



Universidad del Bío-Bío
Facultad de Ciencias Empresariales
Depto. de Sistemas de Información

Título de la Tesis

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

Alumno

Nombre Apellido

Profesor Guía

Nombre Apellido

Lunes 18 de diciembre, 2023

Índice

Dedicatoria	5
Agradecimientos	6
Resumen	7
Introducción	8
1. Estudio del Problema	9
1.1. Definiciones, Siglas y Abreviaciones	9
1.2. Contexto del Problema	9
1.2.1. Diagrama de Situación Actual	9
1.3. Oportunidades de Mejora o Problemáticas	10
1.4. Propuesta de Solución	10
1.5. Soluciones Similares Disponibles	10
1.5.1. Aplicación Similar	10
1.6. Justificación del Proyecto	10
2. Proyecto	11
2.1. Objetivo General del Proyecto	11
2.2. Objetivos Específicos del Proyecto	11
2.3. Metodología de Desarrollo	11
2.4. Técnicas y Notaciones	12
2.5. Estándares de Documentación	12
2.6. Software, Frameworks y Lenguajes Utilizados	12
3. Factibilidad	14
3.1. Factibilidad Técnica	14
3.1.1. Conocimientos de los Usuarios	14
3.1.2. Disponibilidad Profesional	14
3.1.3. Despliegue y Servidor	15
3.2. Factibilidad Operativa	15
3.3. Factibilidad Económica	15
3.3.1. Tablas de Costos	15
3.3.2. Flujo de Caja	16
3.3.2.1. Contexto e Indicadores Económicos	16
3.3.2.2. Puesta en Marcha	16
3.3.2.3. Cálculo del Valor Actual Neto	17
3.4. Conclusión de Factibilidad	18

4. Requerimientos del Software	19
4.1. Límites	19
4.2. Caracterización de los Usuarios	19
4.3. Objetivo General del Software	19
4.3.1. Objetivos Específicos del Software	19
4.4. Requerimientos Funcionales del Software	19
4.5. Requerimientos No Funcionales del Software	20
4.6. Interfaces Internas de Salida	21
4.7. Interfaces Externas de Salida	21
5. Análisis Funcional	22
5.1. Actores	22
5.2. Casos de Uso	22
5.2.1. Diagramas de Casos de Uso	22
5.2.2. Especificación de los Casos de Uso	22
5.2.3. Detalle de los Casos de Uso	23
5.3. Modelo de Datos	23
5.4. Esquema de la Base de Datos	24
5.5. Diseño de Interfaz	24
5.5.1. Paleta de Colores y Tipografía	24
5.6. Diseño de Arquitectura	24
5.7. Estructura del Código	24
5.7.1. Estándres de Codificación	24
5.7.2. Backend	24
5.7.3. Frontend	25
6. Plan de Capacitación, Implantación y Puesta en Marcha	26
6.1. Estado del Proyecto	26
6.2. Implantación y Puesta en Marcha	26
6.3. Plan de Capacitación	26
7. Conclusión del Proyecto	27
8. Anexos	28
8.1. Anexo Estimación de Casos de Uso	28
8.2. Anexos de Recopilación de Información	30
8.3. Anexo Aspectos de Gestión de Proyectos	30
8.3.1. Anexo Resumen de Esfuerzo	30
8.4. Anexos Retrospectiva del Proyecto	31
8.4.1. Anexo Iteraciones en el Desarrollo	31

Índice de figuras

1.1. Figura de ejemplo	9
----------------------------------	---

Índice de tablas

2.1. Tabla de Riesgos	12
3.1. Tabla de equipos físicos	14
3.2. Tabla de software	14
3.3. Tabla de VPS	15
3.4. Tabla de costos de software	15
3.5. Tabla de costos de producción	16
3.6. Tabla de indicadores económicos	16
3.7. Tabla de costos	17
3.8. Tabla de cálculo de VAN	17
4.1. Tabla de requerimientos funcionales del módulo de ejemplo	20
4.2. Tabla de requerimiento no funcional de API	20
4.3. Tabla de interfaces internas de salida	21
4.4. Tabla de interfaces externas internas de salida	21
5.1. Tabla de actores	22
5.2. Tabla de especificación de casos de uso	22
5.3. Tabla del caso de uso CU_01_INICIAR_SESION	23
5.4. Tabla del caso de uso CU_02_CREAR_MODELO_X	23
5.5. Tabla de directorios del backend del proyecto	25
5.6. Tabla de directorios del frontend del proyecto	25
8.1. Tabla de complejidad técnica	29
8.2. Tabla de factores medioambientales	30
8.3. Tabla de resumen de esfuerzo	30
8.4. Tabla de iteraciones en el desarrollo	31

Dedicatoria

En la dedicatoria es normal referirse a la familia y a personas que hayan tenido influencia personal en torno al desarrollo profesional del tesista, o que hayan brindado apoyo moral, inspiración u otros aspectos positivos en torno a la realización del proyecto de título.

Agradecimientos

En los agradecimientos, es de buena educación agradecer a los profesores que guiaron el proyecto de tesis, a la universidad del alumno tesista y a otros profesores que hayan tenido influencia en el trabajo del alumno. Además, no olvidar también incluir agradecimientos a cualquiera que haya aportado a la realización del proyecto de título.

Resumen

En el resumen se entrega una breve descripción da cada capítulo de manera individual. Por ejemplo: el capítulo 1 habla del estudio del problema, entregando detalles de su origen, características, etc. El capítulo 2 detalla los objetivos generales y específicos. Así con el resto de capítulos.

Introducción

En la introducción se describe, a grandes rasgos, el contexto general del proyecto de tesis, y una vista general de todo lo que contiene el informe de tesis.

Capítulo 1

Estudio del Problema

1.1. Definiciones, Siglas y Abreviaciones

En esta sección se deben listar todas las definiciones, siglas y abreviaciones relevantes para la actividad de titulación. Por ejemplo, se pueden definir en una lista como la siguiente:

- ABC: Sigla que significa tal cosa.
- DEF: Otra sigla que significa otra cosa.

1.2. Contexto del Problema

...



Figura 1.1: Figura de ejemplo

1.2.1. Diagrama de Situación Actual

BPMN, u otro tipo de diagrama para ilustrar la situación actual.

1.3. Oportunidades de Mejora o Problemáticas

...

1.4. Propuesta de Solución

...

1.5. Soluciones Similares Disponibles

A continuación, se describen las soluciones disponibles que pueden ser catalogadas como similares al proyecto que se presenta.

1.5.1. Aplicación Similar

...

1.6. Justificación del Proyecto

...

Capítulo 2

Proyecto

2.1. Objetivo General del Proyecto

Objetivo general del proyecto

2.2. Objetivos Específicos del Proyecto

1. Objetivo específico
2. Objetivo específico
3. Objetivo específico

2.3. Metodología de Desarrollo

Para poder definir la metodología de desarrollo a utilizar, primero se debe tener en cuenta la tabla 2.1, que representa los riesgos asociados.

Tabla de Riesgos			
Experiencia en el Problema	Alta	X	Se tienen años de experiencia con la empresa.
	Baja		
Tamaño del Problema	Grande	X	La cantidad de funcionalidades a implementar es muy alta.
	Pequeño		
Complejidad del Problema	Complejo	X	El sistema es difícil de comprender y manejar completamente.
	Simple		
Tamaño del Software	Grande	X	El software a construir requiere muchas funcionalidades.
	Pequeño		
Complejidad Software	Complejo	X	El software debe implementar cálculos complejos (ratios, promedios, etc.).
	Simple		
Experiencia Software	Alta	X	Se tiene una alta experiencia desarrollando software para la empresa.
	Baja		
Modularidad Funcional	Existe	X	Las funcionalidades pueden implementarse por separado y luego integrarse.
	No existe		

Tabla 2.1: Tabla de Riesgos

Aquí es recomendable escribir la interpretación de la tabla anterior y concluir por qué se eligió utilizar una determinada metodología.

2.4. Técnicas y Notaciones

- Diagrama de Casos de Usos.
- BPMN para modelar el proceso de negocio actual.
- Carta Gantt para la planificación inicial del proyecto.
- Patrón de diseño MVC (Modelo, Vista, Controlador).

2.5. Estándares de Documentación

- Adaptación Basada en IEEE Software Test Documentation Std 829-1998.
- Adaptación Basada en IEEE Software Requirements Specifications Std 830-1998.

2.6. Software, Frameworks y Lenguajes Utilizados

A continuación se lista el software, frameworks y lenguajes de programación, marcado y estilos utilizados para la realización de este proyecto.

Para efectos del siguiente listado, los nombres de las herramientas, frameworks y lenguajes se han redactado en **negrita**, seguidos de paréntesis en *itálica* que contienen el número de la versión asociada a cada ítem.

Lenguajes

- **Ruby** (3.2.2): Lenguaje de programación de alto nivel.
- **HAML** (6.2.3): Lenguaje de marcado para la abstracción de HTML.
- **Sass** (6.0): Lenguaje de extensión para CSS.

Software

- **MongoDB** (7.0.3): Base de datos orientada a documentos JSON.
- **Redis** (7.0.12): Almacenamiento en memoria, utilizado para el caché de datos.
- **RubyMine** (2023.2.2): Entorno de desarrollo integrado especializado para el trabajo con aplicaciones en Ruby, específicamente para Ruby on Rails.
- **Rake** (13.1): Librería de Ruby para la definición de tareas interdependientes.
- **MongoDB Compass** (1.39.0): Visor para bases de datos de MongoDB.
- **RedisInsight** (2.30.0): Visor para el almacenamiento del caché en Redis.
- **Docker Desktop** (4.21.0): Visor y gestor de contenedores de Docker, en formato de aplicación de escritorio multiplataforma.
- **NodeJS** (16.13.0): Entorno de servidor multiplataforma utilizado para la conversión de archivos en runtime.
- **Yarn** (1.22.21): Gestor de paquetes para JavaScript.
- **Docker** (24.0.2): Tecnología que permite crear y utilizar contenedores. Para efectos de este proyecto, es utilizado con el fin de probar el software desarrollado en distribuciones de Linux determinadas.
- **Termius** (8.7.2): Cliente SSH.
- **Git/Git Bash** (2.34.1): Sistema de control de versiones.
- **Ubuntu LTS** (18.04.6): Subsistema de Linux para Windows.

Frameworks

- **Ruby on Rails** (7.1): Framework para desarrollo de aplicaciones web full-stack.
- **Jekyll** (4.0.0): Framework para desarrollo de aplicaciones web estáticas escrito en Ruby.
- **Bootstrap** (4.4.1): Framework para la creación de estilos, manejo de elementos visuales y la responsividad en aplicaciones web.

Capítulo 3

Factibilidad

3.1. Factibilidad Técnica

3.1.1. Conocimientos de los Usuarios

¿Qué grado de conocimiento tienen los usuarios con respecto al contexto del problema? ¿Cómo se capacitará a los usuarios? ¿Tendrán disposición al cambio?

3.1.2. Disponibilidad Profesional

¿Se necesita del trabajo de un profesional del desarrollo de software u otra área específica? ¿Qué equipamiento se requiere y con qué equipamiento se cuenta? ¿Qué software se requiere y con qué tipo de acceso a él se cuenta?

Equipamiento	
Nombre	Acceso
Windows PC	Equipo personal
MacBook Pro M1	Equipo personal

Tabla 3.1: Tabla de equipos físicos

Software	
Nombre	Acceso
Notepad++	Software libre
MongoDB Compass	Software libre
RedisInsight	Software libre
Git/Git Bash	Software libre
Ubuntu LTS	Software libre
RubyMine	Licencia de estudiante
Termius	Licencia de estudiante
Microsoft Excel	Licencia de estudiante

Tabla 3.2: Tabla de software

3.1.3. Despliegue y Servidor

Características del servidor de despliegue, cómo y por qué se eligió, etc.

VPS	
Característica	Detalle
Proveedor	DigitalOcean
Región	Nueva York
Sistema Operativo	Ubuntu 22.04 (LTS) x64
Tipo de CPU	Intel Regular
Número de vCPUs	1 CPU
Memoria	2 GB
Almacenamiento (SSD)	50 GB
Transferencia	2 TB

Tabla 3.3: Tabla de VPS

Concluir si el proyecto es factible técnicamente explicando por qué lo es.

3.2. Factibilidad Operativa

¿Los usuarios tienen disposición al cambio? ¿Los usuarios verán una mejora en su experiencia? ¿Por qué?

3.3. Factibilidad Económica

Todo lo que tiene que ver con los flujos de caja, cálculo del VAN proyectado a cierta cantidad de años, etc.

3.3.1. Tablas de Costos

Tablas de costos relacionados con el proyecto. A continuación hay algunas tablas de ejemplo:

Software		
Nombre	Acceso	Precio (anual)
Notepad++	Software libre	\$0
MongoDB Compass	Software libre	\$0
RedisInsight	Software libre	\$0
Git/Git Bash	Software libre	\$0
Ubuntu LTS	Software libre	\$0
RubyMine	Licencia de estudiante	\$0
Termius	Licencia de estudiante	\$0
Microsoft Excel	Licencia de estudiante	\$0

Tabla 3.4: Tabla de costos de software

Costos de Producción		
Nombre	Proveedor	Precio (anual)
VPS	DigitalOcean	\$125.000
Mailer	Postmark	\$156.000
Dominio	Namecheap	\$22.600
Git/Git Bash	Software libre	\$0
Ubuntu LTS	Software libre	\$0
Crowdin	Licencia Open Source	\$0
Sentry	Licencia de estudiante	\$0
RubyMine	Licencia de estudiante	\$0
Termius	Licencia de estudiante	\$0
Microsoft Excel	Licencia de estudiante	\$0

Tabla 3.5: Tabla de costos de producción

3.3.2. Flujo de Caja

...

3.3.2.1. Contexto e Indicadores Económicos

...

La inflación promedio anual reportada por el Banco Central de Chile [1], y la prima de riesgo asociada a proyectos tecnológicos reportada por el Standish Group International en el CHAOS Manifesto del 2011 [2].

...

Inflación Promedio Anual	4.3 %
Tasa Prima de Riesgo	21 %

Tabla 3.6: Tabla de indicadores económicos

$$TMAR = 0.04 + 0.21 + (0.04 \cdot 0.21) = 0.2584 \approx 26 \%$$

3.3.2.2. Puesta en Marcha

En esta sección se puede buscar una referencia del sueldo mensual promedio de un desarrollador de software, analista programador o similar. Luego, a partir de esa referencia, calcular el valor por hora y ajustarlo según las horas de trabajo efectivo.

Ejemplo: (\$900.000 mensual; \$5.538 por hora), ajustado a las horas de trabajo efectivas empleadas en el proyecto, las cuales fueron 4 horas de trabajo efectivo durante 6 días de la semana por mes (4 semanas) de desarrollo.

$$Desarrollo = \$5.538 \cdot (4 \text{ horas} \cdot 6 \text{ días} \cdot 4 \text{ semanas}) = \$531.648$$

Costos	
Desarrollo	\$531.648
Internet	\$10.000
Electricidad	\$20.600
Total	\$561.648

Tabla 3.7: Tabla de costos

Una vez calculado el total de gastos para la puesta en marcha, es posible extrapolar a 4 meses y calcular la inversión inicial del proyecto a través de la siguiente fórmula.

$$I_0 = \$561.648 \cdot 4 \text{ meses} = \$2.246.592$$

3.3.2.3. Cálculo del Valor Actual Neto

Por ejemplo, al tratarse ahora del mantenimiento del software, las horas disminuyen, por lo que el costo también decrece:

$$\text{Mantenimiento} = \$5.538 \cdot (1 \text{ hora} \cdot 5 \text{ días} \cdot 4 \text{ semanas}) = \$110.760$$

$$\frac{\text{Desarrollo}}{\text{Mantenimiento}} = \frac{\$531.648}{\$110.760} = 4.8$$

$$\text{Internet} = \frac{\$10.000}{4.8} = \$2.083$$

$$\text{Electricidad} = \frac{\$20.600}{4.8} = \$5.208$$

	0	1	2	3	4	5
Mantenimiento		\$1.329.120	\$1.329.120	\$1.329.120	\$1.329.120	\$1.329.120
Internet		\$24.996	\$24.996	\$24.996	\$24.996	\$24.996
Electricidad		\$62.496	\$62.496	\$62.496	\$62.496	\$62.496
Hosting		-\$126.312	-\$126.312	-\$126.312	-\$126.312	-\$126.312
Flujo		\$1.290.300	\$1.290.300	\$1.290.300	\$1.290.300	\$1.290.300
Inv. Inicial	\$2.246.592					
Flujo Total	\$6.451.500					

Tabla 3.8: Tabla de cálculo de VAN

$$VAN = I_0 + \sum_{t=i}^n \frac{C_t}{(1+r^t)} = \$5.646.624$$

3.4. Conclusión de Factibilidad

Concluir sobre cada tipo de factibilidad y entregar una conclusión general del capítulo.

Capítulo 4

Requerimientos del Software

4.1. Límites

- El software no permitirá...

4.2. Caracterización de los Usuarios

Describir a quienes apunta este proyecto. Se caracterizan por:

- Pertenencia a un grupo etario entre...
- ¿Competencias técnicas en uso de software similares?
- Familiaridad con...

4.3. Objetivo General del Software

Describir el objetivo general del software.

4.3.1. Objetivos Específicos del Software

- Objetivo específico
- Objetivo específico
- Objetivo específico

4.4. Requerimientos Funcionales del Software

A continuación, en las tablas x a y, se definen los requerimientos funcionales del software.

Módulo de Ejemplo	
Id	Descripción
RF_01	La plataforma contará con un módulo de creación...
RF_02	La plataforma contará con un módulo de visualización...
RF_03	La plataforma contará con un módulo de edición de un...
RF_04	La plataforma contará con un módulo de eliminación de un...
RF_05	La plataforma contará con un módulo de visualización de un...

Tabla 4.1: Tabla de requerimientos funcionales del módulo de ejemplo

4.5. Requerimientos No Funcionales del Software

La presente sección hablará de los requerimientos no funcionales del software desarrollado. 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. A continuación, se presentan los requerimientos no funcionales en las tablas x a y.

RNF_01 (EJEMPLO)		
Descripción	La plataforma contará con una API REST	
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 independientes, 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		

Tabla 4.2: Tabla de requerimiento no funcional de API

4.6. Interfaces Internas de Salida

Id	Nombre	Detalle de Datos
IN_01	Modelo X	field_1, field_2, field_3
IN_02	Modelo Y	field_1, field_2, field_3

Tabla 4.3: Tabla de interfaces internas de salida

4.7. Interfaces Externas de Salida

Id	Nombre	Detalle de Datos	Medio de Salida
OUT_01	Modelo X	field_1, field_2, field_3	Pantalla
OUT_02	Modelo Y	field_1, field_2, field_3	Archivo PDF

Tabla 4.4: Tabla de interfaces externas internas de salida

Capítulo 5

Análisis Funcional

5.1. Actores

La especificación de todos los actores se puede encontrar a continuación en la tabla 5.1.

Actores			
Actor	Función	Conocimientos	Privilegio
Administrador	Cumple con todas las funciones dentro de la empresa, tales como ...	¿Requiere conocimiento sobre como funciona el sistema en su totalidad? Si no es así, ¿qué tanto?	Máximo.
Cliente	Utiliza la página sólo para visualizar la información que esta ofrece.	No requiere conocimientos técnicos más allá de iniciar sesión.	Ninguno.

Tabla 5.1: Tabla de actores

5.2. Casos de Uso

Esta sección contiene todos los diagramas de casos de uso relevantes para el proyecto.

5.2.1. Diagramas de Casos de Uso

Imágenes de todos los casos de uso confeccionados para el proyecto.

5.2.2. Especificación de los Casos de Uso

Casos de Uso		
Id	Actor	Nombre
CU_01	Administrador, Usuario	Iniciar Sesión
CU_02	Administrador	Crear Modelo X

Tabla 5.2: Tabla de especificación de casos de uso

5.2.3. Detalle de los Casos de Uso

A continuación, se presentan tablas de detalle para los casos de uso listados en la tabla 5.2 que fueron marcados en negrita. Estos casos de uso también contarán con un detalle de su flujo de eventos básico.

Los casos de uso seleccionados para ser detallados fueron elegidos porque cumplen funciones fundamentales de los requisitos funcionales de la aplicación desarrollada. El resto de los casos de uso solamente contarán con sus precondiciones y una descripción simple.

CU_01_INICIAR_SESION (Usuario)	
Pre-Condiciones: El usuario debe estar en la página web. El usuario debe haberse registrado en la plataforma.	
Post-Condiciones: El usuario inicia sesión en la plataforma.	
Flujo de Eventos Básicos	
Usuarios: Administrador, Usuario	Sistema
	1. Renderiza la pantalla de inicio de sesión.
2. Ingresa su correo y contraseña, y luego pulsa el botón para iniciar sesión.	3. Valida la información ingresada por el usuario.
	4. Sesión iniciada. Redirecciona al usuario a la página principal.
Flujo de Eventos Alternativo	
Usuarios: Administrador, Organizador, Moderador y Jugador	Sistema
	3 (b). Si las credenciales son incorrectas, el sistema muestra un mensaje de error.
	4 (b). Vuelve al paso 2 del flujo básico.

Tabla 5.3: Tabla del caso de uso CU_01_INICIAR_SESION

CU_02_CREAR_MODELO_X (Administrador)
Pre-Condiciones: El administrador debe haber iniciado sesión con sus credenciales.
Descripción: El sistema guarda el modelo X en la base de datos con los datos ingresados por el administrador.

Tabla 5.4: Tabla del caso de uso CU_02_CREAR_MODELO_X

5.3. Modelo de Datos

Modelos de la base de datos. Pueden ser esquemas de SQL, diagramas para bases de datos no relacionales, etc.

5.4. Esquema de la Base de Datos

Esquemas de definición para los modelos de la base de datos. Por ejemplo, pueden ser los modelos escritos en JSON, directamente en algún lenguaje de programación, etc.

5.5. Diseño de Interfaz

En esta sección se presentan imágenes de mockups o capturas de pantalla del software terminado, para así ilustrar las interfaces realizadas.

5.5.1. Paleta de Colores y Tipografía

Imágenes o descripción de la paleta de colores y tipografía utilizadas.

5.6. Diseño de Arquitectura

Describir el diseño de la arquitectura utilizada para montar el proyecto desarrollado. Por ejemplo: "El proyecto, en su estado actual, hace uso de un servidor propio, el cual contiene los servicios web, bases de datos y caché, todo en una sola máquina ..."

5.7. Estructura del Código

El proyecto es una aplicación hecha en el framework X, por lo tanto sigue el patrón Y...

Insertar alguna imagen o tabla que permita visualizar el árbol de directorios/archivos del proyecto.

5.7.1. Estándres de Codificación

- Toda la base de código y documentación debe estar escrita en inglés.
- Se utilizan linebreaks (EOL - End of Line) CRLF.
- Se utilizan dos espacios para la indentación del código, no tabulaciones.
- La codificación del proyecto es en UTF-8.

5.7.2. Backend

La tabla 5.5 entrega una especificación de los directorios relevantes para el backend del proyecto.

Directorio	Detalle
controllers	Contiene todos los controladores ...
models	Contiene todas las clases que modelan y envuelven los datos almacenados en la base de datos de la aplicación ...

Tabla 5.5: Tabla de directorios del backend del proyecto

5.7.3. Frontend

La tabla 5.6 entrega una especificación de los directorios relevantes para el frontend del proyecto.

Directorio	Detalle
views	Contiene todas las vistas de la aplicación ...
assets	Todas las imágenes, hojas de estilo ...

Tabla 5.6: Tabla de directorios del frontend del proyecto

Capítulo 6

Plan de Capacitación, Implantación y Puesta en Marcha

6.1. Estado del Proyecto

Describir el estado del proyecto y su relación con el plan de capacitación. Por ejemplo:

”Actualmente, el proyecto se encuentra finalizado. Esto quiere decir que todos los módulos, requerimientos funcionales y objetivos propuestos han sido alcanzados con éxito.

Si bien el software del proyecto está completamente finalizado, éste aún no pasa a ser utilizado por el público, por lo que sigue estando en una etapa de puesta en marcha ...”

6.2. Implantación y Puesta en Marcha

...

6.3. Plan de Capacitación

Para este proyecto, se ha considerado un plan de capacitación que consiste en ...

Capítulo 7

Conclusión del Proyecto

Aquí se concluye sobre los objetivos del proyecto, si es que estos se lograron, en qué aportaron al perfil del tesista, etc ...

- ...
- ...
- ...
- ...
- ...

Capítulo 8

Anexos

En este capítulo se listan elementos relacionados directamente con la confección del presente informe y con el desarrollo del software que se ha realizado.

8.1. Anexo Estimación de Casos de Uso

En esta sección se evalúan los factores de complejidad técnica y ambiental. La tabla 8.1 es el detalle utilizado para obtener el "Technical Complexity Factor", o "TCF", y la tabla 8.2 el detalle de "Environment Factors", o "EF".

Technical Complexity Factor (TCF)			
Technical Factor	Multiplier	Relevancia Percibida	Resultado Multip.
Distributed System	2	2	4
Application performance objectives, in either response or throughput	1	2	2
End-user efficiency (on-line)	1	3	3
Complex internal processing	1	3	3
Reusability, the code must be able to reuse in other applications	1	3	3
Installation ease	0,5	1	0,5
Operational ease, usability	0,5	2	1
Portability	2	3	6
Changeability	1	3	3
Concurrency	1	2	2
Special security features	1	3	3
Provide direct access for third parties	1	0	0
Special user training facilities	1,5	4	6

Tabla 8.1: Tabla de complejidad técnica

Se obtiene entonces que el total es 34, con lo que sustituyendo en la fórmula del TCF, quedaría:

$$TCF = 0.6 + (0.01 \cdot 34)$$

$$TCF = 0.94$$

Environment Factors (EF)			
Environmental Factor	Multiplier	Relevancia Percibida	Resultado Multip.
Familiar with Iterative Methods	0,5	5	2,5
Application experience	1	5	5
Object Oriented experience	0,5	5	2,5
Analyst capability	1	5	5
Motivation	2	5	10
Stable requirements	-1	0	0
Difficult programming language	-1	3	-3

Tabla 8.2: Tabla de factores medioambientales

8.2. Anexos de Recopilación de Información

...

8.3. Anexo Aspectos de Gestión de Proyectos

...

8.3.1. Anexo Resumen de Esfuerzo

Actividad	Número de Horas
Preparación del proyecto	20
Desarrollo del módulo de autos	50
Desarrollo del módulo de pistas	50
Desarrollo del módulo de sesiones	80
Desarrollo del módulo de temporadas	80
Corrección de errores de código	40
Despliegue de la aplicación	27
Control de versiones	37
Total	384

Tabla 8.3: Tabla de resumen de esfuerzo

8.4. Anexos Retrospectiva del Proyecto

8.4.1. Anexo Iteraciones en el Desarrollo

Funcionalidad	Fecha	Retroalimentación
Módulo X (1)	12/11/2023	Añadir visualización para todos y uno sólo ...
Módulo X (2)	12/11/2023	Agregar a la navegación ...
Módulo X (3)	21/11/2023	Corregir problemas internos del módulo X

Tabla 8.4: Tabla de iteraciones en el desarrollo

Bibliografía

- [1] Banco Central de Chile, “Informe de Política Monetaria Septiembre 2023.” <https://www.bcentral.cl/web/banco-central/contenido/-/detalle/informe-de-politica-monetaria-septiembre-2023>, 2023. [Online; accedido 08/12/2023].
- [2] Big, Think and Small, Act, “Chaos manifesto 2011,” *CHAOS MANIFESTO*, 2011.