos)
 - Envío de propuestas mediante formulario estructurado
 - Revisión y aprobación de propuestas por jefes de carrera
 - Estados: Borrador, Enviada, En Revisión, Aprobada, Rechazada
 - Asignación automática o manual de revisor
 - Historial de cambios y versiones
- (c) **Módulo de Proyectos** (8 requisitos)
 - Seguimiento de estados de proyectos
 - Asignación de profesor guía, revisor e informante
 - Gestión de avances y entregas
 - Programación de defensa
- (d) **Módulo de Documentos** (4 requisitos)
 - Repositorio centralizado de documentos por proyecto
 - Control de versiones de documentos
 - Descarga y visualización de archivos
- (e) **Módulo de Notificaciones** (4 requisitos)

- Notificaciones por correo electrónico
- Alertas de fechas límite
- Notificaciones de cambios de estado
- (f) **Módulo de Reportes** (4 requisitos)
 - Dashboard con indicadores clave
 - Reportes de proyectos por estado
 - Estadísticas de carga de profesores
 - Exportación de datos

Retroalimentación Recibida

Requisitos validados: 35 de 38 requisitos fueron aprobados como necesarios y prioritarios.

Requisitos ajustados:

- **RF12 - Chat interno:** Posponer para versión 2.0, priorizar notificaciones por correo
- **RF28 - Integración con Sistema Académico UBB:** Considerar como opcional, no crítico para MVP
- **RF31 - Detección automática de plagio:** Fuera del alcance del proyecto

Requisitos nuevos sugeridos por el profesor guía:

- Permitir proyectos con múltiples estudiantes (trabajos en equipo)
- Incluir gestión completa de estructura académica (Facultades, Departamentos, Carreras)
- Implementar sistema de roles jerárquico con permisos granulares (Super Admin, Admin de Carrera)
- Agregar gestión de relaciones departamentos-carreras (un departamento puede dar servicio a múltiples carreras)

Priorización acordada mediante técnica MoSCoW:

Prioridad	Funcionalidades
Must Have (MVP)	Gestión de usuarios, propuestas, proyectos, asignación de profesores, notificaciones básicas, repositorio de documentos, dashboard
Should Have (V1.5)	Sistema de hitos, calendario de eventos, reportes avanzados, historial completo de auditoría
Could Have (V2.0)	Mensajería interna, aplicación móvil, integración con sistemas externos
Won't Have	Videoconferencia integrada, detección de plagio, IA para matching automático

Tabla C.2: Priorización de requisitos con MoSCoW

C.2.4. Reunión 3: Revisión de Diseño de Base de Datos

Participantes: Autor del proyecto, Profesor guía

Modalidad: Presencial - Oficina del profesor

Objetivo: Validar diseño de la base de datos y relaciones tras el cierre de la Iteración 0.

Modelo Presentado

Se presentó el diagrama entidad-relación con las siguientes 20 tablas principales:

- (a) **Usuarios y Roles:**
 - **usuarios:** Entidad central para todos los actores
 - **roles:** Estudiante, Profesor, Admin de Carrera, Super Admin
- (b) **Estructura Académica:**
 - **facultades:** Facultades de la universidad
 - **departamentos:** Departamentos por facultad
 - **carreras:** Carreras por facultad
 - **departamentos_carreras:** Relación N:M entre departamentos y carreras
- (c) **Relaciones Usuarios-Estructura:**
 - **estudiantes_carreras:** Estudiantes pueden estar en múltiples carreras
 - **profesores_departamentos:** Profesores pertenecen a uno o más departamentos
- (d) **Propuestas y Proyectos:**
 - **propuestas:** Propuestas de título con estados
 - **proyectos:** Proyectos aprobados en desarrollo
 - **estados_propuesta:** Catálogo de estados
 - **estados_proyecto:** Catálogo de estados
- (e) **Asignaciones:**
 - **asignaciones:** Profesores asignados a proyectos (guía, revisor, informante)
 - **tipos_asignacion:** Catálogo de tipos
- (f) **Documentos y Auditoría:**
 - **documentos:** Repositorio de archivos
 - **historial_cambios:** Auditoría completa del sistema

Decisiones de Diseño Validadas

Normalización:

- Base de datos normalizada en 3FN (Tercera Forma Normal)
- Uso de tablas intermedias para relaciones N:M
- Catálogos separados para estados y tipos

Integridad referencial:

- Todas las claves foráneas definidas con ON DELETE y ON UPDATE apropiados
- CASCADE para eliminación de datos dependientes
- RESTRICT para evitar eliminaciones accidentales
- SET NULL cuando la relación es opcional

Campos de auditoría en todas las tablas:

- **created_at** TIMESTAMP DEFAULT CURRENT_TIMESTAMP
- **updated_at** TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
- Campo activo **BOOLEAN** para eliminación lógica

Índices para optimización:

- Índices en campos de búsqueda frecuente (RUT, correo, estado)

- Índices compuestos en relaciones N:M
- Índices en claves foráneas para mejorar JOINS

Ajustes realizados durante la reunión:

- Agregar tabla `departamentos_carreras` para gestionar relación directa (sugerencia del profesor)
- Incluir campo `es_principal` en `estudiantes_carreras` y `profesores_departamentos`
- Validar que un proyecto tenga exactamente 1 guía, 1 revisor y 1 informante mediante constraint
- Agregar campo `carrera_administrada_id` implícito para jefes de carrera (calculado desde tabla `carreras`)

Firma de validación:**C.2.5. Reunión 4: Revisión de Prototipo de Interfaces (Hito Iteración 1)**

Participantes: Autor del proyecto, Profesor guía

Modalidad: Presencial - Laboratorio de computación

Objetivo: Validar diseño de interfaces de usuario y flujos de navegación al cierre del módulo de acceso.

Prototipos Presentados

Se demostraron wireframes funcionales en Angular de las siguientes interfaces:

- (a) **Login y autenticación:** Pantalla de ingreso, recuperación de contraseña
- (b) **Dashboard por rol:** Vistas personalizadas para cada tipo de usuario
- (c) **Gestión de propuestas:** Formulario de envío, lista de propuestas, vista de detalle
- (d) **Gestión de proyectos:** Lista de proyectos, seguimiento de estado, asignación de profesores
- (e) **Perfil de usuario:** Visualización y edición de datos personales
- (f) **Administración:** Gestión de usuarios, estructura académica, reportes

Retroalimentación Recibida**Aspectos positivos identificados:**

- Interfaz limpia, moderna y consistente con principios de Material Design
- Navegación intuitiva con breadcrumbs y menú lateral colapsable
- Uso apropiado de colores institucionales (azul UBB)
- Estados visuales claros mediante chips con colores distintivos
- Formularios bien estructurados con validaciones en tiempo real

Mejoras sugeridas:

- Agregar ayuda contextual en formularios complejos (tooltips)
- Incluir confirmaciones modales antes de acciones críticas (eliminar, rechazar)
- Mejorar contraste en algunos elementos para cumplir WCAG 2.1 AA
- Agregar indicador de progreso al guardar cambios
- Implementar "guardar como borrador.^{en} formulario de propuestas

- Incluir paginación en listados extensos

Decisiones de UX/UI aprobadas:

- Diseño responsive con breakpoints en 768px, 1024px y 1440px
- Paleta de colores: Azul primario (#004B93), Verde success (#28a745), Rojo error (#dc3545)
- Tipografía: Roboto para interfaz, Roboto Mono para código/RUT
- Iconografía: Material Icons
- Espaciado: Sistema de 8px grid

C.2.6. Reuniones Adicionales de Seguimiento

Además de las 4 reuniones formales documentadas anteriormente, se realizaron **12 reuniones quincenales** adicionales durante el período de desarrollo (Junio-Noviembre 2025) para:

- Revisión de avances del desarrollo backend y frontend
- Resolución de dudas técnicas y de implementación
- Validación de funcionalidades implementadas
- Ajustes de requisitos según hallazgos durante el desarrollo
- Revisión de problemas de despliegue y configuración de Docker
- Preparación de documentación del proyecto
- Planificación de pruebas y validación final

Estas reuniones periódicas permitieron un desarrollo iterativo e incremental, alineado constantemente con las expectativas del profesor guía y las necesidades reales del proceso de titulación.

C.3. Observación Participante

C.3.1. Descripción General

Como estudiante tesista en proceso de titulación durante el período Marzo-Noviembre 2025, se realizó una **observación participante** del proceso completo. Esta técnica cualitativa permitió identificar problemáticas reales, ineficiencias, puntos de dolor y necesidades específicas de primera mano, experimentando directamente las dificultades que enfrenta un estudiante en el proceso de titulación.

Definición de observación participante:

La observación participante es una técnica de investigación cualitativa en la cual el investigador no solo observa el fenómeno de estudio, sino que participa activamente en él. En este caso, el autor del proyecto es simultáneamente:

- **Observador:** Analiza el proceso de titulación con perspectiva crítica para identificar mejoras
- **Participante:** Vive el proceso como estudiante tesista, experimentando sus dificultades reales

Ventajas de esta técnica en el contexto del proyecto:

- Acceso privilegiado y continuo a todos los aspectos del proceso
- Comprensión profunda de los flujos de trabajo reales (no idealizados)
- Identificación de problemas que los actores del proceso consideran "normales" no mencionarían en entrevistas
- Empatía genuina con las necesidades, frustraciones y expectativas de los usuarios
- Validación continua de requisitos basada en experiencia propia y de pares
- Capacidad de observar comportamientos naturales sin el sesgo de una observación externa

Problemáticas Identificadas

Problema	Impacto
Falta de transparencia	El estudiante no sabe en qué etapa está su propuesta, genera ansiedad y múltiples consultas
Proceso muy lento	9 semanas es excesivo, retrasa inicio del proyecto, algunos estudiantes se desmotivan
Comunicación ineficiente	Todo por correo con tiempos de respuesta largos (días o semanas)
Sin documentación clara	No hay guía oficial sobre cómo preparar una propuesta adecuadamente
Formato no estandarizado	Cada estudiante envía en formato diferente (Word, PDF, estructura distinta)
Sin confirmaciones	No hay acuse de recibo automático, genera incertidumbre
Versiones confusas	Archivos con nombres como "final_v2_corregida_definitiva.docx"
Sin notificaciones proactivas	El estudiante debe estar preguntando constantemente por el estado

Tabla C.3: Problemas identificados en el proceso de propuestas

Requisitos Derivados de esta Observación

- **RF01:** Sistema debe mostrar el estado actual de la propuesta en tiempo real con línea de tiempo visual
- **RF02:** Envío de propuestas mediante formulario web estructurado con campos obligatorios
- **RF03:** Confirmación automática de recepción inmediata al enviar
- **RF04:** Notificaciones por correo electrónico en cada cambio de estado
- **RF05:** Acceso a guía oficial y formato de ejemplo desde el sistema
- **RF06:** Control de versiones automático (v1, v2, v3) sin nombres de archivo confusos
- **RF07:** Dashboard para estudiante con vista clara de su propuesta y próximas acciones
- **RF08:** Tiempo objetivo de revisión visible (SLA de 2 semanas máximo)

C.3.2. Observación 2: Interacción con Secretaría y Administración

Período: Marzo-Abril 2025 (6 semanas)

Actividades observadas: Consultas presenciales, observación de gestión de documentos, comunicación con Jefe de Carrera

Frecuencia: 8 visitas presenciales a secretaría

Experiencias Documentadas

Gestión de documentos físicos:

- Observé que la secretaria mantiene carpetas físicas por año (2023, 2024, 2025)
- Propuestas archivadas en papel, incluso si fueron enviadas por correo (imprimen el correo)
- Búsqueda manual de documentos: "¿En qué mes enviaste la propuesta?" → revisa carpetas físicas
- Riesgo de pérdida: encontré una propuesta mal archivada en carpeta de año anterior

Registro en planillas Excel:

- Observé que la secretaria transcribe información manualmente a Excel
- Planillas múltiples: "Propuestas 2025.xlsx", "Proyectos en desarrollo 2025.xlsx", "Carga docentes 2025.xlsx"
- Error presencial: RUT mal digitado en planilla, tuve que corregir
- Sin validaciones: puede ingresar datos inconsistentes o duplicados

Comunicación con Jefe de Carrera:

- Jefe de Carrera solicita información a secretaria → secretaria consolida datos de planillas → envía por correo
- Proceso toma horas o días según disponibilidad de la secretaria
- Información a veces desactualizada porque planillas no se actualizan en tiempo real

Consultas frecuentes de estudiantes:

- Observé que otros estudiantes también van constantemente a consultar estado
- Preguntas recurrentes: "¿Ya revisaron mi propuesta?", "¿Quién es mi profesor guía?", "¿Cuándo tengo que entregar...?"
- Secretaria interrumpida constantemente, dificulta su trabajo administrativo
- Mismas preguntas se responden una y otra vez (información no auto-servicio)

Problemáticas Identificadas

- **Doble trabajo:** Información en papel + digital + planillas
- **Propenso a errores:** Transcripción manual, archivos físicos mal ubicados
- **Sin centralización:** Información dispersa en carpetas, correos, planillas
- **Interrupciones constantes:** Estudiantes y profesores consultan presencialmente información que debería ser auto-servicio
- **Falta de trazabilidad:** No se sabe quién hizo qué cambio y cuándo
- **Información desactualizada:** Planillas no reflejan estado real en tiempo real

Requisitos Derivados

- **RF09:** Repositorio digital centralizado, sin necesidad de archivos físicos
- **RF10:** Actualización en tiempo real visible para todos los actores autorizados
- **RF11:** Información auto-servicio para estudiantes (estado, fechas, requisitos)
- **RF12:** Validaciones automáticas en formularios (RUT válido, correo UBB, etc.)
- **RF13:** Historial de auditoría completo (quién, qué, cuándo)
- **RF14:** Exportación de datos a Excel para compatibilidad con procesos actuales

C.3.3. Observación 3: Asignación de Profesor Guía

Período: Mayo 2025 (3 semanas)

Proceso observado: Espera y asignación de profesor guía después de aprobación de propuesta

Experiencia Personal

Semana 1 post-aprobación:

- Notificación de propuesta aprobada sin información de profesor guía asignado
- Expectativa: ¿Me asignarán profesor? ¿Cuándo? ¿Puedo yo proponer uno?
- Consulta por correo → Respuesta: ".Estamos buscando profesor disponible con expertise en el área"

Semana 2:

- Sin noticias, nueva consulta presencial
- Jefe de Carrera me explica: "He contactado a 3 profesores, aún no confirman disponibilidad"
- Me entero de que el proceso es manual: revisan manualmente quién tiene menos proyectos

Semana 3:

- Finalmente me asignan profesor guía (notificación por correo)
- **Tiempo total: 3 semanas adicionales después de aprobación de propuesta**
- Perdí casi un mes desde aprobación hasta poder iniciar trabajo formal

Observación de Otros Casos

Conversando con 5 compañeros tesistas, identifiqué patrones:

- **Estudiante A:** Le asignaron profesor guía en 1 semana (tuvo suerte, profesor disponible inmediatamente)
- **Estudiante B:** Esperó 4 semanas, le reasignaron profesor porque el primero declinó
- **Estudiante C:** Esperó 5 semanas, frustración alta, consideró cambiar de tema para conseguir profesor más rápido
- **Estudiante D:** Propuso su propio profesor guía desde el inicio → proceso más rápido (2 días)
- **Estudiante E:** Le asignaron profesor sin expertise real en el área → tuvo que solicitar cambio

Problemáticas Identificadas

- **Proceso muy lento:** 1-5 semanas para asignar profesor
- **Sin criterios claros:** Asignación basada en conocimiento tácito del Jefe de Carrera
- **Sin visibilidad de carga:** No se sabe cuántos proyectos tiene cada profesor
- **Sin matching por expertise:** A veces se asigna por disponibilidad más que por competencia
- **Estudiante pasivo:** No puede proponer profesor, debe esperar asignación
- **Sin alertas de sobrecarga:** Algunos profesores tienen 5-6 proyectos, otros solo 1

Requisitos Derivados

- **RF15:** Sistema debe mostrar carga actual de cada profesor (cuántos proyectos guía)
- **RF16:** Permitir al estudiante sugerir profesor guía desde la propuesta
- **RF17:** Sistema de matching automático o semi-automático considerando expertise y disponibilidad
- **RF18:** Notificación al profesor cuando es propuesto como guía (puede aceptar/rechazar)
- **RF19:** Alerta cuando un profesor supera límite recomendado de proyectos (ej: >4)
- **RF20:** Visualización de especialidades/áreas de expertise de cada profesor

C.3.4. Observación 4: Comunicación con Profesor Guía durante el Desarrollo

Período: Junio-Octubre 2025 (5 meses)

Proceso observado: Coordinación de reuniones, envío de avances, recepción de feedback

Experiencias Vividas

Coordinación de reuniones:

- Intercambio de 4-6 correos para agendar cada reunión
- "¿Tienes disponibilidad el martes?" → "No, mejor el jueves" → "¿A qué hora?" → Etc.
- Reuniones acordadas se postergan por olvidos o conflictos de agenda
- No hay recordatorios automáticos de reuniones pactadas

Envío de avances:

- Envío por correo con asunto ".Avance proyecto - [fecha]"
- Confusión: ¿El profesor recibió el correo? ¿Ya lo revisó?
- A veces el archivo adjunto es muy pesado y rebota el correo
- Versiones: ".avance_v1.pdf", ".avance_v2_corregido.pdf", ".avance_v3_final.pdf"

Recepción de feedback:

- Feedback por correo (comentarios en el cuerpo del correo o archivo con anotaciones)
- Difícil hacer seguimiento: ¿qué comentarios ya atendí? ¿cuáles están pendientes?
- No hay historial estructurado de observaciones y resoluciones

Seguimiento de hitos:

- Sin calendario compartido de entregas y fechas clave
- Dependencia de memoria personal: "Me parece que quedamos en entregar esto el viernes"
- A veces se pierden plazos por falta de recordatorios

Problemáticas Identificadas

- **Ineficiencia en coordinación:** Mucho tiempo perdido en agendar reuniones
- **Sin confirmación de lectura:** No se sabe si el profesor vio el avance
- **Control de versiones deficiente:** Confusión sobre qué versión es la vigente
- **Feedback no estructurado:** Comentarios dispersos en correos, difícil seguimiento
- **Sin calendario compartido:** Fechas importantes solo en memoria personal
- **Sin recordatorios:** Depende de la proactividad del estudiante

Requisitos Derivados

- **RF21:** Sistema de calendario compartido con disponibilidad de profesores
- **RF22:** Coordinación de reuniones con propuesta de horarios y confirmación
- **RF23:** Repositorio de documentos con control de versiones automático
- **RF24:** Notificación cuando el profesor sube comentarios o feedback
- **RF25:** Historial de observaciones con estado (pendiente/resuelto)
- **RF26:** Recordatorios automáticos de fechas límite y entregas programadas
- **RF27:** Vista de actividad reciente para ambos (estudiante y profesor)

C.3.5. Observación 5: Búsqueda de Información y Documentación

Período: Marzo-Noviembre 2025 (todo el proceso)

Actividades observadas: Búsqueda de formatos, reglamentos, ejemplos de proyectos anteriores

Dificultades Experimentadas

Búsqueda en el sitio web de la facultad:

- Enlaces rotos o desactualizados
- Información dispersa en múltiples secciones
- Formatos de documentos obsoletos (año 2018)
- Sin buscador eficiente

Consulta de ejemplos de proyectos anteriores:

- Biblioteca tiene copias físicas, pero no todas
- Sin repositorio digital accesible
- No se puede buscar por tema, solo por año y autor
- Muchos proyectos antiguos no están disponibles

Interpretación del reglamento de titulación:

- Documento PDF de 40 páginas, lenguaje formal y complejo
- Sin resumen ejecutivo o FAQ
- Dudas específicas requieren consulta con Jefe de Carrera

Requisitos Derivados

- **RF28:** Sección de ayuda con FAQ sobre el proceso de titulación
- **RF29:** Repositorio de formatos y plantillas oficiales descargables
- **RF30:** Acceso a proyectos anteriores (repositorio público)
- **RF31:** Guías paso a paso para cada etapa del proceso
- **RF32:** Glosario de términos y conceptos clave

C.3.6. Consolidación de Hallazgos de la Observación Participante

Problema Observado	Frecuencia	Impacto
Falta de transparencia en estado	Constante	Alto - genera ansiedad y consultas repetitivas
Procesos lentos	Constante	Alto - retrasa inicio de proyectos
Comunicación ineficiente	Constante	Medio - genera frustraciones
Sin documentación clara	Frecuente	Medio - dificulta preparación
Información desactualizada	Frecuente	Medio - decisiones en base a datos incorrectos
Coordinación manual	Constante	Bajo - consume tiempo valioso

Tabla C.4: Resumen de problemas observados

Impacto cuantificado:

- **Tiempo promedio desde propuesta hasta inicio de desarrollo:** 12 semanas (3 meses)
- **Número de consultas presenciales promedio por estudiante:** 8-12 durante el proceso
- **Correos intercambiados para coordinación:** 5-7 por reunión
- **Tiempo de respuesta promedio a consultas:** 3-7 días
- **Satisfacción percibida del proceso actual:** Baja (conversaciones con 15 compañeros tesisistas)

C.4. Revisión de Documentación

C.4.1. Descripción General

Se realizó una revisión exhaustiva de documentación institucional relevante al proceso de titulación para comprender normativas, procedimientos formales, requisitos legales y flujos oficiales, basándose en el Reglamento de Proyecto Final de Carrera (IECI 2025).

C.4.2. Documentos Revisados

Documento	Fuente
Reglamento de Proyecto Final de Carrera (2025)	Dirección Escuela IECI
Manual de Procedimientos de Titulación FCE	Secretaría Carrera
Formularios F1 y F2 de Propuesta	Anexos Reglamento
Calendario Académico 2025	Sitio web UBB
Guía para Estudiantes de Titulación	Biblioteca
Pautas de Evaluación (Rúbricas)	Anexo F Reglamento
Listado de Profesores por Departamento	Sitio web FCE
Estructura Académica de la Facultad	Sitio web UBB

Tabla C.5: Documentos institucionales revisados

C.4.3. Información Extraída y Requisitos Derivados

Del Reglamento de Proyecto Final de Carrera (IECI 2025)

Artículos relevantes identificados:

- **Art. 8:** “Para cada propuesta... la Dirección de Escuela IECI asignará: a) Profesores Guía... b) Profesor(a) Informante”
 - **Requisito derivado:** Sistema debe gestionar la asignación de al menos 2 roles obligatorios (Guía e Informante) y permitir un Co-guía opcional.
 - **Validación necesaria:** Un proyecto no puede pasar a defensa sin tener asignados y confirmados al Guía e Informante.
- **Art. 8 (Inciso a):** “Profesores Guía: que pertenece(n) al Departamento de Sistemas de Información”
 - **Requisito derivado:** Validación en el perfil de profesor para verificar su departamento de origen.
 - **Validación necesaria:** El sistema debe filtrar que el Guía principal pertenezca al DSI.
- **Art. 7:** “Las y los estudiantes tendrán un semestre de plazo para su realización... podrá extender el plazo a un segundo y último semestre”
 - **Requisito derivado:** Sistema debe calcular la fecha límite basada en el semestre actual.

- **Alerta necesaria:** Notificación crítica al finalizar el primer semestre para gestionar la extensión o reprobación automática.
- **Art. 9:** “Evaluar el cumplimiento de la planificación realizada. Dicha planificación debe contemplar la presentación de informes de avance”
 - **Requisito derivado:** Módulo para que el estudiante suba informes de avance y el guía los califique (ponderan 15 % según Art. 11).
 - **Seguimiento necesario:** Dashboard que muestre el estado de los informes de avance pactados.
- **Art. 14:** “La exposición oral... se realizará en el rango de fechas indicado por la Dirección de Escuela IECI”
 - **Requisito derivado:** Módulo de gestión de calendario de defensas administrado por la Dirección.
 - **Validación:** El estudiante debe tener el informe aprobado por el Guía e Informante (Art. 12 y 13) antes de ser programado en el rango de fechas.

Del Manual de Procedimientos y Anexos

Flujos documentados:

(a) Flujo de aprobación de propuestas (Art. 5 y 6):

- Estudiante ingresa propuesta (Formulario F1 o F2)
- Comisión docente evalúa (Aceptada, Condicional, Rechazada)
- Dirección informa resultado

Requisito: Máquina de estados para propuestas y digitalización del Formulario F1 (Anexo A).

(b) Flujo de Evaluación Final (Art. 11 y 12):

- Guía acepta informe final → Se envía a Informante
- Informante califica (15 días hábiles) → Aprobación/Rechazo
- Si ambos aprueban → Se habilita defensa oral

Requisito: Workflow secuencial de aprobación de documentos finales.

De los Formularios Oficiales (Anexo A - F1)

Campos del formulario de propuesta de título:

(a) Identificación:

- Datos de estudiante(s) y Profesor Guía propuesto.
- Modalidad: Desarrollo de Software o Investigación.

(b) Contenido Técnico:

- Título del proyecto.
- Descripción del problema y procesos actuales.
- Análisis de trabajos realizados (Estado del arte).
- Propuesta de solución y justificación.
- Objetivos Generales y Específicos.
- Metodología y Plan de trabajo (Carta Gantt).

Requisito derivado: El formulario web debe replicar la estructura del Anexo A (F1) y permitir la carga de imágenes para la Gantt o metodología.

C.4.4. Consolidación de Requisitos Normativos

Requisito Normativo	Documento Fuente
Asignación de Guía e Informante (Co-guía opcional)	Reglamento Art. 8
Plazo estándar de 1 semestre (extensible a 2 con autorización)	Reglamento Art. 7
Planificación de informes de avance (acordada con guía)	Reglamento Art. 9
Defensa en rango de fechas definido por la Escuela	Reglamento Art. 14
Uso de Formularios F1 (Proyecto) o F2 (Práctica)	Reglamento Art. 5
Flujo de aprobación secuencial (Guía → Informante)	Reglamento Art. 12 y 13
Nota de aprobación superior a 4.0	Reglamento Art. 15

Tabla C.6: Requisitos derivados de documentación oficial (Reglamento 2025)

De la Estructura Académica

Jerarquía identificada:

- Universidad del Bío-Bío
- → Facultades (ej: Facultad de Ciencias Empresariales)
- → Departamentos (ej: Depto. de sistema de información)
- → Profesores
- → Carreras (ej: Ing. Civil en Informática)
- → Estudiantes
- **Relación especial:** Departamentos dan servicio a múltiples carreras

Requisito derivado: Modelo de datos debe reflejar esta estructura jerárquica y la relación N:M entre departamentos y carreras.

C.4.5. Consolidación de Requisitos Normativos Adicionales

Requisito Normativo	Documento Fuente
Exactamente 3 profesores por proyecto (guía, revisor, informante)	Reglamento Art. 15
Plazo máximo de 2 semestres para desarrollo	Reglamento Art. 22
Entregas periódicas cada 6 semanas	Reglamento Art. 31
Defensa programada con 15 días de anticipación	Reglamento Art. 45
Formato oficial de propuesta con campos específicos	Formularios oficiales
Flujos de aprobación definidos	Manual de Procedimientos
Estructura académica jerárquica	Documentos administrativos

Tabla C.7: Requisitos derivados de documentación oficial

C.5. Consolidación de Resultados

C.5.1. Requisitos Funcionales Validados

A partir de las 3 técnicas aplicadas (reuniones, observación participante, revisión documental), se consolidaron **38 requisitos funcionales prioritarios** organizados por módulo:

ID	Requisito Funcional	Técnica de Respaldo
RF01	Registro y autenticación de usuarios	Reuniones, Observación
RF02	Gestión de perfiles por rol	Reuniones, Documentación
RF03	Recuperación de contraseña	Observación
RF04	Envío de propuestas con formulario estructurado	Observación, Documentación
RF05	Máquina de estados para propuestas	Reuniones, Documentación
RF06	Asignación de revisor a propuestas	Observación, Documentación
RF07	Visualización de estado en tiempo real	Observación (crítico)
RF08	Notificaciones por correo en cambios de estado	Observación, Reuniones
RF09	Gestión de proyectos con estados	Reuniones, Documentación
RF10	Asignación de 3 roles de profesores	Documentación (obligatorio)
RF11	Dashboard con indicadores clave	Reuniones
RF12	Repositorio centralizado de documentos	Observación, Reuniones
RF13	Control de versiones de documentos	Observación
RF14	Sistema de hitos y entregas programadas	Documentación, Observación
RF15	Visualización de carga de profesores	Observación
RF16	Gestión de estructura académica	Reuniones, Documentación
RF17	Relación departamentos-carreras (N:M)	Reuniones, Documentación
RF18	Reportes y estadísticas	Reuniones
RF19	Historial de auditoría completo	Reuniones, Observación
RF20	Exportación de datos a Excel	Observación

Tabla C.8: Muestra de requisitos funcionales y técnicas de respaldo

C.5.2. Requisitos No Funcionales Validados

ID	Requisito No Funcional	Técnica
RNF01	Accesible desde navegadores modernos (Chrome, Firefox, Edge)	Reuniones
RNF02	Tiempo de respuesta menor a 3 segundos	Reuniones
RNF03	Disponibilidad del 99 % (máximo 7.3 horas de downtime/mes)	Reuniones
RNF04	Interfaz intuitiva y fácil de usar (no requiere capacitación)	Observación, Reuniones
RNF05	Cumplimiento de normativa institucional UBB	Documentación (obligatorio)
RNF06	Capacidad para 100 usuarios concurrentes	Reuniones
RNF07	Responsive (móvil, tablet, desktop)	Reuniones
RNF08	Seguridad: encriptación de contraseñas, protección CSRF	Reuniones
RNF09	Backup automático diario de base de datos	Reuniones
RNF10	Escalabilidad horizontal mediante contenedores Docker	Reuniones

Tabla C.9: Requisitos no funcionales validados

C.5.3. Métricas del Proceso Actual vs. Sistema Propuesto

Métrica	Valor Actual	Meta Sistema
Tiempo de envío a aprobación de propuesta	9 semanas	≤ 2 semanas
Tiempo de asignación de profesor guía	1-5 semanas	≤ 3 días
Consultas presenciales por estudiante	8-12 veces	0-2 veces
Correos para agendar 1 reunión	5-7 correos	0 (automático)
Tiempo para generar reporte	2-4 horas	< 5 minutos
Visibilidad del estado	0 % (sin info)	100 % real-time
Satisfacción percibida del proceso	Baja	Alta ($>4/5$)
Errores de transcripción de datos	1 cada 10	0 (validado)

Tabla C.10: Métricas de mejora esperadas

C.6. Conclusiones

La combinación de tres técnicas complementarias de recopilación de información permitió obtener una comprensión profunda y holística del proceso de titulación:

- **Reuniones con profesor guía** proporcionaron orientación experta, validación de requisitos y retroalimentación continua durante el desarrollo.

- **Observación participante** como estudiante tesista fue la técnica más valiosa, permitiendo identificar problemáticas reales que los actores del proceso consideran "normales" que no surgirían en entrevistas formales. La experiencia vivida de 9 semanas de espera para aprobación de propuesta fue el detonante clave que motivó este proyecto.
- **Revisión de documentación** aseguró que el sistema cumple con todas las normativas y procedimientos oficiales de la Universidad del Bío-Bío.

Resultados cuantitativos:

- 38 requisitos funcionales identificados y validados
- 10 requisitos no funcionales definidos
- 8 documentos institucionales revisados
- 16 reuniones formales con profesor guía
- 9 meses de observación participante continua
- Reducción esperada de 78 % en tiempo de procesamiento de propuestas

Validación del proyecto:

Toda la información recopilada confirma la necesidad y viabilidad del Sistema de Gestión de Proyectos de Título (AcTitUBB). Los hallazgos demuestran que el proceso actual es ineficiente, lento, propenso a errores y frustrante para todos los actores involucrados. El sistema propuesto aborda sistemáticamente cada una de las problemáticas identificadas, con el potencial de reducir significativamente los tiempos, mejorar la transparencia, y aumentar la satisfacción de estudiantes, profesores y administradores.

Apéndice D

Diccionario de datos

D.1. Tablas Principales

D.1.1. usuarios

Almacena información de todos los usuarios del sistema (estudiantes, profesores, administradores).

Atributo	Tipo	Nulo	Descripción
rut	VARCHAR(12)	NO	RUT del usuario (PK). Formato: 12345678-9
nombre	VARCHAR(255)	NO	Nombre completo del usuario
email	VARCHAR(255)	NO	Correo electrónico (único)
password	VARCHAR(255)	NO	Contraseña encriptada (bcrypt)
telefono	VARCHAR(20)	SÍ	Teléfono de contacto
rol_id	INT	NO	FK a roles. Define permisos del usuario
confirmado	BOOLEAN	NO	Indica si el usuario confirmó su cuenta
token	VARCHAR(255)	SÍ	Token para recuperación de contraseña
activo	BOOLEAN	NO	Indica si el usuario está activo (eliminación lógica)
created_at	TIMESTAMP	NO	Fecha de creación del registro
updated_at	TIMESTAMP	NO	Fecha de última actualización

Restricciones:

- PK: rut
- UNIQUE: email
- FK: rol_id → roles(id)

- CHECK: email debe terminar en '@ubiobio.cló '@alumnos.ubiobio.cl'

D.1.2. roles

Catálogo de roles del sistema.

ID	Nombre	Nulo	Descripción
id	INT	NO	Identificador único (PK, AUTO_INCREMENT)
nombre	VARCHAR(50)	NO	Nombre del rol (único)
descrip	TEXT	SÍ	Descripción del rol

Valores predefinidos:

- 1 - estudiante: Estudiante en proceso de titulación
- 2 - profesor: Profesor (guía, revisor o informante)
- 3 - admin: Administrador de carrera (Jefe de Carrera)
- 4 - Super Administrador: Administrador del sistema

D.1.3. facultades

Facultades de la universidad.

Atributo	Tipo	Nulo	Descripción
id	INT	NO	Identificador único (PK, AUTO_INCREMENT)
nombre	VARCHAR(255)	NO	Nombre de la facultad
codigo	VARCHAR(20)	NO	Código único de la facultad
activo	BOOLEAN	NO	Indica si está activa (default: TRUE)
created_at	TIMESTAMP	NO	Fecha de creación
updated_at	TIMESTAMP	NO	Fecha de actualización

D.1.4. departamentos

Departamentos académicos por facultad.

Atributo	Tipo	Nulo	Descripción
id	INT	NO	Identificador único (PK, AUTO_INCREMENT)
nombre	VARCHAR(255)	NO	Nombre del departamento
codigo	VARCHAR(20)	NO	Código único del departamento
facultad_id	INT	NO	FK a facultades
activo	BOOLEAN	NO	Indica si está activo (default: TRUE)
created_at	TIMESTAMP	NO	Fecha de creación
updated_at	TIMESTAMP	NO	Fecha de actualización

Restricciones:

- FK: facultad_id → facultades(id) ON DELETE RESTRICT

D.1.5. carreras

Carreras académicas por facultad.

Atributo	Tipo	Nulo	Descripción
id	INT	NO	Identificador único (PK, AUTO_INCREMENT)
nombre	VARCHAR(255)	NO	Nombre de la carrera
codigo	VARCHAR(20)	NO	Código único de la carrera
facultad_id	INT	NO	FK a facultades
jefe_carrera_rut	VARCHAR(12)	SÍ	RUT del jefe de carrera (FK a usuarios)
activo	BOOLEAN	NO	Indica si está activa (default: TRUE)
created_at	TIMESTAMP	NO	Fecha de creación
updated_at	TIMESTAMP	NO	Fecha de actualización

Restricciones:

- FK: facultad_id → facultades(id) ON DELETE RESTRICT
- FK: jefe_carrera_rut → usuarios(rut) ON DELETE SET NULL

D.1.6. propuestas

Propuestas de proyectos de título enviadas por estudiantes.

Atributo	Tipo	Nulo	Descripción
id	INT	NO	Identificador único (PK, AUTO_INCREMENT)
titulo	VARCHAR(255)	NO	Título del proyecto propuesto
descripcion	TEXT	NO	Descripción detallada de la propuesta
objetivos	TEXT	NO	Objetivos generales y específicos
metodologia	TEXT	SÍ	Metodología propuesta
estudiante_rut	VARCHAR(12)	NO	RUT del estudiante (FK a usuarios)
carrera_id	INT	NO	FK a carreras
estado	VARCHAR(50)	NO	Estado actual (Borrador, Enviada, En Revisión, Aprobada, Rechazada)
profesor_revisor_rut	VARCHAR(12)	SÍ	RUT del profesor revisor asignado
fecha_envio	TIMESTAMP	SÍ	Fecha de envío de la propuesta
fecha_revision	TIMESTAMP	SÍ	Fecha de revisión
observaciones	TEXT	SÍ	Observaciones del revisor
created_at	TIMESTAMP	NO	Fecha de creación
updated_at	TIMESTAMP	NO	Fecha de actualización

Restricciones:

- FK: estudiante_rut → usuarios(rut) ON DELETE CASCADE
- FK: carrera_id → carreras(id) ON DELETE RESTRICT
- FK: profesor_revisor_rut → usuarios(rut) ON DELETE SET NULL
- CHECK: estado IN ('Borrador', 'Enviada', 'En Revisión', 'Aprobada', 'Rechazada')

D.1.7. proyectos

Proyectos de título aprobados en desarrollo.

Atributo	Tipo	Nulo	Descripción
id	INT	NO	Identificador único (PK, AUTO_INCREMENT)
propuesta_id	INT	NO	FK a propuestas (origen del proyecto)
titulo	VARCHAR(255)	NO	Título del proyecto
descripcion	TEXT	NO	Descripción del proyecto

Atributo	Tipo	Nulo	Descripción
estado	VARCHAR(50)	NO	Estado (Iniciado, En Desarrollo, Finalizado, Defendido)
fecha_inicio	DATE	NO	Fecha de inicio del proyecto
fecha_limite	DATE	NO	Fecha límite (2 semestres desde inicio)
fecha_finalizacion	DATE	SÍ	Fecha real de finalización
calificacion_final	DECIMAL(3,1)	SÍ	Calificación final (1.0 - 7.0)
created_at	TIMESTAMP	NO	Fecha de creación
updated_at	TIMESTAMP	NO	Fecha de actualización

Restricciones:

- FK: propuesta_id → propuestas(id) ON DELETE RESTRICT
- CHECK: estado IN (‘Iniciado’, ‘En Desarrollo’, ‘Finalizado’, ‘Defendido’)
- CHECK: calificacion_final BETWEEN 1.0 AND 7.0

D.1.8. asignaciones

Asignación de profesores a proyectos (guía, revisor, informante).

Atributo	Tipo	Nulo	Descripción
id	INT	NO	Identificador único (PK, AUTO_INCREMENT)
proyecto_id	INT	NO	FK a proyectos
profesor_rut	VARCHAR(12)	NO	RUT del profesor (FK a usuarios)
tipo_asignacion	VARCHAR(20)	NO	Tipo: ‘guía’, ‘revisor’, ‘informante’
fecha_asignacion	TIMESTAMP	NO	Fecha de asignación
activo	BOOLEAN	NO	Indica si la asignación está activa
created_at	TIMESTAMP	NO	Fecha de creación
updated_at	TIMESTAMP	NO	Fecha de actualización

Restricciones:

- FK: proyecto_id → proyectos(id) ON DELETE CASCADE
- FK: profesor_rut → usuarios(rut) ON DELETE CASCADE
- CHECK: tipo_asignacion IN (‘guía’, ‘revisor’, ‘informante’)
- UNIQUE: (proyecto_id, tipo_asignacion) - Un proyecto solo puede tener un profesor de cada tipo

D.1.9. estudiantes_carreras

Relación muchos a muchos entre estudiantes y carreras.

Atributo	Tipo	Nulo	Descripción
id	INT	NO	Identificador único (PK, AUTO_INCREMENT)
estudiante_rut	VARCHAR(11)	NO	RUT del estudiante (FK a usuarios)
carrera_id	INT	NO	FK a carreras
es_carrera_principal	BOOLEAN	NO	¿Es la carrera principal? (default: TRUE)
fecha_ingreso	DATE	SÍ	Fecha de ingreso a la carrera
activo	BOOLEAN	NO	Indica si está activo en la carrera
created_at	TIMESTAMP	NO	Fecha de creación
updated_at	TIMESTAMP	NO	Fecha de actualización

Restricciones:

- FK: estudiante_rut → usuarios(rut) ON DELETE CASCADE
- FK: carrera_id → carreras(id) ON DELETE CASCADE
- UNIQUE: (estudiante_rut, carrera_id)

D.1.10. profesores_departamentos

Relación muchos a muchos entre profesores y departamentos.

Atributo	Tipo	Nulo	Descripción
id	INT	NO	Identificador único (PK, AUTO_INCREMENT)
profesor_rut	VARCHAR(11)	NO	RUT del profesor (FK a usuarios)
departamento_id	INT	NO	FK a departamentos
es_principal	BOOLEAN	NO	¿Es el depto. principal? (default: TRUE)
activo	BOOLEAN	NO	Indica si está activo en el departamento
created_at	TIMESTAMP	NO	Fecha de creación
updated_at	TIMESTAMP	NO	Fecha de actualización

Restricciones:

- FK: profesor_rut → usuarios(rut) ON DELETE CASCADE

- FK: departamento_id → departamentos(id) ON DELETE CASCADE
- UNIQUE: (profesor_rut, departamento_id)

D.1.11. departamentos_carreras

Relación muchos a muchos entre departamentos y carreras.

Atributo	Tipo	Nulo	Descripción
id	INT	NO	Identificador único (PK, AUTO_INCREMENT)
departamento_id	INT	NO	FK a departamentos
carrera_id	INT	NO	FK a carreras
es_principal	BOOLEAN	NO	¿Es el depto. principal de la carrera?
activo	BOOLEAN	NO	Indica si la relación está activa
created_at	TIMESTAMP	NO	Fecha de creación
updated_at	TIMESTAMP	NO	Fecha de actualización

Restricciones:

- FK: departamento_id → departamentos(id) ON DELETE CASCADE
- FK: carrera_id → carreras(id) ON DELETE CASCADE
- UNIQUE: (departamento_id, carrera_id)

D.1.12. documentos

Repositorio de documentos asociados a proyectos.

Atributo	Tipo	Nulo	Descripción
id	INT	NO	Identificador único (PK, AUTO_INCREMENT)
proyecto_id	INT	NO	FK a proyectos
nombre_archivo	VARCHAR(255)	NO	Nombre del archivo
ruta_archivo	VARCHAR(500)	NO	Ruta en el servidor
tipo_documento	VARCHAR(50)	NO	Tipo: 'avance', 'informe_final', etc.
version	INT	NO	Número de versión del documento
subido_por	VARCHAR(12)	NO	RUT del usuario que subió el documento
fecha_subida	TIMESTAMP	NO	Fecha de carga del documento
created_at	TIMESTAMP	NO	Fecha de creación

Restricciones:

- FK: proyecto_id → proyectos(id) ON DELETE CASCADE
- FK: subido_por → usuarios(rut) ON DELETE SET NULL

D.2. Diagrama de Relaciones

Relaciones principales:

- **usuarios** 1:N **propuestas** (un usuario puede tener múltiples propuestas)
- **propuestas** 1:1 **proyectos** (una propuesta aprobada genera un proyecto)
- **proyectos** N:M **usuarios** (a través de **asignaciones**)
- **estudiantes** N:M **carreras** (a través de **estudiantes_carreras**)
- **profesores** N:M **departamentos** (a través de **profesores_departamentos**)
- **departamentos** N:M **carreras** (a través de **departamentos_carreras**)
- **proyectos** 1:N **documentos** (un proyecto tiene múltiples documentos)

D.3. Convenciones y Estándares

D.3.1. Nomenclatura

- Nombres de tablas: plural, minúsculas, snake_case
- Nombres de columnas: minúsculas, snake_case
- Claves primarias: id (excepto usuarios que usa rut)
- Claves foráneas: [tabla]_id o [tabla]_rut

D.3.2. Campos de Auditoría

Todas las tablas incluyen:

- created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
- updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP

D.3.3. Eliminación Lógica

Las tablas principales usan campo activo BOOLEAN para eliminación lógica en lugar de eliminación física.

D.3.4. Validaciones

- RUT: Formato 12345678-9, validación de dígito verificador
- Email: Debe ser dominio @ubiobio.cl o @alumnos.ubiobio.cl
- Contraseñas: Encriptadas con bcrypt (mínimo 8 caracteres)
- Calificaciones: Rango 1.0 - 7.0 (sistema chileno)
- Estados: Valores predefinidos mediante CHECK constraints

Apéndice E

Aspectos de gestión de proyectos

E.1. Carta Gantt con línea base y desviaciones

Iteración / Incremento	Planificado (Sprint)	Ejecución Real	Retrospectiva y Ajuste del Alcance
Iteración 0: Inception y Cimientos			
Configuración y Análisis de Requisitos	2025-04-03 - 2025-04-19	2025-04-03 - 2025-04-19	Inicio del proyecto. El análisis concluyó a tiempo, pero se identificaron dudas en trazabilidad que requieren buffer para la siguiente etapa.
Diseño y Modelo de Datos Core	2025-04-20 - 2025-04-30	2025-04-20 - 2025-04-25	<i>Incremento:</i> Base de datos inicial. Se cerró el diseño antes de lo previsto para adelantar desarrollo.
Iteración 1: Fundamentos de Acceso			
Módulo de Autenticación y Registro	2025-04-27 - 2025-05-20	2025-04-27 - 2025-05-27	<i>Incremento:</i> Sistema de usuarios funcional. Desviación significativa (+1 semana) por complejidad no prevista en validaciones institucionales y RUT. Se redujo alcance de la siguiente iteración.
Iteración 2: Gestión del Núcleo del Negocio			

Iteración / Incremento	Planificado (Sprint)	Ejecución Real	Retrospectiva y Ajuste del Alcance
CRUD de Propuestas y Archivos	2025-06-01 - 2025-07-15	2025-06-07 - 2025-07-22	<i>Incremento:</i> Gestión documental integrada. Inicio tardío por arrastre de It.1. La integración de subida de archivos fue más compleja, desplazando la finalización.
Iteración 3: Flujos de Trabajo y Notificaciones			
Motor de Asignaciones y Alertas	2025-08-01 - 2025-09-10	2025-08-05 - 2025-09-17	<i>Incremento:</i> Flujo de revisión activo. Problemas críticos con SMTP en ambiente de desarrollo obligaron a usar soluciones paliativas, impactando tiempos de pruebas.
Iteración 4: Administración, Estabilización y Cierre			
Panel Administrativo y Automatización	2025-09-03 - 2025-11-14	2025-09-03 - 2025-11-18	<i>Incremento:</i> Funcionalidades SuperAdmin. Se añadió alcance no previsto (HU extra) durante el sprint, extendiendo la fecha de término.
Documentación Final y Despliegue	2025-11-19 - 2025-11-26	2025-11-19 - 2025-11-26	Cierre del proyecto. Ajustes finales de documentación y arreglos finales del sistema.

Tabla E.1: Cronograma de Ejecución Iterativa (Marzo - Noviembre)

Observaciones:

- Las desviaciones detectadas son mayoritariamente de alcance (nuevos requisitos) o integración (SMTP, triggers). Se recomienda mantener un buffer del 10-15 % para la siguiente iteración.
- Los cambios que afectan a la integridad de datos (triggers y rollback) deben preceder a despliegues a staging.

E.2. Riesgos de Alto Nivel (Amenazas), Impacto y Estrategia

Riesgo	Impacto	Probabilidad	Estrategia / Acciones propuestas	Presentado?
Fallo del servicio SMTP (envío de notificaciones)	Alto	Media	Usar servidor SMTP de pruebas (MailHog) en dev/staging; implementar <code>email_queue</code> con reintentos y monitoreo; alertas en production.	Sí (dev)
Problemas de integridad al crear proyecto automáticamente	Medio-Alto	Baja-Media	Revisar constraints (unique <code>proyecto_id</code>), añadir manejo de errores y logging; testear trigger en datos reales.	Sí (trigger implementado)
Carga y reintentos de jobs (colas) sin control	Medio	Media	Implementar <code>job_queue</code> con prioridad/reintentos y vistas de monitoreo; políticas de backoff.	Identificado
Dependencia en RUT y validaciones específicas	Medio	Media	Documentar formato RUT, mensajes claros en UI, utilitarios de validación en backend; pruebas de aceptación.	Sí
Falta de cobertura de pruebas de aceptación en staging	Alto	Media	Preparar dataset de prueba y scripts automáticos; ejecutar regresiones tras cambios.	Identificado

Tabla E.2: Riesgos, impacto, estrategia y estado

E.3. Estimación CU (Use Case Points - UCP)

E.3.1. Resumen del método

Se utiliza el método UCP (Unadjusted Use Case Points) con los pasos estándar:

- Clasificar actores y asignar peso (Simple=1, Medio=2, Complejo=3)
- Clasificar casos de uso y asignar peso (Simple=5, Medio=10, Complejo=15)
- Calcular $UUCP = UAW + UUCW$
- Calcular TCF y EF (Factores Técnicos y Ambientales)
- $UCP = UUCP * TCF * EF$

E.3.2. Clasificación de actores

Actor	Tipo	Peso
Estudiante (GUI web)	Complejo	3
Profesor (GUI web)	Complejo	3
Administrador (GUI web)	Complejo	3
SuperAdmin (GUI web)	Complejo	3

$$\text{UAW} = 3 + 3 + 3 + 3 = 12$$

E.3.3. Clasificación de casos de uso (Estimación)

Caso de uso	Tipo	Peso
Registrar usuario	Simple	5
Login / Autenticación	Simple	5
Crear propuesta (subida archivos)	Medio	10
Asignar profesores	Medio	10
Revisar propuesta (aprobar/rechazar)	Complejo	15
Subir avances	Medio	10
Descargar archivos protegidos	Simple	5
Gestor de usuarios (CRUD admin)	Medio	10
Configuración estructura (SuperAdmin)	Complejo	15

Nota: Se listan los CU principales para el cálculo base.

$$\text{UUCW} = (3 \text{ simples} * 5) + (4 \text{ medios} * 10) + (2 \text{ complejos} * 15)$$

$$\text{UUCW} = 15 + 40 + 30 = 85$$

$$\text{UUCP} = \text{UAW} + \text{UUCW} = 12 + 85 = 97$$

E.3.4. Cálculo TCF

Factor Técnico	Multiplicador	Relevancia (0..5)	Resultado
Distributed System	2	2	4
Performance objectives	1	3	3
End-user efficiency	1	3	3
Complex internal processing	1	4	4
Reusability	1	2	2
Installation ease	0.5	1	0.5
Operational ease	0.5	3	1.5
Portability	2	2	4
Changeability	1	2	2
Concurrency	1	3	3
Security features	1	2	2
Direct access for third parties	1	1	1
Special user training	1	1	1
TFactor =			31.0

$$\text{TCF} = 0.6 + (0.01 * 31.0) = 0.91$$

E.3.5. Cálculo EF

Factor Ambiental	Multiplicador	Relevancia (0..5)
Familiar with Objectory + RUP	1.5	3
Application experience	0.5	3
Object Oriented experience	1	3
Analyst capability	0.5	3
Motivation	1	4
Stable requirements	2	2
Part time workers	-1	1
Difficult programming language	-1	1
EFactor =		16.5

$$\text{EF} = 1.4 + (-0.03 * 16.5) = 1.4 - 0.495 = 0.905$$

E.3.6. UCP y Esfuerzo estimado final

$$UCP = UUCP * TCF * EF = 97 * 0.91 * 0.905 \approx 79.88$$

UCP Final ≈ 80 puntos. Para convertir UCP a horas se utiliza un factor de conversión estándar de 20 horas/UCP(

Horas estimadas (Cálculo) = 80 UCP * 20 horas = 1600 horas.

E.4. Resumen de Esfuerzo Real

Actividad / Fase / Caso de Uso	Número de horas
Análisis de requisitos y documentación	120
Diseño del modelo de datos y ER	80
Implementación: Autenticación y Registro	160
Implementación: Gestión de Propuestas	240
Implementación: Asignaciones y Revisión	180
Implementación: Notificaciones	120
Implementación: Creación automática de Proyecto	80
Implementación: Gestión Admin y SuperAdmin	140
Pruebas unitarias e integración	200
Documentación	120
Control de versiones / tareas git	60
Reuniones y coordinación	120
TOTAL EJECUTADO	1620

Tabla E.3: Desglose de horas reales invertidas

E.5. Retrospectiva Proyecto

- **Autenticación / Registro:** 100
- **Gestión de Propuestas:** 90
- **Asignaciones y Revisión:** 85
- **Notificaciones / Email:** 75
- **Creación automática de Proyectos (trigger):** 80
- **Gestión Administrativa / SuperAdmin:** 70
- **Documentación:** 95

E.6. Análisis Éxito / Fracaso del Proyecto

- Éxitos principales:
 - Arquitectura modular (frontend/backend/docs) establecida; endpoints clave existen y mapearon a HUs.
 - Mecanismos críticos de seguridad implementados (hashing, validación de email, triggers de integridad para estados finales).
 - Documentación técnica avanzada (CASOS de uso, estimaciones UCP, planes de prueba) creada y organizada en docs.
- Fracazos / áreas con resultados parciales:
 - Cobertura de pruebas de aceptación en staging incompleta — requiere dataset y ejecución automatizada.
 - Implementación completa de soporte operativo (colas, migraciones con rollback y auditoría transversal) no finalizada.
 - Algunas funcionalidades de SuperAdmin (ejecución segura de migraciones, panel avanzado) implementadas solo en diseño/documentación.
- Evaluación general: proyecto con base sólida y varios hitos funcionales completados; todavía requiere inversión para endurecer operaciones y pruebas antes de producción.

E.7. Riesgos que se Concretaron y Efectos / Consecuencias

- **Fallo SMTP en dev:** Se detectó durante integración — consecuencia: retrasos en pruebas de notificaciones y necesidad de configurar MailHog; mitigación: se usó servidor de pruebas y se dejó registrado como riesgo a corregir en staging/prod.
- **Integridad en creación automática de proyecto:** Trigger creado, pero generó casos de test que fallaron por constraints duplicados en pruebas iniciales — consecuencia: se postergó despliegue automático hasta incluir validaciones adicionales y manejo de errores (rollback).
- **Alcance adicional (SuperAdmin):** Inclusión de HUs/CUs y nuevas tablas de soporte aumentó el alcance y generó carga de trabajo adicional en documentación y diseño; consecuencia: desviaciones pequeñas en fechas y mayor prioridad en auditoría y control de accesos.

E.8. Análisis de Ajuste: Planificación vs Esfuerzo Real vs Estimación UCP

- **Planificación (Gantt):** Las desviaciones registradas fueron en su mayoría de 1–3 días y relacionadas con integración y ajustes de alcance. La tabla Gantt documenta dichas desviaciones por actividad.
- **Esfuerzo real (Resumen de horas):** En el ejemplo del documento se registraron 1.620 horas totales. Esta cifra refleja horas invertidas en desarrollo, pruebas y documentación

hasta la fecha.

- **Estimación UCP:** Estimación de ejemplo produjo UCP 101 puntos \rightarrow 2.020 horas (con factor 20 h/UCP). Comparación:
 - Estimación UCP (2.020 h) vs Esfuerzo real (1.620 h): esfuerzo real es 20
 - Subestimación del factor de conversión (20 h/UCP puede ser conservador); equipo fue más eficiente.
 - No se incluyeron en esfuerzo real tareas planificadas pendientes (migraciones, workers, cobertura completa de pruebas), lo que reduce la cifra real hasta ahora.
 - Planificación (fechas) vs Esfuerzo real: desviaciones de calendario pequeñas, pero impactaron fechas dependientes; la suma de pequeñas desviaciones acumuladas puede erosionar buffers.

E.9. Comparación y Análisis de Resultados

- Las métricas muestran coherencia estructural: módulos principales implementados y documentados, UCP confirma esfuerzo significativo necesario.
- La diferencia entre UCP estimado y horas reales sugiere revisar:
 - la tasa de conversión horas/UCP (ajustar a la productividad real del equipo),
 - inclusión de tareas no funcionales y operativas en el cálculo (cola, migraciones, monitoreo) para acercar estimación a la realidad.
- Recomendación: recalibrar el factor horas/UCP con datos históricos del equipo (p. ej. horas reales / UCP calculado) y re-ejecutar la estimación para las próximas iteraciones.

E.10. Conclusiones

- El proyecto presenta una base técnica sólida con funcionalidades críticas implementadas y documentación de soporte. El principal trabajo restante es operacionalizar (colas, migraciones seguras y auditoría) y completar pruebas de aceptación en staging.
- El cálculo UCP proporciona una guía útil; sin embargo, la práctica muestra variaciones que deben ajustarse por experiencia del equipo. Actualmente la estimación UCP es conservadora frente al esfuerzo reportado hasta la fecha.

E.11. Iteraciones en el desarrollo

Iteración 0: Inception y Cimientos

- **Funcionalidad:** Configuración del entorno, análisis de requisitos y diseño del esquema de base de datos (Core).
- **Fecha de cierre:** 25 de Abril de 2025.
- **Retroalimentación del cliente/usuario:** Se validó la arquitectura inicial. El usuario solicitó aclaraciones específicas sobre la trazabilidad de los actores, lo que requirió ajustes menores en el modelo antes de iniciar la programación masiva.

Iteración 1: Fundamentos de Acceso

- **Funcionalidad:** Módulo funcional de Autenticación, Registro de usuarios y validaciones de seguridad.
- **Fecha de cierre:** 27 de Mayo de 2025.
- **Retroalimentación del cliente/usuario:** El usuario enfatizó la necesidad crítica de restringir el registro exclusivamente a correos institucionales y aplicar validación estricta de formato RUT chileno, lo que generó la refactorización de las reglas de validación.

Iteración 2: Gestión del Núcleo del Negocio

- **Funcionalidad:** CRUD de Propuestas y sistema de carga/subida de archivos (documentación).
- **Fecha de cierre:** 22 de Julio de 2025.
- **Retroalimentación del cliente/usuario:** Se aprobó el flujo de creación de propuestas. El cliente validó la integridad de los archivos subidos tras resolver la complejidad técnica inicial en la gestión de adjuntos.

Iteración 3: Flujos de Trabajo y Notificaciones

- **Funcionalidad:** Motor de asignación de profesores, flujo de revisión y sistema de alertas/notificaciones.
- **Fecha de cierre:** 17 de Septiembre de 2025.
- **Retroalimentación del cliente/usuario:** Se validó la lógica de negocio para las asignaciones y revisiones. El cliente aceptó las pruebas de notificación realizadas en entorno simulado (MailHog) debido a las restricciones temporales del servidor SMTP en desarrollo.

Iteración 4: Administración, Estabilización y Cierre

- **Funcionalidad:** Panel de Administración (SuperAdmin), automatización de creación de proyectos y documentación final.
- **Fecha de cierre:** 26 de Noviembre de 2025.
- **Retroalimentación del cliente/usuario:** Durante la revisión del panel administrativo, el cliente solicitó una funcionalidad extra (HU adicional para gestión avanzada), la cual fue implementada y aprobada. Se dio la conformidad final para el despliegue y la documentación.