

**“AcTitUBB: Solución software para la coordinación de las actividades de titulación para las carreras de informática de la Facultad de Cs. Empresariales”**

Proyecto de título para optar al título de Ingeniero de Ejecución en Computación e Informática

**Alumno(s)**

Daniel Abraham Aguayo Muñoz

**Profesora guía**

20 de julio de 2025

Concepción – Chile

Dedicatoria y/o Agradecimientos

Dedicar o agradecer….

Tabla de contenido

[Capítulo I Introducción 7](#_Toc179816661)

[I.1 Definiciones, siglas y abreviaciones del negocio 7](#_Toc179816662)

[I.2 Presentación del contexto o área de su problemática. 8](#_Toc179816663)

[I.2.1 Presentación Institución/Empresa 8](#_Toc179816664)

[I.2.2 Descripción de problemas u Oportunidades de Mejora 8](#_Toc179816665)

[I.3 Propuesta de solución 11](#_Toc179816666)

[I.4 Análisis de los principales Trabajos realizados o soluciones disponibles en el área o tema de la propuesta 13](#_Toc179816667)

[I.5 Justificación del proyecto 14](#_Toc179816668)

[I.6 Composición del Informe 17](#_Toc179816669)

[Capítulo II Proyecto 19](#_Toc179816670)

[II.1 Objetivo general del proyecto 19](#_Toc179816671)

[II.2 Objetivos específicos del proyecto 19](#_Toc179816672)

[II.3 Metodología de desarrollo 19](#_Toc179816673)

[II.4 Descripción de las actividades para lograr los objetivos Específicos 20](#_Toc179816674)

[II.5 Estándares de documentación 22](#_Toc179816675)

[II.6 Técnicas y notaciones 22](#_Toc179816676)

[II.7 Herramientas, framework, lenguaje usados en el desarrollo del proyecto 23](#_Toc179816677)

[Capítulo III Digitalización en la I M SPedro 24](#_Toc179816678)

[III.1 Proceso de sergio 24](#_Toc179816679)

[III.2 Proceso de DTransito 24](#_Toc179816680)

[III.3 Discusión 24](#_Toc179816681)

[Capítulo IV Factibilidad 25](#_Toc179816682)

[IV.1 Factibilidad técnica 25](#_Toc179816683)

[IV.2 Factibilidad operativa 26](#_Toc179816684)

[IV.3 Factibilidad económica 26](#_Toc179816685)

[IV.3.1 Flujo de caja 27](#_Toc179816686)

[Capítulo V Definición de la solución Software 31](#_Toc179816687)

[V.1 Objetivo General del Software propuesto. 31](#_Toc179816688)

[V.2 Objetivos Específicos del Software propuesto. 31](#_Toc179816689)

[V.3 Límites 31](#_Toc179816690)

[V.4 Restricciones técnicas 32](#_Toc179816691)

[V.1 Roles de Usuario 32](#_Toc179816692)

[V.2 Requerimientos Funcionales del Sw 32](#_Toc179816693)

[V.3 Requerimientos No Funcionales del Sw 33](#_Toc179816694)

[V.4 Interfaces externas de entrada 35](#_Toc179816695)

[V.5 Interfaces externas de Salida 36](#_Toc179816696)

[Capítulo VI Análisis y Diseño 37](#_Toc179816697)

[VI.1 Descripción de los servicios web - necesarios. 37](#_Toc179816698)

[VI.2 Modelo de datos 38](#_Toc179816699)

[VI.3 Casos de uso 41](#_Toc179816700)

[VI.3.1 Actores 41](#_Toc179816701)

[VI.3.2 Diagrama de casos de uso 41](#_Toc179816702)

[VI.3.3 Especificación de los Casos de Uso 42](#_Toc179816703)

[VI.4 Diseño interfaz y navegación 44](#_Toc179816704)

[VI.4.1 Guías de estilos 44](#_Toc179816705)

[VI.4.2 Guía de colores 44](#_Toc179816706)

[VI.4.3 Composición de las interfaces 46](#_Toc179816707)

[Capítulo VII Desarrollo del Trabajo 47](#_Toc179816708)

[VII.1 Diseño de arquitectura 47](#_Toc179816709)

[VII.2 Estructura del código 48](#_Toc179816710)

[VII.2.1 Backend 48](#_Toc179816711)

[VII.2.2 Frontend 49](#_Toc179816712)

[Capítulo VIII Implantación y puesta en marcha 51](#_Toc179816713)

[VIII.1 Plan de Capacitación/entrenamiento 51](#_Toc179816714)

[VIII.2 Estrategia de implantación. 52](#_Toc179816715)

[VIII.3 Estado del Proyecto 52](#_Toc179816716)

[Capítulo IX Conclusión del proyecto 53](#_Toc179816717)

[Capítulo X Referencias 54](#_Toc179816718)

Índice Tablas

[Tabla 1: Requerimientos Funcionales 32](#_Toc122015021)

[Tabla 2: interfaces de entrada 35](#_Toc122015022)

[Tabla 3: interfaces de Salida 36](#_Toc122015023)

[Tabla 4: Especificación de actores del sistema 41](#_Toc122015024)

[Tabla 5: Especificación CU\_01 43](#_Toc122015025)

[Tabla 6: Especificación CU\_01 crear publicación mejorada desde Tesis Tomas Montecinos IECI 43](#_Toc122015026)

[Tabla 7: Estructura de código – Backend ejemplo Tesis Tomas Montecinos IECI 48](#_Toc122015027)

[Tabla 8: Estructura de código – Frontend ejemplo Tesis Tomas Montecinos IECI 49](#_Toc122015028)

[Tabla 9: Detalle de pruebas de requerimiento 55](#_Toc122015029)

Índice de Ilustraciones

[Ilustración 1 malla curricular carrera IECI, FACE, https://ubiobio.cl/admision/Todas\_las\_Carreras/21/Ingenieria\_de\_Ejecucion\_en\_Computacion\_e\_Informatica/ 9](#_Toc202906598)

[Ilustración 2 malla curricular carrera INCIF, FACE, https://ubiobio.cl/admision/Todas\_las\_Carreras/17/Ingenieria\_Civil\_en\_Informatica\_Concepcion/ 9](#_Toc202906599)

[Ilustración 3 Datos de alumnos IECI titulados 2024 sede concepción FACE, https://dgai.ubiobio.cl/modelos-de-gestion/caracterizacion-estudiantil/ , Consulta 06-07-2025 10](#_Toc202906600)

[Ilustración 4 Datos de alumnos INCIF titulados 2024 sede concepción FACE, https://dgai.ubiobio.cl/modelos-de-gestion/caracterizacion-estudiantil/ , Consulta 06-07-2024 10](#_Toc202906601)

[Ilustración 5 docentes 2025 depto. de sistemas de información FACE., https://dgai.ubiobio.cl/modelos-de-gestion/personal-institucional/, CONSULTADA EL 06-07-2025 10](#_Toc202906602)

[Ilustración 6 modelo desarrollo iterativo e incremental , https://www.pngwing.com/es/free-png-vxojg 21](#_Toc202906603)

[Ilustración 7: Diagrama de Casos de Uso publicaciones – ejemplo tesis Tomás Montecinos IECI 42](#_Toc202906604)

# Introducción

## Definiciones, siglas 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. |
| **IPROCH** | 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 Tesis. Documento y desarrollo que los estudiantes deben realizar para finalizar su proceso de titulación. |
| **JC** | Jefatura de Carrera. 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. |
| **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 tesis ante una comisión académica, como parte del proceso de evaluación final. |
| **Acta** | Documento oficial que registra los resultados de la defensa y las calificaciones obtenidas. |
| **PDF** | Formato de Documento Portátil (Portable Document Format), utilizado para la entrega digital de las propuestas e informes. |

## Presentación del contexto o área de su problemática.

### Presentación Institución/Empresa

La Universidad del Bío-Bío (UBB por sus siglas en español) es una institución estatal y pública de educación superior en la Región del Bío-Bío, que tiene sus raíces en la Universidad Técnica del Estado (UTE por sus siglas en español), fundada en 1947. En 1952, la UTE comenzó oficialmente sus actividades, desempeñando un papel clave en el desarrollo industrial de la región mediante la docencia y la investigación. En 1980, la sede de la UTE en Concepción se transformó en una universidad autónoma, adoptando el nombre de Universidad del Bío-Bío al consolidar la carrera de Arquitectura. Posteriormente, en 1988, la UBB fortalece su posición tras fusionarse con el Instituto Profesional de Chillán (IPROCH por sus siglas en español), convirtiéndose en una institución de referencia en la formación académica y el desarrollo regional. [1]

El proyecto se enmarca en la Facultad de Ciencias Empresariales (FACE por sus siglas en español) de la Universidad del Bío-Bío, una unidad académica que se distingue por su compromiso con la excelencia en la formación de profesionales y en la investigación aplicada en las áreas de economía, administración y gestión.

1.2.2. Presentación de los procesos del ámbito

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. [2]



Ilustración 1 malla curricular carrera IECI, FACE, https://ubiobio.cl/admision/Todas\_las\_Carreras/21/Ingenieria\_de\_Ejecucion\_en\_Computacion\_e\_Informatica/



Ilustración 2 malla curricular carrera INCIF, FACE, https://ubiobio.cl/admision/Todas\_las\_Carreras/17/Ingenieria\_Civil\_en\_Informatica\_Concepcion/

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 [3], 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 [4]. 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.

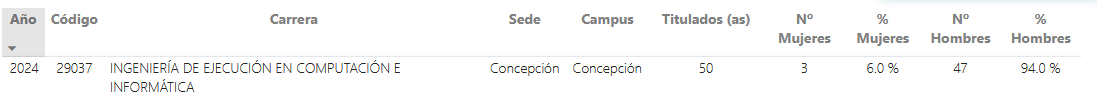


Ilustración 3 Datos de alumnos IECI titulados 2024 sede concepción FACE, https://dgai.ubiobio.cl/modelos-de-gestion/caracterizacion-estudiantil/ , Consulta 06-07-2025

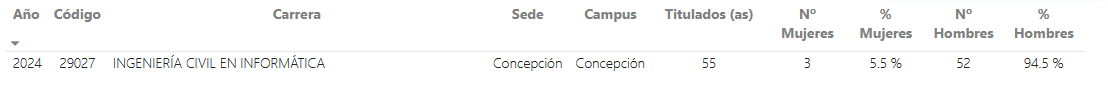


Ilustración 4 Datos de alumnos INCIF titulados 2024 sede concepción FACE, https://dgai.ubiobio.cl/modelos-de-gestion/caracterizacion-estudiantil/ , Consulta 06-07-2024

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, 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. (esta cantidad es menor dado que los académicos ejercen cargos centrales no pueden dirigir proyectos).

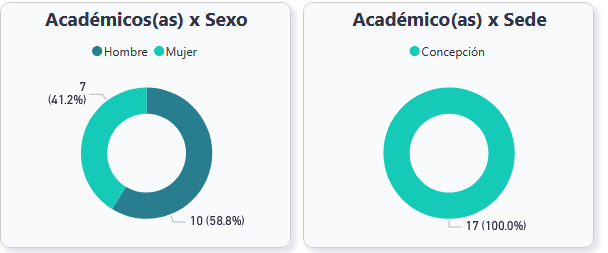


Ilustración 5 docentes 2025 depto. de sistemas de información FACE., https://dgai.ubiobio.cl/modelos-de-gestion/personal-institucional/, CONSULTADA EL 06-07-2025

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 Decana, jefes 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.

* **Inicio de clases:** Se informa sobre la charla informativa y el formato para la presentación de propuestas de Proyecto de Tesis (PT).
* **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:** La Jefatura de Carrera 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**.** Las propuestas sin profesor guía reciben la asignación de uno, idealmente por la afinidad con el tema tratado.
* **Se da inicio al desarrollo del proyecto**
* **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/o Software:** Semanas de revisión están divididas en períodos específicos para la revisión del informe y/o software entregado.
* **Defensa:** una vez revisado el informe y el producto software, el alumno junto a los profesores guía, informante y sala fijan horario y sala de defensa. Las notas de la defensa se formalizan en un acta enviada a la JC.
* **Cierre de Actas y Calificaciones:** Finalización de las evaluaciones y cierre de calificaciones para el Proyecto de Tesis**.**
* **Segunda Fecha para Entrega de Informe Final (con costo):** Opción para los estudiantes que necesiten más tiempo y estén dispuestos a pagar la Cuota Básica de Matrícula.

### Descripción de problemas u Oportunidades de Mejora

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.

## 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:**

**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.

**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.

**Sistematización de Notificaciones:** El sistema enviará notificaciones automáticas 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.

**Facilidad en la Comunicación:** Incluye una herramienta interna de mensajería que permitirá una comunicación fluida entre los estudiantes y sus profesores guías, evitando la dependencia de otros medios de comunicación externos que pueden generar confusión o pérdida de información.

**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.

**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.

**Historial de Revisión:** El sistema almacenará el historial de revisiones y comentarios, de modo que tanto estudiantes como profesores puedan acceder a versiones anteriores de los documentos y revisar los cambios solicitados.

**Escalabilidad:** El proyecto será escalable, lo que permitirá agregar más funcionalidades o ajustarlo para su implementación en otras áreas académicas o instituciones educativas.

## Análisis de los principales Trabajos realizados o soluciones disponibles en el área o tema 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:

1. **Trello (versión gratuita y de pago)** [5]
   * **Funciones**: Trello permite la organización de tareas a través de tableros, listas y tarjetas, donde se pueden gestionar proyectos individuales o grupales. Incluye etiquetas, fechas límite y permite la colaboración entre equipos.
   * **Características**: Colaboración en tiempo real, integraciones con otras herramientas, personalización de tableros, notificaciones y la posibilidad de adjuntar archivos.
   * **Costos**: Versión gratuita con funciones limitadas, y versiones de pago a partir de $5 USD por usuario al mes.
   * **Valoración de Usuarios**: Alta valoración por su simplicidad, aunque es limitado para proyectos más complejos que requieren un seguimiento más estructurado y específico.
   * **Diferencias**: A diferencia de Trello, el software propuesto estará específicamente adaptado a las necesidades académicas de la Universidad del Bío-Bío, incluyendo notificaciones automáticas sobre el estado de revisión, la gestión de plazos y visibilidad total del progreso de las tesis, lo que no es un enfoque central de Trello.

**Asana (versión gratuita y de pago)** [6]

* **Funciones**: Asana permite la gestión de proyectos y tareas, asignación de responsables, seguimiento del progreso y generación de informes sobre el estado del proyecto.
* **Características**: Ofrece múltiples vistas de proyectos (lista, cronograma, tablero), integración con otras herramientas de gestión y colaboración en tiempo real.
* **Costos**: Versión gratuita con funcionalidades básicas y planes de pago a partir de $10.99 USD por usuario al mes.
* **Valoración de Usuarios**: Los usuarios valoran su capacidad para gestionar proyectos complejos, pero mencionan que puede ser complicado de usar para tareas simples o individuales.
* **Diferencias**: Asana es una herramienta de gestión de proyectos genérica, mientras que el software propuesto estará personalizado para el proceso de titulación de la universidad, con funcionalidades específicas como la documentación automática de cada fase del proyecto y la trazabilidad de revisiones

**Notion (versión gratuita y de pago)** [7]

* **Funciones**: Gestión de notas, bases de datos, wikis, tareas y proyectos. Permite la colaboración en tiempo real y la organización de información estructurada.
* **Características**: Totalmente personalizable, integración con herramientas externas, plantillas para educación y proyectos, múltiples vistas (tabla, calendario, galería, lista).
* **Costos**: Plan gratuito con limitaciones mínimas. Planes pagos desde $8 USD por usuario al mes.
* **Valoración de Usuarios**: Muy valorado por su flexibilidad y diseño minimalista, aunque algunos usuarios indican que puede volverse complejo en equipos grandes sin estructura clara.
* **Diferencias**: Notion es generalista y requiere configuración manual. El sistema propuesto se enfoca específicamente en el proceso de titulación, automatizando procesos definidos como revisiones, validaciones y comentarios.

**Monday.com (versión gratuita y de pago)**

* **Funciones**: Gestión de proyectos con automatización de tareas, asignación de miembros, vistas personalizadas, seguimiento del tiempo y recursos.
* **Características**: Muy visual, permite tableros con columnas personalizadas, automatizaciones complejas y buena integración con otras apps.
* **Costos**: Plan gratuito limitado a 2 usuarios. Planes pagos desde $9 USD por usuario al mes.
* **Valoración de Usuarios**: Buena valoración por flexibilidad y diseño moderno. Curva de aprendizaje algo elevada si se usa con muchas automatizaciones.
* **Diferencias**: Monday.com está pensado para múltiples industrias. El sistema propuesto está adaptado al flujo académico con control por fases y usuarios universitarios.

**ClickUp (versión gratuita y de pago)**

* **Funciones**: Gestión de proyectos, tareas, documentos, tiempo, objetivos y automatización.
* **Características**: Vista tipo lista, Gantt, calendario, tablero kanban. Personalización avanzada de espacios y tareas.
* **Costos**: Plan gratuito muy completo. Planes pagos desde $7 USD por usuario al mes.
* **Valoración de Usuarios**: Muy positivo por su amplitud de funciones incluso en plan gratuito. Algunos mencionan que puede resultar abrumador.
* **Diferencias**: ClickUp es muy completo pero generalista. El sistema UBB tiene un flujo más rígido y específico, con control de estados, roles y seguimiento por parte de jefatura

## Justificación del proyecto

El desarrollo de un sistema centralizado que realice el seguimiento de proyectos de título es una respuesta directa a los desafíos que enfrentan tanto estudiantes como profesores en el proceso actual. La falta de visibilidad sobre el estado de las revisiones, el seguimiento de los avances y la comunicación entre las partes genera incertidumbre, retrasos y una carga administrativa significativa para los profesores guías. Estos problemas pueden afectar negativamente la planificación y el progreso de los proyectos de titulación, impactando tanto el rendimiento académico de los estudiantes como la eficiencia del proceso para los docentes.

Aunque existen herramientas actuales como Google Classroom [8], Moodle [9]y plataformas de gestión de proyectos como Trello [10] [10] que permiten cierta organización y comunicación, estas soluciones no están diseñadas específicamente para el proceso de titulación. Si bien pueden ser útiles para tareas generales, no abordan de manera eficiente la necesidad de seguimiento detallado de informes y plazos, la gestión de revisiones múltiples o las notificaciones automatizadas específicas del proceso de titulación.

El sistema propuesto está específicamente diseñado para cubrir las necesidades de la Universidad del Bío-Bío en el seguimiento de proyectos de título. A diferencia de herramientas genéricas, este sistema ofrecerá:

* **Seguimiento automatizado y detallado** de cada fase del proyecto, desde la propuesta inicial hasta la defensa final.
* **Notificaciones específicas** relacionadas con el estado de las entregas y revisiones, adaptadas a los plazos establecidos.
* **Gestión centralizada de revisiones y avances**, facilitando a los profesores la revisión y seguimiento sin depender de herramientas externas que requieren más pasos o procesos manuales.

Al centralizar la información y automatizar las notificaciones, se reducirá la carga administrativa sobre los profesores, minimizando el riesgo de errores y retrasos en la revisión de los informes. A su vez, los estudiantes podrán gestionar mejor sus plazos, mejorar la planificación de sus actividades y tener mayor confianza en el seguimiento de sus proyectos.

Este sistema contribuirá a un proceso de titulación más transparente, eficiente y organizado, beneficiando a todos los actores involucrados, y asegurando que los proyectos se completen de manera oportuna y con la calidad requerida. Además, el proyecto tiene el potencial de ser escalable y adaptable a otras instituciones o procesos académicos que enfrenten problemas similares.

## 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: Solución software para la coordinación de las actividades de titulación para las carreras de informática de la Facultad de Cs. Empresariales”**

El documento se organiza en 10 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.

# Proyecto

## Objetivo general del proyecto

Desarrollar un sistema centralizado que apoye la asignación, seguimiento y visibilidad del estado de los proyectos de titulación de las carreras de informática de la FACE. A través de este sistema se espera facilitar la comunicación entre los actores involucrados y optimizar la gestión de revisiones, evaluaciones y entregas.

## Objetivos específicos del proyecto

**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.

**Diseñar** un sistema de software adaptado a los requerimientos institucionales de la Universidad del Bío-Bío, incluyendo funcionalidades como notificaciones automáticas y seguimiento

**Implementar** el sistema de gestión de proyectos de titulación en la Universidad del Bío-Bío, asegurando su correcta integración con las plataformas académicas existentes.

## Metodología de desarrollo

Luego de analizar a fondo el proyecto, se concluyó que en relación con el problema se cuenta con una experiencia alta, pero la complejidad es moderada, ya que el proceso de seguimiento de tesis tiene varios actores y etapas a considerar. Sin embargo, el tamaño del problema es mediano, ya que se trata de un entorno controlado dentro de la Universidad del Bío-Bío. Esto representa un riesgo bajo en cuanto al problema.

La siguiente tabla caracteriza el proyecto a desarrollar, en distintos aspectos:

Tabla 1: Características del proyecto

| **Ítem** | **Nivel** | **Descripción** |
| --- | --- | --- |
| Experiencia en el problema | Medio | Se tiene conocimiento sobre el tipo de problema y se ha trabajado en procesos similares relacionados con la gestión académica |
| 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. |

Tras evaluar el proyecto, se determinó que el riesgo total es muy bajo, lo que ofrece la flexibilidad de utilizar una metodología de desarrollo. En este contexto, se ha seleccionado una metodología iterativa que permite el desarrollo por ciclos. Estos ciclos facilitan que el software evolucione de acuerdo con la retroalimentación recibida de los usuarios, corrigiendo errores y mejorando las funcionalidades conforme avanza el desarrollo.

Dado que los estudiantes y profesores estarán directamente involucrados en el uso del sistema, su participación en el proceso de pruebas será crucial para asegurar que el software cumpla con sus necesidades.

Este enfoque permitirá un desarrollo continuo y ágil del sistema, asegurando que el producto final sea una herramienta efectiva para el seguimiento de tesis en la Universidad del Bío-Bío.

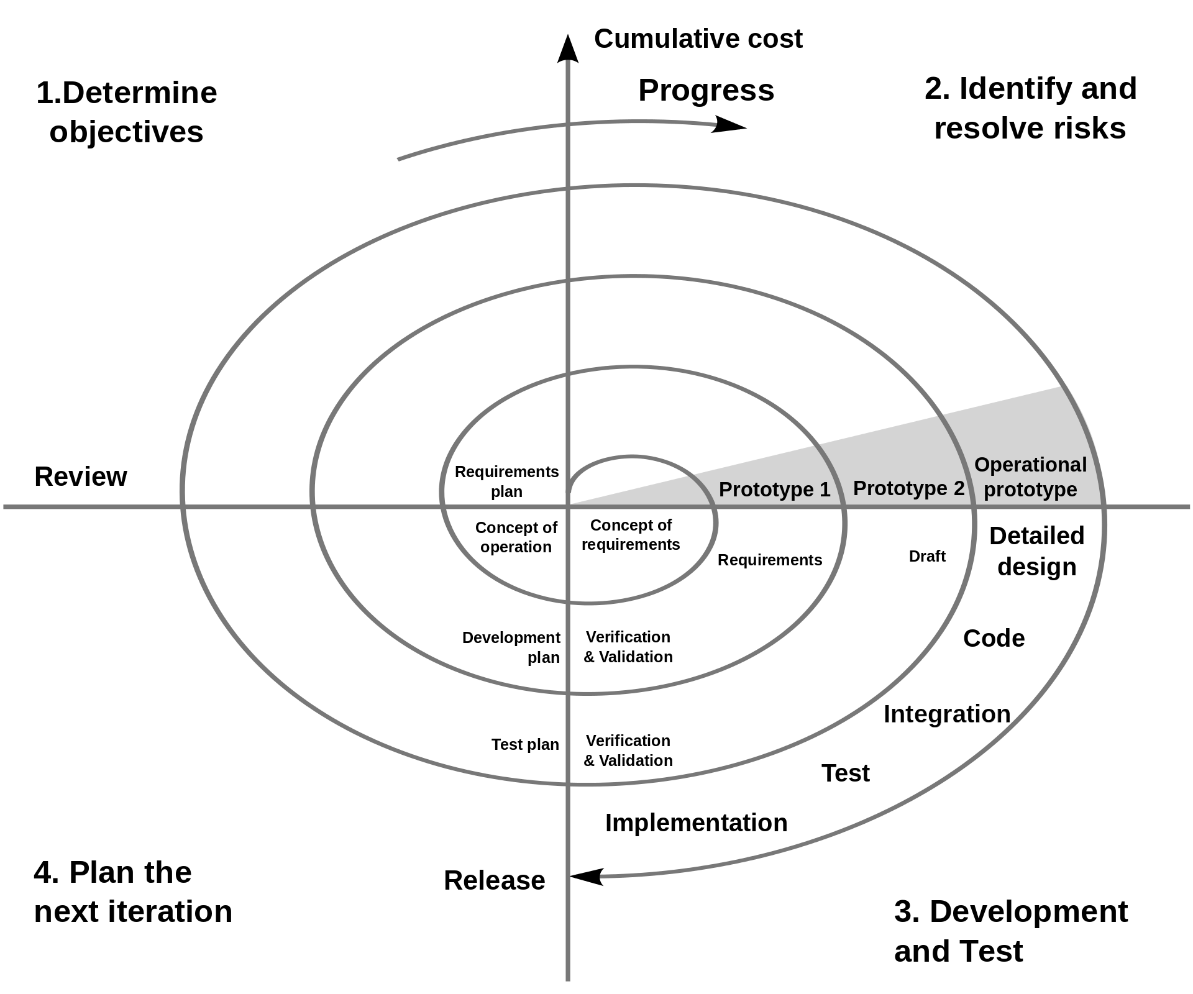


Ilustración 6 modelo desarrollo iterativo e incremental , https://www.pngwing.com/es/free-png-vxojg

## Descripción de las actividades para lograr los objetivos Específicos

**Objetivo específico 1: Levantar los requisitos funcionales y no funcionales del sistema**

1. Revisión de sistemas de gestión académica utilizados en otras universidades.
2. Revisión de antecedentes de proyectos similares desarrollados en la UBB.
3. Entrevistas y cuestionarios dirigidos a estudiantes, profesores y jefes de carrera.
4. Priorización de requisitos en base a criticidad y frecuencia de uso.
5. Validación de la lista final de requisitos con usuarios clave.

**Objetivo específico 2: Diseñar la arquitectura del sistema**

1. Análisis de patrones de diseño adecuados para una aplicación web modular.
2. Elaboración de diagrama de arquitectura general del sistema.
3. Diseño de la estructura de base de datos y relaciones entre entidades.
4. Validación del diseño arquitectónico con el equipo técnico.

**Objetivo específico 3: Implementar el sistema web de gestión para apoyar el seguimiento de los proyectos de titulación**

1. Implementación del backend con Node.js y Express.js siguiendo estructura en capas.
2. Desarrollo del frontend con Angular usando componentes standalone.
3. Configuración de servicios para el manejo de autenticación y autorización.
4. Implementación de flujos diferenciados para estudiantes, profesores y jefe de carrera.
5. Integración de subida de archivos y manejo de estados de propuesta.

**Objetivo específico 4: Validar el sistema con usuarios finales**

1. Desarrollo y ejecución de pruebas funcionales con estudiantes y profesores.
2. Corrección de errores detectados en pruebas de usuario.
3. Ajuste de funcionalidades según retroalimentación recibida.
4. Pruebas de aceptación supervisadas por usuarios clave.

## Estándares de documentación

Adaptación basada en IEEE Software Test Documentation Std 829-1998

Adaptación basada en IEEE Software Requirements Specifications Std 830-1998

## Técnicas y notaciones

**UML**: Diagramas de casos de uso, diagramas de clases, diagramas de secuencia.

**BPMN**: Para modelar los procesos de revisión, asignación y seguimiento de propuestas de tesis.

**Cuestionarios y entrevistas**: Técnicas utilizadas para levantar requisitos con usuarios finales.

**Carta Gantt**: Para la planificación temporal de actividades del proyecto.

## Herramientas, framework, lenguaje usados en el desarrollo del proyecto

**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

# Proceso de Digitalización en Municipalidad (ejemplo)

Capítulos para detallar temas importantes, explicar un modelo que será utilizado.

## Proceso de Transformación Digital

Etapas

## Proceso de D. Transito

Estado actual planilla

Que proceos declkarados

El estados reci

## Discusión

Estado actual planilla

Que proceos declkarados

El estados

# Factibilidad

## 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 por un equipo que posee las competencias técnicas necesarias para su correcta ejecución. Los integrantes cuentan con 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, manejan 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.

*Tabla 2: Especificación software requerido en desarrollo en el 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 (version lts) |
| 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) |

*Tabla 3: Especificación Hw requerido en desarrollo el 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 |

*Tabla 4: Especificación hw servidor 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**

Considerando los aspectos analizados, se concluye que el desarrollo del sistema es técnicamente factible.  
El equipo cuenta con las competencias técnicas requeridas y con acceso al hardware necesario para la programación, pruebas y ejecución del sistema.  
Asimismo, las herramientas de software seleccionadas son de libre acceso o de código abierto, lo que permite su uso sin restricciones económicas.

Por otra parte, todas las tecnologías empleadas —tales como Node.js, Angular, MySQL, Docker— cuentan con comunidades activas de usuarios y documentación oficial, lo que garantiza soporte, actualización continua y resolución de posibles problemas durante el desarrollo y la explotación del sistema.

Por todo lo anterior, se valida la viabilidad técnica del proyecto.

## 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.

Por lo anterior, se concluye que el sistema es **operativamente factible**, ya que tanto los clientes como los usuarios finales valoran su importancia, están dispuestos a utilizarlo y cuentan con las habilidades mínimas requeridas, o bien, se tomarán las medidas necesarias para asegurar su correcta adopción.

## Factibilidad económica

Describa:

*Tabla 5: Licencias.*

|  |  |  |
| --- | --- | --- |
| Software | Licencia | Costo licencia |
| Visual Studio Code | MIT License | $0 |
| Node.js | MIT License | $0 |
| Angular CLI | MIT License | $0 |
| MySQL | Licencia Pública General de GNU (GPL v2) | $0 |
| MySQL Workbench | Licencia Pública General de GNU (GPL v2) | $0 |
| Docker (uso personal/educativo) | Licencia Docker Subscription Service Agreement | $0 |
| Git | Licencia Pública General de GNU (GPL v2) | $0 |
|  |  |  |

*Tabla 6: Costo hosting y dominio, Este servicio tiene un costo anual obtenido de https://www.nic.cl/dominios/tarifas.html*

|  |  |  |
| --- | --- | --- |
| **item** | **Precio mensual aproximado en CLP** | **Precio anual aproximado en CLP** |
| Hosting | $0 | $0 |
| Dominio | - | $10.000 |
| **Total** | **-** | **$10.000** |

*Tabla 7: Calculo costo de desarrollo y soporte.*

|  |  |  |  |
| --- | --- | --- | --- |
| Recursos humanos | Cantidad personal | Sueldo aproximado en CLP por mes | Salario aproximado por duración de proyecto (12 meses) en CLP |
| Desarrolladores | 1 | $950.000 | $11.400.000 |
| Total | | | $11.400.000 |

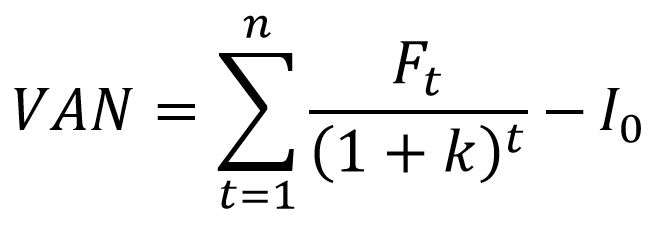
### Flujo de caja

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:

Tabla 8: Flujo de caja del proyecto

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Año 0 | Año 1 | Año 2 | Año 3 | Año 4 | Año 5 |
| (+) Ingresos |  |  |  |  |  |  |
| Beneficios |  | * $ reducción de costos (ahorro) | $ reducción de costos (ahorro) | $ reducción de costos (ahorro) | $ reducción de costos (ahorro) | $ reducción de costos (ahorro) |
| (-) Costos |  |  |  |  |  |  |
| Servicios y equipo | * $ costo desarrollo MO * $ (%luz + internet+depreciación equipos existentes) * $ costo equipos comprados | * $ hosting anual * $ dominio anual | * $ hosting anual * $ dominio anual | * $ hosting anual   $ dominio anual | * $ hosting anual   $ dominio anual | * $ hosting anual   $ dominio anual |
| Soporte y mantención |  | * $ costo mantención MO | $ costo mantención MO | $ costo mantención MO | $ costo mantención MO | $ costo mantención MO |
| TOTAL | F0 | F1 | F2 | F3 | F4 | F5 |

#### Cálculo del V.A.N



Donde cada uno de los términos, se especifican en la Tabla:

Tabla 9: Términos de la fórmula de VAN

|  |  |
| --- | --- |
| Término | Significado |
| t | Intervalo de tiempo |
| n | Duración en años |
| I0 | Inversión inicial (t=0) |
| K | Tasa de descuento |
| Vt | Flujos de caja obtenidos en el intervalo de tiempo t |

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

Tabla 10: Cálculo del VAN

|  |  |
| --- | --- |
| Año | Flujo de caja |
| Año 0 | $f0 / (1 + 0.10) ^0 = $ |
| Año 1 | $f1 / (1 + 0.10) ^1 = $ |
| Año 2 | $f2 / (1 + 0.10) ^2 = $ |
| Año 3 | $f3 / (1 + 0.10) ^3 = $ |
| Año 4 | $f4/ (1 + 0.10) ^4 = $ |
| Año 5 | $f5/ (1 + 0.10) ^5 = $ |

**VAN (10%)** = Año 0 + Año 1 + Año 2 + Año 3 + Año 4 + Año 5

= $

**VAN (10%)** = $

En este caso, el VAN obtenido es positivo ($ ), lo que indica que el proyecto es viable desde una perspectiva económica.

#### Conclusión de la factibilidad

Gracias al análisis realizado en los puntos anteriores, se puede concluir que el proyecto es

# Definición de la solución Software

## Objetivo General del Software propuesto.

El sistema gestionará la información del proceso de seguimiento y revisión de propuestas de proyectos de titulación para que la institución optimice el uso de sus recursos administrativos y académicos, reduciendo los tiempos, asignación, evaluación y aprobación de trabajos de titulación.

## Objetivos Específicos del Software propuesto.

* El sistema **automatizará** el registro y envío de las propuestas de los proyectos de título para que la institución **reduzca** el tiempo dedicado a la recepción y validación de solicitudes.
* El sistema **facilitará** la asignación y seguimiento de revisores para que la institución **mejore** la trazabilidad y control de las etapas de evaluación, disminuyendo la necesidad de gestiones administrativas repetitivas.
* El sistema **notificará** a los usuarios sobre el estado de sus propuestas/proyectos para que la institución **optimice** la comunicación entre estudiantes, profesores y jefes de carrera, reduciendo las gestiones informales y consultas administrativas.

## Límites

* El sistema **no permitirá** la aprobación automática de propuestas sin la revisión previa por parte de un profesor asignado.
* El sistema **no permitirá** la modificación de propuestas una vez que hayan sido aprobadas o rechazadas formalmente.
* El sistema **no permitirá** el acceso a información de otras propuestas o usuarios sin la debida autorización de acuerdo con el perfil asignado (estudiante, profesor o jefe de carrera).
* El sistema **no permitirá** la carga de archivos fuera de los formatos permitidos (.pdf, .docx) ni la carga de documentos que excedan el tamaño máximo definido.
* El sistema **no permitirá** el acceso o gestión de las funciones administrativas desde usuarios no autenticados o sin los permisos correspondientes.

## Restricciones técnicas

* El sistema **no contempla integración directa con plataformas institucionales existentes**, salvo que se realicen desarrollos adicionales posteriores a su implementación.

## Roles de Usuario

El sistema será utilizado por los siguientes tipos de usuarios, cada uno con funciones y responsabilidades específicas dentro del proceso de seguimiento de tesis:

* **Estudiantes**  
  Son los principales solicitantes del proceso. Utilizarán el sistema para registrar, enviar y dar seguimiento a sus propuestas de tesis, además de recibir notificaciones sobre el estado de revisión y observaciones realizadas por los revisores.
* **Profesores Revisores**  
  Son los encargados de revisar, comentar y aprobar o rechazar las propuestas asignadas. También podrán realizar observaciones y solicitar modificaciones a los estudiantes antes de aprobar una propuesta.
* **Jefe de Carrera o Coordinador Académico**  
  Será el encargado de asignar profesores revisores a las propuestas recibidas, supervisar el estado de estas y validar el flujo general del proceso. Además, podrá consultar reportes y estadísticas para la gestión académica.

Cada uno de estos roles responde a necesidades detectadas en la gestión actual del proceso de titulación, permitiendo con el sistema la formalización, trazabilidad y control del flujo de propuestas de tesis en la institución.

## Requerimientos Funcionales del Sw

La lista de los requerimientos funcionales específicos se presenta en la Tabla 11.

Tabla 11: Requerimientos Funcionales

| id | el sistema debe |
| --- | --- |
| RF\_01 | Permitir que el **estudiante** registre una nueva propuesta para su proyecto de título , siempre que haya iniciado sesión, completando todos los campos obligatorios del formulario. Al registrar la propuesta, esta quedará en estado “Pendiente de revisión” y se notificará al jefe de carrera. |
| RF\_02 | Permitir que el **jefe de carrera** asigne un profesor revisor a cada propuesta registrada, solo si la propuesta se encuentra en estado “Pendiente de revisión”. La asignación se notificará al profesor asignado y al estudiante. |
| RF\_03 | Permitir que el **profesor revisor** visualice las propuestas que le han sido asignadas, revise su contenido y registre observaciones o comentarios, solo si la propuesta no ha sido aún aprobada o rechazada. |
| RF\_04 | Permitir que el **profesor revisor** cambie el estado de la propuesta a “Aprobada” o “Rechazada”, solo después de haber ingresado al menos una revisión. Al cambiar el estado, el sistema notificará al estudiante y al jefe de carrera. |
| RF\_05 | Permitir que el **estudiante** consulte en todo momento el estado actual de su propuesta y las observaciones realizadas por el profesor revisor. |
| RF\_06 | Restringir el acceso a la edición o eliminación de propuestas una vez que la propuesta haya sido “Aprobada” o “Rechazada”. |
| RF\_07 | Notificar automáticamente a los usuarios involucrados (estudiante, profesor revisor, jefe de carrera) cada vez que cambie el estado de la propuesta o se registre una nueva observación. |
| RF\_08 | Registrar un historial de revisiones y cambios de estado por cada propuesta, accesible para el estudiante, el profesor revisor y el jefe de carrera. |

## Requerimientos No Funcionales del Sw

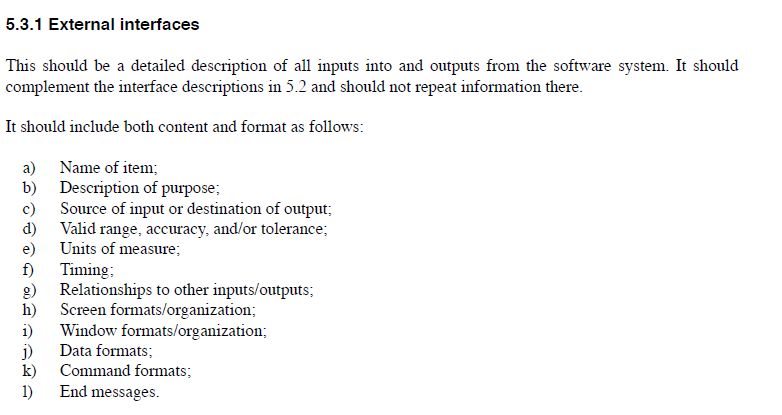
La presente sección hablará de los requerimientos no funcionales de la aplicación desarrollada. Todos los requerimientos no funcionales se relacionarán con uno o más atributos. Si un atributo aplica a un requerimiento no funcional, eso quiere decir que el requerimiento contribuye a la calidad del software desarrollado a través de ese atributo. Todos los atributos listados están basados en la norma ISO 25010

|  |  |  |
| --- | --- | --- |
| RNF 01 | | |
| Descripción | La plataforma contará con una API REST, la cual estará diseñada para  ser consumida por aplicaciones de terceros. Todos los endpoints estarán disponibles sólo para lectura por parte de terceros. | |
| Atributo | Aplica | Especificación |
| Adecuación  Funcional | X | La API contribuye a la corrección funcional, ya que facilita la obtención de datos precisos del sistema a terceros. |
| Eficiencia de  Desempeño |  |  |
| Compatibilidad | X | La API contribuye a la coexistencia con otras piezas de  software independiente, ya que permite a dicho software consumir información del sistema en tiempo real. |
| Usabilidad |  |  |
| Fiabilidad | X | La API contribuye a la madurez del software, ya que  es gracias a ella que el sistema puede satisfacer las necesidades de los usuarios que consumen información del mismo. |
| Seguridad | X | Gracias al diseño de la API, sólo se exponen endpoints de  lectura, por lo que esta contribuye a la confidencialidad e integridad de la información. |
| Mantenibilidad |  |  |
| Portabilidad |  |  |

|  |  |  |
| --- | --- | --- |
| RNF 02 | | |
| Descripci´on | El c´odigo estar´a versionado a trav´es de Git, utilizando GitHub como plataforma de almacenamiento en la nube para el control de versiones. El proyecto ser´a de c´odigo abierto, y se contar´a con un sistema de integraci´on continua para el software implementado directamente en GitHub. A trav´es de este sistema se podr´a ser desplegar el software de manera au- tom´atica cuando se detecten cambios en una rama de Git determinada. | |
| Atributo | Aplica | Especificaci´on |
| Adecuaci´on  Funcional |  |  |
| Eficiencia de  Desempen˜o |  |  |
| Compatibilidad | X | Contribuye a la coexistencia del software con otros sis-  temas, ya que, gracias a estar versionado en Git y ser de c´odigo abierto, los mantenedores y desarrolladores de otras piezas de software podr´an tener acceso a toda la informaci´on t´ecnica que necesiten para poder integrar sus productos con la plataforma de RVA exitosamente. |
| Usabilidad | X | Existe una contribuci´on al reconocimiento de la adecua-  ci´on, ya que un usuario, al poder visualizar el historial de desarrollo completo en Git, podr´a entender si el soft- ware, a nivel t´ecnico, es adecuado para sus necesidades. |
| Fiabilidad | X | Gracias al sistema de despliegue que detalla, este re-  querimiento no funcional contribuye a la madurez, dis- ponibilidad y tolerancia a fallos del sistema. Al contar con un sistema de despliegue y versionado del proyecto, los desarrolladores pueden trabajar en el y hacer llegar actualizaciones much´ısimo m´as r´apido al entorno de pro- ducci´on, y por ende a los usuarios finales del software. |
| Seguridad | X | Contribuye al no repudio ya que, gracias a que el control  de versiones lleva registro de todo lo que se modifica en la base de c´odigo, ser´a posible rastrear malas decisiones o errores con facilidad. |
| Mantenibilidad | X | Este requerimiento contribuye a la modularidad, reusa-  bilidad analizabilidad y a la capacidad del software para ser modificado y probado. Todo esto gracias a que los cambios realizados en el software son documentados a trav´es del control de versiones. |
| Portabilidad | X | Contribuye a la adaptabilidad del software, y a la capaci-  dad del mismo para ser instalado. Como todo el software es pu´blico y se encuentra en Git, analizarlo para adap- tarlo resulta muy sencillo. Por otra parte, el sistema de despliegue facilita enormemente el instalar la aplicaci´on en un entorno determinado. |

## Interfaces externas de entrada

Cada interfaz externa, es una especificación tomada desde IEEE Std 830-1998 página 22, tal como lo muestra la figura. Se separa en ENTRADA Y SALIDA.



Interfaz externa de entrada, es decir un conjunto de datos que serán ingresados al sistema independiente del medio de ingreso, es decir datos que provienen de un usuario a través de teclado, lector de código barra, etc.

En la tabla se incluyen los ítems de datos 1 sola vez, por ejemplo, no es necesario repetir el rut del proveedor o el código del producto, en esta tabla importan QUE DATOS INGRESARÁN y no importa cuántas veces ingresen.

Tabla 12: interfaces de entrada

| Identificador | Nombre del ítem. | Detalle de Datos contenidos en ítem |
| --- | --- | --- |
| Ejemplo: DE\_01 | Datos del proveedor | NOMBRE, RUT\_PROV, GIRO, DIRECCION,TELEFONO….. |
| DE\_02 | Datos de factura | ~~RUT\_PROV,~~FECHA\_FACT, TIPO\_PAGO~~, COD\_PROD~~, CANT\_COMPRADA, PRECIO ……. |
| DE\_03 | Datos de productos | COD\_PROD, NOMBRE |
|  |  |  |

## Interfaces externas de Salida

Se especifica cada salida del sistema, conjuntos de datos que se sacan del sistema para los usuarios u otros sistema, indicando en cada caso el formato o medio de salida.

Tabla 13: interfaces de Salida

| Identificador | Nombre del ítem. | Detalle de Datos contenidos en ítem | Medio Salida |
| --- | --- | --- | --- |
| Ejemplo  IS\_01 | Informe de los proveedores | NOMBRE, RUT, CODIGO,GIRO,DIRECCION,TELEFONO | Archivo XLS  Impresora  Pantalla |
|  |  |  |  |
|  |  |  |  |

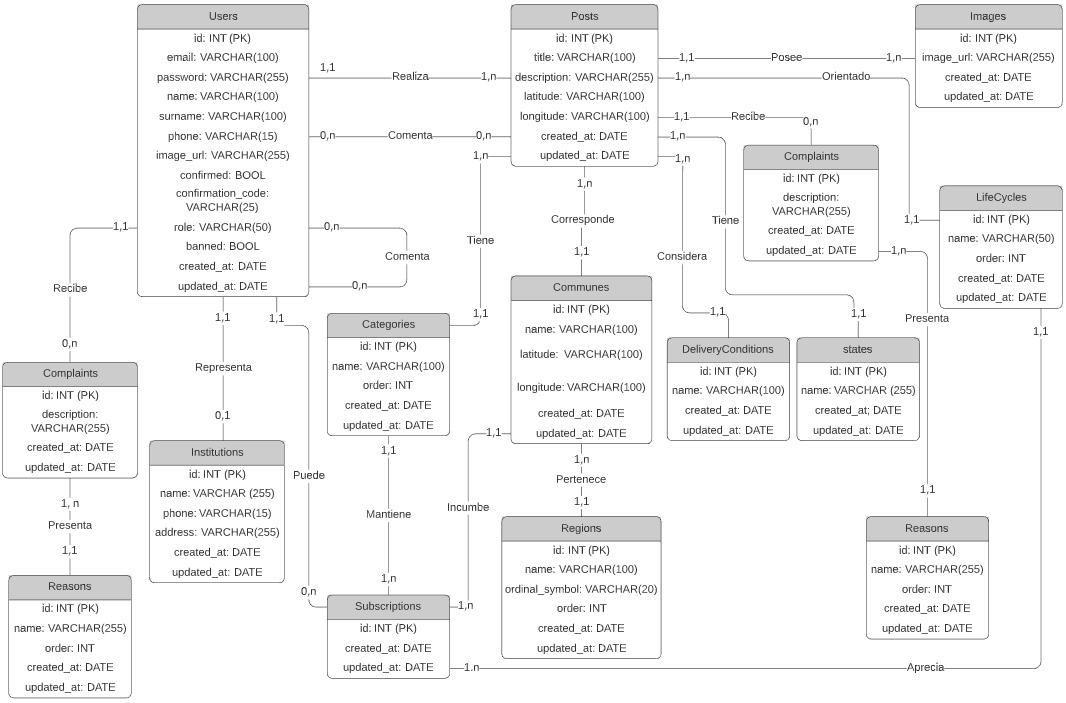
# Análisis y Diseño

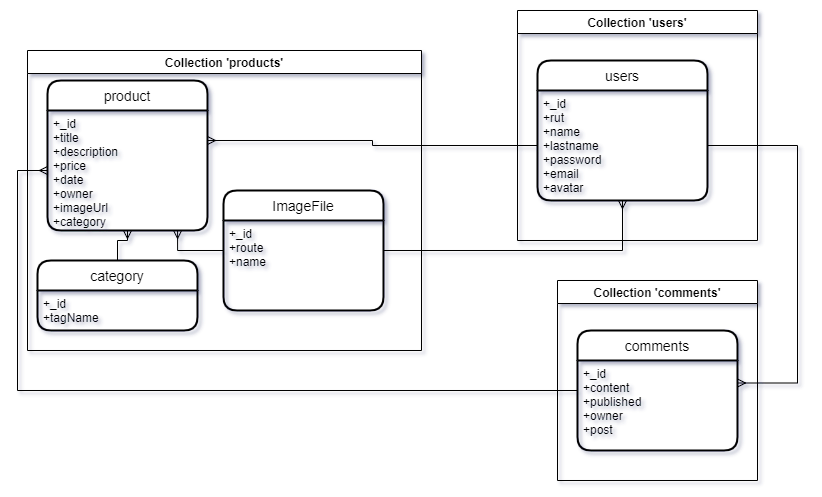
## Descripción de los servicios web - necesarios.

## Modelo de datos

Debe explicar el modelo de datos, relacional o no-relacional, es decir leerlo destacando las entidades, colecciones y/o relaciones -referencias fundamentales en la problemática.

Diagrama con Modelo de datos relacional o no relacional





Esquema BD no relacional/\*opcional

A continuación, se describen los datos y tipos de la BD, en formato JSON.

Esquema products :

{

  title: {

    type: String,

    required: true

  },

  description: {

    type: String,

    default: null

  },

  price: {

    type: Number,

    required: true

  },

  date: {

    type: Date,

    required: true

  },

  imageUrl: [{

    type: Schema.ObjectId,

    ref: "imageFile"

  }],

  owner: {

    type: Schema.ObjectId,

    ref: "user"

  },

  category: [{

    type: Schema.ObjectId,

    ref: "category"

  }],

}

## Casos de uso

### Actores

Los actores que interactúan con el sistema se detallan en la Tabla 14.

Tabla 14: Especificación de actores del sistema

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ACTOR |  |  | Nivel de conocimientos técnicos requeridos | Nivel privilegio en el sistema |
|  |  |  |  |  |
|  |  |  |  |  |

En el caso de que existen generalizaciones de actores incluya los diagramas correspondientes.

### Diagrama de casos de uso

Incluya más de 1 diagrama para que quede claro el modelo.

Diagrama(s) de CU

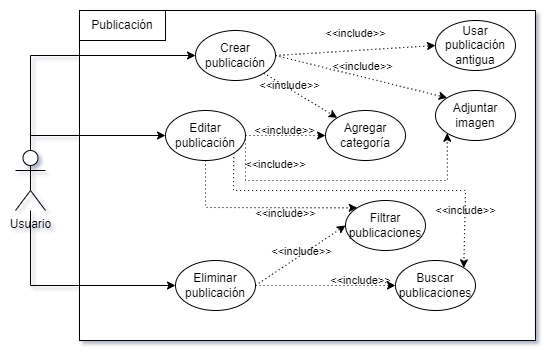


Ilustración 7: Diagrama de Casos de Uso publicaciones – ejemplo tesis Tomás Montecinos IECI

### Especificación de los Casos de Uso

Liste los CU que están en su (s) diagramas destacando cuales serán detallados.

Considerando funcionalidad RELEVANTE del negocio especifique con la tabla sólo los CU relacionados. Para los CU restantes sólo incluya una descripción y precondiciones.

|  |
| --- |
| Pre-Condiciones: Se listan las condiciones que deben cumplirse para que ocurra el caso de Uso 1 |
| Includes: a que CU hace include |
| Extends: de que otro caso de uso SE EXTIENDE este CU |
| Flujo de Eventos Básicos |
| 1.-. <nombre del actor> y/o sistema… |
| 2 (a)Si ( )sistema hace … |
| 3 |
| 4 |
| Flujo de Eventos Alternativo: |
| <nombre del actor> |
| 2(b) Si no( ) el sistema hace … |
|  |
| Post-Condiciones: Lista de posibles condiciones o estados de la base de datos que pueden o deben cumplirse al terminar el Caso de Uso. |

Tabla 15: Especificación CU\_01

Ejemplos:

|  |  |
| --- | --- |
| CU\_01: Crear publicación | |
| Precondiciones El usuario debe estar **registrado** previamente en el sistema, se encuentra logeado como usuario UBB. | |
| Flujo de eventos | |
| 1)La app despliega una pantalla con un formulario de creación  2.1) El usuario debe rellenar los campos requeridos, y agregar imagen y desplegar la lista de categorías.  3.1) La app valida campos de texto y formato de estos, si **validación esta correcta** se llama al “CU\_04: Agregar Imagen” y/o al “CU\_06: Agregar categoría” según el usuario requiera.  4) el usuario debe confirmar la publicación  5) La aplicación agrega esta nueva publicación a la base de datos. | |
| Flujo de eventos alternativos | |
| 2.2) El usuario desea “Usar publicación antigua”  3.2) La app llama al “CU\_05: Usar publicación antigua” y pasa hacia el paso 3.1 |  |
| Post condiciones la nueva publicación se visualiza en el catálogo | |

Tabla 16: Especificación CU\_01 crear publicación (mejorada desde Tesis Tomas Montecinos IECI)

## Diseño interfaz y navegación

### Guías de estilos

La guía de estilo marcará las pautas a seguir para el diseño de la web. Por tanto, servirá

de consulta para visualizar los objetivos de la aplicación en cuanto a su estilo. El principal

objetivo es dotar al estilo de la aplicación de la máxima sencillez posible, a fin de que al

usuario le resulte intuitivo el uso de la aplicación.

A continuación, se señalarán aspectos relativos al estilo como son el logotipo, los colores

principales y secundarios, la tipografía y la composición de las interfaces.

#### Logotipo

El logotipo es el símbolo que representa la temática de la aplicación. Esta imagen podrá ser presentada de diferentes formas, como son el imagotipo y el isotipo. Además, se mostrarán dos versiones de cada representación: una en color y otra en blanco y negro.

Por lo que respecta al isotipo, en la figura 8.8, este se caracteriza por no tener ningún texto. El icono representativo se caracteriza por estar compuesto por dos elementos. El principal es un coche con 5 pasajeros, simbolizando el hecho de compartir coche, mientras que el secundario se trata del icono representativo de la ubicación con un birrete, que simboliza el hecho de encontrar estudiantes.

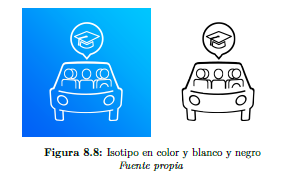


Figura 1

### Guía de colores

Los colores corporativos de la aplicación componen una gama cromática fría, orientada a

tonos azules. En primer lugar, los colores corporativos que se utilizarán mayoritariamente en el diseño de las interfaces están reunidos en la paleta de la figura 8.10. Por una parte, “Bluetiful” y “Vivid Sky Blue” serán los colores de los elementos con los que el usuario podrá interactuar, como botones y enlaces. Y, por otra parte, “Black Chocolate” y “Azyre X Web Color” serán los correspondientes al texto y fondo de la aplicación.



Figura 2

No obstante, a estos colores principales se les añadirán 3 más que estarán relacionados con la acción que realizan los botones en los que se apliquen. Se trata del color “Rufous” para acciones como “Eliminar” y “Cancelar”, el “Maximum Yellow” para “Editar” y “Leaf Green” para “Enviar” y “Aceptar”.



Figura 3

#### Tipografía

La tipografía utilizada es un tipo de letra sencilla y universal que facilite la legibilidad. Se ha optado por emplear una fuente gratuita de Google Fonts, como es Roboto. En la figura se incluye una representación de dicha fuente con los caracteres más comunes.

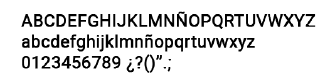


Figura 4

### Composición de las interfaces

La composición de las interfaces expuesta a continuación corresponde a la versión adaptada para móvil, que será la versión implementada en primer lugar. La adaptación para pantallas de mayor tamaño se realizará en un futuro.

Distinguiremos entre 3 tipos de interfaces según su composición: lista, tarjeta y formulario. La comparación entre ellas puede observarse en la figura

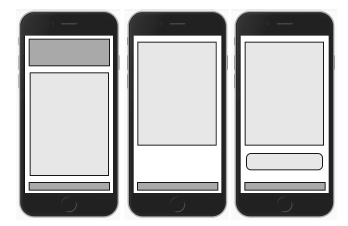


Figura 5

<Incluye al menos **3 mockups o screenshot** de las interfaces propuestas que representan el estándar que será seguido en el sw>.

Recuerde: El diseño de la interfaz de usuario debe considerar un diseño estándar que será respetado en todas las pantallas. En el diseño se considera la organización y el aspecto de la interfaz. El aspecto considera muchos elementos, entre ellos, los colores, imágenes de fondo, uso de iconos entre otros. La organización de una pantalla considera la ubicación de cada uno de los tipos de elementos de la interfaz, considerando por ejemplo las siguientes áreas: De ingresos de datos, De Botones de opción general, De botones de opciones específicas a la ventana, De Menús, De títulos, De Barras de Herramientas, De pie de página, De Encabezados, y De Logos

# Desarrollo del Trabajo

## Diseño de arquitectura

Especificar la decisión relacionada respecto a servidores de datos y aplicación (web)

* Servidores propios,
* Hosting,
* Cloud, otros o mezcla de ellos.

Incluya un esquema que represente la integración de estos elementos, desde el punto de vista físico y lógico. Este diagrama debe ser explicado. Por ejemplo:

Tal como se representa en la **Error! Reference source not found.** la arq. que da soporte al sw “mi software” se divide en 2 servidores físicos, ATLANTA y MARCUS.

EL primero aloja el server web sobre un SO apache xxx, para el sw **mio.midominio.cl** <http://146.83.99.99>, puerto ssh 999 u puerto apache 999. El segundo servidor físico aloja 2 componentes lógicos, servidor de archivos user/carpeta/ y de base de datos con Mysql 192.168.l.0 (ejemplo). Los usuarios locales acceden a través de los sistemas distribuidos por red local y al sistema web vía internet…. etc.

El escenario es distinto si se trabaja con contenedores, si se utilizan servicios web, API, o la nube amazon ws, azure, etc

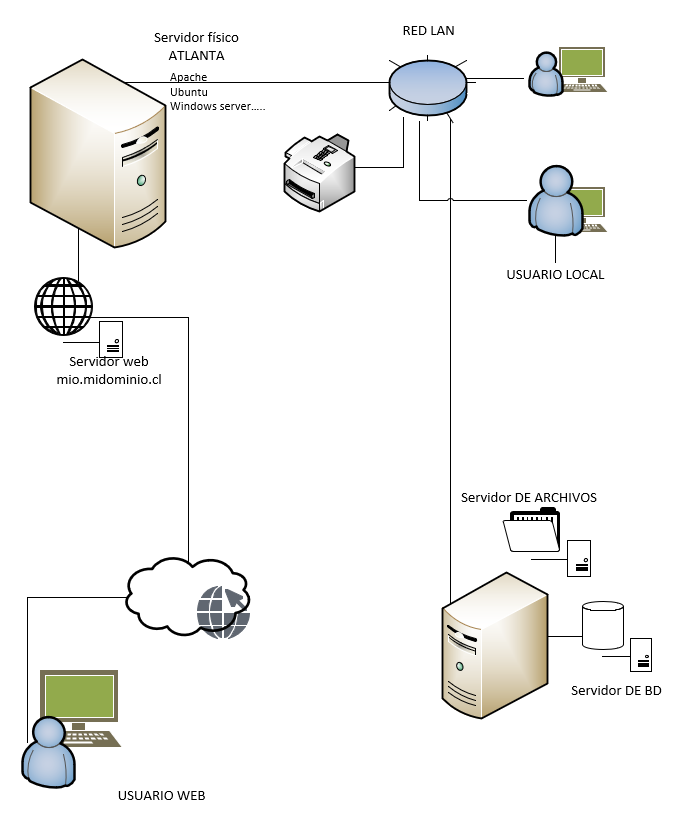


Figura 6: esquema arquitectura de sistemas para software

## Estructura del código

Si se utiliza framework o se programa en lenguaje web puro indicar la forma como se organizan físicamente los archivos y si estos respetan alguna arquitectura de programación como modelo vista controlador, o 3 capas, etc. Incluya una imagen del árbol de directorio.

### Backend

Tabla 17: Estructura de código – Backend ejemplo Tesis Tomas Montecinos IECI

| Directorio | Detalle |
| --- | --- |
| Controllers | Funciones con las que cuenta la aplicación al momento de comunicarse con el servidor de base de datos y viceversa. |
|  |  |
|  |  |

Funcionalidad- Endpoints a utilizar:

1. http://”URLPROYECTO”.cl/”RUTA A UTILIZAR”
   1. Tipo de petición:
      1. “get,post,put,delete”
   2. Parámetros a ingresar, el tipo de datos y restricciones
      1. Nombre
         1. String
         2. Requerido
         3. Tamaño mínimo de 1 caracteres
         4. Tamaño máximo de 100 caracteres
      2. Precio
         1. Number
         2. No requerido - Default 10
2. http://”URLPROYECTO”.cl/”RUTA A UTILIZAR”
   1. Tipo de petición:
      1. “get,post,put,delete”
   2. Parámetros a ingresar, el tipo de datos y restricciones
      1. Nombre
         1. String
         2. Requerido
         3. Tamaño mínimo de 1 caracteres
         4. Tamaño máximo de 100 caracteres
      2. Precio
         1. Number
         2. No requerido - Default 10

### Frontend

Tabla 18: Estructura de código – Frontend ejemplo Tesis Tomas Montecinos IECI

| Directorio | Detalle |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Implantación y puesta en marcha

## Plan de Capacitación/entrenamiento

La capacitación de contenidos y conceptos relacionados al sw, entrenamiento en el uso del software, en la resolución de problemas, por ejemplo. Considerando a los distintos usuarios y su nivel de expertiz. Estas actividades pueden ser online o presencial y requieren que el Sw se encuentre disponible.

Implantación considera el proceso en el que la empresa adopta el sw y la nueva forma de hacer las cosas, existen estrategias revise cuál de ellas y justifique por que será utilizada. Por ejemplo, radical/directa, paralelo, entre otras. Los tipos de implantación son:

* Sistemas paralelos: es el método más seguro, el cual consiste en poner a trabajar los dos sistemas en paralelo, de esta manera los usuarios siguen utilizando el sistema anterior de manera acostumbrada, aunque van teniendo más contacto con el otro. La data va a ser poco a poco migrada de un sistema a otro y sin que el usuario se dé cuenta vamos obligándolo a usar poco a poco más el nuevo sistema. Una de las desventajas es que al estar operando los dos sistemas los costos se duplicaran debido a que pudiera ser que se tenga que contratar personal para que opere los dos sistemas, puede que también el nuevo sistema sea rechazado por los usuarios y se vuelva al sistema anterior.
* Conversión directa: este tipo de conversión se hace de manera radical debido que se hace de un día a otro obligando tanto físico como psicológicamente al usuario que no existe otro sistema y debe usar ese. Esto tiene una desventaja ya que al eliminar por completo el sistema antiguo se quedan sin respaldo, y si el sistema nuevo llegase a tener problemas este quedara parando a la empresa hasta que se solucione, también la empresa se retrasa varias semanas debido que toda la captura de datos debe empezarse de nuevo y los departamentos deben ponerse a trabajar con eso. una vez que empiece este proceso debe seguirse a pesar de las frustraciones que puede haber por cuestión de tiempo perdido. Este método necesita una buena planificación, para que así no exista perdida de ningún tipo.
* Enfoque piloto: este método funciona de la siguiente manera, tenemos el sistema, pero solo se lo aplicamos a un departamento a manera de prueba para así también ir probándolo y mejorándolo una vez capaces de trabajar con él, y saber que el sistema está trabajando en su plenitud y no tiene errores y ha minimizado tareas en ese departamento tanto como costos, tiempo etc. se va a implementar en toda la empresa.
* Modelo por etapas: este método se da debido a la tardanza de la llegada del nuevo sistema que pasara de días a meses y es por eso que solo algunos tendrán acceso a él. Ejemplo: soy un empresario, tengo 15 tiendas de ropa, automatizar a las 15 tiendas es muy costoso y es por eso que la implanto primero en 5 tiendas y luego en el resto.

## Estrategia de implantación.

Preparación de datos / Migración /Poblamiento debe calendarizar esta etapa y documentar el proceso en caso que deba ser repetido.

Puesta en marcha planificar tiempo de monitoreo y la forma como se atenderán las consultas del usuario hasta que finalmente sea liberado el sw.

Todos estos elementos deben ser definido y justificado y luego calendarizado en una Gantt.

## Estado del Proyecto

En que etapa se encuentra, que falta ¿??

# Conclusión del proyecto

* Estado del Proyecto.
* Trabajo Futuro.
* Concluir respecto a el logro de cada uno de los objetivos específicos del proyecto, y por ende se concluye respecto al logro del objetivo general “aún aquellos proyectos que según el capítulo 6 les falta implantación”
* Concluir respecto al tiempo y esfuerzo estimado y real
* Concluir respecto a logro de competencias, desarrollo de competencias del perfil de su carrera (buscar en sitio de la carrera)
* Concluir respecto a percepciones u opiniones personales obtenidas del proyecto

# Referencias

|  |  |
| --- | --- |
| [1] | « Universidad del Bío-Bío, única universidad pública de las regiones del Biobío y Ñuble. Acreditada por la CNA.,» [En línea]. Available: https://ubiobio.cl/. [Último acceso: 12 09 2024]. |
| [2] | «FACULTAD DE CIENCIAS EMPRESARIALES,» REGLAMENTO ACTIVIDAD DE TITULACIÓN CARRERAS DE LA FACULTAD DE CIENCIAS EMPRESARIALES, 13 05 2009. [En línea]. Available: https://ubiobio.cl/miweb/webfile/media/32/Documentos/Reglamento%20Acttividad%20de%20titulaci%C3%B3n.pdf. |
| [3] | «Admisión a la Universidad del Bío-Bío,» [En línea]. Available: https://ubiobio.cl/admision/Todas\_las\_Carreras/21/Ingenieria\_de\_Ejecucion\_en\_Computacion\_e\_Informatica/. [Último acceso: 12 06 2025]. |
| [4] | «Admisión a la Universidad del Bío-Bío,» [En línea]. Available: https://ubiobio.cl/admision/Todas\_las\_Carreras/17/Ingenieria\_Civil\_en\_Informatica\_Concepcion/. [Último acceso: 12 06 2025]. |
| [5] | «¿Qué plan de Trello es perfecto para ti? Averígualo con nuestra guía de precios?,» Trello, [En línea]. Available: https://trello.com/es/pricing. [Último acceso: 15 06 2025]. |
| [6] | «Asana,» Precios de Asana | Planes Personal, Starter, Advanced y Enterprise, [En línea]. Available: https://asana.com/es/pricing. [Último acceso: 15 06 2025]. |
| [7] | «Notion,» Planes y características, [En línea]. Available: https://www.notion.com/es/pricing. [Último acceso: 15 06 2025]. |
| [8] | «Herramientas y recursos para la administración del aula - Google for Education,» Google for Education, [En línea]. Available: https://edu.google.com/intl/es-419\_ALL/workspace-for-education/classroom/. [Último acceso: 12 06 2025]. |
| [9] | «Learning Management System - Moodle LMS - Best LMS Platform,» Moodle. [En línea]. [Último acceso: 12 06 2025]. |
| [10] | «Gestiona los proyectos de tu equipo desde cualquier lugar | Trello,» [En línea]. [Último acceso: 12 06 2025]. |

Anexos de Pruebas de aceptación

Este anexo SE ENTREGA EN LA REVISIÓN DEL SW.

Se prueban todos los requisitos.

Por cada requisito indique los pre-requisitos o configuraciones necesarios para la prueba y luego la tabla con los datos de prueba.

Por ejemplo, Requerimiento REGISTRAR NUEVO PERSONAL. El usuario se logea con la cuenta usuario ENCARGADO DE PERSONAL, rut 11111111-1 contraseña clave 123, y se prueban los casos indicados, en la tabla.

Tabla 19: Detalle de pruebas de requerimiento

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NOMBRE | Rut | FECHA NACTO | CERTNACTO | Resultado esperado, actualización en las tablas | |
| JUAN PEREZ | 12.676436-1 | 01/12/1990 | CERT1.JPG | Trabajador, documentos | Registro exitoso, verificar en la opción ver trabajadores |
| JUAN PEREZ | 12.676436-1 | 01/12/1990 | CERT1.JPG | Trabajador, documentos | Advertencia Trabajador ya existe, se habilita opción de editar |

Anexos de recopilación de información

<Todas las técnicas aplicadas y el respaldo de la información recopilada. Por ejemplo, entrevista, cuestionarios, observación en terreno, revisión de documentación, talleres grupales, etc.

a. En cuestionarios se indica: a quien, para que, cuando se aplicó, y las preguntas. Después se incluyen las tablas de respuestas y resumen de los resultados.

b. En entrevistas se indica: a quien, para que, cuando se aplicó, y preguntas. Después se incluyen las respuestas y firma del cliente.

c. Observación en terreno: donde se realizó, que proceso observó, que usuarios, que información recopiló.

d. Revisión de documentos (internos o externos): que documentos obtuvo, desde donde, cuando los obtuvo, que información útil extrajo.

Toda la información que se recopila de las técnicas debe estar relacionada con el contenido del informe, es decir con la etapa del desarrollo del software. Por ejemplo, si no estamos interesados de los tipos de usuarios del software, no debe preguntar o extraer información al respecto.>

Anexo Diccionario de datos

Entidades/ colecciones: que representan

Atributos: que información contienen, formato, valores por defecto, reglas y validaciones

Anexo aspectos de gestión de proyectos

Anexo Carta Gantt con línea base y desviaciones

Incluir cada Gantt con la explicación del cambio y el efecto en la planificación global

Riesgos de Alto nivel (Amenazas), Impacto, estrategia.

Explique los Riesgos que tengan impacto en su proyecto. Ordene los riesgos y defina las acciones (estrategia)que se proponen para abordarles.

Incluir columna con los riesgos que se presentaron en el proyecto.

Anexo Estimación CU

Estimación de tamaño de Sw: Puntos de Casos de Uso

* Clasificar Actores
* Clasificar casos de uso
* Factores técnicos
* Factores del entorno
* Calcular puntos de Casos de uso

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo de caso de uso | |  |  |
| 5 | Simple | Menos de 5 clases 5 | 3 transacciones o menos |
| 10 | Medio | 5 a 10 clases 10 | 4 a 7 transacciones |
| 15 | Complejo | Más de 10 clases 18 | Más de 7 transacciones |

|  |  |  |
| --- | --- | --- |
| Tipo de actor D | |  |
| 1 | Simple | Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API). |
| 2 | Medio | Otro sistema interactuando a través de un protocolo (ej. TCP/IP) o una persona interactuando a través de una interfaz en modo texto |
| 3 | Complejo | Una persona que interactúa con el sistema mediante una interfaz gráfica (GUI). |

Calcular UUCP (Unadjusted Use Case Point)

UUCP= UAW+UUCW

Calcular TCF (Technical Complexity Factor)

TCF=0.6+(0.01\*TFactor)

Calcular EF (Environmental Factor)

EF=1.4+(-0.03\*EFactor)

UCP = UUCP \* TCF \* EF

|  |  |
| --- | --- |
| Evaluación de relevancia de factores técnicos y ambientales | Valor |
| Irrelevante De | 0 a 2. |
| Medio De | 3 a 4. |
| Esencial | 5 |

Calculate TCF (Technical Complexity Factor)

| Technical Factor | Multiplier | Relevancia percibida | Resultado multiplicación |
| --- | --- | --- | --- |
| Distributed System | 2 |  |  |
| Application performance objectives, in either response or throughput | 1 |  |  |
| End-user efficiency (on-line) | 1 |  |  |
| Complex internal processing | 1 |  |  |
| Reusability, the code must be able to reuse in other applications | 1 |  |  |
| Installation ease | 0,5 |  |  |
| Operational ease, usability | 0,5 |  |  |
| Portability | 2 |  |  |
| Changeability | 1 |  |  |
| Concurrency | 1 |  |  |
| Special security features | 1 |  |  |
| Provide direct access for third parties | 1 |  |  |
| Special user training facilities | 1 |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Environmental Factor | Multiplier | Relevancia percibida | Resultado multiplicación |
| Familiar with Objectory + RUP | 1,5 |  |  |
| Application experience | 0,5 |  |  |
| Object Oriented experience | 1 |  |  |
| Analyst capability | 0,5 |  |  |
| Motivation | 1 |  |  |
| Stable requirements | 2 |  |  |
| Par time workers | -1 |  |  |
| Difficult programming language | -1 |  |  |

Level of Effort. Schneider and Winters, proponen que: Si la suma entre (el número de factores de entorno (F1 a F6) inferiores a 3 y el número de factores de entorno (F7 a F8) superiores a 3).

* es menor o igual a 2 entonces LOE=20,
* es 3 o 4 LOE=28.
* es mayor a 4 reconsiderar el proyecto. Por ejemplo, reducir los riesgos relacionados con los factores de entorno.

Anexo Resumen Esfuerzo

El final de este documento se debe indicar las horas destinadas en realizar cada una de las fases del desarrollo del software, las horas corresponden a la suma de las horas gastadas por cada integrante y del equipo en conjunto.

|  |  |
| --- | --- |
| Actividades/fases/casos de Uso | N° Horas |
| Cuantas horas se dedicaron en) |  |
| Cuantas horas se dedicaron en |  |
| Cuantas horas se dedicaron en |  |
| Cuantas horas se dedicaron en programar |  |
| Cuantas horas se dedicaron en informe completo (preparar y corregir) |  |
| Cuantas horas se dedicaron a git |  |
| TOTAL | 320 |

Resumen de Level of Effort

esfuerzo gastado en meses versus estimación CU, no importa si es distinto (está bien... porque ahora podrán sacar un cálculo real ..... por ejemplo:

puntos de casos de uso ajustados 60 UCP

mi esfuerzo me dio 320 hrs

Entonces 320/60 =5.33 hrsxCU

Anexo Retrospectiva Proyecto

Síntesis del porcentaje de cumplimiento de los requerimientos por cada módulo.

Análisis éxito/fracaso del proyecto

Riesgos que se concretaron en el proyecto y efectos/consecuencias

Análisis de ajuste entre planificación, esfuerzo real gastado y estimación de CU

Compare y analice los resultados extraídos desde los tiempos de la Gantt, de esfuerzo requerido y estimación de CU.

Concluya respecto a los resultados.

Anexo Iteraciones en el desarrollo

Por cada incremento y/o iteración:

Funcionalidad

Fecha

Retroalimentación del cliente/usuario