



UNIVERSIDAD DE LOS LAGOS

# TÍTULO

SUBTÍTULO

DEPARTAMENTO DE CIENCIAS DE LA INGENIERÍA  
INGENIERÍA CIVIL EN INFORMÁTICA  
PROYECTO DE TÍTULO  
CAMPUS ???, CHILE

Autor  
Autor

Docente Guía: Guía  
Docente Guía: Guía

4 de febrero de 2026



**5** años  
UNIVERSIDAD ACREDITADA  
Septiembre de 2021 - Septiembre de 2026  
Gestión Institucional - Docencia de Pregrado  
Investigación - Vinculación con el Medio  
**AVANZADA**

[www.ulagos.cl](http://www.ulagos.cl)

# Agradecimientos

Sección personal donde el autor agradece a las personas e instituciones que apoyaron el desarrollo del trabajo.

## Resumen

Un solo párrafo de máximo 350 palabras que presenta de forma clara y motivadora el proyecto. No debe incluir citas, referencias o conclusiones detalladas. Debe ser preciso:

- El **problema**: ¿Qué necesidad o problema se aborda?
- **Importancia**: ¿Por qué es interesante o relevante resolverlo?
- **Logros y desafíos**: ¿Qué se consiguió y cuáles fueron los principales retos?

**Ejemplo 1.** *El actual proceso de inscripción de asignaturas en la Universidad X es manual y propenso a errores, generando demoras y sobrecarga administrativa. Este proyecto aborda dicha problemática mediante el desarrollo de un sistema web centralizado que automatiza el registro académico. El principal logro es una plataforma funcional que reduce los tiempos de inscripción en un 80 % y garantiza la integridad de los datos. El mayor desafío fue la integración con los sistemas legados de la universidad, superado mediante la creación de una API RESTful.*

**Palabras Clave**— incluir 5, palabras clave, que describan, el proyecto

# Índice general

<b>1. Formulación del Proyecto</b>	<b>2</b>
1.1. Propósito . . . . .	2
1.2. Identificación del Problema . . . . .	2
1.3. Justificación y Aporte . . . . .	2
1.3.1. Utilidad del Sistema . . . . .	3
1.3.2. Aporte del Sistema a ..... . . . .	3
1.4. Marco Teórico . . . . .	3
1.5. Estado del Arte . . . . .	4
1.5.1. Comparativa propuesta con estado del arte . . . . .	4
<b>2. Metodología de Trabajo</b>	<b>5</b>
2.1. Alcance . . . . .	5
2.2. Objetivos . . . . .	5
2.2.1. Objetivo General . . . . .	6
2.2.2. Objetivos Específicos . . . . .	6
2.3. Desglose de Actividades y Tareas . . . . .	6
2.3.1. Objetivo Específico 1: ... . . . .	7
2.4. Planificación Temporal . . . . .	7
2.4.1. Carta Gantt . . . . .	8
2.4.2. Desglose de Actividades . . . . .	9
2.5. Análisis de Viabilidad del Proyecto . . . . .	12
2.5.1. Viabilidad Técnica . . . . .	12
2.5.2. Viabilidad Operativa . . . . .	12
2.5.3. Viabilidad de Plazos (Cronograma) . . . . .	12
2.6. Metodología de Desarrollo ( <b>OBLIGATORIO</b> ) . . . . .	12
<b>3. Descripción General del Sistema</b>	<b>13</b>
3.1. Perspectiva del Producto . . . . .	13
3.2. Funciones del Sistema . . . . .	13
3.3. Características de los Usuarios . . . . .	13
3.4. Restricciones . . . . .	13
3.5. Suposiciones y Dependencias . . . . .	14

3.5.1. Suposiciones . . . . .	14
3.5.2. Dependencias . . . . .	14
<b>4. Requisitos del Proyecto</b>	<b>15</b>
4.1. Requisitos de Interfaces Externos ( <b>OPCIONAL</b> ) . . . . .	15
4.1.1. Interfaces de Usuario . . . . .	15
4.1.2. Interfaces de Hardware . . . . .	15
4.1.3. Interfaces de Software . . . . .	15
4.1.4. Interfaces de Comunicación . . . . .	15
4.2. Requisitos Funcionales ( <b>Obligatorio</b> ) . . . . .	15
4.2.1. Requisito funcional x . . . . .	16
4.2.2. Requisito funcional y . . . . .	16
4.3. Requisitos de Rendimiento ( <b>Opcional</b> ) . . . . .	16
4.4. Requisitos de Desarrollo ( <b>Opcional</b> ) . . . . .	16
4.5. Requisitos de Tecnológicos ( <b>Obligatorio</b> ) . . . . .	16
4.6. Seguridad ( <b>Obligatorio</b> ) . . . . .	16
4.7. Modelo de Datos ( <b>Obligatorio</b> ) . . . . .	16
<b>5. Análisis Económico</b>	<b>19</b>
5.1. Estudio de Mercado . . . . .	19
5.1.1. Consumidores y análisis de la demanda . . . . .	19
5.1.2. Análisis de la oferta y los precios . . . . .	19
5.1.3. Comercialización . . . . .	19
5.2. Evaluación de Proyectos de Inversión . . . . .	20
5.2.1. Criterios de evaluación . . . . .	20
5.2.2. Valor Actual Neto (VAN) . . . . .	20
5.2.3. Tasa Interna de Retorno (TIR) . . . . .	20
5.2.4. Análisis de riesgo e incertidumbre . . . . .	20
5.3. Viabilidad Económica . . . . .	20
5.3.1. Costos de Desarrollo . . . . .	20
5.3.2. Costos Operativos (Post-Implementación) . . . . .	21
5.3.3. Retorno de la Inversión (ROI) y Beneficios . . . . .	21
<b>6. Desarrollo</b>	<b>23</b>
6.1. Lenguaje de Programación Elegido . . . . .	23
6.1.1. Propiedades . . . . .	23
6.1.2. Ventajas y Desventajas . . . . .	23
6.2. Propuesta de Solución . . . . .	23
<b>7. Aseguramiento de la Calidad (<b>Opcional</b>)</b>	<b>24</b>
7.1. Modelo . . . . .	24
7.2. Técnica . . . . .	24
7.3. Implementación . . . . .	24

<b>8. Proceso de Prueba del Software (Opcional)</b>	<b>25</b>
8.1. Criterios de Prueba . . . . .	25
8.1.1. Enfoque de Prueba de Caja Negra . . . . .	25
8.1.2. Enfoque de Prueba de Caja Blanca . . . . .	25
8.2. Prueba de Unidad . . . . .	25
8.3. Prueba de Integración . . . . .	25
8.3.1. Integración Descendente . . . . .	26
8.3.2. Integración Ascendente . . . . .	26
8.3.3. Prueba de Regresión . . . . .	26
8.4. Prueba de Sistema . . . . .	26
8.4.1. Prueba de Recuperación . . . . .	26
8.4.2. Prueba de Seguridad . . . . .	26
8.4.3. Prueba de Resistencia . . . . .	26
8.4.4. Prueba de Rendimiento . . . . .	26
8.5. Prueba de Aceptación . . . . .	26
<b>9. Conclusión</b>	<b>27</b>
9.1. Principales aportes . . . . .	27
9.2. Contraste de resultados . . . . .	27
9.3. Trabajos futuros . . . . .	27
<b>A. Definciones, Acronimos y Abreviaturas</b>	<b>29</b>
A.1. Definiciones . . . . .	29
A.2. Acrónimos . . . . .	29
A.3. Abreviaturas . . . . .	29
<b>B. Configuraciones</b>	<b>30</b>
<b>C. Ejemplos de Usos de <math>\text{\LaTeX}</math></b>	<b>31</b>
C.1. Códigos de Programación . . . . .	31
C.2. Entornos . . . . .	35
C.3. Tablas . . . . .	37
C.4. Formulas Matemáticas . . . . .	38
C.5. Items, Descripciones y Enumeraciones . . . . .	38
C.6. Figuras . . . . .	39

# Índice de figuras

2.1. Carta Gantt del Proyecto XYZ . . . . .	8
2.2. Duración de tareas y dependencias . . . . .	9
2.3. Grafo de Actividades del Proyecto XYZ . . . . .	10
2.4. Grafo de Actividades con duración del Proyecto XYZ . . . . .	11
2.5. Cálculo del diagrama de actividades . . . . .	12
4.1. Modelo Entidad Relación . . . . .	17
4.2. Modelo Relación Relacional . . . . .	18
5.1. Grafo de Actividades con duración y caminos críticos . . . . .	22
C.1. Logo Universidad de Los Lagos . . . . .	39
C.2. Insertar subfiguras en $\LaTeX$ . . . . .	39
C.3. Foto entre texto . . . . .	39

# Índice de tablas

C.1. Información General de GNU/Linux . . . . .	37
C.2. Ejemplo tabla con barras . . . . .	37
C.3. Ejemplo tabla con barras . . . . .	38
C.4. Títulos de varias columnas y filas . . . . .	38



# Índice de algoritmos

C.1. Código en C de una sumatoria . . . . .	31
C.2. Ejemplo de SQL . . . . .	31
C.3. Código LISP de una Lista . . . . .	34
C.4. Código PROLOG de un árbol genealógico . . . . .	35
C.5. Código JAVA de una clase . . . . .	35

INDICACIONES BÁSICAS:

- CADA IMÁGEN O LOGO, TABLA Y ALGORITMO DEBEN TENER CAPTION (PIE DE...) Y DEBEN SER REFERENCIADAS EN EL TEXTO.
- CADA CITE DEBE SER ANTES DE UN PUNTO.

# Capítulo 1

## Formulación del Proyecto

Esta sección es la introducción formal del proyecto. Debe guiar al lector desde el contexto general hasta el problema específico, justificando la necesidad del trabajo y presentando la solución propuesta.

Introduce la terminología, destaca las contribuciones del documento y da una breve descripción de la organización de éste

### 1.1. Propósito

Contextualiza el proyecto en relación con otros trabajos o investigaciones existentes. Señala qué lo hace diferente o cómo se basa en investigaciones previas. Largo aproximado: 2 páginas.

**Ejemplo 2.** *Mientras que existen múltiples sistemas de gestión académica, la mayoría son soluciones genéricas. Este proyecto se diferencia al proponer un sistema diseñado específicamente para los flujos de trabajo del Departamento de Ciencias de la Ingeniería de la Universidad de Los Lagos, integrando funcionalidades únicas de seguimiento de tesis que no se encuentran en alternativas comerciales.*

### 1.2. Identificación del Problema

Define clara y detalladamente el problema que el proyecto busca resolver. Expone las deficiencias de los métodos actuales y por qué es necesario encontrar una nueva solución.

### 1.3. Justificación y Aporte

Argumenta por qué el proyecto es importante y conveniente. Responde a preguntas como: ¿Para qué sirve? , ¿quiénes se benefician? , ¿resuelve un problema práctico?

### **1.3.1. Utilidad del Sistema**

Explica cómo el sistema mejorará procesos como la centralización de información o la optimización de tareas.

Los siguientes subtítulos son algunas ideas:

**Centralización e Integración de la Información**

**Optimización de Procesos**

**Facilitación del Acceso a la Información**

**Soporte para la Generación de Informes y Estadísticas**

### **1.3.2. Aporte del Sistema a .....**

Detalla los beneficios institucionales, como la mejora en la toma de decisiones o la modernización tecnológica.

Los siguientes subtítulos son algunas ideas:

**Mejora de la Eficiencia Operativa**

**Fortalecimiento de la Toma de Decisiones Estratégicas**

**Impulso a la Gestión**

**Mejora en la Gestión y Apoyo**

**Incremento de la Transparencia y Rendición de Cuentas**

**Modernización Tecnológica y Base para el Futuro**

## **1.4. Marco Teórico**

Presenta la base teórica que sustenta el proyecto. Aquí se definen los conceptos, modelos y tecnologías fundamentales que se utilizarán para interpretar los resultados. Ayuda a centrar la investigación y a prevenir errores. Se construye a partir de una revisión bibliográfica de fuentes primarias, secundarias y terciarias.

- **Primarias (directas):** Libros, artículos, antologías, tesis, disertaciones, entre otros.
- **Secundarias:** Compilaciones, resúmenes de listados de referencias publicadas en un área en particular, bases de datos.
- **Terciarias:** Documentos que reúnen nombres y títulos de revistas y otras publicaciones.

Ayuda a prevenir errores que se han cometido en otros estudios, conduce al establecimiento de la hipótesis o afirmaciones que se someterán a prueba.

**Ejemplo 3.** *Para el desarrollo de este proyecto, se aplicará la metodología de desarrollo ágil Scrum, ya que permite una adaptación continua a los requisitos cambiantes. El sistema se fundamentará en una arquitectura de microservicios para garantizar la escalabilidad. La base de datos utilizará el modelo relacional, específicamente la Tercera Forma Normal (3FN), para asegurar la integridad y evitar la redundancia de datos...*

## **1.5. Estado del Arte**

Investiga y describe las soluciones, herramientas o tecnologías existentes que ya intentan resolver el mismo problema o uno similar. Al final, se debe incluir una comparativa directa entre esas soluciones y la propuesta del proyecto, destacando las ventajas y novedades de esta última.

### **1.5.1. Comparativa propuesta con estado del arte**

## Capítulo 2

# Metodología de Trabajo

Detalla el plan y la estrategia que se seguirán para desarrollar el proyecto, desde su justificación hasta la asignación de recursos.

### 2.1. Alcance

Define con precisión los límites del proyecto. Debe quedar claro qué hará y qué no hará el software.

- Identificar el producto por su nombre.
- Describir la aplicación, sus beneficios y objetivos.

Esta subdivisión debe:

1. Identifique el producto del software para ser diseñado por el nombre (por ejemplo, Anfitrión DBMS, el Generador del Reporte, etc.);
2. Explique eso que el producto (del software hará y que no hará.
3. Describe la aplicación del software especificándose los beneficios pertinentes, objetivos, y metas;
4. Sea consistente con las declaraciones similares en las especificaciones de niveles superiores (por ejemplo, las especificaciones de los requisitos del sistema), si ellos existen.

### 2.2. Objetivos

Expresan los fines concretos del proyecto. Deben ser medibles, factibles y comenzar con un verbo en infinitivo (ej. Analizar, Diseñar, Implementar)

### 2.2.1. Objetivo General

Declara el propósito principal y final del proyecto en una sola frase.

**Ejemplo 4.** *Desarrollar e implementar un sistema de base de datos web integral y eficiente que satisfaga los requisitos de gestión de información académica y de investigación de [Nombre del Cliente o Empresa Cliente], permitiendo la administración centralizada y estructurada de datos relativos a profesores, alumnos, proyectos, departamentos y asignaturas, y facilitando el acceso y la operatividad a los usuarios designados.*

### 2.2.2. Objetivos Específicos

Son los pasos o hitos necesarios para alcanzar el objetivo general.

- Ejemplo 5.**
- 1. Formulación y Modelado Conceptual:** *Analizar exhaustivamente los requisitos funcionales y de datos proporcionados por el cliente, para formular el presente informe técnico y diseñar un modelo de datos conceptual y lógico que represente fielmente las entidades, atributos y relaciones identificadas, sirviendo como base para la estructura de la base de datos.*
  - 2. Diseño e Implementación de la Base de Datos:** *Traducir el modelo de datos lógico a un diseño físico de base de datos optimizado, seleccionando un sistema gestor de base de datos (SGBD) adecuado, y proceder con la creación de la estructura de la base de datos, incluyendo tablas, relaciones, restricciones de integridad y los índices necesarios para asegurar la eficiencia y consistencia de los datos.*
  - 3. Diseño y Desarrollo del Sistema Web con Integración de Datos:** *Diseñar la arquitectura del sistema web y desarrollar una interfaz de usuario intuitiva y funcional que permita la interacción con la base de datos implementada. Esto incluye la implementación de los procesos CRUD (Crear, Leer, Actualizar, Eliminar) para todas las entidades gestionadas, asegurando una correcta integración entre la capa de presentación y la capa de datos.*
  - 4. Pruebas, Validación y Despliegue:** *Ejecutar un plan de pruebas exhaustivo que abarque pruebas unitarias, de integración, de sistema y de aceptación del usuario, con el fin de validar el correcto funcionamiento del sistema web y la base de datos, asegurar el cumplimiento de los requisitos iniciales, corregir posibles errores y preparar el sistema para su despliegue en el entorno productivo del cliente.*

## 2.3. Desglose de Actividades y Tareas

Por cada objetivo específico, se listan las actividades y tareas concretas que se realizarán para cumplirlo.

### **Ejemplo 6. 2.3.1. Objetivo Específico 1: ...**

#### **Actividades y Tareas**

*1. Actividad 1:*

*1.1 Tarea 1:*

*1.2 Tarea 2:*

*2. Actividad 2:*

*2.1 Tarea 1:*

*2.2 Tarea 2:*

## **2.4. Planificación Temporal del Proyecto**

Presenta el cronograma del proyecto.



### 2.4.1. Carta Gantt

Diagrama visual de las actividades en el tiempo. Si el proyecto es ágil no se recomienda usar Gantt.

Figura 2.1: Carta Gantt del Proyecto XYZ

### 2.4.2. Desglose de Actividades

En esta sección presentan las actividades de la Sección 2.3 con su duración, dependencias, caminos críticos, entre otras y se debe dar una conclusión de lo mismo.

Actividad	Duración	Después de	Simultanea	Antes de

Figura 2.2: Duración de tareas y dependencias

Figura 2.3: Grafo de Actividades del Proyecto XYZ

Figura 2.4: Grafo de Actividades con duración del Proyecto XYZ

Actividad	Duración	Inicio		Termino		Holgura Total	Crítico
		Temprano	Tardío	Temprano	Tardío		

Figura 2.5: Cálculo del diagrama de actividades

## 2.5. Análisis de Viabilidad del Proyecto

Evalúa si el proyecto puede llevarse a cabo con los recursos y plazos disponibles.

### 2.5.1. Viabilidad Técnica

¿Se cuenta con la tecnología y el conocimiento necesarios?  
Los siguientes subtítulos son algunas ideas:

**Tecnologías Maduras y Disponibles**

**Complejidad Manejable**

**Recursos Humanos Calificados**

**Infraestructura Requerida**

**Escalabilidad y Rendimiento**

### 2.5.2. Viabilidad Operativa

¿El sistema será aceptado y utilizado por los usuarios?

### 2.5.3. Viabilidad de Plazos (Cronograma)

¿El cronograma propuesto es realista?

## 2.6. Metodología de Desarrollo (**OBLIGATORIO**)

Aquí se debe explicar en detalle la metodología de desarrollo elegida (ej. Scrum, Kanban, XP, Cascada) y justificar por qué es la más adecuada para este proyecto en particular.

## Capítulo 3

# Descripción General del Sistema

Ofrece una vista general del producto de software, sus funciones, usuarios y las condiciones bajo las cuales operará.

### 3.1. Perspectiva del Producto

Describe el producto en relación con otros sistemas y el entorno en el que funcionará.

### 3.2. Funciones del Sistema

Acá se describen las partes más relevantes que tendrá el sistema.

### 3.3. Características de los Usuarios

Define los diferentes tipos de usuarios que interactuarán con el sistema (ej. administrador, usuario final, etc.) y sus niveles de conocimiento técnico.

### 3.4. Restricciones

Enumera las limitaciones o restricciones que afectan el desarrollo, como el uso de un sistema operativo específico, políticas de la empresa o un presupuesto limitado.

## **3.5. Suposiciones y Dependencias**

### **3.5.1. Suposiciones**

Factores que se consideran verdaderos pero que no se pueden verificar al inicio del proyecto (ej. "Se asume que la universidad proporcionará acceso a la API del sistema de matrículas")

### **3.5.2. Dependencias**

Factores externos que el proyecto necesita para poder completarse (ej. "El proyecto depende de la compra de una licencia para el motor de base de datos Oracle").

# Capítulo 4

## Requisitos del Proyecto

Especifica en detalle qué debe hacer el sistema. Es recomendable complementar esta sección con diagramas UML para mayor claridad.

### 4.1. Requisitos de Interfaces Externos (**OPCIONAL**)

Detalla cómo el sistema interactuará con usuarios y otros sistemas.

#### 4.1.1. Interfaces de Usuario

#### 4.1.2. Interfaces de Hardware

#### 4.1.3. Interfaces de Software

#### 4.1.4. Interfaces de Comunicación

### 4.2. Requisitos Funcionales (**Obligatorio**)

Describen las funciones que el sistema debe ser capaz de realizar.

**Ejemplo 7.**    ■ **RF-01:** *El sistema debe permitir al administrador registrar nuevos usuarios.*

■ **RF-02:** *El sistema debe generar un reporte de inscripciones en formato PDF.*



#### 4.2.1. Requisito funcional x

#### 4.2.2. Requisito funcional y

### 4.3. Requisitos de Rendimiento (**Opcional**)

Especifica qué tan bien debe funcionar el sistema en términos de velocidad, capacidad de respuesta y uso de recursos.

**Ejemplo 8.** *El sistema debe cargar la página principal en menos de 2 segundos con 100 usuarios concurrentes.*

### 4.4. Requisitos de Desarrollo (**Opcional**)

Impone restricciones sobre el proceso de construcción del sistema, como el uso de un estándar de codificación específico o la obligación de usar un sistema de control de versiones.

### 4.5. Requisitos de Tecnológicos (**Obligatorio**)

Detalla las tecnologías específicas (lenguajes, frameworks, bases de datos) que se utilizarán.

**Ejemplo 9.**     ■ *RT-01: El backend se desarrollará en Python con el framework Django.*

■ *RT-02: La base de datos a utilizar será PostgreSQL.*

### 4.6. Seguridad (**Obligatorio**)

Define los requisitos para garantizar la seguridad del sistema y sus datos.

**Ejemplo 10.**     ■ *RS-01: Todas las contraseñas de usuario deben almacenarse de forma cifrada*

### 4.7. Modelo de Datos (**Obligatorio**)

Describe la estructura de los datos. Se debe incluir un Modelo Entidad-Relación y el Modelo Relacional normalizado. Si se usan bases de datos no relacionales, se debe mostrar la estructura (ej. JSON).

Figura 4.1: Modelo Entidad Relación

Figura 4.2: Modelo Relación Relacional

# Capítulo 5

## Análisis Económico

**Juaramir:** Este capítulo es obligatorio para todo proyecto que implique desarrollo de software y no sea de investigación **END.**

Este capítulo evalúa la viabilidad del proyecto desde una perspectiva de negocio y financiera. Es especialmente útil si el software tiene un potencial comercial.

### 5.1. Estudio de Mercado

Valida la idea del software como un producto viable en el mercado. Incluye un análisis de los posibles consumidores (demanda), la competencia (oferta y precios) y la estrategia de comercialización.

#### 5.1.1. Consumidores y análisis de la demanda

Identificar el público objetivo, estimar el tamaño del mercado potencial y proyectar la demanda futura del software. Esto es fundamental para justificar que el proyecto es necesario.

#### 5.1.2. Análisis de la oferta y los precios

Investigar a la competencia (otras soluciones de software) y definir un modelo de negocio y precios (ej. licencia, suscripción, freemium)

#### 5.1.3. Comercialización

Describir cómo se dará a conocer y se distribuirá el software a los clientes

## **5.2. Evaluación de Proyectos de Inversión**

Utiliza indicadores financieros para medir la rentabilidad del proyecto.

### **5.2.1. Criterios de evaluación**

Utilizar los flujos de caja para calcular indicadores clave de rentabilidad

### **5.2.2. Valor Actual Neto (VAN)**

Determinar si el proyecto crea valor, es decir, si los beneficios futuros superan la inversión inicial, a una tasa de descuento definida.

### **5.2.3. Tasa Interna de Retorno (TIR)**

Calcular la rentabilidad intrínseca del proyecto para compararla con el costo de oportunidad del capital

### **5.2.4. Análisis de riesgo e incertidumbre**

Plantear diferentes escenarios (optimista, pesimista) para evaluar la robustez financiera del proyecto de software ante cambios en la demanda o en los costos

## **5.3. Viabilidad Económica**

Detalla todos los costos asociados al proyecto y los beneficios esperados.

### **5.3.1. Costos de Desarrollo**

Inversión inicial en personal, hardware y software.

#### **Personal**

#### **Software**

Estimar la infraestructura necesaria (servidores, capacidad de base de datos) en función de la demanda de usuarios proyectada

## **Hardware/Infraestructura**

Cuantificar la inversión inicial en hardware (computadores, servidores), software (licencias de sistemas operativos, IDEs) y otros activos

### **5.3.2. Costos Operativos (Post-Implementación)**

Estimar los costos recurrentes como salarios del equipo de desarrollo y mantención, costos de hosting, dominios, soporte técnico, etc.

- Mantenimiento del software y la base de datos.
- Soporte técnico a usuarios.
- Costos continuos de infraestructura (energía, conectividad, cloud si aplica).
- Capacitación continua.

### **5.3.3. Retorno de la Inversión (ROI) y Beneficios**

Cuantifica el retorno financiero y describe los beneficios cualitativos (ej. mejora de la eficiencia, reducción de errores). Aunque muchos beneficios son cualitativos, su impacto económico es considerable:

**Reducción de Costos Directos**

**Eficiencia Mejorada**

**Mejora en la Toma de Decisiones**

**Cumplimiento y Acreditación**

**Reducción de Errores**

Aquí hay que añadir secciones según materia vista en el curso de formulación y evaluación de proyectos.

deben añadir análisis económico de carta planificación temporal como la gantt (Figura 2.1), y el desglose de actividades de la Sección 2.4.2 y malla CPM (Figura 5.1)

Figura 5.1: Grafo de Actividades con duración y caminos críticos

# Capítulo 6

## Desarrollo

En este capítulo se añade todo el desarrollo del proyecto; pueden añadir capítulos extra.

### 6.1. Lenguaje de Programación Elegido

Justifica la elección del lenguaje principal, mencionando sus propiedades, ventajas y desventajas en el contexto del proyecto.

#### 6.1.1. Propiedades

#### 6.1.2. Ventajas y Desventajas

### 6.2. Propuesta de Solución

Explica la arquitectura del sistema y cómo los componentes interactúan para resolver el problema. Se puede apoyar con diagramas de arquitectura.



## Capítulo 7

# Aseguramiento de la Calidad (**Opcional**)

Define el plan y las técnicas que se utilizarán para garantizar que el software cumpla con los estándares de calidad definidos. Incluye el modelo de calidad, las técnicas a aplicar y cómo se implementarán.

### 7.1. Modelo

### 7.2. Técnica

### 7.3. Implementación

## Capítulo 8

# Proceso de Prueba del Software (Opcional)

Describe la estrategia de pruebas para verificar que el software funciona correctamente y sin errores. Detalla los diferentes niveles de prueba que se aplicarán.

### 8.1. Criterios de Prueba

Define los enfoques, como pruebas de Caja Negra (sin ver el código) o Caja Blanca (conociendo el código).

#### 8.1.1. Enfoque de Prueba de Caja Negra

#### 8.1.2. Enfoque de Prueba de Caja Blanca

### 8.2. Prueba de Unidad

Probar componentes individuales.

### 8.3. Prueba de Integración

Probar cómo interactúan los componentes entre sí.

### **8.3.1. Integración Descendente**

### **8.3.2. Integración Ascendente**

### **8.3.3. Prueba de Regresión**

## **8.4. Prueba de Sistema**

Probar el sistema completo contra los requisitos.

### **8.4.1. Prueba de Recuperación**

### **8.4.2. Prueba de Seguridad**

### **8.4.3. Prueba de Resistencia**

### **8.4.4. Prueba de Rendimiento**

## **8.5. Prueba de Aceptación**

Realizada por el usuario final para validar que el sistema cumple sus necesidades

## Capítulo 9

# Conclusión

Cierra el informe destacando los resultados obtenidos y los principales hallazgos relacionándolo a los objetivos específicos.

### 9.1. Principales aportes

Resume las contribuciones más importantes del proyecto.

### 9.2. Contraste de resultados

Compara los resultados obtenidos con los objetivos planteados al inicio, evaluando el grado de cumplimiento.

### 9.3. Trabajos futuros

Menciona posibles mejoras, nuevas funcionalidades o líneas de investigación que podrían continuar a partir de este proyecto.

# Bibliografía

## **Anexos A**

# **Definiciones, Acrónimos y Abreviaturas**

Funciona como un glosario para que cualquier lector pueda entender la terminología específica, los acrónimos (ej. UML, API) y las abreviaturas usadas en el informe.

### **A.1. Definiciones**

### **A.2. Acrónimos**

### **A.3. Abreviaturas**

## **Anexos B**

## **Configuraciones**

## Anexos C

# Ejemplos de Usos de L<sup>A</sup>T<sub>E</sub>X

### C.1. Códigos de Programación

En esta sección se presenta la inserción del Código C.1, El Código C.1. El Código C.4. El Código C.3.

Código C.1: Código en C de una sumatoria

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 /* Algoritmo para realizar la sumatoria */
4 /* S = 2+4+6+...+2n */
5
6 int main(void){
7     int i,s,n;
8
9     /* inicializar el valor de la sumatoria en 0 */
10    s=0;
11    printf("ingrese la cantidad de elementos de la sumatoria=");
12    scanf("%d", &n);
13    /* Realiza la iteracion n veces , y el indice "i" lo multiplica ←
        por */
14    /* 2 y lo va sumando a s */
15    for(i=1;i<=n;i++){
16        s = s+ 2*i;
17    }
18    printf("el resultado de la sumatoria es=%d\n",s);
19
20    return (0);
21 }
```

Código C.2: Ejemplo de SQL

```

1 -- Database : acuario
```



```
2
3  -- DROP DATABASE  acuario ;
4
5 CREATE DATABASE  acuario
6   WITH OWNER = postgres;
7
8
9 CREATE TABLE  especies(
10     sno integer PRIMARY KEY,
11     nombre character varying(20),
12     alimento character varying(20)
13 );
14
15 CREATE TABLE  tanques(
16     tno integer PRIMARY KEY,
17     nombre_tanque character varying(20),
18     color_tanque character varying(20),
19     volumen integer NOT NULL
20 );
21
22 CREATE TABLE  peces(
23     pno integer PRIMARY KEY,
24     nombre_peces character varying(20),
25     color_peces character varying(20),
26     tno integer NOT NULL,
27     sno integer NOT NULL,
28     FOREIGN KEY (tno) REFERENCES tanques (tno) ON UPDATE CASCADE ↵
29     ON DELETE CASCADE,
30     FOREIGN KEY (sno) REFERENCES especies (sno) ON UPDATE CASCADE↵
31     ON DELETE CASCADE
32 );
33
34 CREATE TABLE  eventos(
35     eno integer PRIMARY KEY,
36     pno integer NOT NULL,
37     fecha date,
38     FOREIGN KEY (pno) REFERENCES peces (pno) ON UPDATE CASCADE ON↵
39     DELETE CASCADE
40 );
```

```
41 INSERT INTO especies VALUES(17, 'delfin', 'arenque');
42 INSERT INTO especies VALUES(22, 'tiburon', 'cualquier cosa');
43 INSERT INTO especies VALUES(74, 'olomina', 'gusano');
44 INSERT INTO especies VALUES(93, 'ballena', 'mantequilla de mani');
45 INSERT INTO especies VALUES(100, 'pez espada', 'gusano');
46 INSERT INTO especies VALUES(120, 'pez globo', 'gusano');
47
48 -- select * from especies
49
50 INSERT INTO tanques VALUES(55, 'charco', 'verde', 300);
51 INSERT INTO tanques VALUES(42, 'letrina', 'azul', 100);
52 INSERT INTO tanques VALUES(35, 'laguna', 'rojo', 400);
53 INSERT INTO tanques VALUES(85, 'letrina', 'azul', 100);
54 INSERT INTO tanques VALUES(38, 'playa', 'azul', 200);
55 INSERT INTO tanques VALUES(44, 'laguna', 'verde', 200);
56
57 -- select * from tanques
58
59
60 INSERT INTO peces VALUES (164, 'charlie', 'naranja', 42, 74);
61 INSERT INTO peces VALUES (347, 'flipper', 'negro', 35, 17);
62 INSERT INTO peces VALUES (228, 'killer', 'blanco', 42, 22);
63 INSERT INTO peces VALUES (281, 'albert', 'rojo', 55, 17);
64 INSERT INTO peces VALUES (119, 'bonnie', 'azul', 42, 22);
65 INSERT INTO peces VALUES (388, 'cory', 'morado', 35, 93);
66 INSERT INTO peces VALUES (700, 'maureen', 'blanco', 44, 100);
67 INSERT INTO peces VALUES (800, 'beni', 'rojo', 55, 17);
68 INSERT INTO peces VALUES (900, 'nemo', 'rojo', 44, 74);
69 INSERT INTO peces VALUES (150, 'vicky', 'rojo', 55, 100);
70 INSERT INTO peces VALUES (160, 'mati', 'amarillo', 42, 100);
71 INSERT INTO peces VALUES (110, 'rafa', 'azul', 85, 100);
72 INSERT INTO peces VALUES (222, 'jimmy', 'amarillo', 38, 100);
73 INSERT INTO peces VALUES (144, 'bisho', 'rojo', 42, 93);
74 INSERT INTO peces VALUES (125, 'chris', 'azul', 38, 93);
75 INSERT INTO peces VALUES (183, 'sable', 'amarillo', 44, 93);
76 INSERT INTO peces VALUES (241, 'taz', 'rojo', 55, 93);
77 INSERT INTO peces VALUES (300, 'baltazar', 'azul', 85, 100);
78 INSERT INTO peces VALUES (200, 'cash', 'azul', 85, 100);
79 INSERT INTO peces VALUES (424, 'bandido', 'verde', 35, 100);
80 INSERT INTO peces VALUES (454, 'romo', 'blanco', 85, 93);
81
82
```

```

83 -- select * from peces
84
85 INSERT INTO eventos VALUES
86 (3456 , 347 , '2010-01-26'),
87 (6653 , 164 , '2010-05-14'),
88 (5644 , 347 , '2010-05-15'),
89 (5645 , 347 , '2010-05-30'),
90 (6789 , 281 , '2010-04-30'),
91 (5211 , 228 , '2010-08-20'),
92 (6719 , 700 , '2010-10-22'),
93 (4555 , 164 , '2011-11-03'),
94 (9647 , 281 , '2011-12-06'),
95 (5347 , 281 , '2011-01-01');
96
97 -- INSERT INTO eventos VALUES (3456 , 164 , '2010-01-26');
98 -- INSERT INTO eventos VALUES (6653 , 347 , '2010-05-14');
99 -- INSERT INTO eventos VALUES (5644 , 347 , '2010-05-15');
100 -- INSERT INTO eventos VALUES (5645 , 347 , '2010-05-30');
101 -- INSERT INTO eventos VALUES (6789 , 228 , '2010-04-30');
102 -- INSERT INTO eventos VALUES (5211 , 119 , '2010-08-20');
103 -- INSERT INTO eventos VALUES (6719 , 388 , '2010-10-22');
104 -- INSERT INTO eventos VALUES (4555 , 164 , '2011-11-03');
105 -- INSERT INTO eventos VALUES (9647 , 281 , '2011-12-21');
106 -- INSERT INTO eventos VALUES (5369 , 281 , '2011-01-01');
107
108
109 -- ALTER TABLE tanques ADD medida character varying(2);
110
111 -- UPDATE tanques SET medida = 'ml';
112
113 -- select * from tanques;
114
115 -- ALTER TABLE tanques DROP medida;
116
117 -- SELECT * FROM especies;
118 -- SELECT * FROM tanques;

```

### Código C.3: Código LISP de una Lista

```

1 (define (length x)
2   (if (list? x) (length-aux x)
3       (error "x no es una lista")))
4
5 (define (length-aux x)

```

```
6 | (if (null? x) 0 (+1 (length-aux (cdr x)))))
```

#### Código C.4: Código PROLOG de un árbol genealógico

```
1 | % Arbol genealogico version 1.
2 | % padre(A,B) significa que B es el padre de A.
3 |
4 | padre(juan,alberto).
5 | padre(luis,alberto).
6 | padre(alberto,leoncio).
7 | padre(geronimo,leoncio).
8 | padre(luisa,geronimo).
9 |
10 | % Ahora se define las condiciones para que dos individuos sean ↵
    hermanos hermano(A,B), significa que A es hermano de B.
11 | hermano(A,B) :-
12 |     padre(A,P),
13 |     padre(B,P),
14 |     A \== B.
15 | % Ahora se define el parentesco abuelo - nieto. nieto(A,B) ↵
    significa que A es nieto de B.
16 | nieto(A,B) :-
17 |     padre(A,P),
18 |     padre(P,B).
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

#### Código C.5: Código JAVA de una clase

```
1 | class <Nombre>{
2 |     public static void main(String[] args){
3 |         instrucciones;
4 |     }
5 | }
```

## C.2. Entornos

Ahora presentamos los entornos Ejemplo 11, Solución 1, Prueba 1, Definición 1, Demostración 1, Observación 1.

**Ejemplo 11.** *Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra*

*metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.*

**Solución 1.** *Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.*

**Definición 1.** *Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.*

**Prueba 1.** *Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.*

**Demostración 1.** *Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh*

*mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.*

**Observación 1.** *Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.*

### C.3. Tablas

Por otro lado también se muestra un ejemplo de Tabla C.1 llenada, además de la Tabla C.2 que tiene 3 columnas y filas con barras, la Tabla C.3 que tiene 4 filas y 4 columnas sin barras, por último la Tabla C.4 posee títulos que usan más de una columna y fila.

Información general	
<b>Modelo de desarrollo</b>	desarrollo Software libre y código abierto
<b>Última versión estable</b>	Kernel: 4.11.3 (info) 25 de mayo de 2017 (10 días)
<b>Última versión en pruebas</b>	4.12.rc2 (info) 22 de mayo de 2017 (13 días)
<b>Escrito en</b>	C
<b>Núcleo</b>	Núcleo Linux
<b>Plataformas soportadas</b>	DEC Alpha, ARM, AVR32, Blackfin, ETRAX CRIS, FR-V, H8/300, Itanium, M32R, m68k, Microblaze, MIPS, MN103, PA-RISC, PowerPC, s390, S+core, SuperH, SPARC, TILE64, Unicore32, x86, Xtensa
<b>Licencia</b>	GNU General Public License y otras
<b>Estado actual</b>	En desarrollo
<b>En español</b>	Sí

Tabla C.1: Información General de GNU/Linux


Tabla C.2: Ejemplo tabla con barras

1	2	3	4
5	6	7	8
9	10	11	12

Tabla C.3: Ejemplo tabla con barras

S	SCT	Asignatura	Total Horas		Previatura
			TP	TA	
	a	b	c	d	r

Tabla C.4: Títulos de varias columnas y filas

## C.4. Formulas Matemáticas

A continuación se presentan entornos matemáticos con la Ecuación C.1 y la Ecuación C.2.

$$C_L = \frac{(S_{22} - \delta S_{11}^*)^*}{|II S_{22}|^2 = -|\pi|^2} \quad (C.1)$$

$$R_S = \frac{\sqrt{1 - g_s} \cdot (1 - |S_{11}|^2)}{1 - (1 - g_s) \cdot |S_{11}|^2} \quad (C.2)$$

## C.5. Items, Descripciones y Enumeraciones

Ejemplo de `itemize`:

- Item sin números
  - nivel 2
    - nivel 3

Ejemplo de `enumerate`:

1. Item Numerado
  - 1.1 Nivel 2
    - 1.1.1 Nivel 3

Ejemplo de `description`:

**Descripción** Texto descrito

**Nivel 2** Texto

Ejemplo de uso intercalado

**Descripción** Texto descrito

- Nivel 2
  1. Nivel 3

**C.6. Figuras**

En la Figura C.1 se muestra el logo de la Universidad. En cambio en la Figura C.2 se pueden apreciar 3 imágenes, la primera sería la Figura C.2a, la segunda la Figura C.2b y la tercera la Figura C.2c.



Figura C.1: Logo Universidad de Los Lagos



(a) Primera figura



(b) Segunda figura



(c) Tercera figura

Figura C.2: Insertar subfiguras en  $\text{\LaTeX}$ .

A continuación se presenta la Figura C.3 entre texto, esta figura debe estar antes del texto y la ubicación puede ser L: izquierda; C: centrado; R: derecha. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus.



Figura C.3: Foto entre texto



Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.