| DUOC UC - Escuela de informática y telecomunicaciones |
| --- |
| Propuesta de Proyecto y Especificación de Requisitos de Software |
| *Proyecto: [Trackify]* |
|  |
| **Revisión*: [01]*** |
| *23/08/2025* |

| Planificación y Especificación de Requisitos según estándares; IEEE 830, ISO9000 y PMI. |
| --- |

**Contenido**

[*DUOC UC - Escuela de informática y telecomunicaciones 1*](#_heading=h.mfj5vmpk5of0)

[**Ficha del documento 4**](#_heading=h.lamk0hdunlrp)

[**1. Introducción 5**](#_heading=h.mj3af18dtdel)

[1.1.](#_heading=h.zezin0u91zfa) Propósito 5

[1.2.](#_heading=h.rptpgw658pg1) Ámbito del Sistema 5

[1.3.](#_heading=h.vxg8ek5sdlnh) Definiciones, Acrónimos y Abreviaturas 5

[1.4.](#_heading=h.mljwhc301i81) Referencias 5

[1.5.](#_heading=h.y8cw5hf9qkx3) Visión General del Documento 5

[**2.**](#_heading=h.lae70w1dcalj) **Descripción General 6**

[2.1.](#_heading=h.1bywtrg4i6ch) Perspectiva del Producto 6

[2.2.](#_heading=h.pal52xdmpkt4) Funciones del Producto 6

[2.3.](#_heading=h.rj4dg9gopz3i) Características de los Usuarios 6

[2.4.](#_heading=h.599uxxmjvwku) Restricciones 6

[2.5.](#_heading=h.x7r819u2rjl0) Suposiciones y Dependencias 7

[2.6.](#_heading=h.kdyb6e9h2trk) Requisitos Futuros 7

[**3.**](#_heading=h.aua2ott8czsn) **Requisitos Específicos 8**

[3.1](#_heading=h.fmm48ak4wbmp) Requisitos comunes de las interfaces 8

[*3.1.1*](#_heading=h.6nhty6nhqkmj) *Interfaces de usuario 8*

[*3.1.2*](#_heading=h.emxo6wb0e32z) *Interfaces de hardware 8*

[*3.1.3*](#_heading=h.z7srxu2z1jdt) *Interfaces de software 8*

[3.2](#_heading=h.hkg8hs1wqpgj) Requisitos funcionales 9

[3.3](#_heading=h.r6i6ihaqqt2g) Requisitos no funcionales (Organización) y de calidad (Producto) 9

[*3.3.1*](#_heading=h.uqemi780y4rp) *Requisitos de Rendimiento o Eficiencia 9*

[*3.3.2*](#_heading=h.ew61ygdcjury) *Requisitos de Seguridad 9*

[*3.3.3*](#_heading=h.32rbuus5hpb2) *Requisitos de Usabilidad 9*

[*3.3.4*](#_heading=h.5y97mex2cn0) *Requisitos de Disponibilidad 9*

[*3.3.5*](#_heading=h.5b95s24i4pds) *Requisitos de Portabilidad 9*

[*3.3.6*](#_heading=h.mon8779180kk) *Requisitos de Mantenibilidad 9*

[*3.3.7*](#_heading=h.qws9qcbkrvw0) *Requisitos de Funcionalidad 9*

[3.4](#_heading=h.krvwvy7a18bo) Requisitos No funcionales Organizacionales 10

[**4. Propuesta de Planificación 11**](#_heading=h.cwtw70lwh86y)

[4.1 Descripción general acerca de la Planificación 11](#_heading=h.jg111zd4od9g)

[*4.1.2 Definición del Equipo de Trabajo 11*](#_heading=h.q2xvti6p2f6v)

[*4.1.3 Definición de Actividades principales del Proyecto 11*](#_heading=h.gcoryd6cbr9z)

[*4.1.4 Resumen Costos del Desarrollo del Proyecto 11*](#_heading=h.8l81ciouestd)

[*4.1.6 Carta Gantt 11*](#_heading=h.wpoj82xl5phy)

[5. Anexos 12](#_heading=h.mxvd4c5dleo3)

[*5.1 Acta de Proyecto 12*](#_heading=h.xnb1r233kchx)

[*5.2 Matriz Especificación de Requerimientos 12*](#_heading=h.v2314zabsikc)

[*5.3 Prototipado de Software 12*](#_heading=h.wqdhundkyizd)

[*5.4 Matriz EDT. Planilla Detallada Cálculo de Esfuerzo 12*](#_heading=h.6ocv70e0kera)

[*5.5 Planilla Carta Gantt 12*](#_heading=h.f4ajdsqjitc2)

# Ficha del documento

| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| --- | --- | --- | --- |
| *23/08/2025* | *28/08/2025* | *Duan Guzmán* | *Actualización del informe ERS* |
|  |  |  |  |

Documento validado por las partes en fecha:

**Integrantes:**

| **Nombre Integrante del Equipo** | **Rol Definido** |
| --- | --- |
| *Fernanda Donoso* | *Jefe de proyecto* |
| *Felipe Carrera* | *Analista programador* |
| *Duan Guzmán* | *Analista QA* |
|  |  |
|  |  |

# 1. Introducción

El presente documento corresponde a la **Especificación de Requerimientos de Software (ERS)** de la plataforma de gestión de desempeño laboral. El propósito es describir de manera estructurada los requerimientos funcionales y no funcionales que guiarán el diseño, desarrollo, implementación y validación del sistema.

El sistema propuesto consiste en una solución compuesta por una **aplicación web** y una **aplicación móvil**, ambas conectadas a un back-end central desarrollado en **Django y Django REST Framework**. La aplicación web estará dirigida principalmente a gerentes y supervisores, quienes podrán gestionar usuarios, proyectos, tareas y reportes, mientras que la aplicación móvil estará destinada a trabajadores, permitiéndoles visualizar y actualizar sus tareas, recibir notificaciones y consultar su historial de desempeño.

## 1.1. Propósito

El propósito de este documento es establecer de manera clara, estructurada y detallada los objetivos, funcionalidades, restricciones y características de la plataforma de desempeño laboral compuesto por una aplicación web y una aplicación móvil, nombrada **Trackify.**

El documento tiene como finalidad garantizar que los participantes comprendan qué se espera del sistema, cuáles son sus alcances y limitaciones, y cómo debe responder a las necesidades operativas de la organización, esto busca alinear los distintos interesados en el proyecto.

## 1.2. Ámbito del Sistema

El sistema, nombrado Trackify permitirá a las organizaciones gestionar de manera centralizada tareas, avances, evoluciones del trabajo y métricas de productividad de sus trabajadores.

* **Aplicación Web (Principal):** Utilizada por los administradores y supervisores. Permitirá gestionar usuarios, roles, áreas, proyectos y tareas; como también la generación de reportes exportables en PDF/Excel; visualizar dashboards con métricas claves y calificar el desempeño de los trabajadores.
* **Aplicación Móvil (Secundaria):** Será utilizada por empleados. Permitirá Visualizar y actualizar el estado de sus tareas y recibir notificaciones en tiempo real sobre nuevas asignaciones o evaluaciones.

Lo que realizará el sistema es:

* Administración de usuarios, roles y permisos.
* La gestión de proyecto, tareas y equipos de trabajo.
* Evaluación de desempeño con calificaciones y métricas.
* Reportes exportables y dashboards.
* Comunicación mediante notificaciones.

Lo que no realizara el sistema es:

* No contempla control biométrico de asistencia.
* No incluirá gestión de remuneraciones o pagos.

Beneficios esperados:

* Mejora de trazabilidad de tareas y evoluciones.
* Transparencia en la medición de desempeño.
* Optimización en la toma de decisiones estratégicas.

## 1.3. Definiciones, Acrónimos y Abreviaturas

**Django:** Framework de desarrollo web basado en Python, utilizado para la construcción de la aplicación web y la lógica de negocio en el back-end.

**Django** **Framework (DRF):** Herramienta de Django para construir Apis RESTful

**Api REST:** Interfaz de comunicación entre aplicaciones basada en HTTP.

**SQLite:** Motor de base de datos relacional considerado para producción.

**PostgreSQL:** Motor de base de datos liviano utilizado en etapa de desarrollo.

**React Native:** Framework para desarrollo de aplicaciones móviles multiplataforma.

**Gerente:** Usuario con control total sobre proyectos y reportes.

**Recursos humanos:** Usuario con control sobre los usuarios y roles

**Supervisor:** Usuario con permisos para gestionar tareas, calificar desempeño y generar reportes.

**Empleado:** Usuario que recibe, ejecuta y actualiza tareas desde la aplicación móvil

## 1.4. Referencias

* Planilla de requerimientos
* Carta Gantt y control del cambio
* Acta kick off
* Casos de uso
* Planilla EDT, tiempos y costos

## 1.5. Visión General del Documento

El documento está estructurado de la siguiente manera:

* **Sección 1 (Introducción)** Se establece el propósito, ámbito, definiciones y referencias
* **Sección 2 (Descripción General)** Se describe el contexto del sistema, sus funciones generales, características de usuarios, restricciones y requisitos futuros.
* **Sección 3 (Requisitos específicos)** detalla los requerimientos funcionales, no funcionales y de calidad del producto.
* **Sección 4 (planificación)** Expone la planificación, equipo de trabajo, actividades principales, costos estimados y cronograma del proyecto.
* **Sección 5 (Anexos)** Incluye la documentación complementaria, como el acta de constitución, matriz de requerimientos, prototipos y carta Gantt.

# 2. Descripción General

Esta propuesta consiste en el desarrollo de una plataforma de gestión de desempeño laboral compuesta por dos aplicaciones: una aplicación web y una aplicación móvil, ambas conectadas a un back-end central desarrollado en **Django** y **Django REST Framework**.

La **aplicación web** estará destinada a administradores y supervisores, quienes podrán gestionar usuarios, roles, áreas, proyectos y tareas. Además, contará con módulos de reportes exportables (PDF/Excel). La **aplicación móvil** será utilizada por los empleados, quienes podrán visualizar y actualizar sus tareas, registrar comentarios y avances, así como recibir notificaciones en tiempo real sobre nuevas asignaciones o evaluaciones de desempeño.

El sistema busca centralizar la información de tareas, avances y evaluaciones en una única plataforma, permitiendo mejorar la trazabilidad del trabajo, optimizar la comunicación entre los equipos y apoyar la toma de decisiones estratégicas basadas en datos.

## 2.1. Perspectiva del Producto

El sistema se basa en una arquitectura cliente-servidor compuesta por una aplicación web y una aplicación móvil que comparten un mismo back-end.

* La aplicación web será la herramienta principal de administración y supervisión, desarrollada con Django y Django Template, permitiendo la gestión integral de usuarios, proyectos, tareas, roles, reportes y evaluaciones.
* La aplicación móvil será utilizada por empleados, desarrollada en React Native con Expo, conectada con back-end mediante la API REST implementada en Django REST Framework.
* La base de datos será SQLite en la fase de desarrollo y PostgreSQL en ambientes productivos.

## 2.2. Funciones del Producto

**Gestión de usuarios y roles:** Registro, autentificación, recuperación de contraseñas, asignación de roles (administrador, supervisor, empleado).

**Gestión de proyectos y tareas:** Creación, modificación, eliminación y asignación de proyectos y tareas a los empleados.

**Seguimiento de avances:** Actualización del estado de tareas, registro de comentarios y avances.

**Evaluación de desempeño:** Calificación de desempeño de los empleados, generación de puntajes e historial de evaluaciones.

**Reportes y métricas:** Generación de reportes filtrados por fecha, estado y responsable, exportación a PDF/Excel y dashboards con métricas clave de productividad.

**Notificaciones:** Alertas por correo y aplicación móvil sobre asignaciones, plazos próximos a vencer y evaluaciones.

## 2.3. Características de los Usuarios

El sistema está diseñado para cuatro tipos de usuarios:

**Gerente:**

* **Perfil:** nivel técnico medio-alto, encargado de la gestión de equipos.
* **Funciones:** administración de usuarios, roles, áreas, proyectos y configuración general.
* **Plataforma:** Web

**Recursos humanos:**

* **Perfil:** nivel técnico medio-alto, encargado de la gestión del personal.
* **Funciones:** administración de usuarios, roles, áreas, proyectos.
* **Plataforma:** Web

**Supervisor:**

* **Perfil:** nivel técnico medio, con experiencia en gestión de equipos.
* **Funciones:** Creación y asignación de tareas, evaluación de desempeño, generación de reportes y análisis de métricas.
* **Plataforma:** Web.

**Empleado:**

* **Perfil:** nivel técnico básico, usuario final.
* **Funciones:** Recepción y actualización de tareas, registro de avances, recepción de notificaciones y comentarios
* **Plataforma:** Móvil.

## 2.4. Restricciones

**Tecnológicas:**

* El back-end debe desarrollarse en Django con Django REST Framework.
* La aplicación móvil debe desarrollarse en React Native con Expo.
* La aplicación web debe garantizar un diseño responsivo.

**Base de datos:**

* Uso de SQLite en desarrollo.
* Migración PostgreSQL en producción.

**Seguridad:**

* Enmascaramiento y cifrado de contraseñas.
* Manejo de sesiones y autenticación segura.

**Disponibilidad:**

* El sistema debe estar disponible 24/7.

**Alcance:**

* El sistema no contempla, en su versión inicial, integración con nómina o sistema de remuneración.

## 2.5. Suposiciones y Dependencias

* Se asume que todos los empleados tendrán acceso a un dispositivo móvil con conexión a internet.
* Dependencia de servicios de correo electrónico para el envío de notificaciones.
* Dependencia de librerías y frameworks externos (Django, DRF, React Native).
* Dependencia de la infraestructura de red de la organización para asegurar conectividad constante.

## 2.6. Requisitos Futuros

El sistema, inicialmente orientado a los sectores de hotelería y retail, podrá expandirse a otros nichos de mercado, como manufactura, salud, educación o logística. Esta expansión se logrará gracias a la arquitectura modular de la plataforma, que permite adaptar fácilmente la gestión de tareas, reportes y evaluaciones a diferentes contextos organizacionales.

# 3. Requisitos Específicos

En esta sección se presenta de manera detallada los requisitos que debe cumplir el sistema propuesto para garantizar su correcto funcionamiento. Los requisitos serán calificados en **Funcionales** y **no funcionales,** considerando tanto las necesidades del negocio como los estándares de calidad de software.

* Los **requerimientos funcionales (RF)** describen las acciones y servicios que el sistema proporcionará a los diferentes tipos de usuarios (gerentes, supervisores y trabajadores), tales como la gestión de usuarios, proyectos, tareas, reportes y notificaciones.
* Los requerimientos no funciona**les (RNF)** establecen las condiciones de calidad, seguridad, usabilidad, rendimiento y disponibilidad que debe cumplir la plataforma en sus versiones web y móvil.

## 3.1 Requisitos comunes de las interfaces

Descripción detallada de todas las entradas y salidas del sistema de software.

### 3.1.1 Interfaces de usuario

* La aplicación web se desarrollará Django con Django Templates, utilizando HTML5, CSS3 y Javascript.
* La interfaz web debe ser responsiva, accesible desde computadores y dispositivos móviles mediante navegadores modernos (Chrome, Firefox, Edge)
* La aplicación móvil se desarrollará en React Native conExpo, con interfaz simple y optimizada para trabajadores, disponibles para Android e IOS.
* Los formularios de ingreso deben validar datos en tiempo real, mostrar mensajes de error claros y enmascarar contraseñas.
* Los dashboards incluirán gráficos dinámicos y tablas con filtros de datos.

### 3.1.2 Interfaces de hardware

* La aplicación web podrá ejecutarse en cualquier PC o dispositivo móvil con un navegador moderno y conexión a internet.
* La aplicación móvil debe funcionar en dispositivos Android (versión 8.0 o superior) e IOS (versión 12.0 o superior).
* Los servidores deberán contar con un entorno de ejecución Python 3.10+ y PostgreSQL en ambiente productivo.

### 3.1.3 Interfaces de software

Indicar si hay que integrar el producto con otros productos de software.

* **Back-end:** Django 5.x con Django REST Framework (DRF).
* **Base de datos:** SQLite en desarrollo, PostgreSQL en producción.
* **Notificaciones:** Integración con servicios de mensajería push (Expo Push Notifications).
* **Reportes:** Uso de librerías de exportación (ReportLab para PDF, openpyxl para Excel).
* **Control de versiones:** Repositorio en GitHub/GitLab.
* **API REST:** expuesta para consumo tanto desde la aplicación web como móvil

## 3.2 Requisitos funcionales

Los requisitos funcionales deben estar enumerados (diagrama caso uso) y pueden ser divididos en sub-secciones.

**Gestión de usuarios y autenticación**

* RF01: El sistema permite registrar gerentes.
* RF02: El sistema permite inicio de sesión.
* RF03: El sistema permite cerrar sesión.
* RF04: El sistema debe diferenciar el acceso según roles (gerente, supervisor y trabajador).
* RF05: Los usuarios pueden modificar sus credenciales.
* RF06: El sistema debe permitir recuperar la contraseña mediante SMS.

**Gestión empresarial y organizacional**

* RF07: El gerente puede modificar la información de la empresa (nombre, departamento y áreas)
* RF08: El gerente puede gestionar supervisores y trabajadores (crear, modificar, eliminar y listar)
* RF09: El gerente puede gestionar departamentos/áreas.
* RF10: El gerente puede crear proyectos y asignarlos a supervisores
* RF11: El gerente puede crear, asignar, modificar y eliminar tareas para los supervisores
* RF12: El gerente puede generar reportes globales de desempeño
* RF13: El gerente puede visualizar dashboards basado en tareas completadas, retrasos y desempeño por departamento

**Gestión de supervisores y tareas**

* RF14: El supervisor puede crear, asignar, modificar y eliminar tareas para los trabajadores de su área
* RF15: El supervisor puede monitorear el estado de las tareas asignadas
* RF16: El supervisor califica el desempeño de los trabajadores.
* RF17: El supervisor puede generar reportes por departamento, trabajador o estado de las tareas.
* RF18: El supervisor puede dejar comentarios en las tareas de los trabajadores.

**Gestión de trabajadores y aplicación móvil**

* RF19: El trabajador debe poder visualizar las tareas que se le asignaron en la aplicación móvil.
* RF20: El trabajador puede actualizar el estado de las tareas en la aplicación móvil (pendiente, en progreso, atrasada y finalizada).
* RF21: El trabajador debe recibir notificaciones de la aplicación móvil de nuevas tareas o evaluaciones.
* RF22: El trabajador puede consultar su historial de desempeño en la aplicación móvil.
* RF23: La aplicación móvil debe permitir que el trabajador visualice sus evaluaciones.

**Reportes y notificaciones**

* RF24: El sistema debe permitir generar reportes de desempeño individual y grupal.
* RF25: Los reportes deben incluir filtros por fecha, área, estado de tareas y responsables.
* RF26: Los reportes deben exportarse en PDF y Excel.
* RF27:  El sistema debe enviar notificaciones por la aplicación móvil al asignar tareas.
* RF28: El sistema debe enviar notificaciones por la aplicación móvil cuando una tarea esté por vencer.
* RF29: El sistema debe enviar notificaciones por la aplicación móvil cuando se reciba una nueva evaluación.

## 3.3 Requisitos no funcionales (Organización) y de calidad (Producto)

### 3.3.1 Requisitos de Rendimiento o Eficiencia

* RNF04: El tiempo de respuesta del sistema no debe superar los 3 segundos en operaciones comunes (inicio de sesión, carga de reportes, actualización de tareas).
* RNF13:La aplicación móvil debe funcionar con una conexión mínima de 3G o superior

### 3.3.2 Requisitos de Seguridad

* RNF05: Las contraseñas deben almacenarse de manera encriptada utilizando algoritmos seguros.
* RNF14: El sistema debe diferenciar acceso según roles (gerente, supervisor, trabajador), restringiendo funcionalidades de acuerdo con el perfil de usuario.
* RNF07: El sistema debe mostrar mensajes de error claros, sin revelar información sensible.
* RNF15: Todas las comunicaciones entre la aplicación web/móvil y el servidor debe realizarse a través de HTTPS para proteger la información.

### 3.3.3 Requisitos de Usabilidad

* RNF02: La aplicación web debe ser responsiva y funcionar en distintos navegadores (Chrome, Firefox, Opera, Safari).
* RNF03: La aplicación móvil debe ser responsiva y ejecutarse correctamente en distintos dispositivos con Android e IOS.
* RNF06: La aplicación móvil debe contar con una interfaz simple y optimizada para trabajadores.
* RNF16:Los formularios deben incluir validaciones en tiempo real para reducir errores en ingreso de datos.

### 3.3.4 Requisitos de Disponibilidad

* RNF11: El sistema debe estar disponible al menos el 95% del tiempo anual.
* RNF17:El sistema debe poder recuperarse en menos de 1 hora en caso de falla crítica del servidor
* RNF18:El servidor debe contar con respaldo automático diario de la base de datos

### 3.3.5 Requisitos de Portabilidad

* RNF01: El sistema debe ser accesible tanto desde la aplicación web como desde la aplicación móvil.
* RNF10: El sistema debe exponer un API REST, lo que permitirá su integración con aplicaciones de terceros en el futuro
* RNF19: La aplicación móvil debe ejecutarse en Android (8.0 o superior) y iOS (12.0 o superior).

### 3.3.6 Requisitos de Mantenibilidad

* RNF08: El sistema debe estar diseñado bajo una arquitectura cliente-servidor que permita separar la lógica del negocio, la interfaz y la base de datos.
* RNF09: El código fuente debe estar versionado en un repositorio Git, lo que permitirá mantener un control de versiones y facilitar futuras actualizaciones.
* RNF20: El sistema debe desarrollarse de forma modular, de manera que nuevas funcionalidades puedan integrarse sin afectar los módulos existentes.
* RNF21:Debe existir documentación técnica y manual de usuario para asegurar la continuidad del mantenimiento

### 3.3.7 Requisitos de Funcionalidad

* RNF12: El sistema debe contar con un servidor web y una base de datos centralizada para su correcto funcionamiento.
* RNF22: La aplicación debe garantizar la correcta ejecución de todas las operaciones definidas en los requisitos funcionales (RF).
* RNF23: El sistema debe permitir la trazabilidad completa de tareas, proyectos y evaluaciones, asegurando que no existan inconsistencias en la información.

## 3.4 Requisitos No funcionales Organizacionales

1. El sistema debe incluir los colores e imágenes definidas por la organización.
2. La documentación técnica y manual de usuario deben entregarse al finalizar el proyecto

# 4. Propuesta de Planificación

## 4.1 Descripción general acerca de la Planificación

El proyecto tendrá una duración total de 5 meses (150 días hábiles), iniciando el 11 de agosto de 2025 y finalizando el 30 de diciembre de 2025. La planificación está organizada en fases que incluyen levantamiento de requerimientos, análisis y diseño, desarrollóo pruebas, implementación y cierre.

### 4.1.2 Definición del Equipo de Trabajo

El equipo del proyecto estará conformado por profesionales con roles definidos de acuerdo con sus responsabilidades en el desarrollo:

* **Gerente de Proyecto:** Planificación, control de hitos, comunicación con stakeholders y supervisión general
* **Programador Analista:** Desarrollo de módulos críticos del sistema, integración de back-end con API REST.
* **Desarrollador de Base de Datos:** Modelamiento, diseño e implementación de la base de datos PostgreSQL.
* **Diseñador:** Desarrollo de interfaces gráficas (web y móvil), asegurando usabilidad y diseño responsivo.
* **Tester / QA:** Elaboración de plan de pruebas, ejecución de casos de prueba y aseguramiento de calidad.

### 4.1.3 Definición de Actividades principales del Proyecto

Las actividades principales que se desarrollarán en el proyecto se encuentran detalladas en la Carta Gantt. Estas incluyen, entre otras:

* Levantamiento de requerimientos.
* Diseño de la arquitectura del sistema.
* Desarrollo de la aplicación web.
* Desarrollo de la aplicación móvil.
* Desarrollo del back-end.
* Pruebas e integración.
* Documentación y entrega final.

### 4.1.4 Resumen Costos del Desarrollo del Proyecto

El costo estimado del proyecto se sitúa en torno a **$25 millones de pesos chilenos (29.070 USD)**. Este monto considera gastos de personal, infraestructura tecnológica, herramientas de software y otros costos asociados a la correcta ejecución e implementación del sistema.

* **Costos por fase**: Incluye planificación, análisis y diseño, desarrollo, pruebas/QA, implementación y cierre.
* **Costos por rol**: Se contemplan los perfiles de Gerente de Proyecto, Programador Analista, Desarrollador de Base de Datos, Diseñador y QA, alcanzando un subtotal de **$16.351.503 CLP**.
* **Infraestructura tecnológica**: Servidores de desarrollo, pruebas y producción, junto con servicios de base de datos administrada, con un subtotal de **$515.530 CLP**.
* **Herramientas y software**: Licencias y suscripciones para repositorios, pruebas, documentación y gestión, con un subtotal de **$700.000 CLP**.
* **Otros costos asociados**: Capacitación a usuarios y soporte inicial post-implementación, por **$500.000 CLP**.

**Total estimado: $18.067.033 CLP**

### 4.1.6 Carta Gantt

### 

| **Hito** | **Fecha tope** |
| --- | --- |
| Fase de Planificación. | 11-08-2025 |
| Fase de Análisis y diseño. | 25-08-2025 |
| Fase de desarrollo web. | 01-09-2025 |
| Implementación módulo del administrador del sistema - web. | 15-09-2025 |
| Implementación módulos principales. | 29-09-2025 |
| Fase de Desarrollo App Móvil | 06-10-2025 |
| Implementación módulo del administrador del sistema - APP. | 20-10-2025 |
| Implementación módulos principales. | 03-11-2025 |
| Fase de gestión de RRHH. | 24-11-2025 |
| Fase de Pruebas y QA. | 08-12-2025 |
| Fase de implementación y cierre. | 22-12-2025 |

## 5. Anexos

### 5.1 Acta de Proyecto

[1.Acta de constitución.docx](https://docs.google.com/document/d/1jgjEjECk1SkfxEqxEURZzx_duSQrCYHY/edit?usp=sharing&ouid=116557405496578001357&rtpof=true&sd=true)

### 5.2 Matriz Especificación de Requerimientos

[2. Planilla de Requerimientos.xlsx](https://docs.google.com/spreadsheets/d/1qSMe7oYpTVEUh6G2m9qhwpz5GXbzL0bs/edit?usp=sharing&ouid=116557405496578001357&rtpof=true&sd=true)

### 5.3 Prototipado de Software

[7. Documento Mockups.docx](https://docs.google.com/document/d/1GeC_xK7yqh93hMFKN0wzBvOTdCZVyTtJ/edit?usp=sharing&ouid=116557405496578001357&rtpof=true&sd=true)

### 5.4 Matriz EDT. Planilla Detallada Cálculo de Esfuerzo

[4. Matriz EDT.xlsx](https://docs.google.com/spreadsheets/d/1ryLsoN6onYBFgW0eog5aVCKJw_vVafXD/edit?usp=sharing&ouid=116557405496578001357&rtpof=true&sd=true)

### 5.5 Planilla Carta Gantt

[3. Carta Gantt.xlsx](https://docs.google.com/spreadsheets/d/1_VPxPJOORSxdYJvZzkIfEvjRrFUhtHka/edit?usp=sharing&ouid=116557405496578001357&rtpof=true&sd=true)