Evidencias de Diseño

Proyecto TaskControl

Integrantes:

Michael Cid Molina

Luciano Pino Aguilar

Vania Vargas Dinen,

Fecha:

17/10/2025

Contenido

[1. Introducción 3](#_Toc211804849)

[2. Casos de Uso 4](#_Toc211804850)

[3. Prototipo de la Aplicación 8](#_Toc211804851)

[a. Figura 1 – pantalla de login. 9](#_Toc211804852)

[b. Figura 2 – vista del panel principal. 10](#_Toc211804853)

[c. Figura 3 – vista del módulo de gestión de usuarios. 11](#_Toc211804854)

[d. Figura 4 – módulo de tareas. 12](#_Toc211804855)

[e. Figura 5 – carga de evidencias. 13](#_Toc211804856)

[4. Diagramas BPMN 14](#_Toc211804857)

[a. Figura 1 – proceso de carga de evidencias. 14](#_Toc211804858)

[b. Figura 2 – proceso de programación mensual. 15](#_Toc211804859)

[c. Figura 3 – generación de reportes. 16](#_Toc211804860)

[5. Modelo Entidad–Relación (ERD) 17](#_Toc211804861)

[6. Arquitectura de Software 18](#_Toc211804862)

[a. Figura 1 - Arquitectura de software Sistema TaskControl 19](#_Toc211804863)

[19](#_Toc211804864)

[7. Carta Gantt / Planificación 20](#_Toc211804865)

[8. Conclusión 21](#_Toc211804866)

# Introducción

El desarrollo del proyecto TaskControl comenzó con una fase de análisis y diseño, orientada a definir los requerimientos funcionales y no funcionales del sistema.

Durante esta etapa, el equipo elaboró distintos artefactos de documentación que sirvieron como guía para la construcción posterior del sistema.

Las evidencias presentadas en este documento reflejan la planificación, modelamiento y diseño técnico que dieron forma a la solución.

Incluyen los diagramas de procesos en BPMN, el modelo entidad-relación, la arquitectura de software en tres capas y la planificación registrada en la carta Gantt.

Cada uno de estos elementos permitió mantener coherencia entre las etapas de desarrollo y asegurar la correcta implementación del sistema.

Estas evidencias documentales sirvieron como base para la implementación técnica del sistema y la validación de sus funcionalidades en las siguientes fases del proyecto.

# Casos de Uso

Los casos de uso del sistema TaskControl describen las principales interacciones entre los usuarios y las funcionalidades del sistema.

Entre ellos destacan la gestión de tareas, carga de evidencias, programación de actividades y generación de reportes.

El diagrama de casos de uso permite visualizar las relaciones entre los actores (Administrador, Proveedor y Colaborador) y las operaciones del sistema, sirviendo como base para el desarrollo de los módulos de frontend y backend.

El análisis de casos de uso permitió establecer los requerimientos funcionales del sistema y guiar el diseño de la base de datos y la arquitectura de software.

Tabla

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

# Prototipo de la Aplicación

El prototipo fue desarrollado en Figma, representando la estructura visual y la navegación de la aplicación TaskControl.

En esta etapa se definieron las principales pantallas del sistema, incluyendo el login, panel de actividades, creación de tareas, carga de evidencias y reportes.

Este diseño permitió validar la usabilidad y coherencia visual antes de implementar la versión funcional en React.

El prototipo definió la lógica de navegación y fue validado por el equipo antes del desarrollo del frontend en React, asegurando consistencia entre diseño y funcionalidad.

## Figura 1 – pantalla de login.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.Permite el acceso de usuarios mediante autenticación. Fue desarrollada en React con validación desde el backend en FastAPI.

## Figura 2 – vista del panel principal.

Muestra la navegación entre módulos, indicadores y tareas asignadas.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

## Figura 3 – vista del módulo de gestión de usuarios.

Permite agregar, editar o eliminar usuarios, además de asignar roles y dependencias.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

## Figura 4 – módulo de tareas.

Presenta la creación, actualización y visualización de tareas, con indicadores de estado (En proceso, Finalizada).

Imagen de la pantalla de un celular con letras

El contenido generado por IA puede ser incorrecto.

## Figura 5 – carga de evidencias.

Muestra el formulario de carga de archivos y su asociación a tareas específicas.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

# Diagramas BPMN

Los diagramas BPMN representan los principales procesos del sistema TaskControl, detallando la interacción entre los usuarios y las funciones operativas de la plataforma.

En el proceso “Carga de evidencias”, se detalla el flujo desde que el usuario inicia sesión, registra una tarea y adjunta los archivos correspondientes, hasta su validación por parte del sistema.

En “Programación mensual”, se muestra la planificación periódica de tareas y su asignación automática según dependencias.

Finalmente, en “Reportes”, se describe la generación de indicadores de avance, los cuales permiten visualizar resultados y estadísticas del mantenimiento realizado.

Estos flujos fueron esenciales para el modelado del backend y la integración de la lógica de negocio en FastAPI.

## Figura 1 – proceso de carga de evidencias.

Representa las actividades desde la subida de archivos hasta la validación por el administrador.

Diagrama

El contenido generado por IA puede ser incorrecto.

## Figura 2 – proceso de programación mensual.

Describe el flujo de asignación de tareas y planificación por parte del administrador

Diagrama

El contenido generado por IA puede ser incorrecto. .

## Figura 3 – generación de reportes.

Imagen que contiene Diagrama

El contenido generado por IA puede ser incorrecto.Detalla la obtención de datos y visualización de reportes

# Modelo Entidad–Relación (ERD)

El modelo entidad–relación define la estructura de la base de datos implementada en Supabase/PostgreSQL.

Contiene las entidades principales:

* Users: usuarios del sistema (administradores y proveedores).
* Tasks: registro de tareas de mantenimiento y sus estados.
* Providers: empresas o personas asignadas a la ejecución de tareas.
* Dependencies: áreas o unidades dentro de la organización.

Las relaciones establecidas entre ellas garantizan la integridad de los datos y la trazabilidad de cada tarea registrada.

Este modelo fue implementado directamente en Supabase, garantizando coherencia entre el diseño lógico y la estructura física de la base de datos.

Diagrama

El contenido generado por IA puede ser incorrecto.

# Arquitectura de Software

La arquitectura en tres capas utilizada en el desarrollo del sistema TaskControl, organizada bajo el modelo cliente-servidor.

* Capa de Presentación: Implementada con React, HTML, CSS y JavaScript, permite la interacción con el usuario desde el navegador web. La autenticación se realiza mediante Google API SSO (Single Sign-On) para garantizar seguridad y facilidad de acceso.
* Capa de Negocio: Contiene la lógica del sistema desarrollada con FastAPI y Django (Python), además del soporte de Node.js para manejo de dependencias y ejecución del servidor. Esta capa procesa las solicitudes recibidas desde el frontend y gestiona la comunicación con la base de datos.
* Capa de Datos: Se utiliza Supabase como servicio backend que facilita la conexión segura con PostgreSQL, encargado del almacenamiento y administración de la información.

Este diseño modular permite escalabilidad, mantenimiento y seguridad, asegurando la separación clara entre presentación, lógica y persistencia de datos.

## Figura 1 - Arquitectura de software Sistema TaskControl

# Interfaz de usuario gráfica, Diagrama El contenido generado por IA puede ser incorrecto.

# Carta Gantt / Planificación

La carta Gantt muestra la planificación general del proyecto, con las etapas de análisis, diseño, desarrollo, pruebas y documentación.

En ella se observa que algunas tareas se desarrollaron en paralelo para optimizar tiempos, especialmente entre el diseño del sistema, la implementación del frontend y la definición de la base de datos.

Esta planificación permitió cumplir los plazos establecidos y mantener un seguimiento continuo del avance.

Tabla

El contenido generado por IA puede ser incorrecto.La correcta gestión del cronograma permitió mantener control sobre el avance del proyecto y cumplir los hitos propuestos dentro del plazo establecido.

# Conclusión

El presente documento reúne las evidencias de documentación desarrolladas durante el proyecto TaskControl, reflejando el trabajo realizado en las etapas de análisis, diseño y planificación técnica del sistema.

Los artefactos generados como los diagramas BPMN, casos de uso, modelo entidad relación y arquitectura de software permitieron definir claramente la estructura y funcionamiento del proyecto.

La documentación elaborada facilitó la coordinación del equipo y aseguró la coherencia entre las fases de desarrollo, garantizando que la implementación técnica respondiera a los objetivos definidos.

En conjunto, estas evidencias demuestran un proceso de trabajo organizado, colaborativo y alineado con las buenas prácticas de la ingeniería de software.