

**Germán Alberto Hernández Mairal**

**DESARROLLO DE UNA PLATAFORMA WEB DE PRODUCTIVIDAD Y  
ORGANIZACIÓN DE TAREAS**

**TRABAJO FINAL DE GRADO**

**Dirigido por Carlos García-Barroso Villalonga**

**Grado de Técnicas de Desarrollo de Aplicaciones Web y Móviles**



**UNIVERSITAT ROVIRA I VIRGILI**

**Tarragona**

**2024**

**Resum.****Justificació del Projecte**

El projecte desenvolupat té com a objectiu proporcionar una solució eficient i automatitzada per a la gestió de projectes en agències de màrqueting digital.

En un entorn on l'organització i el seguiment de les tasques són crucials per a l'èxit, aquest projecte neix de la necessitat d'integrar i analitzar les dades rellevants de les agències emmagatzemades en diferents fonts, com Trello, en una única plataforma.

Això permet a les agències millorar la seva productivitat i prendre decisions informades basades en el rendiment real dels seus projectes.

**Metodologia Emprada**

El desenvolupament del projecte es va dur a terme seguint un enfocament personalitzat, centrat a complir amb els terminis establerts definits de manera independent juntament amb el suport del meu tutor.

Es van organitzar i planificar les tasques i objectius al llarg del temps, assegurant que cada funcionalitat principal es desenvolupés i provés abans de passar a la següent fase.

La integració de tecnologies com Node.js per al backend, PostgreSQL per a la gestió de bases de dades, i Chart.js per a la visualització de dades en el frontend, es van provar i ajustar amb l'objectiu d'assolir els objectius desitjats.

**Resultats**

Els resultats obtinguts han estat satisfactoris, aconseguint una primera versió de l'aplicació, simple i funcional, que permet a un usuari gestionar i visualitzar el rendiment dels seus projectes de manera eficient.

La integració amb Trello ha demostrat ser robusta, facilitant la sincronització de dades i millorant la capacitat d'anàlisi.

L'aplicació permet als usuaris generar informes detallats sobre costos, rendibilitat i temps dedicats a cada projecte, contribuint així a una millor presa de decisions estratègiques dins de l'agència.

**Resumen.****Justificación del Proyecto**

El proyecto desarrollado tiene como objetivo proporcionar una solución eficiente y automatizada para la gestión de proyectos en agencias de marketing digital.

En un entorno donde la organización y el seguimiento de las tareas son cruciales para el éxito, este proyecto nace de la necesidad de integrar y analizar los datos relevantes de las agencias guardadas en diferentes fuentes, como Trello, en una única plataforma.

Esto permite a las agencias mejorar su productividad y tomar decisiones informadas basadas en el rendimiento real de sus proyectos.

### **Metodología Empleada**

El desarrollo del proyecto se llevó a cabo siguiendo un enfoque personalizado, centrado en cumplir con los plazos establecidos definidos de forma independiente junto con el apoyo de mi tutor.

Se organizaron y planificaron las tareas y objetivos a lo largo del tiempo, asegurando que cada funcionalidad principal se desarrollara y probara antes de pasar a la siguiente fase.

La integración de tecnologías como Node.js para el backend, PostgreSQL para la gestión de bases de datos, y Chart.js para la visualización de datos en el frontend, iban siendo probadas y ajustadas con el fin de conseguir los objetivos deseados.

### **Resultados**

Los resultados obtenidos han sido satisfactorios, logrando una primera versión de la aplicación, simple y funcional que permite a un usuario gestionar y visualizar el rendimiento de sus proyectos de manera eficiente.

La integración con Trello ha demostrado ser robusta, facilitando la sincronización de datos y mejorando la capacidad de análisis.

La aplicación permite a los usuarios generar informes detallados sobre costos, rentabilidad y tiempos dedicados a cada proyecto, lo que contribuye a una mejor toma de decisiones estratégicas dentro de la agencia.

## **Abstract.**

### **Project Justification**

The developed project aims to provide an efficient and automated solution for project management in digital marketing agencies.

In an environment where organization and task tracking are crucial for success, this project arises from the need to integrate and analyze relevant agency data stored in various sources, such as Trello, into a single platform.

This allows agencies to improve their productivity and make informed decisions based on the actual performance of their projects.

### **Methodology Used**

The developed project aims to provide an efficient and automated solution for project management in digital marketing agencies.

In an environment where organization and task tracking are crucial for success, this project arises from the need to integrate and analyze relevant agency data stored in various sources, such as Trello, into a single platform.

This allows agencies to improve their productivity and make informed decisions based on the actual performance of their projects.

### **Results**

The results obtained have been satisfactory, achieving a first version of the application, simple and functional, that allows a user to manage and visualize the performance of their projects efficiently.

The integration with Trello has proven to be robust, facilitating data synchronization and improving analysis capabilities.

The application allows users to generate detailed reports on costs, profitability, and time dedicated to each project, contributing to better strategic decision-making within the agency.

# Índice

<b>1</b>	<b>INTRODUCCIÓN .....</b>	<b>9</b>
<b>2</b>	<b>PLANIFICACIÓN.....</b>	<b>10</b>
2.1	DIAGRAMA DE GANTT.....	10
2.2	DESARROLLO DEL PROYECTO .....	10
2.2.1	<i>Fase de Planificación y Preparación (4 semanas) – Enero.....</i>	<i>10</i>
2.2.2	<i>Fase de Diseño (3 semanas) – Febrero.....</i>	<i>10</i>
2.2.3	<i>Fase de Preparación de la Infraestructura (1 semana) – Marzo.....</i>	<i>11</i>
2.2.4	<i>Fase de Obtención de Datos (3 semanas) – Marzo.....</i>	<i>11</i>
2.2.5	<i>Fase de Desarrollo (12 semanas) - Finales de Abril, Mayo, Junio, Julio y Agosto.....</i>	<i>11</i>
2.2.6	<i>Fase de Documentación y Finalización (4 Semanas) - Agosto.....</i>	<i>11</i>
<b>3</b>	<b>REQUISITOS DEL PROYECTO.....</b>	<b>13</b>
3.1	REQUISITOS FUNCIONALES .....	13
3.1.1	<i>Casos de Uso de la Pantalla Bienvenida .....</i>	<i>13</i>
3.1.2	<i>Casos de Uso de la Pantalla de Inicio de Sesión .....</i>	<i>14</i>
3.1.3	<i>Casos de Uso de la 1ª Pantalla (Informe de Costos por Proyecto).....</i>	<i>16</i>
3.1.4	<i>Casos de Uso de la 2ª Pantalla (Análisis de Rentabilidad por Cliente) .....</i>	<i>20</i>
3.1.5	<i>Casos de Uso de la 3ª Pantalla (Visualización de Cargas de Trabajo).....</i>	<i>23</i>
3.1.6	<i>Casos de Uso de la 4ª Pantalla (Análisis de Servicio por Cliente).....</i>	<i>26</i>
3.1.7	<i>Casos de Uso de la 5ª Pantalla (Estadísticas de Actividades por Servicio) .</i>	<i>30</i>
3.1.8	<i>Casos de Uso de la 6ª Pantalla (Seguimiento del Tiempo).....</i>	<i>33</i>
3.1.9	<i>Casos de Uso de la 7ª Pantalla (Back-Office).....</i>	<i>37</i>
3.2	REQUISITOS NO FUNCIONALES .....	41
3.2.1	<i>Rendimiento.....</i>	<i>41</i>
3.2.2	<i>Seguridad.....</i>	<i>41</i>
3.2.3	<i>Mantenibilidad .....</i>	<i>41</i>
3.2.4	<i>Compatibilidad.....</i>	<i>42</i>
3.2.5	<i>Integridad de Datos.....</i>	<i>42</i>
<b>4</b>	<b>DISEÑO.....</b>	<b>43</b>
4.1	ARQUITECTURA DE LA APLICACIÓN .....	43
4.1.1	<i>Descripción General de la Arquitectura .....</i>	<i>43</i>
4.1.2	<i>Arquitectura de Tres Capas.....</i>	<i>43</i>
4.1.3	<i>Seguridad y Escalabilidad.....</i>	<i>44</i>
4.1.4	<i>Conclusión.....</i>	<i>44</i>
4.2	DIAGRAMAS DE USO .....	45
4.2.1	<i>Identificación de Roles de Usuario .....</i>	<i>45</i>
4.2.2	<i>Identificación de Casos de Uso .....</i>	<i>45</i>
4.2.3	<i>Diagramas.....</i>	<i>45</i>
4.3	DISEÑO DE CLASES.....	63
4.3.1	<i>Diagrama de Clases .....</i>	<i>63</i>
4.3.2	<i>Lógica de la Organización.....</i>	<i>66</i>
4.4	DISEÑO DE LA INTERFAZ DE USUARIO .....	67
4.4.1	<i>Navegación y Disposición de los Elementos en Pantalla.....</i>	<i>67</i>
4.4.2	<i>Usabilidad y Accesibilidad.....</i>	<i>67</i>
4.4.3	<i>Principios de Diseño Centrado en el Usuario.....</i>	<i>67</i>
4.5	DISEÑO DE LA PERSISTENCIA DE DATOS.....	80
4.5.1	<i>Estructura de la Base de Datos.....</i>	<i>80</i>
4.5.2	<i>Decisiones de Diseño detrás de la Elección del Modelo de Base de Datos..</i>	<i>85</i>
4.5.3	<i>Razones para la Elección del Modelo Relacional.....</i>	<i>85</i>
4.5.4	<i>Relación con los Requisitos de la Aplicación.....</i>	<i>86</i>

4.5.5	<i>Implementación de Operaciones de Lectura, Escritura y Consulta en la Base de Datos</i> .....	87
<b>5</b>	<b>IMPLEMENTACIÓN</b> .....	<b>92</b>
5.1	TECNOLOGÍAS UTILIZADAS.....	92
5.1.1	<i>Lenguajes de Programación</i> .....	92
5.1.2	<i>Frameworks y Librerías del Backend</i> .....	92
5.1.3	<i>Frameworks y Librerías del Frontend</i> .....	92
5.1.4	<i>Base de Datos</i> .....	92
5.1.5	<i>Integraciones de API</i> .....	92
5.1.6	<i>Seguridad</i> .....	92
5.2	ALGORITMOS ESPECÍFICOS .....	93
5.2.1	<i>Procesamiento de Tareas y Proyectos desde Trello (app.js)</i> .....	93
5.2.2	<i>Carga y Procesamiento de Archivos CSV (loadcsv.js)</i> .....	93
5.2.3	<i>Manejo de Autenticación y Seguridad (app.js)</i> .....	94
<b>6</b>	<b>EVALUACIÓN</b> .....	<b>95</b>
6.1	DISEÑO DE LOS CASOS DE PRUEBA .....	95
6.2	RESULTADOS DETALLADOS DE LAS PRUEBAS .....	95
6.3	EVALUACIÓN DEL PROYECTO .....	96
<b>7</b>	<b>EVALUACIÓN DE COSTES</b> .....	<b>97</b>
7.1	COSTES DE PERSONAL.....	97
7.1.1	<i>Personal Involucrado</i> .....	97
7.1.2	<i>Costo Total del Personal Anual</i> .....	97
7.2	COSTES DE MATERIAL.....	98
7.2.1	<i>Material Necesario</i> .....	98
7.2.2	<i>Costo Total del Material Anual</i> .....	99
7.3	RESUMEN DE EVALUACIÓN DE COSTES .....	99
<b>8</b>	<b>LEGISLACIÓN Y PROTECCIÓN DE DATOS</b> .....	<b>100</b>
8.1	IDENTIFICACIÓN DE DATOS PERSONALES EN LA APLICACIÓN .....	100
8.2	NORMATIVA DE PROTECCIÓN DE DATOS APLICABLE.....	100
8.2.1	<i>Reglamento General de Protección de Datos</i> .....	100
8.2.2	<i>Ley Orgánica de Protección de Datos Personales y Garantía de los Derechos Digitales</i> .....	101
8.2.3	<i>Leyes Sectoriales y de Confidencialidad</i> .....	101
8.3	MEDIDAS DE PROTECCIÓN Y CUMPLIMIENTO NORMATIVO (FUTURAS IMPLEMENTACIONES).....	101
8.4	CONCLUSIÓN .....	102
<b>9</b>	<b>CONCLUSIONES</b> .....	<b>103</b>
9.1	EVALUACIÓN DEL TRABAJO REALIZADO.....	103
9.2	APORTACIÓN A LA FORMACIÓN PROFESIONAL.....	104
9.3	CONCLUSIÓN .....	104
<b>10</b>	<b>RECURSOS UTILIZADOS</b> .....	<b>105</b>
10.1	BIBLIOGRAFÍA Y DOCUMENTACIÓN TÉCNICA .....	105
10.2	SOFTWARE UTILIZADO.....	105
10.3	HARDWARE UTILIZADO .....	105
10.4	SERVICIOS EN LA NUBE Y HOSTING .....	105
<b>11</b>	<b>ANEXOS</b> .....	<b>106</b>
11.1	INSTALACIÓN Y REQUISITOS .....	106
11.2	DOCUMENTACIÓN PARA EL USO Y MANTENIMIENTO.....	107
11.3	CASOS DE PRUEBA DETALLADOS .....	108
11.3.1	<i>Casos de Prueba de la Pantalla de Bienvenida</i> .....	108
11.3.2	<i>Casos de Prueba de la Pantalla de Inicio de Sesión</i> .....	108
11.3.3	<i>Casos de Prueba de la 1ª Pantalla (Informe de Costos por Proyecto)</i> .....	110

11.3.4	<i>Casos de Prueba de la 2ª Pantalla (Análisis de Rentabilidad por Cliente)</i>	111
11.3.5	<i>Casos de Prueba de la 3ª Pantalla (Visualización de Cargas de Trabajo).</i>	113
11.3.6	<i>Casos de Prueba de la 4ª Pantalla (Análisis de Servicio por Cliente).....</i>	114
11.3.7	<i>Casos de Prueba de la 5ª Pantalla (Estadísticas de Actividades por Servicio)</i> .....	117
11.3.8	<i>Casos de Prueba de la 6ª Pantalla (Seguimiento del Tiempo).....</i>	118
11.3.9	<i>Casos de Prueba de la 7ª Pantalla (Back-Office) .....</i>	120
11.3.10	<i>Casos de Prueba Para Requisitos No Funcionales .....</i>	122

## Índice de Figuras

FIGURA 1. CRONOGRAMA DEL PROYECTO.....	10
FIGURA 2. DIAGRAMA 1: PANTALLA DE BIENVENIDA Y PANTALLA DE INICIO DE SESIÓN.....	44
FIGURA 3. DIAGRAMA 2: PANTALLA DE INFORME DE COSTOS POR PROYECTO .....	46
FIGURA 4. DIAGRAMA 3: PANTALLA DE ANÁLISIS DE RENTABILIDAD POR CLIENTE .....	48
FIGURA 5. DIAGRAMA 4: PANTALLA DE VISUALIZACIÓN DE CARGAS DE TRABAJO .....	50
FIGURA 6. DIAGRAMA 5: PANTALLA DE ANÁLISIS DE SERVICIO POR CLIENTE .....	53
FIGURA 7. DIAGRAMA 6: PANTALLA DE ESTADÍSTICAS DE ACTIVIDADES POR SERVICIO .....	55
FIGURA 8. DIAGRAMA 7: PANTALLA DE SEGUIMIENTO DEL TIEMPO .....	57
FIGURA 9. DIAGRAMA 8: PANTALLA DE BACK-OFFICE.....	59
FIGURA 10. DIAGRAMA DE CLASES .....	62
FIGURA 11. PANTALLA DE BIENVENIDA .....	65
FIGURA 12. PANTALLA DE INICIO DE SESIÓN .....	66
FIGURA 13. PANTALLA DE REGISTRO DE SESIÓN .....	66
FIGURA 14. PANTALLA DE SUBIR IMAGEN DE PERFIL.....	67
FIGURA 15. PANTALLA DE INGRESAR CREDENCIALES DEL TRELLO.....	67
FIGURA 16. PANTALLA DE CREDENCIALES INTRODUCIDAS .....	68
FIGURA 17. PANTALLA DE EDICIÓN DE INFORMACIÓN DEL USUARIO .....	68
FIGURA 18. PANTALLA DE EDICIÓN DE LA CONTRASEÑA DEL USUARIO .....	69
FIGURA 19. PANTALLA DE SUBIR IMAGEN DE PERFIL (MODO EDICIÓN).....	69
FIGURA 20. PANTALLA DE INGRESAR CREDENCIALES DEL TRELLO (MODO EDICIÓN).....	70
FIGURA 21. PANTALLA DE INFORME DE COSTOS POR PROYECTO (CARGA INICIAL).....	70
FIGURA 22. PANTALLA DE INFORME DE COSTOS POR PROYECTO (INFORMACIÓN CARGADA) .....	71
FIGURA 23. PANTALLA DE ANÁLISIS DE RENTABILIDAD POR CLIENTE (CARGA INICIAL) .....	71
FIGURA 24. PANTALLA DE ANÁLISIS DE RENTABILIDAD POR CLIENTE (INFORMACIÓN CARGADA) .....	72
FIGURA 25. PANTALLA DE VISUALIZACIÓN DE CARGAS DE TRABAJO (CARGA INICIAL) .....	72
FIGURA 26. PANTALLA DE VISUALIZACIÓN DE CARGAS DE TRABAJO (INFORMACIÓN CARGADA).....	73
FIGURA 27. PANTALLA DE ANÁLISIS DE SERVICIO POR CLIENTE (CARGA INICIAL) .....	73
FIGURA 28. PANTALLA DE ANÁLISIS DE SERVICIO POR CLIENTE (INFORMACIÓN CARGADA).....	74
FIGURA 29. PANTALLA DE ESTADÍSTICAS DE ACTIVIDADES POR SERVICIO (CARGA INICIAL) .....	74
FIGURA 30. PANTALLA DE ESTADÍSTICAS DE ACTIVIDADES POR SERVICIO (INFORMACIÓN CARGADA) .....	75
FIGURA 31. PANTALLA DE SEGUIMIENTO DEL TIEMPO (CARGA INICIAL).....	75
FIGURA 32. PANTALLA DE SEGUIMIENTO DEL TIEMPO (INFORMACIÓN CARGADA).....	76
FIGURA 33. PANTALLA DE BACK OFFICE (CARGA INICIAL) .....	76
FIGURA 34. PANTALLA DE BACK OFFICE (REGISTRO DE INGRESOS DEL CLIENTE DE UN PROYECTO) .....	77

## 1 Introducción

Este Trabajo de Fin de Grado se centra en el desarrollo de una Aplicación Web de Análisis de Rendimiento, diseñada específicamente para Agencias de Marketing Digital.

Al analizar las necesidades de las agencias de marketing digital, identifiqué la importancia de tener claridad sobre ciertas variables críticas que sustentan sus servicios. Esta claridad es esencial para optimizar sus procesos y maximizar la eficiencia.

Aprovechando mi buena relación con el fundador y CEO de la agencia **Kiwop**, desarrollé una aplicación que considera variables clave como el rendimiento de los empleados, la efectividad de los servicios ofrecidos y la gestión de tareas.

La aplicación está diseñada para proporcionar información útil sobre los puntos fuertes y áreas de mejora en los procesos de la agencia: la carga de trabajo de los empleados, su rendimiento a lo largo del tiempo, la rentabilidad de los servicios y la eficiencia en la realización de tareas, entre otros.

Además, permite a la agencia acceder a esta información y contar con un entorno seguro para almacenar datos privados, como los costos de proyectos y los ingresos generados por cada servicio.

El objetivo principal de este proyecto es ofrecer a la agencia información esencial que le permita identificar de manera clara y efectiva qué partes de sus procesos y sistemas deben ser optimizadas, automatizadas o delegadas.

Creo firmemente que la creación de sistemas sólidos es clave para aumentar la productividad, tanto para empleados como para líderes de negocios.

El desarrollo de esta aplicación me ha brindado una experiencia valiosa en áreas donde antes tenía conocimientos teóricos, pero poca práctica como desarrollador web: el uso de bases de datos, la programación Front-End y Back-End, entre otros muchos aspectos.

Esta experiencia ha sido crucial para darme la confianza necesaria para iniciar mi carrera profesional en este campo.

## 2 Planificación

### 2.1 Diagrama de Gantt



Figura 1. Cronograma del Proyecto

## 2.2 Desarrollo del Proyecto

### 2.2.1 Fase de Planificación y Preparación (4 semanas) – Enero

**Generación de Ideas para el Proyecto (2 semanas):** Durante las dos primeras semanas de enero, me enfoqué en la generación de ideas, explorando diferentes enfoques y posibilidades para el proyecto, asegurándome de elegir un tema que fuera relevante y desafiante.

**Recopilación de Información y Definición de Necesidades (2 semanas):** En la segunda quincena de enero, recopilé toda la información necesaria y analicé las necesidades específicas de la aplicación. Para ello, utilicé formularios enviados a diversas agencias de Tarragona y realicé preguntas directas a **Kiwop**, la agencia con la que tengo una relación cercana con el propietario, Josep Purroy. Además, identifiqué una herramienta similar a la que quería desarrollar, **Screenful** (<https://screenful.com/>), lo que me ayudó a entender mejor las funcionalidades y el valor que una aplicación como esta podría ofrecer. Esta fase fue crucial para comprender tanto los requerimientos del mercado como las expectativas de los usuarios finales.

### 2.2.2 Fase de Diseño (3 semanas) – Febrero

**Creación de Mock-ups (3 semanas):** Durante todo febrero, me dediqué a la creación de mock-ups detallados, los cuales sirvieron como la primera representación

visual de la aplicación. Durante esta fase, diseñé 6 de las 9 pantallas que finalmente se han creado para la aplicación. Estos diseños fueron clave para visualizar la estructura y el flujo de la aplicación antes de iniciar el desarrollo.

### ***2.2.3 Fase de Preparación de la Infraestructura (1 semana) – Marzo***

**Creación de la Base de Datos (1 semana):** Durante la primera semana de marzo, me enfoqué en la creación de la base de datos, asegurándome de que estuviera bien estructurada y optimizada para gestionar eficientemente la información que la aplicación necesitaría procesar. Para este propósito, utilicé PGAdmin, una herramienta que me permitió diseñar y administrar la base de datos de manera efectiva.

### ***2.2.4 Fase de Obtención de Datos (3 semanas) – Marzo***

**Recopilación de Datos de la Agencia (3 semanas):** Durante las últimas tres semanas de marzo, me dediqué a recopilar datos reales para la aplicación, pero al darme cuenta de que la única agencia con la que tenía más contacto y de la que pude obtener información (**Kiwop**) no contaba con suficientes datos, tomé la decisión de crear una empresa ficticia bajo el nombre de mi propia agencia de marketing, Germán Hernández Mairal. Esta agencia ficticia me permitió avanzar en el desarrollo del Mínimo Producto Viable (MVP) de la aplicación.

Con la ayuda de ChatGPT, generé 9 archivos CSV que cargué en Trello mediante código, simulando proyectos para integrarlos en la aplicación. De esos 9 archivos, decidí cargar solo 3 en mi aplicación, manteniendo el enfoque lo más simple y funcional posible para este primer Mínimo Producto Viable.

La seguridad de los datos cargados se basó en las credenciales, sin implementar encriptación específica para los datos financieros.

### ***2.2.5 Fase de Desarrollo (12 semanas) - Finales de Abril, Mayo, Junio, Julio y Agosto***

**Desarrollo Backend y Frontend de la Aplicación:** Desde el 22 de abril hasta el 8 de agosto, me dediqué al desarrollo tanto del backend como del frontend de la aplicación.

Durante la fase de desarrollo del backend, implementé el código necesario para cargar los datos desde Trello hacia mi base de datos, estableciendo la lógica del servidor y gestionando la base de datos para asegurar la correcta manipulación y almacenamiento de la información. También desarrollé el código responsable de realizar los cálculos y proporcionar la información necesaria para el lado del cliente, garantizando que los datos se presentaran de manera coherente y útil al usuario. Además, traté de implementar medidas de seguridad en el código para proteger los datos involucrados en la aplicación, asegurando su integridad y confidencialidad.

En la fase de desarrollo del frontend, trabajé en la creación de una interfaz de usuario que no solo fuera visualmente atractiva, sino también intuitiva y funcional en web. Presté especial atención a la experiencia del usuario (UX), asegurándome de que la interacción con la aplicación fuera fluida y que cumpliera con las expectativas de los usuarios finales. Para lograr esto, me aseguré de que el frontend recibiera correctamente los datos del backend, garantizando que la comunicación entre ambos fuera fluida y precisa.

### ***2.2.6 Fase de Documentación y Finalización (4 Semanas) - Agosto***

**Redacción de la Memoria del Trabajo (2 semanas):** Durante las dos primeras semanas de agosto, me enfoqué en la redacción de la memoria del trabajo, documentando

en detalle todo el proceso de desarrollo, los desafíos enfrentados y las soluciones que implementé. Esta fase fue crucial para dejar constancia del trabajo realizado y para asegurar que todo el conocimiento adquirido durante el proyecto quedara bien registrado.

Simultáneamente, realicé una **revisión exhaustiva de la interfaz de la aplicación**. Aproveché el insight obtenido durante la documentación para identificar y realizar pequeños ajustes que mejoraran la estética y simplicidad de la aplicación, haciendo que fuera más atractiva y fácil de usar

**Preparación de la Presentación y del Vídeo (2 semanas):** En las últimas dos semanas de agosto, me dediqué a la preparación de la presentación y el vídeo, destacando los aspectos más importantes del proyecto, mostrando su funcionamiento y resaltando los logros alcanzados durante el desarrollo de la aplicación.

### 3 Requisitos del Proyecto

#### 3.1 Requisitos Funcionales

##### 3.1.1 Casos de Uso de la Pantalla Bienvenida

###### **Caso de Uso 1: Visualización de la Pantalla de Bienvenida**

**Título:** Visualización de la Pantalla de Bienvenida

**Actores:** Usuario

**Descripción:** Al cargar la aplicación, el usuario ve la pantalla de bienvenida, que proporciona una introducción a la aplicación y destaca sus características principales.

**Flujo Principal:**

- El usuario accede a la URL de la aplicación.
- El sistema carga la pantalla de bienvenida.
- Se muestra un mensaje de bienvenida al usuario.
- Se despliega un "hook" que resalta el valor y los resultados que ofrece la aplicación.
- El sistema muestra un botón etiquetado como "Empezar".
- Se presentan las características principales de la aplicación, destacando lo que el usuario puede esperar al usarla.

**Precondiciones:**

- El usuario no ha iniciado sesión.
- La aplicación se carga correctamente en el navegador.

**Postcondiciones:**

- El usuario permanece en la pantalla de bienvenida hasta que decida interactuar con ella.

###### **Caso de Uso 2: Navegación a la Pantalla de Inicio de Sesión**

**Título:** Navegación a la Pantalla de Inicio de Sesión

**Actores:** Usuario

**Descripción:** El usuario, tras haber visualizado la pantalla de bienvenida, selecciona el botón "Empezar", lo que lo redirige a la pantalla de inicio de sesión para que pueda autenticarse en la aplicación.

**Flujo Principal:**

- El usuario visualiza la pantalla de bienvenida.
- El usuario hace clic en el botón "Empezar".
- El sistema redirige al usuario a la pantalla de inicio de sesión.

- Se carga la pantalla de inicio de sesión, donde el usuario puede ingresar sus credenciales.

**Precondiciones:**

- El usuario está en la pantalla de bienvenida.
- El botón "Empezar" está visible y habilitado.

**Postcondiciones:**

- El usuario es llevado a la pantalla de inicio de sesión y puede proceder a autenticarse en la aplicación.

### *3.1.2 Casos de Uso de la Pantalla de Inicio de Sesión*

#### **Caso de Uso 1: Cargar la Pantalla de Inicio de Sesión**

**Título:** Cargar la Pantalla de Inicio de Sesión

**Actores:** Usuario

**Descripción:** El usuario accede a la pantalla de inicio de sesión para ingresar sus credenciales o registrarse en la aplicación.

**Flujo Principal:**

- El usuario carga la URL de la pantalla de inicio de sesión.
- El sistema muestra un formulario donde el usuario puede iniciar sesión o registrarse si no tiene una cuenta.
- Si el usuario tiene una cuenta, se le permite ingresar sus credenciales (nombre de usuario y contraseña).
- Si el usuario no tiene una cuenta, se le da la opción de registrarse.

**Precondiciones:**

- El usuario no ha iniciado sesión o ha sido redirigido a la pantalla de inicio de sesión.
- La pantalla de inicio de sesión carga correctamente.

**Postcondiciones:**

- El usuario no ha iniciado sesión o ha sido redirigido a la pantalla de inicio de sesión.
- La pantalla de inicio de sesión carga correctamente.

#### **Caso de Uso 2: Registro de Usuario**

**Título:** Registro de Usuario

**Actores:** Usuario

**Descripción:** El usuario, que no tiene una cuenta, puede registrarse proporcionando los detalles necesarios.

**Flujo Principal:**

- El usuario selecciona la opción de registrarse.
- Se muestra un formulario donde el usuario ingresa su nombre de usuario, contraseña, y confirma la contraseña.
- El usuario puede optar por subir una imagen de perfil.
- El usuario ingresa su API Key y Token de Trello.
- El sistema valida que el nombre de usuario no esté en uso y que las contraseñas coincidan.
- El sistema guarda las credenciales y carga los proyectos del usuario desde Trello.
- El sistema guarda la imagen de perfil si se proporcionó.
- El sistema genera un token para el usuario y lo guarda en localStorage.
- El usuario es redirigido a la pantalla principal de la aplicación.

### **Precondiciones:**

- El usuario no tiene una cuenta en la aplicación.
- El formulario de registro está visible y funcional.

### **Postcondiciones:**

- El usuario está registrado, con su perfil configurado y proyectos cargados.

## **Caso de Uso 3: Iniciar Sesión**

**Título:** Iniciar Sesión

**Actores:** Usuario

**Descripción:** El usuario ingresa sus credenciales para iniciar sesión en la aplicación.

### **Flujo Principal:**

- El usuario ingresa su nombre de usuario y contraseña.
- El sistema verifica las credenciales ingresadas.
- Si las credenciales son correctas, el sistema carga la imagen de perfil, API Key y Token del usuario.
- El usuario es redirigido a la pantalla principal de la aplicación, donde se muestra toda la información relevante a su cuenta.

### **Precondiciones:**

- El usuario tiene una cuenta en la aplicación.
- El formulario de inicio de sesión está visible y funcional.
- El usuario ha iniciado sesión y es redirigido a la pantalla principal.

### **Postcondiciones:**

- El usuario ha iniciado sesión y es redirigido a la pantalla principal.

## **Caso de Uso 4: Gestión de la Información del Usuario**

**Título:** Gestión de la Información del Usuario

**Actores:** Usuario

**Descripción:** El usuario puede modificar su información personal como el nombre, contraseña e imagen de perfil.

**Flujo Principal:**

- El usuario, tras iniciar sesión, selecciona la opción para editar su información personal.
- El sistema muestra un formulario donde el usuario puede modificar su nombre, contraseña e imagen de perfil.
- El usuario realiza los cambios y los guarda.
- El sistema actualiza la información en la base de datos y refleja los cambios en todas las tablas relacionadas.
- El sistema redirige al usuario a la pantalla principal con la información actualizada.

**Precondiciones:**

- El usuario está registrado e iniciado sesión en la aplicación.
- El formulario de edición de información está visible y funcional.

**Postcondiciones:**

- La información del usuario se actualiza y se refleja en todas las instancias relevantes de la aplicación.

### **Caso de Uso 5: Manejo de Errores en Inicio de Sesión**

**Título:** Manejo de Errores en Inicio de Sesión

**Actores:** Usuario

**Descripción:** El sistema maneja los errores cuando el usuario ingresa credenciales incorrectas.

**Flujo Principal:**

- El usuario ingresa su nombre de usuario y contraseña.
- El sistema verifica las credenciales ingresadas.
- Si las credenciales son incorrectas, el sistema muestra un mensaje de error indicando el problema.
- El usuario puede volver a intentar iniciar sesión.

**Precondiciones:**

- El usuario intenta iniciar sesión en la aplicación.
- Las credenciales ingresadas son incorrectas.

**Postcondiciones:**

- El usuario es informado del error y puede corregirlo o intentarlo nuevamente

### **3.1.3 Casos de Uso de la 1ª Pantalla (Informe de Costos por Proyecto)**

#### **Caso de Uso 1: Cargar la Pantalla de Informe de Costos por Proyecto**

**Título:** Cargar la Pantalla de Informe de Costos por Proyecto

**Actores:** Usuario

**Descripción:** El usuario carga la pantalla de Informe de Costos por Proyecto, donde puede seleccionar y visualizar datos sobre diferentes proyectos.

**Flujo Principal:**

- El usuario accede a la URL de la pantalla de Informe de Costos por Proyecto.
- El sistema muestra la lista de proyectos disponibles en la base de datos.
- Las tres secciones de la pantalla aparecen vacías, esperando que el usuario seleccione un proyecto para mostrar los datos correspondientes.

**Precondiciones:**

- El usuario ha iniciado sesión y está autorizado para ver los informes de costos

**Postcondiciones:**

- La lista de proyectos está cargada y visible, pero sin información específica mostrada hasta que se seleccione un proyecto.

## **Caso de Uso 2: Seleccionar un Proyecto**

**Título:** Seleccionar un Proyecto

**Actores:** Usuario

**Descripción:** El usuario selecciona un proyecto de la lista para ver el informe de costos detallado

**Flujo Principal:**

- El usuario selecciona un proyecto de la lista presentada en la pantalla.
- El sistema muestra una ventana emergente solicitando al usuario que seleccione el período para el cual desea ver la información del proyecto (uno en concreto o todos los períodos).
- El usuario selecciona un período o decide ver todos los datos sin filtrar por un período específico.
- El sistema recupera y muestra la información del proyecto seleccionado en las tres secciones de la pantalla, incluyendo las horas trabajadas y los costos específicos al período seleccionado.

**Precondiciones:**

- La lista de proyectos está cargada en la pantalla.

**Postcondiciones:**

- Las tres secciones de la pantalla se actualizan con la información del proyecto seleccionado, según el período elegido.

## **Caso de Uso 3: Deseleccionar un Proyecto**

**Título:** Deseleccionar un Proyecto

**Actores:** Usuario

**Descripción:** El usuario deselecciona un proyecto previamente seleccionado para volver al estado inicial de la pantalla.

**Flujo Principal:**

- El usuario deselecciona un proyecto que previamente había seleccionado.
- El sistema restablece las tres secciones de la pantalla a su estado base, sin mostrar información específica

**Precondiciones:**

- El usuario ha seleccionado previamente un proyecto.

**Postcondiciones:**

- La pantalla vuelve a su estado base, sin mostrar información de ningún proyecto.

#### **Caso de Uso 4: Seleccionar Múltiples Proyectos**

**Título:** Seleccionar Múltiples Proyectos

**Actores:** Usuario

**Descripción:** El usuario selecciona varios proyectos consecutivamente para comparar sus informes de costos.

**Flujo Principal:**

- El usuario selecciona un segundo proyecto mientras ya tiene uno seleccionado.
- El sistema actualiza las dos primeras secciones de la pantalla con la información del último proyecto seleccionado.
- En la tercera sección, el sistema añade la información del segundo proyecto seleccionado al gráfico.

**Precondiciones:**

- El usuario ha seleccionado un proyecto y desea seleccionar otro adicional.

**Postcondiciones:**

- Las dos primeras secciones muestran la información del último proyecto seleccionado, mientras que la tercera sección compara ambos proyectos.

#### **Caso de Uso 5: Deseleccionar un Proyecto al Comparar Múltiples Proyectos**

**Título:** Deseleccionar un Proyecto al Comparar Múltiples Proyectos

**Actores:** Usuario

**Descripción:** El usuario deselecciona un proyecto mientras compara múltiples proyectos para modificar la visualización.

**Flujo Principal:**

- El usuario deselectiona uno de los proyectos que ha seleccionado.
- El sistema actualiza las tres secciones:
  - Las dos primeras secciones mantienen la información del último proyecto que permanece seleccionado.
  - La tercera sección elimina la información del proyecto deselectionado del gráfico.

### **Precondiciones:**

- El usuario ha seleccionado múltiples proyectos y desea modificar la comparación.

### **Postcondiciones:**

- La pantalla muestra la información de los proyectos restantes, sin el proyecto deselectionado.

## **Caso de Uso 6: Seleccionar Otro Proyecto Después de Deseleccionar**

**Título:** Seleccionar Otro Proyecto Después de Deseleccionar

**Actores:** Usuario

**Descripción:** El usuario selecciona otro proyecto después de haber deselectionado todos los anteriores.

### **Flujo Principal:**

- El usuario selecciona un nuevo proyecto tras haber deselectionado otros.
- El sistema actualiza las dos primeras secciones con la información del nuevo proyecto seleccionado.
- En la tercera sección, el sistema agrega el nuevo proyecto al gráfico.

### **Precondiciones:**

- El usuario ha deselectionado previamente todos los proyectos.

### **Postcondiciones:**

- La pantalla muestra la información del nuevo proyecto seleccionado.

## **Caso de Uso 7: Navegar a Otra Página**

**Título:** Navegar a Otra Página

**Actores:** Usuario

**Descripción:** El usuario decide navegar a otra sección de la aplicación para ver información adicional.

### **Flujo Principal:**

- El usuario selecciona una opción de menú o botón para navegar a otra página de la aplicación.
- El sistema redirige al usuario a la nueva página seleccionada.

### **Precondiciones:**

- El usuario tiene acceso a otras secciones de la aplicación.

**Postcondiciones:**

- El usuario es redirigido a la nueva página.

### ***3.1.4 Casos de Uso de la 2ª Pantalla (Análisis de Rentabilidad por Cliente)***

#### **Caso de Uso 1: Cargar la Pantalla de Análisis de Rentabilidad por Cliente**

**Título:** Cargar la Pantalla de Análisis de Rentabilidad por Cliente

**Actores:** Usuario

**Descripción:** El usuario carga la pantalla de Análisis de Rentabilidad por Cliente, donde puede seleccionar un cliente para ver los ingresos generados y la rentabilidad específica.

**Flujo Principal:**

- El usuario accede a la URL de la pantalla de Análisis de Rentabilidad por Cliente.
- El sistema muestra la lista de clientes disponibles en la base de datos.
- Las tres secciones de la pantalla están vacías, esperando que el usuario seleccione un cliente para mostrar los datos correspondientes.

**Precondiciones:**

- El usuario ha iniciado sesión y está autorizado para ver los análisis de rentabilidad.

**Postcondiciones:**

- La lista de clientes está cargada y visible, pero sin información específica mostrada hasta que se seleccione un cliente.

#### **Caso de Uso 2: Seleccionar un Cliente**

**Título:** Seleccionar un Cliente

**Actores:** Usuario

**Descripción:** El usuario selecciona un cliente de la lista para ver el análisis de rentabilidad específico de ese cliente.

**Flujo Principal:**

- El usuario selecciona un cliente de la lista presentada en la pantalla.
- El sistema muestra una ventana emergente solicitando al usuario que seleccione el período para el cual desea ver la información de ingresos generados por ese cliente (uno en específico o todos los períodos).
- El usuario selecciona un período o decide ver todos los datos sin filtrar por un período específico.

- El sistema recupera y muestra la información del cliente seleccionado en las tres secciones de la pantalla, incluyendo los ingresos generados y la rentabilidad para el período seleccionado.

**Precondiciones:**

- La lista de clientes está cargada en la pantalla.

**Postcondiciones:**

- Las tres secciones de la pantalla se actualizan con la información del cliente seleccionado, según el período elegido.

**Caso de Uso 3: Deseleccionar un Cliente**

**Título:** Deseleccionar un Cliente

**Actores:** Usuario

**Descripción:** El usuario deselectiona un cliente previamente seleccionado para volver al estado inicial de la pantalla.

**Flujo Principal:**

- El usuario deselectiona un cliente que previamente había seleccionado.
- El sistema restablece las tres secciones de la pantalla a su estado base, sin mostrar información específica.

**Precondiciones:**

- El usuario ha seleccionado previamente un cliente.

**Postcondiciones:**

- La pantalla vuelve a su estado base, sin mostrar información de ningún cliente.

**Caso de Uso 4: Seleccionar Múltiples Clientes**

**Título:** Seleccionar Múltiples Clientes

**Actores:** Usuario

**Descripción:** El usuario selecciona varios clientes consecutivamente para comparar sus rentabilidades e ingresos generados.

**Flujo Principal:**

- El usuario selecciona un segundo cliente mientras ya tiene uno seleccionado.
- El sistema actualiza la primera y segunda sección de la pantalla con la información del último cliente seleccionado.
- La tercera sección de la pantalla muestra la información combinada de ambos clientes, permitiendo la comparación de la rentabilidad.

**Precondiciones:**

- El usuario ha seleccionado un cliente y desea seleccionar otro adicional.

**Postcondiciones:**

- La pantalla muestra la información combinada de los clientes seleccionados.

### **Caso de Uso 5: Deseleccionar un Cliente al Comparar Múltiples Clientes**

**Título:** Deseleccionar un Cliente al Comparar Múltiples Clientes

**Actores:** Usuario

**Descripción:** El usuario deselectiona un cliente mientras compara múltiples clientes para modificar la visualización.

**Flujo Principal:**

- El usuario deselectiona uno de los clientes que ha seleccionado.
- El sistema actualiza las tres secciones:
  - La primera sección mantiene la información del último cliente que permanece seleccionado.
  - La segunda y tercera secciones eliminan la información del cliente deselectionado del análisis, manteniendo la información del cliente restante.

**Precondiciones:**

- El usuario ha seleccionado múltiples clientes y desea modificar la comparación.

**Postcondiciones:**

- La pantalla muestra la información de los clientes restantes, sin el cliente deselectionado.

### **Caso de Uso 6: Seleccionar Otro Cliente Después de Deseleccionar**

**Título:** Seleccionar Otro Cliente Después de Deseleccionar

**Actores:** Usuario

**Descripción:** El usuario selecciona otro cliente después de haber deselectionado todos los anteriores.

**Flujo Principal:**

- El usuario selecciona un nuevo cliente tras haber deselectionado otros.
- El sistema actualiza la primera sección con la información del nuevo cliente seleccionado.
- La segunda y tercera secciones se actualizan con la información del nuevo cliente.

**Precondiciones:**

- El usuario ha deselectionado previamente todos los clientes.

**Postcondiciones:**

- La pantalla muestra la información del nuevo cliente seleccionado.

### **Caso de Uso 7: Navegar a Otra Página**

**Título:** Navegar a Otra Página

**Actores:** Usuario

**Descripción:** El usuario decide navegar a otra sección de la aplicación para ver información adicional.

**Flujo Principal:**

- El usuario selecciona una opción de menú o botón para navegar a otra página de la aplicación.
- El sistema redirige al usuario a la nueva página seleccionada

**Precondiciones:**

- El usuario tiene acceso a otras secciones de la aplicación.

**Postcondiciones:**

- El usuario es redirigido a la nueva página.

### **3.1.5 Casos de Uso de la 3ª Pantalla (Visualización de Cargas de Trabajo)**

#### **Caso de Uso 1: Cargar la Pantalla de Visualización de Cargas de Trabajo**

**Título:** Cargar la Pantalla de Visualización de Cargas de Trabajo

**Actores:** Usuario

**Descripción:** El usuario carga la pantalla de Visualización de Cargas de Trabajo, donde puede seleccionar miembros de la agencia para ver su productividad y tareas completadas.

**Flujo Principal:**

- El usuario accede a la URL de la pantalla de Visualización de Cargas de Trabajo.
- El sistema muestra la lista de miembros de la agencia disponibles.
- Los gráficos están vacíos y muestran un mensaje que indica al usuario que seleccione un miembro para visualizar su productividad.

**Precondiciones:**

- El usuario ha iniciado sesión y está autorizado para ver la visualización de cargas de trabajo.

**Postcondiciones:**

- La lista de miembros está cargada, y los gráficos están en estado inicial, esperando la selección de un miembro.

#### **Caso de Uso 2: Seleccionar un Miembro**

**Título:** Seleccionar un Miembro

**Actores:** Usuario

**Descripción:** El usuario selecciona un miembro de la lista para ver su productividad y las tareas que ha completado en un período de tiempo específico

**Flujo Principal:**

- El usuario selecciona un miembro de la lista presentada en la pantalla.
- El sistema muestra una ventana emergente solicitando al usuario que seleccione el período de tiempo para el cual desea ver la información de productividad del miembro (uno en específico o todos los períodos).
- El usuario selecciona un período o decide ver todos los datos sin filtrar por un período específico.
- El sistema recupera y muestra la información del miembro seleccionado en los dos gráficos:
  - El número de tareas completadas por el miembro.
  - La duración que ha tardado en completar esas tareas.

**Precondiciones:**

- La lista de miembros está cargada en la pantalla.

**Postcondiciones:**

- Los gráficos se actualizan para mostrar la información de productividad del miembro seleccionado.

### **Caso de Uso 3: Deseleccionar un Miembro**

**Título:** Deseleccionar un Miembro

**Actores:** Usuario

**Descripción:** El usuario deselectiona un miembro previamente seleccionado, volviendo a dejar los gráficos en estado inicial.

**Flujo Principal:**

- El usuario deselectiona un miembro que previamente había seleccionado.
- El sistema elimina el cuadro de selección del miembro deselectionado.
- Los gráficos vuelven a su estado inicial, mostrando el mensaje de que se seleccione un miembro para visualizar su productividad.

**Precondiciones:**

- El usuario ha seleccionado previamente un miembro.

**Postcondiciones:**

- Los gráficos vuelven a su estado base, sin mostrar información específica.

### **Caso de Uso 4: Seleccionar un Segundo Miembro**

**Título:** Seleccionar un Segundo Miembro

**Actores:** Usuario

**Descripción:** El usuario selecciona un segundo miembro de la lista para comparar su productividad con la del primer miembro seleccionado.

**Flujo Principal:**

- El usuario selecciona un segundo miembro de la lista mientras ya tiene uno seleccionado.
- El sistema mantiene el cuadro de selección en ambos miembros.
- Los gráficos se actualizan para incluir la información del segundo miembro seleccionado:
  - El primer gráfico muestra el número de tareas completadas y la duración para ambos miembros, sumando las duraciones si coinciden.
  - El segundo gráfico diferencia las tareas completadas por cada miembro con colores distintos, mostrando una barra única con secciones de colores diferentes si las tareas coinciden en duración.

**Precondiciones:**

- El usuario ha seleccionado un primer miembro.

**Postcondiciones:**

- Los gráficos muestran la productividad combinada de los miembros seleccionados, diferenciando sus tareas por colores en el segundo gráfico.

### **Caso de Uso 5: Deseleccionar un Miembro al Comparar Múltiples Miembros**

**Título:** Deseleccionar un Miembro al Comparar Múltiples Miembros

**Actores:** Usuario

**Descripción:** El usuario deselecta uno de los miembros mientras compara múltiples miembros para modificar la visualización.

**Flujo Principal:**

- El usuario deselecta uno de los miembros que ha seleccionado.
- El sistema elimina el cuadro de selección del miembro deselectado.
- Los gráficos se actualizan eliminando la información del miembro deselectado, mientras mantienen la información del miembro restante.

**Precondiciones:**

- El usuario ha seleccionado múltiples miembros y desea modificar la comparación.

**Postcondiciones:**

- Los gráficos muestran la información de los miembros restantes, sin el miembro deselectado.

### **Caso de Uso 6: Navegar a Otra Página**

**Título:** Navegar a Otra Página

**Actores:** Usuario

**Descripción:** El usuario decide navegar a otra sección de la aplicación para ver información adicional.

**Flujo Principal:**

- El usuario selecciona una opción de menú o botón para navegar a otra página de la aplicación.
- El sistema redirige al usuario a la nueva página seleccionada.

**Precondiciones:**

- El usuario tiene acceso a otras secciones de la aplicación.

**Postcondiciones:**

- El usuario es redirigido a la nueva página.

### ***3.1.6 Casos de Uso de la 4ª Pantalla (Análisis de Servicio por Cliente)***

#### **Caso de Uso 1: Cargar la Página de Análisis de Servicio por Cliente**

**Título:** Cargar la Página de Análisis de Servicio por Cliente

**Actores:** Usuario

**Descripción:** El usuario carga la página de Análisis de Servicio por Cliente.

**Flujo Principal:**

- El usuario accede a la 4ª pantalla de la aplicación.
- El sistema carga la lista de servicios y clientes disponibles.
- Las secciones de gráficos muestran el mensaje "Selecciona un Servicio y un Cliente para Visualizar su Rentabilidad".

**Precondiciones:**

- El usuario ha iniciado sesión correctamente.
- La aplicación tiene acceso a los datos de servicios y clientes.

**Postcondiciones:**

- El usuario es redirigido a la nueva página.

#### **Caso de Uso 2: Seleccionar un Servicio**

**Título:** Seleccionar un Servicio

**Actores:** Usuario

**Descripción:** El usuario selecciona un servicio de la lista disponible.

**Flujo Principal:**

- El usuario selecciona un servicio en la lista de servicios.
- El sistema resalta el servicio seleccionado con un rectángulo de color.
- Las secciones de gráficos actualizan el mensaje a "Selecciona un Cliente para Visualizar su Rentabilidad".

**Precondiciones:**

- La lista de servicios está cargada y visible.

**Postcondiciones:**

- El servicio seleccionado se resalta y está listo para ser asociado a un cliente

**Caso de Uso 3: Seleccionar un Cliente**

**Título:** Seleccionar un Cliente

**Actores:** Usuario

**Descripción:** El usuario selecciona un cliente después de haber seleccionado un servicio.

**Flujo Principal:**

- El usuario selecciona un cliente de la lista disponible.
- El sistema muestra una ventana para que el usuario seleccione un período de tiempo.
- El usuario selecciona un período específico o decide no seleccionar ninguno.
- El sistema actualiza los gráficos para mostrar los ingresos generados por el servicio seleccionado para el cliente en el período especificado.

**Precondiciones:**

- Un servicio ya ha sido seleccionado.
- La lista de clientes está cargada y visible.

**Postcondiciones:**

- Los gráficos muestran los ingresos generados según las selecciones del usuario.

**Caso de Uso 4: Deseleccionar el Cliente**

**Título:** Deseleccionar el Cliente

**Actores:** Usuario

**Descripción:** El usuario deselectiona un cliente previamente seleccionado.

**Flujo Principal:**

- El usuario deselectiona el cliente seleccionado.
- El sistema elimina los datos del cliente de los gráficos.
- Los gráficos vuelven a su estado inicial, mostrando el mensaje "Selecciona un Cliente para Visualizar su Rentabilidad".

**Precondiciones:**

- Un cliente ya ha sido seleccionado.

**Postcondiciones:**

- Los gráficos regresan al estado de espera inicial.

### **Caso de Uso 5: Deseleccionar el Servicio**

**Título:** Deseleccionar el Servicio

**Actores:** Usuario

**Descripción:** El usuario deselecciona el servicio previamente seleccionado.

**Flujo Principal:**

- El usuario deselecciona el servicio seleccionado.
- El sistema elimina los gráficos asociados a ese servicio.
- El mensaje de "Selecciona un Servicio y un Cliente para Visualizar su Rentabilidad" reaparece.

**Precondiciones:**

- Un servicio ya ha sido seleccionado.

**Postcondiciones:**

- Los gráficos se eliminan y la página vuelve al estado inicial de espera.

### **Caso de Uso 6: Usuario Ha Deseleccionado un Servicio o un Cliente, Estando Seleccionado el Otro**

**Título:** Usuario Ha Deseleccionado un Servicio o un Cliente, Estando Seleccionado el Otro

**Actores:** Usuario

**Descripción:** El usuario deselecciona un servicio o un cliente mientras el otro elemento (servicio o cliente) sigue seleccionado.

**Flujo Principal:**

- El usuario deselecciona el servicio o cliente seleccionado.
- El sistema elimina los gráficos asociados.
- Las secciones de gráficos muestran el mensaje "Selecciona un Servicio y un Cliente para Visualizar su Rentabilidad".

**Precondiciones:**

- Un servicio y un cliente ya han sido seleccionados.

**Postcondiciones:**

- Los gráficos desaparecen y el sistema indica que se debe seleccionar un servicio y un cliente.

### **Caso de Uso 7: Seleccionar un Segundo Cliente**

**Título:** Seleccionar un Segundo Cliente

**Actores:** Usuario

**Descripción:** El usuario selecciona un segundo cliente después de haber seleccionado un primer cliente y un servicio.

**Flujo Principal:**

- El usuario selecciona un segundo cliente en la lista de clientes.
- El sistema actualiza los gráficos para incluir los ingresos generados por ambos clientes para el servicio seleccionado.
- Si los ingresos para ambos clientes ocurren en el mismo período, se apilan en la misma barra, diferenciados por color.

**Precondiciones:**

- Un primer cliente y un servicio ya han sido seleccionados.

**Postcondiciones:**

- Los gráficos muestran los ingresos de ambos clientes, apilados por período si corresponde.

### **Caso de Uso 8: Deseleccionar el Segundo Cliente**

**Título:** Deseleccionar el Segundo Cliente

**Actores:** Usuario

**Descripción:** El usuario deselectiona el segundo cliente seleccionado.

**Flujo Principal:**

- El usuario deselectiona el segundo cliente.
- El sistema elimina la información del segundo cliente de los gráficos.
- Los gráficos se actualizan para mostrar únicamente los datos del primer cliente.

**Precondiciones:**

- Un segundo cliente ya ha sido seleccionado.

**Postcondiciones:**

- Los gráficos eliminan los datos del segundo cliente, manteniendo los del primer cliente.

### **Caso de Uso 9: Navegar a Otra Página**

**Título:** Navegar a Otra Página

**Actores:** Usuario

**Descripción:** El usuario decide navegar a otra sección de la aplicación para ver información adicional.

**Flujo Principal:**

- El usuario selecciona una opción de menú o botón para navegar a otra página de la aplicación.
- El sistema redirige al usuario a la nueva página seleccionada.

**Precondiciones:**

- El usuario tiene acceso a otras secciones de la aplicación.

**Postcondiciones:**

- El usuario es redirigido a la nueva página.

**3.1.7 Casos de Uso de la 5ª Pantalla (Estadísticas de Actividades por Servicio)**

**Caso de Uso 1: Cargar la Página de Estadísticas de Actividades por Servicio**

**Título:** Cargar la Página de Estadísticas de Actividades por Servicio

**Actores:** Usuario

**Descripción:** El usuario carga la página de Estadísticas de Actividades por Servicio para visualizar información sobre las tareas del cliente.

**Flujo Principal:**

- El usuario accede a la 5ª pantalla de la aplicación.
- El sistema carga y muestra toda la información de los Servicios del cliente para el que trabaja.

**Precondiciones:**

- El usuario ha iniciado sesión correctamente.
- La aplicación tiene acceso a los datos de los servicios del cliente.

**Postcondiciones:**

- La página muestra correctamente la información de los servicios del cliente.

**Caso de Uso 2: Seleccionar un Servicio**

**Título:** Seleccionar un Servicio

**Actores:** Usuario

**Descripción:** El usuario selecciona un servicio específico para filtrar las tareas mostradas en función de ese servicio.

**Flujo Principal:**

- El usuario selecciona un servicio de la lista disponible.
- El sistema muestra una ventana donde el usuario puede seleccionar un período de tiempo para visualizar las tareas completadas.
- El usuario selecciona un período específico o decide no seleccionar ninguno.
- El sistema filtra y muestra información y gráficos de las tareas correspondientes al servicio seleccionado y al período especificado (si corresponde).
- El servicio seleccionado queda visualmente resaltado en la pantalla.

**Precondiciones:**

- La lista de servicios está cargada y visible.
- Los datos de tareas están disponibles.

**Postcondiciones:**

- Las tareas mostradas corresponden al servicio y período seleccionados.

**Caso de Uso 3: Navegar a Otra Página**

**Título:** Navegar a Otra Página

**Actores:** Usuario

**Descripción:** El usuario decide navegar a otra sección de la aplicación desde la pantalla de Estadísticas de Actividades por Servicio.

**Flujo Principal:**

- El usuario selecciona una opción de menú o botón para navegar a otra página de la aplicación.
- El sistema redirige al usuario a la nueva página seleccionada

**Precondiciones:**

- El usuario tiene acceso a otras secciones de la aplicación

**Postcondiciones:**

- El usuario es redirigido a la nueva página.

**Caso de Uso 4: Usuario Selecciona un Período de Tiempo**

**Título:** Usuario Selecciona un Período de Tiempo

**Actores:** Usuario

**Descripción:** El usuario selecciona un período de tiempo para ver las tareas completadas dentro de ese intervalo para un servicio específico.

**Flujo Principal:**

- El usuario selecciona un servicio.
- El sistema muestra una ventana con la opción de seleccionar un período de tiempo.
- El usuario elige un período específico o decide ver todos los datos sin filtrado temporal.
- El sistema filtra las tareas según el período seleccionado y actualiza la visualización.

**Precondiciones:**

- Un servicio ha sido seleccionado.
- Los datos de tareas están disponibles para filtrado.

**Postcondiciones:**

- Las tareas se muestran en función del período seleccionado o sin filtrado temporal si no se selecciona un período.

### **Caso de Uso 5: Usuario No Selecciona un Período de Tiempo**

**Título:** Usuario No Selecciona un Período de Tiempo

**Actores:** Usuario

**Descripción:** El usuario decide no seleccionar un período de tiempo y visualiza todas las tareas completadas para el servicio seleccionado.

**Flujo Principal:**

- El usuario selecciona un servicio.
- El sistema muestra una ventana con la opción de seleccionar un período de tiempo.
- El usuario opta por no seleccionar ningún período y solicita ver todas las tareas disponibles.
- El sistema muestra todas las tareas completadas para el servicio seleccionado sin filtrar por período.

**Precondiciones:**

- Un servicio ha sido seleccionado.
- Los datos de tareas están disponibles.

**Postcondiciones:**

- Todas las tareas asociadas al servicio seleccionado se muestran sin restricción de período.

### **Caso de Uso 6: Guardar la Selección del Período de Tiempo**

**Título:** Guardar la Selección del Período de Tiempo

**Actores:** Usuario

**Descripción:** La selección del período de tiempo por parte del usuario se guarda para futuras interacciones dentro de la misma pantalla.

**Flujo Principal:**

- El usuario selecciona un período de tiempo al visualizar tareas para un servicio.
- El sistema guarda esta selección de período.
- El usuario continúa interactuando con la pantalla, y el sistema sigue utilizando el período seleccionado para futuras acciones.
- Si el usuario sale de la pantalla y vuelve a entrar, se le pide nuevamente seleccionar un período de tiempo.

**Precondiciones:**

- El usuario ha seleccionado un período de tiempo.

**Postcondiciones:**

- El sistema recuerda el período seleccionado mientras el usuario permanezca en la misma pantalla, solicitando una nueva selección si el usuario abandona y regresa.

### **3.1.8 Casos de Uso de la 6ª Pantalla (Seguimiento del Tiempo)**

#### **Caso de Uso 1: Cargar la Página de Seguimiento del Tiempo**

**Título:** Cargar la Página de Seguimiento del Tiempo

**Actores:** Usuario

**Descripción:** El usuario carga la página de Seguimiento del Tiempo para visualizar información sobre los proyectos y miembros de la agencia.

**Flujo Principal:**

- El usuario accede a la 6a pantalla de la aplicación.
- El sistema carga la lista de todos los proyectos de la agencia en la barra lateral.
- Las pantallas muestran mensajes indicando que falta que el usuario interactúe con la página (elegir un proyecto y un miembro).

**Precondiciones:**

- El usuario ha iniciado sesión correctamente.
- La aplicación tiene acceso a los datos de proyectos y miembros.

**Postcondiciones:**

- La página muestra correctamente la lista de proyectos y los mensajes en las pantallas que requieren interacción del usuario.

#### **Caso de Uso 2: Seleccionar un Proyecto**

**Título:** Seleccionar un Proyecto

**Actores:** Usuario

**Descripción:** El usuario selecciona un proyecto de la lista para visualizar el tiempo dedicado a ese proyecto por los miembros de la agencia.

**Flujo Principal:**

- El usuario selecciona un proyecto de la lista disponible.
- El sistema muestra una ventana para que el usuario seleccione un período de tiempo para visualizar el tiempo dedicado.
- El usuario selecciona un período específico o decide no seleccionar ninguno.
- La 1a pantalla se actualiza con el nombre del proyecto y el tiempo total dedicado al proyecto por todos los miembros.
- La 2a pantalla muestra los miembros que han trabajado en el proyecto.
- La 3a pantalla muestra el gráfico correspondiente al proyecto seleccionado.

**Precondiciones:**

- La lista de proyectos está cargada y visible.
- Los datos de tiempo y miembros están disponibles.

**Postcondiciones:**

- La página muestra el tiempo total dedicado al proyecto seleccionado y los gráficos correspondientes.

**Caso de Uso 3: Deseleccionar el Proyecto**

**Título:** Deseleccionar el Proyecto

**Actores:** Usuario

**Descripción:** El usuario deselectiona el proyecto previamente seleccionado para volver al estado inicial de la página.

**Flujo Principal:**

- El usuario deselectiona el proyecto seleccionado.
- El sistema elimina la información del proyecto de todas las pantallas.
- Todas las pantallas vuelven al estado base, mostrando los mensajes de espera de interacción.

**Precondiciones:**

- Un proyecto ya ha sido seleccionado.

**Postcondiciones:**

- La página regresa al estado inicial sin información del proyecto.

**Caso de Uso 4: Seleccionar un Miembro**

**Título:** Seleccionar un Miembro

**Actores:** Usuario

**Descripción:** El usuario selecciona un miembro específico para visualizar el tiempo que ha dedicado al proyecto seleccionado.

**Flujo Principal:**

- El usuario selecciona un miembro de la lista de miembros que han trabajado en el proyecto.
- La 2a pantalla se actualiza con el tiempo total dedicado por el miembro al proyecto.
- La 3a pantalla se actualiza con el gráfico mostrando la información del miembro seleccionado.

**Precondiciones:**

- Un proyecto ya ha sido seleccionado.
- La lista de miembros que han trabajado en el proyecto está cargada y visible

**Postcondiciones:**

- La página muestra el tiempo dedicado por el miembro seleccionado al proyecto y actualiza los gráficos.

### **Caso de Uso 5: Seleccionar un Segundo Miembro**

**Título:** Seleccionar un Segundo Miembro

**Actores:** Usuario

**Descripción:** El usuario selecciona un segundo miembro para comparar el tiempo dedicado por múltiples miembros al proyecto seleccionado.

**Flujo Principal:**

- El usuario selecciona un segundo miembro de la lista.
- La 2a pantalla se actualiza para mostrar el tiempo dedicado por el último miembro seleccionado.
- La 3a pantalla se actualiza para incluir el tiempo dedicado por ambos miembros, diferenciados por color en el gráfico.

**Precondiciones:**

- Un primer miembro ya ha sido seleccionado.
- La lista de miembros que han trabajado en el proyecto está cargada y visible.

**Postcondiciones:**

- La página muestra el tiempo dedicado por ambos miembros al proyecto y actualiza los gráficos.

### **Caso de Uso 6: Seleccionar un Segundo Proyecto**

**Título:** Seleccionar un Segundo Proyecto

**Actores:** Usuario

**Descripción:** El usuario selecciona un segundo proyecto para visualizar y comparar el tiempo dedicado por los miembros en múltiples proyectos.

**Flujo Principal:**

- El usuario selecciona un segundo proyecto de la lista.
- La 1a pantalla se actualiza con el nombre del segundo proyecto y el tiempo total dedicado al proyecto por todos los miembros.
- La 3a pantalla se actualiza para incluir el tiempo dedicado al segundo proyecto en el gráfico.

**Precondiciones:**

- Un primer proyecto ya ha sido seleccionado.
- La lista de proyectos está cargada y visible.

**Postcondiciones:**

- La página muestra el tiempo dedicado al segundo proyecto seleccionado y actualiza los gráficos.

### **Caso de Uso 7: Deseleccionar un Miembro**

**Título:** Deseleccionar un Miembro

**Actores:** Usuario

**Descripción:** El usuario deselectiona un miembro para eliminar su información de los gráficos.

**Flujo Principal:**

- El usuario deselectiona el miembro seleccionado.
- El sistema elimina los datos del miembro de la 2a y 3a pantalla.
- La 2a pantalla vuelve a su estado inicial de espera para la selección de un miembro.
- La 3a pantalla actualiza el gráfico eliminando los datos del miembro deselectionado.

**Precondiciones:**

- Un miembro ya ha sido seleccionado.

**Postcondiciones:**

- La página elimina la información del miembro deselectionado y actualiza los gráficos.

### **Caso de Uso 8: Deseleccionar un Proyecto**

**Título:** Deseleccionar un Proyecto

**Actores:** Usuario

**Descripción:** El usuario deselectiona un proyecto específico para eliminar su información de los gráficos.

**Flujo Principal:**

- El usuario deselectiona el proyecto seleccionado.
- El sistema elimina los datos del proyecto de la 1a y 3a pantalla.
- La pantalla superior vuelve a su estado inicial de espera para la selección de un proyecto.
- La pantalla inferior actualiza el gráfico eliminando los datos del proyecto deselectionado.

**Precondiciones:**

- Un proyecto ya ha sido seleccionado.

**Postcondiciones:**

- La página elimina la información del proyecto deselectionado y actualiza los gráficos.

### **Caso de Uso 9: Navegar a Otra Página**

**Título:** Navegar a Otra Página

**Actores:** Usuario

**Descripción:** El usuario decide navegar a otra sección de la aplicación desde la pantalla de Seguimiento del Tiempo.

**Flujo Principal:**

- El usuario selecciona una opción de menú o botón para navegar a otra página de la aplicación.
- El sistema redirige al usuario a la nueva página seleccionada.

**Precondiciones:**

- El usuario tiene acceso a otras secciones de la aplicación.

**Postcondiciones:**

- El usuario es redirigido a la nueva página

### **3.1.9 Casos de Uso de la 7ª Pantalla (Back-Office)**

#### **Caso de Uso 1: Cargar la Página de Back-Office**

**Título:** Cargar la Página de Back-Office

**Actores:** Usuario

**Descripción:** El usuario carga la página de Back-Office para gestionar la información relacionada con los proyectos, clientes y miembros de la agencia.

**Flujo Principal:**

- El usuario accede a la 7ª pantalla de la aplicación.
- El sistema carga la lista de proyectos de la agencia.
- Un proyecto se selecciona por defecto, y la información del proyecto, cliente, y miembros relacionados se muestra en las respectivas pantallas.

**Precondiciones:**

- El usuario ha iniciado sesión correctamente.
- La aplicación tiene acceso a los datos de proyectos, clientes y miembros.

**Postcondiciones:**

- La página muestra correctamente la información del proyecto seleccionado por defecto y sus relaciones.

#### **Caso de Uso 2: Seleccionar un Proyecto**

**Título:** Seleccionar un Proyecto

**Actores:** Usuario

**Descripción:** El usuario selecciona un proyecto diferente para actualizar la información mostrada en la pantalla de Back-Office..

**Flujo Principal:**

- El usuario selecciona un proyecto de la lista de proyectos.
- El sistema muestra una ventana para que el usuario seleccione un período de tiempo para visualizar la información relacionada con el proyecto.
- El usuario selecciona un período específico o decide no seleccionar ninguno.
- La información del proyecto, cliente, miembros y miembro específico se actualiza en las respectivas pantallas.

**Precondiciones:**

- La lista de proyectos está cargada y visible.

**Postcondiciones:**

- La lista de proyectos está cargada y visible.

### **Caso de Uso 3: Seleccionar un Miembro**

**Título:** Seleccionar un Miembro

**Actores:** Usuario

**Descripción:** El usuario selecciona un miembro específico para visualizar su información relacionada con el proyecto seleccionado.

**Flujo Principal:**

- El usuario selecciona un miembro de la lista horizontal de miembros.
- La pantalla específica del miembro se actualiza para mostrar la información del miembro seleccionado.

**Precondiciones:**

- Un proyecto ya ha sido seleccionado.
- La lista de miembros relacionados con el proyecto está cargada y visible.

**Postcondiciones:**

- La página muestra la información del miembro seleccionado.

### **Caso de Uso 4: Deseleccionar un Proyecto**

**Título:** Deseleccionar un Proyecto

**Actores:** Usuario

**Descripción:** El usuario deselecta el proyecto actualmente seleccionado, volviendo la pantalla a su estado inicial.

**Flujo Principal:**

- El usuario deselecta el proyecto.
- El sistema elimina la información del proyecto de todas las pantallas.

- Las pantallas vuelven a su estado base, mostrando mensajes de espera para la selección de un nuevo proyecto.

**Precondiciones:**

- Un proyecto ya ha sido seleccionado.

**Postcondiciones:**

- La página regresa al estado inicial sin información del proyecto.

**Caso de Uso 5: Deseleccionar un Miembro**

**Título:** Deseleccionar un Miembro

**Actores:** Usuario

**Descripción:** El usuario deselectiona un miembro previamente seleccionado para volver al estado base de la pantalla de miembros.

**Flujo Principal:**

- El usuario deselectiona el miembro.
- El sistema elimina la información del miembro de la pantalla específica.
- La pantalla del miembro vuelve a su estado inicial, mostrando un mensaje de espera para la selección de un nuevo miembro.

**Precondiciones:**

- Un miembro ya ha sido seleccionado.

**Postcondiciones:**

- La información del miembro deselectionado se elimina y la pantalla vuelve a su estado inicial.

**Caso de Uso 6: Editar Información del Cliente**

**Título:** Editar Información del Cliente

**Actores:** Usuario

**Descripción:** El usuario selecciona la opción de editar la información del cliente asociado al proyecto seleccionado.

**Flujo Principal:**

- El usuario selecciona la opción para editar la información del cliente.
- El sistema permite la edición de los datos del cliente.
- El usuario modifica los datos necesarios y los guarda.
- El sistema actualiza la información del cliente en la base de datos y la pantalla correspondiente.

**Precondiciones:**

- Un cliente ya ha sido seleccionado.
- El usuario tiene permisos para editar la información del cliente.

**Postcondiciones:**

- La información del cliente se actualiza correctamente en la base de datos y en la pantalla.

**Caso de Uso 7: Editar Información del Miembro**

**Título:** Editar Información del Miembro

**Actores:** Usuario

**Descripción:** El usuario selecciona la opción de editar la información de un miembro del equipo asociado al proyecto seleccionado.

**Flujo Principal:**

- El usuario selecciona la opción para editar la información de un miembro.
- El sistema permite la edición de los datos del miembro.
- El usuario modifica los datos necesarios y los guarda.
- El sistema actualiza la información del miembro en la base de datos y en la pantalla correspondiente.

**Precondiciones:**

- Un miembro ya ha sido seleccionado.
- El usuario tiene permisos para editar la información del miembro.

**Postcondiciones:**

- La información del miembro se actualiza correctamente en la base de datos y en la pantalla.

**Caso de Uso 8: Registrar Ingresos Generados por un Cliente**

**Título:** Registrar Ingresos Generados por un Cliente

**Actores:** Usuario

**Descripción:** El usuario selecciona la opción de registrar los ingresos generados por un cliente para un proyecto específico.

**Flujo Principal:**

- El usuario selecciona la opción para registrar los ingresos generados por un cliente.
- El sistema permite la entrada de los datos de ingresos.
- El usuario introduce los datos necesarios y los guarda.
- El sistema actualiza la información de ingresos en la base de datos y en la pantalla correspondiente.

**Precondiciones:**

- Un cliente ya ha sido seleccionado.
- El usuario tiene permisos para registrar ingresos.

**Postcondiciones:**

- La información de ingresos se actualiza correctamente en la base de datos y en la pantalla.

### **Caso de Uso 9: Navegar a Otra Página**

**Título:** Navegar a Otra Página

**Actores:** Usuario

**Descripción:** El usuario decide navegar a otra sección de la aplicación desde la pantalla de Back-Office

**Flujo Principal:**

- El usuario selecciona una opción de menú o botón para navegar a otra página de la aplicación.
- El sistema redirige al usuario a la nueva página seleccionada.

**Precondiciones:**


- El usuario tiene acceso a otras secciones de la aplicación.

**Postcondiciones:**


- El usuario es redirigido a la nueva página.

## **3.2 Requisitos No Funcionales**


### **3.2.1 Rendimiento**

**Tiempo de Respuesta:** La aplicación debe responder a las acciones del usuario en menos de 2 segundos en el 95% de las interacciones, medido bajo condiciones de carga normal. 


- Este tiempo de respuesta será monitoreado en futuras versiones usando herramientas como Google Lighthouse o New Relic.


**Optimización de Consultas:** Las consultas a la base de datos deben estar optimizadas para manejar grandes volúmenes de datos sin degradar el rendimiento, manteniendo un tiempo de respuesta de menos de 500 ms en el 95% de las consultas realizadas. 

### **3.2.2 Seguridad**


**Almacenamiento de Contraseñas:** Las contraseñas deben ser almacenadas utilizando algoritmos de hashing seguros como **bcrypt**, con un costo de hashing adecuado para balancear seguridad y rendimiento. 

### **3.2.3 Mantenibilidad**

**Documentación del Código:** El código de la aplicación debe estar bien documentado, siguiendo unas buenas prácticas de desarrollo, y debe incluir comentarios en áreas críticas para facilitar el mantenimiento. 


**Arquitectura Modular:** La arquitectura debe ser modular, permitiendo la adición o modificación de funcionalidades con un impacto mínimo en otras partes del sistema. 

### ***3.2.4 Compatibilidad***

**Compatibilidad entre Navegadores:** La aplicación debe ser compatible con los principales navegadores web (Chrome, Firefox, Safari, Edge), asegurando que todas las funcionalidades sean accesibles y que la experiencia de usuario sea consistente. 

### ***3.2.5 Integridad de Datos***

**Mantenimiento de la Integridad Referencial:** La base de datos debe mantener la integridad referencial mediante el uso de llaves foráneas y restricciones adecuadas, asegurando que las relaciones entre tablas sean correctas y que no existan inconsistencias.

**Uso de Transacciones:** Las transacciones deben ser utilizadas para asegurar que las operaciones críticas (como actualizaciones de datos) se completen correctamente y de manera consistente, garantizando que el sistema esté siempre en un estado válido. 

## 4 Diseño

### 4.1 Arquitectura de la Aplicación

#### 4.1.1 Descripción General de la Arquitectura

El objetivo principal de la arquitectura de la aplicación es proporcionar una estructura robusta, modular y escalable que permita una gestión eficiente y segura de proyectos, clientes y tareas.

Esta arquitectura busca garantizar una separación clara de responsabilidades entre las diferentes capas de la aplicación: Presentación, Lógica de Negocio y Acceso a Datos.

Cada capa interactúa de manera coherente para asegurar que la información fluya correctamente, manteniendo la integridad y la seguridad de los datos.

La **Capa de Presentación** ofrece una interfaz de usuario intuitiva y accesible que se comunica con la **Capa de Lógica de Negocio** para aplicar las reglas de negocio definidas. A su vez, la Capa de Lógica de Negocio se conecta con la **Capa de Acceso a Datos** para realizar operaciones de almacenamiento y recuperación de información, garantizando que los datos sean consistentes y estén disponibles en todo momento.

La arquitectura está diseñada para soportar la escalabilidad del sistema, permitiendo que nuevos módulos y funcionalidades se integren fácilmente sin comprometer el rendimiento o la seguridad.

En resumen, la arquitectura de la aplicación está orientada a facilitar el desarrollo, mantenimiento y evolución del sistema, asegurando al mismo tiempo una experiencia de usuario fluida y una gestión eficiente de los recursos y datos.

#### 4.1.2 Arquitectura de Tres Capas

La arquitectura de tres capas divide la aplicación en tres secciones principales:

**Capa de Presentación:** Gestiona la interfaz de usuario y la interacción con el cliente. Aquí, el HTML y JavaScript en el lado del cliente muestran y manipulan los datos, permitiendo la interacción del usuario con el sistema.

**Capa de Lógica de Negocio:** Contiene las reglas de negocio y la lógica de procesamiento. Esta capa se encarga de manejar las solicitudes recibidas desde la capa de presentación, aplicando las reglas de negocio necesarias.

**Capa de Acceso a Datos:** Se encarga de la interacción con la base de datos y otros servicios externos, como la API utilizada para generar gráficos. Esta capa proporciona los datos requeridos por la capa de lógica de negocio.

Las capas interactúan de manera fluida para garantizar el correcto funcionamiento del sistema:

La **Capa de Presentación** envía solicitudes HTTP a la **Capa de Lógica de Negocio** para obtener y enviar datos necesarios para las operaciones que realiza el usuario.

La **Capa de Lógica de Negocio** recibe estas solicitudes, procesa la información y, si es necesario, se comunica con la **Capa de Acceso a Datos** para recuperar o almacenar información.

La **Capa de Acceso a Datos** gestiona las consultas a la base de datos y proporciona la información solicitada por la **Capa de Lógica de Negocio**.

Esta arquitectura asegura una clara separación de responsabilidades y facilita la escalabilidad y el mantenimiento de la aplicación.

#### **4.1.3 Seguridad y Escalabilidad**

La aplicación implementa la autenticación mediante **JSON Web Tokens (JWT)**. Cuando un usuario se autentica exitosamente, se genera un token JWT que se utiliza para acceder a rutas protegidas. Este token se verifica en cada solicitud a través de un middleware (**authenticateJWT**) que garantiza que solo usuarios autenticados puedan acceder a recursos específicos.

Las contraseñas de los usuarios se almacenan en la base de datos utilizando **bcrypt**, que aplica un algoritmo de hashing seguro para proteger las credenciales. El número de rondas (**saltRounds**) está configurado en 10 para equilibrar seguridad y rendimiento.

Se configura el CORS para permitir que la aplicación web interactúe de manera segura con el servidor, restringiendo el acceso a solo los dominios autorizados (**origin: 'http://192.168.1.35:8081'**), lo que ayuda a prevenir ataques de CSRF (Cross-Site Request Forgery).

Las rutas críticas que manejan datos sensibles, como la actualización de usuarios, clientes, y miembros, están protegidas mediante autenticación JWT, asegurando que solo los usuarios autenticados puedan realizar estas operaciones.

Las consultas a la base de datos están optimizadas para manejar grandes volúmenes de datos. Se utilizan índices en columnas clave para mejorar el rendimiento de las consultas. Además, se aplican filtros por año (**year**) en varias consultas para reducir la carga de datos innecesarios.

El servidor utiliza un **Pool** de conexiones para interactuar con PostgreSQL, lo que permite manejar múltiples solicitudes simultáneas de manera eficiente sin sobrecargar la base de datos.

La aplicación utiliza **express.static** para servir archivos estáticos, optimizando la entrega de recursos al cliente y reduciendo la latencia en la carga de páginas.

#### **4.1.4 Conclusión**

La arquitectura de la aplicación está diseñada con un enfoque modular y escalable que asegura la eficiencia en la gestión de proyectos, clientes y tareas.

Al implementar una arquitectura de tres capas, se logra una clara separación de responsabilidades, permitiendo que la Capa de Presentación, la Capa de Lógica de Negocio y la Capa de Acceso a Datos operen de manera independiente pero coordinada.

Esta estructura no solo facilita el desarrollo y el mantenimiento, sino que también mejora la escalabilidad del sistema, permitiendo la integración de nuevos módulos y funcionalidades sin comprometer el rendimiento o la seguridad.

Es importante señalar que, aunque se ha implementado la protección de credenciales mediante técnicas como la encriptación con bcrypt, la encriptación de datos sensibles, como la información financiera de los clientes, no ha sido implementada en esta primera versión de la aplicación.

Esto representa una limitación importante en términos de seguridad que se abordará en futuras versiones.

Aún así, en conjunto, esta arquitectura proporciona una base sólida para el crecimiento y la evolución continua del sistema

## 4.2 Diagramas de Uso

### 4.2.1 Identificación de Roles de Usuario

Para esta aplicación, he identificado un rol clave:

**CEO de la Agencia:** Este usuario tiene acceso total a la aplicación, incluyendo la visualización y gestión de todos los datos de clientes, proyectos, servicios, y miembros del equipo

El CEO puede realizar análisis detallados de rentabilidad, gestionar información de usuarios y clientes, y visualizar cargas de trabajo y tiempos dedicados por los miembros del equipo

### 4.2.2 Identificación de Casos de Uso

Los casos de uso han sido establecidos en los requisitos funcionales y se agrupan según las diferentes pantallas de la aplicación.

Para obtener más detalles específicos sobre cada caso de uso, consulta el apartado de **Requisitos Funcionales**, donde se detallan los flujos de trabajo y las interacciones para cada pantalla

### 4.2.3 Diagramas

**División de Diagramas:** Para una mejor visualización y claridad, los diagramas de casos de uso se han dividido en función de las diferentes pantallas de la aplicación. Esto permite un enfoque más organizado y manejable para representar las interacciones y flujos de trabajo específicos de cada sección.

## Diagrama 1: Pantalla de Bienvenida y Pantalla de Inicio de Sesión

### Pantalla de Bienvenida:

#### Caso de Uso 1: Visualización de la Pantalla de Bienvenida.

Este es un caso de uso autónomo, por lo que no requiere flechas especiales de relación.

#### Caso de Uso 2: Navegación a la Pantalla de Inicio de Sesión.

Se utiliza <<include>> desde "Visualización de la Pantalla de Bienvenida". La navegación a la pantalla de inicio de sesión se incluye como una parte del flujo principal después de que el usuario visualiza la pantalla de bienvenida.

### Pantalla de Inicio de Sesión:

#### Caso de Uso 1: Cargar la Pantalla de Inicio de Sesión.

Este es un caso de uso autónomo y no requiere flechas especiales de relación.

**Caso de Uso 2: Registro de Usuario.**

Se utiliza <<include>> desde "Cargar la Pantalla de Inicio de Sesión". El registro de usuario es una opción dentro del flujo de carga de la pantalla de inicio de sesión.

**Caso de Uso 3: Iniciar Sesión.**

Se utiliza <<include>> desde "Cargar la Pantalla de Inicio de Sesión". Iniciar sesión es otra acción disponible dentro del flujo de carga de la pantalla de inicio de sesión.

**Caso de Uso 4: Gestión de la Información del Usuario.**

Se utiliza <<extend>> desde "Iniciar Sesión". Una vez que el usuario ha iniciado sesión, tiene la opción de gestionar su información personal.

**Caso de Uso 5: Manejo de Errores en Inicio de Sesión.**

Se utiliza <<extend>> desde "Iniciar Sesión". Este caso de uso se activa si ocurre un error al intentar iniciar sesión, como en el caso de ingresar credenciales incorrectas.

**Caso de Uso 6: Navegar a Otra Página**

Se utiliza <<extend>> desde "Iniciar Sesión". Una vez que el usuario ha iniciado sesión, tiene la opción de navegar en otras páginas de la aplicación.

**Justificación:** Agrupar estas dos pantallas en un solo diagrama es conveniente, ya que ambas se relacionan con el flujo de acceso y autenticación del usuario. Además, los casos de uso de estas pantallas son relativamente pocos y están interconectados.



Figura 2. Diagrama 1: Pantalla de Bienvenida y Pantalla de Inicio de Sesión

## Diagrama 2: Pantalla de Informe de Costos por Proyecto

### Pantalla de Informe de Costos por Proyecto:

#### Caso de Uso 1: Cargar la Pantalla de Informe de Costos por Proyecto

Este es un caso de uso autónomo y no requiere flechas especiales de relación

#### Caso de Uso 2: Seleccionar un Proyecto

Se utiliza <<extend>> desde "Seleccionar un Proyecto". Este caso de uso se activa si el usuario decide deseleccionar un proyecto previamente seleccionado.

#### Caso de Uso 4: Seleccionar Múltiples Proyectos

Se utiliza <<extend>> desde "Seleccionar un Proyecto". Este caso de uso permite al usuario seleccionar varios proyectos para compararlos. Se extiende desde la acción inicial de selección de un proyecto.

#### Caso de Uso 5: Deseleccionar un Proyecto al Comparar Múltiples Proyectos

Se utiliza <<extend>> desde "Seleccionar Múltiples Proyectos". Este caso de uso se activa si el usuario decide deseleccionar un proyecto mientras compara varios proyectos.

#### Caso de Uso 6: Navegar a Otra Página

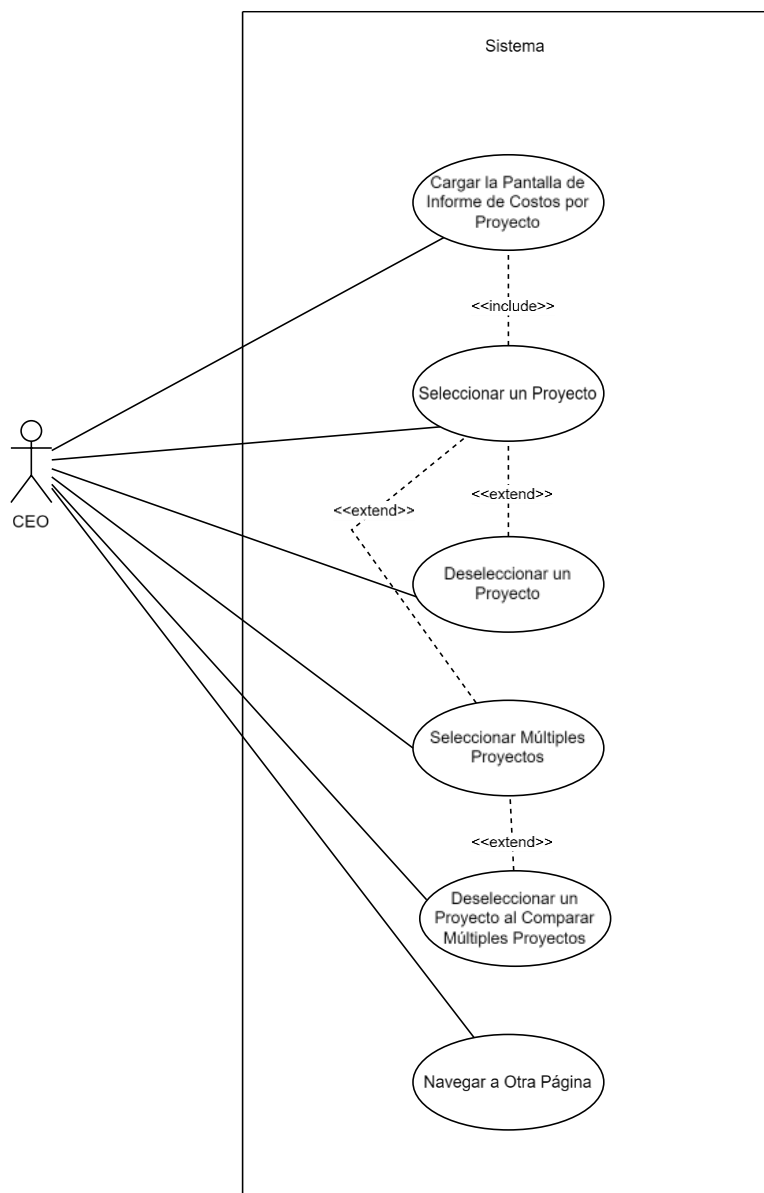
Este caso de uso es un comportamiento común que puede ser activado desde cualquier otro caso de uso principal dentro de esta pantalla.

Aunque en un diagrama convencional se utilizarían flechas con la etiqueta <<include>> para mostrar que este caso de uso está incluido en todos los demás, he decidido simplificar el diagrama al omitir estas flechas.

En su lugar, se indica explícitamente en la documentación que la acción de navegar a otra página está disponible desde cualquier flujo de trabajo dentro de la pantalla.

**Justificación:** Este diagrama está dedicado exclusivamente a la pantalla de Informe de Costos por Proyecto porque los casos de uso están altamente interrelacionados y enfocados en la selección y comparación de proyectos.

Esto permite mantener el enfoque claro y organizado dentro de un único diagrama.



**Figura 3.** Diagrama 2: Pantalla de Informe de Costos por Proyecto

### Diagrama 3: Pantalla de Análisis de Rentabilidad por Cliente

#### Pantalla de Análisis de Rentabilidad por Cliente:

##### Caso de Uso 1: Cargar la Pantalla de Análisis de Rentabilidad por Cliente

Este es un caso de uso autónomo y no requiere flechas especiales de relación.

##### Caso de Uso 2: Seleccionar un Cliente

Se utiliza <<include>> desde "Cargar la Pantalla de Análisis de Rentabilidad por Cliente". La acción de seleccionar un cliente está incluida dentro del flujo principal después de cargar la pantalla.

##### Caso de Uso 3: Deseleccionar un Cliente

Se utiliza <<extend>> desde "Seleccionar un Cliente". Este caso de uso se activa si el usuario decide deseleccionar un cliente previamente seleccionado.

##### Caso de Uso 4: Seleccionar Múltiples Clientes

Se utiliza <<extend>> desde "Seleccionar un Cliente". Este caso de uso permite al usuario seleccionar varios clientes para compararlos, extendiendo la funcionalidad de selección de un cliente.

##### Caso de Uso 5: Navegar a Otra Página

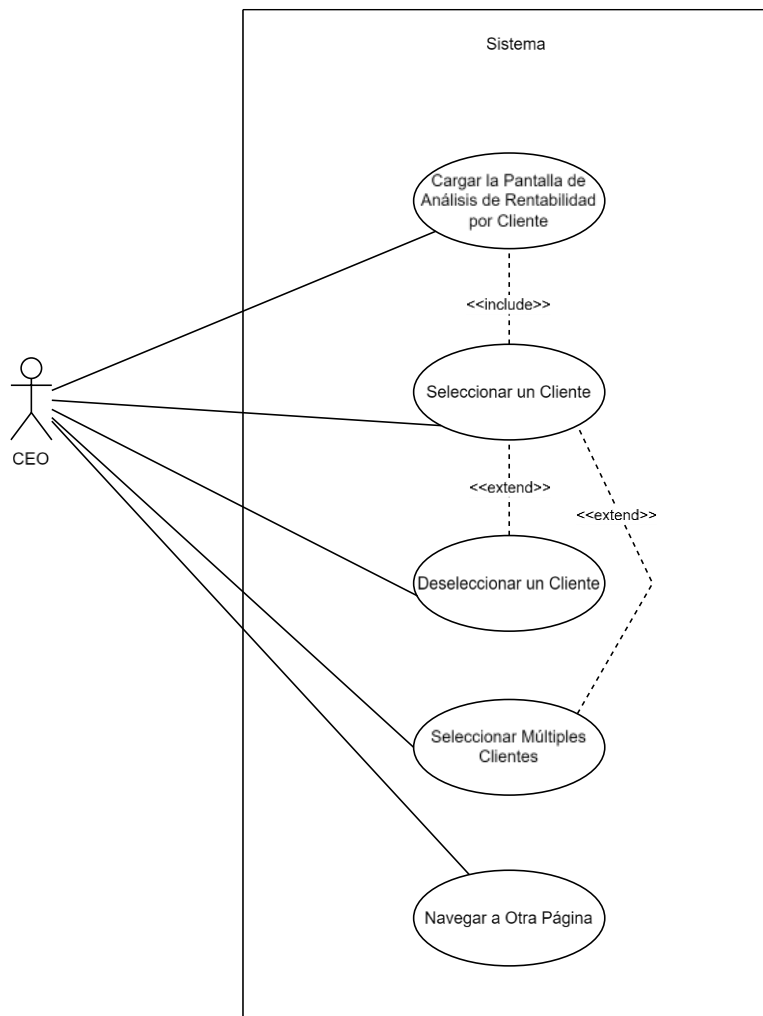
Este caso de uso es un comportamiento común que puede ser activado desde cualquier otro caso de uso principal dentro de esta pantalla.

Aunque en un diagrama convencional se utilizarían flechas con la etiqueta <<include>> para mostrar que este caso de uso está incluido en todos los demás, he decidido simplificar el diagrama al omitir estas flechas.

En su lugar, se indica explícitamente en la documentación que la acción de navegar a otra página está disponible desde cualquier flujo de trabajo dentro de la pantalla.

**Justificación:** Agrupar todos los casos de uso relacionados con la pantalla de Análisis de Rentabilidad por Cliente en un solo diagrama facilita la comprensión de las interacciones posibles dentro de esta pantalla.

Al mantener el enfoque en un único aspecto de la aplicación, se asegura una mayor claridad y organización en la presentación del diagrama.



**Figura 4.** Diagrama 3: Pantalla de Análisis de Rentabilidad por Cliente

## Diagrama 4: Pantalla de Visualización de Cargas de Trabajo

### Pantalla de Visualización de Cargas de Trabajo:

#### Caso de Uso 1: Cargar la Pantalla de Visualización de Cargas de Trabajo

Este caso de uso es autónomo y no requiere flechas especiales de relación.

#### Caso de Uso 2: Seleccionar un Miembro

Se utiliza <<include>> desde "Cargar la Pantalla de Visualización de Cargas de Trabajo". La selección de un miembro está incluida dentro del flujo principal después de cargar la pantalla.

#### Caso de Uso 3: Deseleccionar un Miembro

Se utiliza <<extend>> desde "Seleccionar un Miembro". Este caso de uso se activa si el usuario decide deseleccionar un miembro previamente seleccionado.

#### Caso de Uso 4: Seleccionar un Segundo Miembro

Se utiliza <<extend>> desde "Seleccionar un Miembro". Este caso de uso extiende la funcionalidad de selección permitiendo al usuario comparar la carga de trabajo de dos miembros simultáneamente.

#### Caso de Uso 5: Navegar a Otra Página

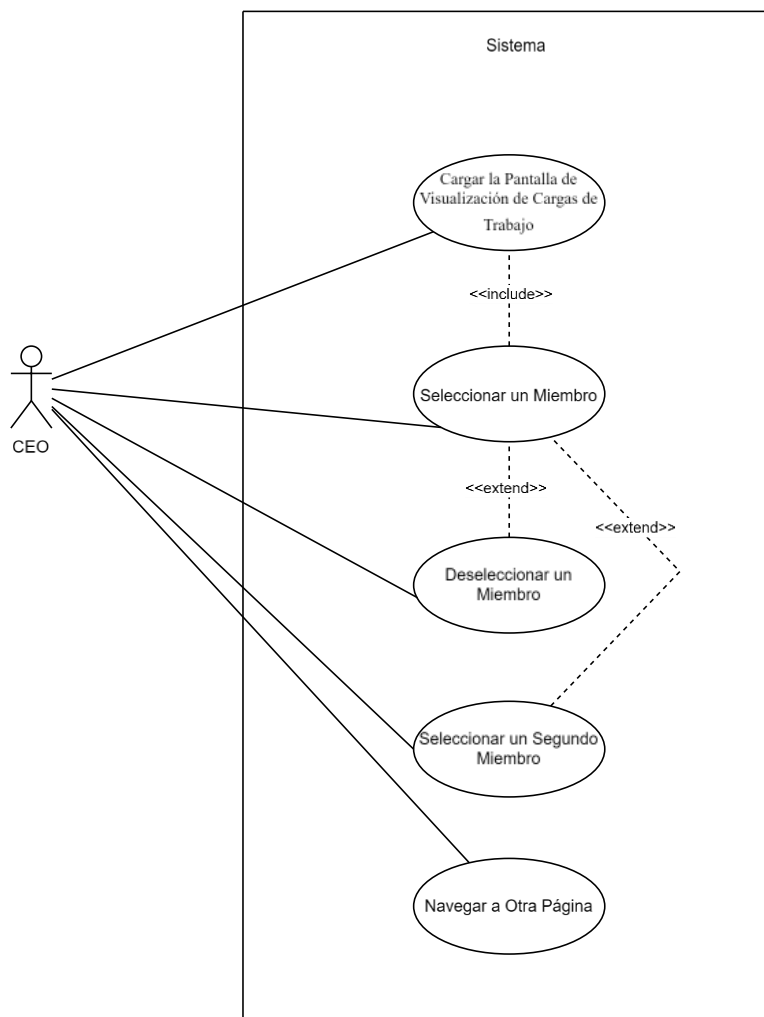
Este caso de uso es un comportamiento común que puede ser activado desde cualquier otro caso de uso principal dentro de esta pantalla.

Aunque en un diagrama convencional se utilizarían flechas con la etiqueta <<include>> para mostrar que este caso de uso está incluido en todos los demás, he decidido simplificar el diagrama al omitir estas flechas.

En su lugar, se indica explícitamente en la documentación que la acción de navegar a otra página está disponible desde cualquier flujo de trabajo dentro de la pantalla.

**Justificación:** Este diagrama se enfoca exclusivamente en la funcionalidad de visualización de cargas de trabajo, que es una parte esencial y específica de la aplicación.

Al dedicar un diagrama exclusivamente a este módulo, se asegura que todas las posibles interacciones relacionadas con la gestión y comparación de cargas de trabajo sean claramente representadas y comprendidas.



**Figura 5.** Diagrama 4: Pantalla de Visualización de Cargas de Trabajo

## Diagrama 5: Pantalla de Análisis de Servicio por Cliente

### Pantalla de Análisis de Servicio por Cliente:

#### Caso de Uso 1: Cargar la Pantalla de Análisis de Servicio por Cliente

Este caso de uso es autónomo y no requiere flechas especiales de relación.

#### Caso de Uso 2: Seleccionar un Servicio

Se utiliza <<include>> desde "Cargar la Página de Análisis de Servicio por Cliente". La selección de un servicio está incluida dentro del flujo principal después de cargar la página.

#### Caso de Uso 3: Seleccionar un Cliente

Se utiliza <<include>> desde "Seleccionar un Servicio". Este caso de uso se incluye después de seleccionar un servicio, permitiendo al usuario asociar un cliente al servicio.

#### Caso de Uso 4: Usuario Ha Seleccionado un Servicio, pero No un Cliente

Se utiliza <<extend>> desde "Seleccionar un Servicio". Este caso de uso se activa cuando un servicio ha sido seleccionado, pero no se ha asociado un cliente, manteniendo el estado de espera en la interfaz.

#### Caso de Uso 5: Deseleccionar el Cliente

Se utiliza <<extend>> desde "Seleccionar un Cliente". Este caso de uso se activa cuando el usuario decide deseleccionar un cliente previamente asociado a un servicio.

#### Caso de Uso 6: Usuario Ha Deseleccionado un Servicio o un Cliente, Estando Seleccionado el Otro

Se utiliza <<extend>> desde "Deseleccionar el Cliente" o "Deseleccionar el Servicio". Este caso de uso se activa si uno de los dos elementos (servicio o cliente) es deseleccionado mientras el otro sigue seleccionado, llevando la interfaz a un estado de espera.

#### Caso de Uso 7: Seleccionar un Segundo Cliente

Se utiliza <<extend>> desde "Seleccionar un Cliente". Este caso de uso permite al usuario comparar la rentabilidad de múltiples clientes dentro del mismo servicio.

#### Caso de Uso 8: Deseleccionar el Segundo Cliente

Se utiliza <<extend>> desde "Seleccionar un Segundo Cliente". Este caso de uso se activa cuando el usuario decide eliminar el segundo cliente de la comparación, regresando a la visualización de un solo cliente.

#### Caso de Uso 9: Deseleccionar el Servicio

Se utiliza <<extend>> desde "Seleccionar un Servicio". Este caso de uso se activa cuando el usuario decide eliminar el servicio seleccionado, regresando a un estado de espera para la selección de un nuevo servicio.

#### Caso de Uso 10: Navegar a Otra Página

Este caso de uso es un comportamiento común que puede ser activado desde cualquier otro caso de uso principal dentro de esta pantalla.

Aunque en un diagrama convencional se utilizarían flechas con la etiqueta <<include>> para mostrar que este caso de uso está incluido en todos los demás, he decidido simplificar el diagrama al omitir estas flechas.

En su lugar, se indica explícitamente en la documentación que la acción de navegar a otra página está disponible desde cualquier flujo de trabajo dentro de la pantalla.

**Justificación:** Debido a la cantidad considerable de casos de uso en esta pantalla, todos ellos enfocados en el análisis de servicio por cliente, es recomendable tener un diagrama separado para mantener la claridad y organizar adecuadamente las interacciones posibles dentro de este módulo.

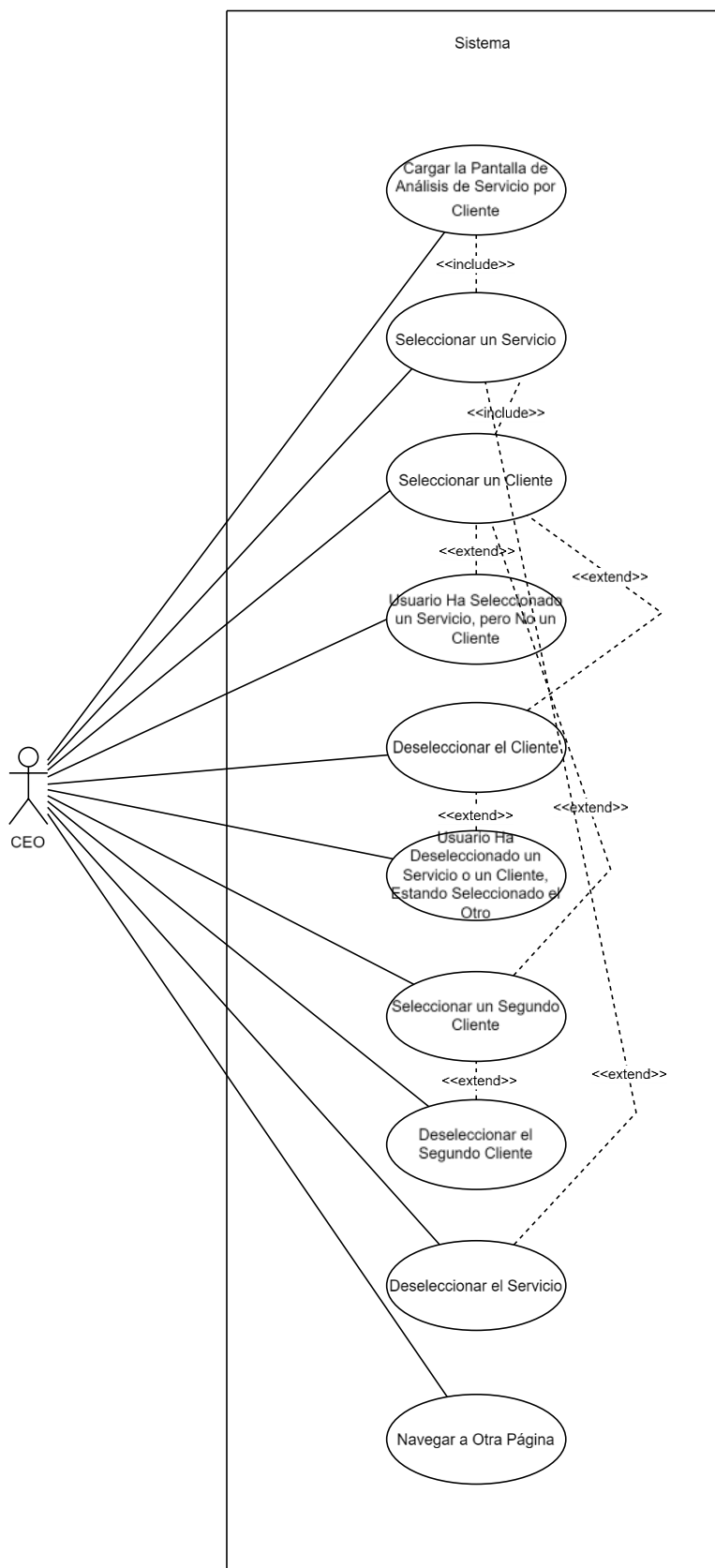


Figura 6. Diagrama 5: Pantalla de Análisis de Servicio por Cliente

## Diagrama 6: Pantalla de Estadísticas de Actividades por Servicio

### Pantalla de Estadísticas de Actividades por Servicio:

#### Caso de Uso 1: Cargar la Pantalla de Estadísticas de Actividades por Servicio

Este caso de uso es autónomo y no requiere flechas especiales de relación.

#### Caso de Uso 2: Seleccionar un Servicio

Se utiliza <<include>> desde "Cargar la Página de Estadísticas de Actividades por Servicio". La selección de un servicio está incluida como parte del flujo principal una vez que la página ha sido cargada.

#### Caso de Uso 3: Usuario Selecciona un Período de Tiempo

Se utiliza <<include>> desde "Seleccionar un Servicio". Este caso de uso permite al usuario filtrar las actividades por un período específico.

#### Caso de Uso 4: Usuario No Selecciona un Período de Tiempo

Se utiliza <<extend>> desde "Seleccionar un Período de Tiempo". Este caso de uso se activa si el usuario opta por no seleccionar un período de tiempo, mostrando todas las actividades sin filtrado.

#### Caso de Uso 5: Guardar la Selección del Período de Tiempo

Se utiliza <<include>> desde "Usuario Selecciona un Período de Tiempo". Este caso de uso guarda la selección del período de tiempo para futuras interacciones dentro de la misma pantalla.

#### Caso de Uso 6: Navegar a Otra Página

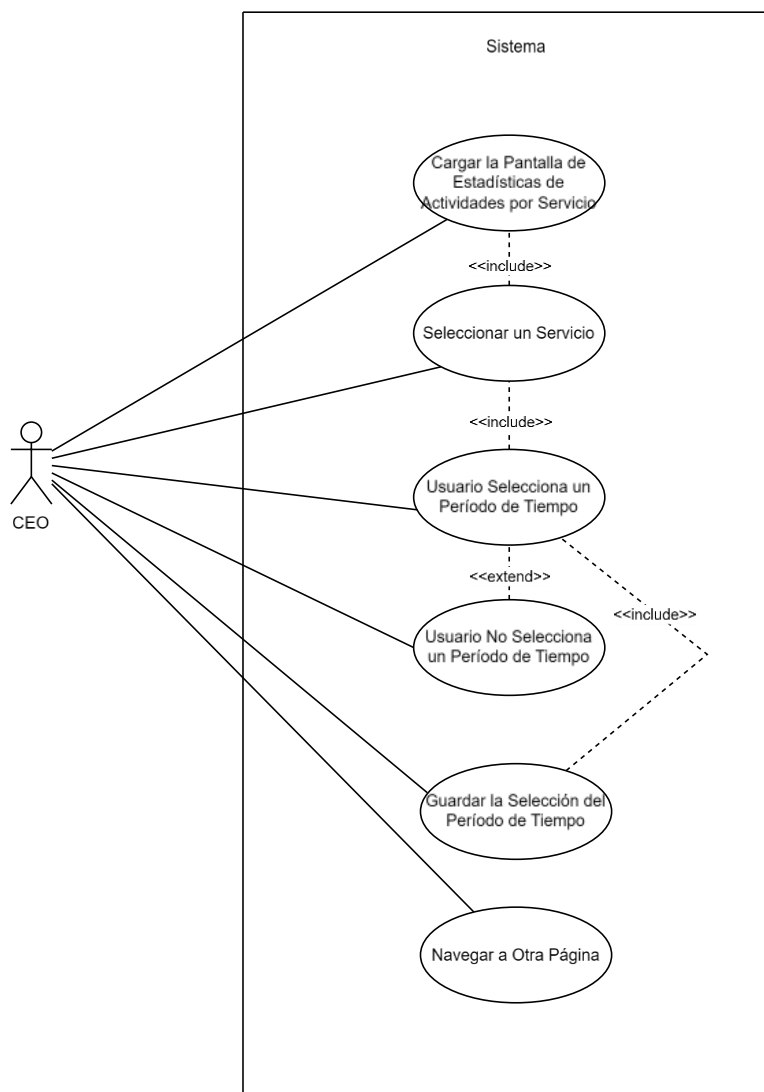
Este caso de uso es un comportamiento común que puede ser activado desde cualquier otro caso de uso principal dentro de esta pantalla.

Aunque en un diagrama convencional se utilizarían flechas con la etiqueta <<include>> para mostrar que este caso de uso está incluido en todos los demás, he decidido simplificar el diagrama al omitir estas flechas.

En su lugar, se indica explícitamente en la documentación que la acción de navegar a otra página está disponible desde cualquier flujo de trabajo dentro de la pantalla.

**Justificación:** Este diagrama se puede dedicar exclusivamente a la funcionalidad de estadísticas de actividades, que es lo suficientemente específica y cuenta con un número adecuado de casos de uso.

Esto permite una representación clara y organizada de las posibles interacciones dentro de este módulo.



**Figura 7.** Diagrama 6: Pantalla de Estadísticas de Actividades por Servicio

## Diagrama 7: Pantalla de Seguimiento del Tiempo

### Pantalla de Seguimiento del Tiempo:

#### Caso de Uso 1: Cargar la Pantalla de Seguimiento del Tiempo

Este caso de uso es autónomo y no requiere flechas especiales de relación.

#### Caso de Uso 2: Seleccionar un Proyecto

Se utiliza <<include>> desde "Cargar la Página de Seguimiento del Tiempo". La selección de un proyecto está incluida como parte del flujo principal después de cargar la página.

#### Caso de Uso 3: Deseleccionar el Proyecto

Se utiliza <<extend>> desde "Seleccionar un Proyecto". Este caso de uso se activa cuando el usuario decide deseleccionar un proyecto previamente seleccionado.

#### Caso de Uso 4: Seleccionar un Miembro

Se utiliza <<include>> desde "Seleccionar un Proyecto". Este caso de uso permite al usuario visualizar el tiempo dedicado por un miembro específico al proyecto seleccionado.

#### Caso de Uso 5: Seleccionar un Segundo Miembro

Se utiliza <<extend>> desde "Seleccionar un Miembro". Este caso de uso se activa cuando el usuario selecciona un segundo miembro para comparar su tiempo dedicado al proyecto.

#### Caso de Uso 6: Seleccionar un Segundo Proyecto

Se utiliza <<extend>> desde "Seleccionar un Proyecto". Este caso de uso permite al usuario comparar el tiempo dedicado a dos proyectos diferentes.

#### Caso de Uso 7: Deseleccionar un Miembro

Se utiliza <<extend>> desde "Seleccionar un Miembro". Este caso de uso se activa cuando el usuario deselecciona un miembro previamente seleccionado.

#### Caso de Uso 8: Navegar a Otra Página

Este caso de uso es un comportamiento común que puede ser activado desde cualquier otro caso de uso principal dentro de esta pantalla.

Aunque en un diagrama convencional se utilizarían flechas con la etiqueta <<include>> para mostrar que este caso de uso está incluido en todos los demás, he decidido simplificar el diagrama al omitir estas flechas.

En su lugar, se indica explícitamente en la documentación que la acción de navegar a otra página está disponible desde cualquier flujo de trabajo dentro de la pantalla.

**Justificación:** Dada la complejidad y cantidad de casos de uso relacionados con la gestión del tiempo, un diagrama separado es ideal para mantener la organización y legibilidad, permitiendo una representación clara de las interacciones posibles en este módulo.

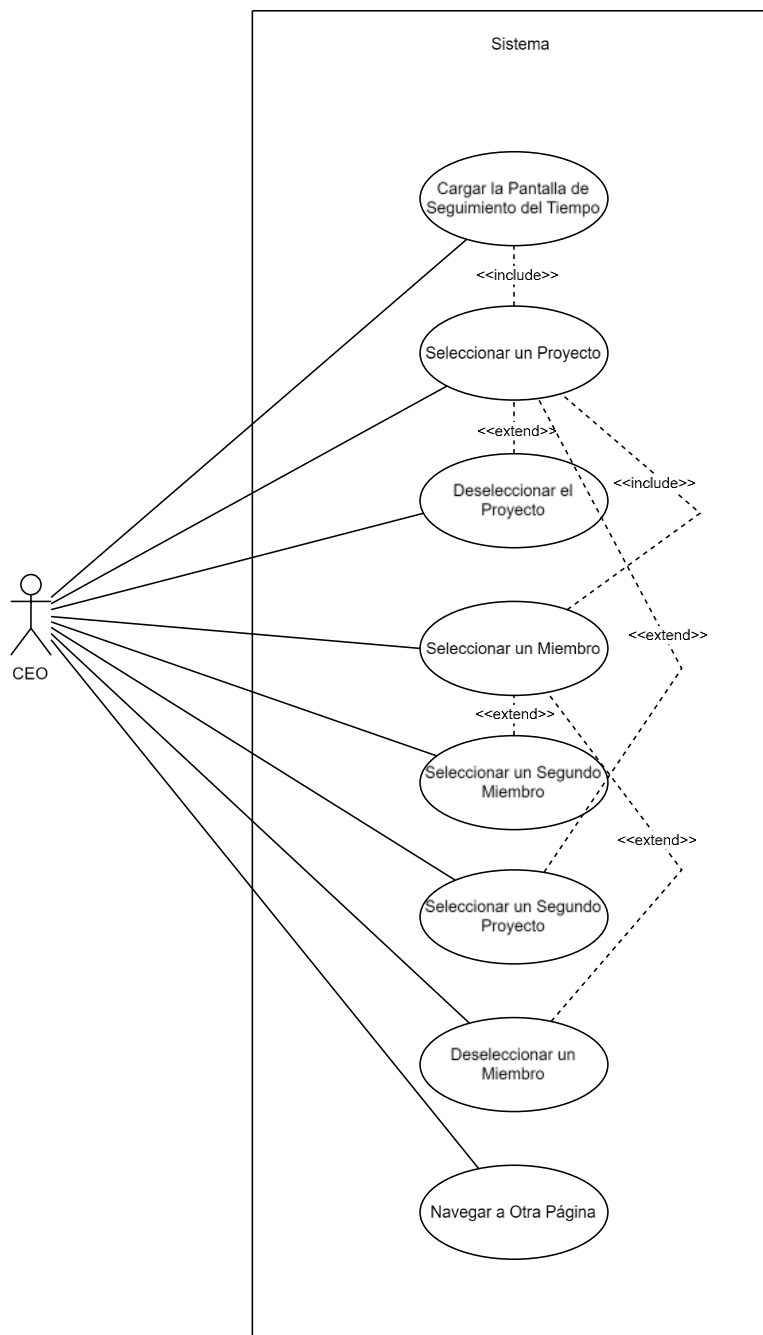


Figura 8. Diagrama 7: Pantalla de Seguimiento del Tiempo

## Diagrama 8: Pantalla de Back-Office

### Pantalla de Back-Office:

#### Caso de Uso 1: Cargar la Pantalla de Back-Office

Este caso de uso es autónomo y no requiere flechas especiales de relación.

#### Caso de Uso 2: Seleccionar un Proyecto

Se utiliza <<include>> desde "Cargar la Página de Back-Office". La selección de un proyecto está incluida dentro del flujo principal después de cargar la página.

#### Caso de Uso 3: Seleccionar un Miembro

Se utiliza <<include>> desde "Seleccionar un Proyecto". Este caso de uso permite al usuario visualizar y gestionar la información de un miembro específico asociado al proyecto seleccionado.

#### Caso de Uso 4: Deseleccionar un Proyecto

Se utiliza <<extend>> desde "Seleccionar un Proyecto". Este caso de uso se activa cuando el usuario decide deseleccionar un proyecto previamente seleccionado.

#### Caso de Uso 5: Deseleccionar un Miembro

Se utiliza <<extend>> desde "Seleccionar un Miembro". Este caso de uso se activa cuando el usuario decide deseleccionar un miembro previamente seleccionado.

#### Caso de Uso 6: Editar Información del Cliente

Se utiliza <<include>> desde "Seleccionar un Proyecto". Este caso de uso permite al usuario editar la información del cliente asociado al proyecto seleccionado.

#### Caso de Uso 7: Editar Información del Miembro

Se utiliza <<include>> desde "Seleccionar un Miembro". Este caso de uso permite al usuario editar la información de un miembro específico.

#### Caso de Uso 8: Registrar Ingresos Generados por un Cliente

Se utiliza <<include>> desde "Seleccionar un Proyecto". Este caso de uso permite al usuario registrar los ingresos generados por un cliente para un proyecto específico.

#### Caso de Uso 9: Navegar a Otra Página

Este caso de uso es un comportamiento común que puede ser activado desde cualquier otro caso de uso principal dentro de esta pantalla.

Aunque en un diagrama convencional se utilizarían flechas con la etiqueta <<include>> para mostrar que este caso de uso está incluido en todos los demás, he decidido simplificar el diagrama al omitir estas flechas.

En su lugar, se indica explícitamente en la documentación que la acción de navegar a otra página está disponible desde cualquier flujo de trabajo dentro de la pantalla.

**Justificación:** Este módulo de administración tiene varios casos de uso relacionados con la gestión de proyectos, clientes y miembros.

Un diagrama exclusivo es necesario para cubrir adecuadamente esta funcionalidad, permitiendo una representación organizada y clara de las interacciones posibles dentro de esta pantalla.

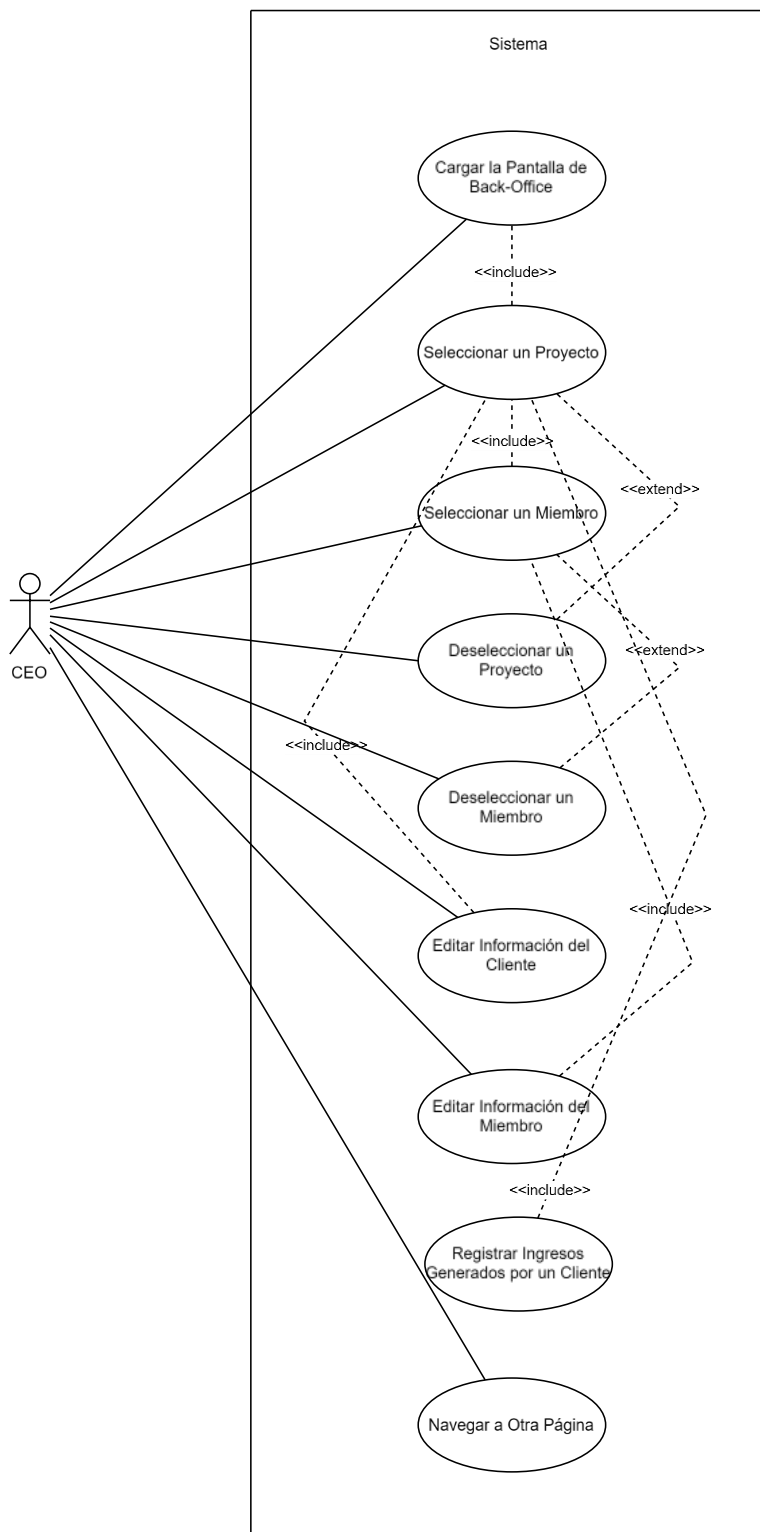


Figura 9. Diagrama 8: Pantalla de Back-Office

## 4.3 Diseño de Clases

### 4.3.1 Diagrama de Clases

#### Clases Principales:

##### Usuario:

**Atributos:** username, password, profileImage, trelloApiKey, trelloApiToken

**Métodos:** iniciarSesion(), registrar(), guardarCredencialesTrello(), cambiarContraseña(), cambiarNombreUsuario(), subirImagenPerfil()

##### Relaciones:

Relación 1 a muchos con Proyecto (un usuario puede tener varios proyectos).

##### Proyecto:

**Atributos:** proyectoId, nombre, costInicial, costRecursos, costFinal, clientId, dataUltimaActividad, dataUltimaVista, costRd, costEquipamientoInicial, costConfiguracion, costesAdicionales, username

**Métodos:** crearProyecto(), actualizarProyecto(), eliminarProyecto(), obtenerProyectosPorUsuario()

##### Relaciones:

Relación 1 a 1 con Cliente (un proyecto está asociado a un cliente).

Relación 1 a muchos con Tarea (un proyecto tiene varias tareas).

Relación 1 a muchos con Material (un proyecto puede utilizar varios materiales).

##### Cliente:

**Atributos:** clientId, nombre, informacionContacto, ingresosGenerados, username

**Métodos:** crearCliente(), actualizarCliente(), eliminarCliente(), obtenerClientesPorUsuario()

##### Relaciones:

Relación 1 a muchos con Servicio (un cliente puede tener varios servicios).

Relación 1 a muchos con IngresosServicioCliente (un cliente puede generar varios ingresos a través de diferentes servicios).

##### Servicio:

**Atributos:** serveiId, nombre, categoria, username

**Métodos:** crearServicio(), actualizarServicio(), eliminarServicio(), obtenerServiciosPorUsuario()

**Relaciones:**

Relación 1 a muchos con Tarea (a través de Clasificación, un servicio puede estar asociado con varias tareas).

Relación 1 a muchos con IngresosServicioCliente (un servicio puede generar varios ingresos para un cliente).

**Tarea:**

**Atributos:** tascaId, nombre, descripcion, estado, dataCreacio, dataLimit, dataFinalitzacio, tempsDestinat, tempsEstimat, projecteId, membres, username

**Métodos:** crearTarea(), actualizarTarea(), eliminarTarea(), asignarMiembro(), obtenerTareasPorProyecto()

**Relaciones:**

Relación de muchos a muchos con Miembro a través de Asignación (una tarea puede ser asignada a varios miembros).

Relación 1 a 1 con Proyecto (una tarea pertenece a un proyecto).

**Miembro:**

**Atributos:** membreId, nombre, actividad, rol, salario, username

**Métodos:** crearMiembro(), actualizarMiembro(), eliminarMiembro(), obtenerMiembrosPorProyecto()

**Relaciones:**

Relación de muchos a muchos con Tarea a través de Asignación (un miembro puede estar asignado a varias tareas).

**Clases de Relación:**

**Asignación:**

**Atributos:** assignacioId, tascaId, membreId, tempsDedicat

**Métodos:** asignarTareaAMiembro(), actualizarAsignacion()

**Relaciones:**

Relación de muchos a muchos entre Tarea y Miembro.

**IngresosServicioCliente:**

**Atributos:** ingresoId, clientId, serveiId, fecha, ingresos, username

**Métodos:** registrarIngreso(), obtenerIngresosPorCliente(), obtenerIngresosPorServicio()

**Relaciones:**

Relación de muchos a muchos entre Cliente y Servicio.

**Material:**

**Atributos:** materialId, projecteId, nomMaterial, cost, fecha, username

**Métodos:** registrarMaterial(), obtenerMaterialesPorProyecto()

**Relaciones:**

Relación 1 a muchos con Proyecto.

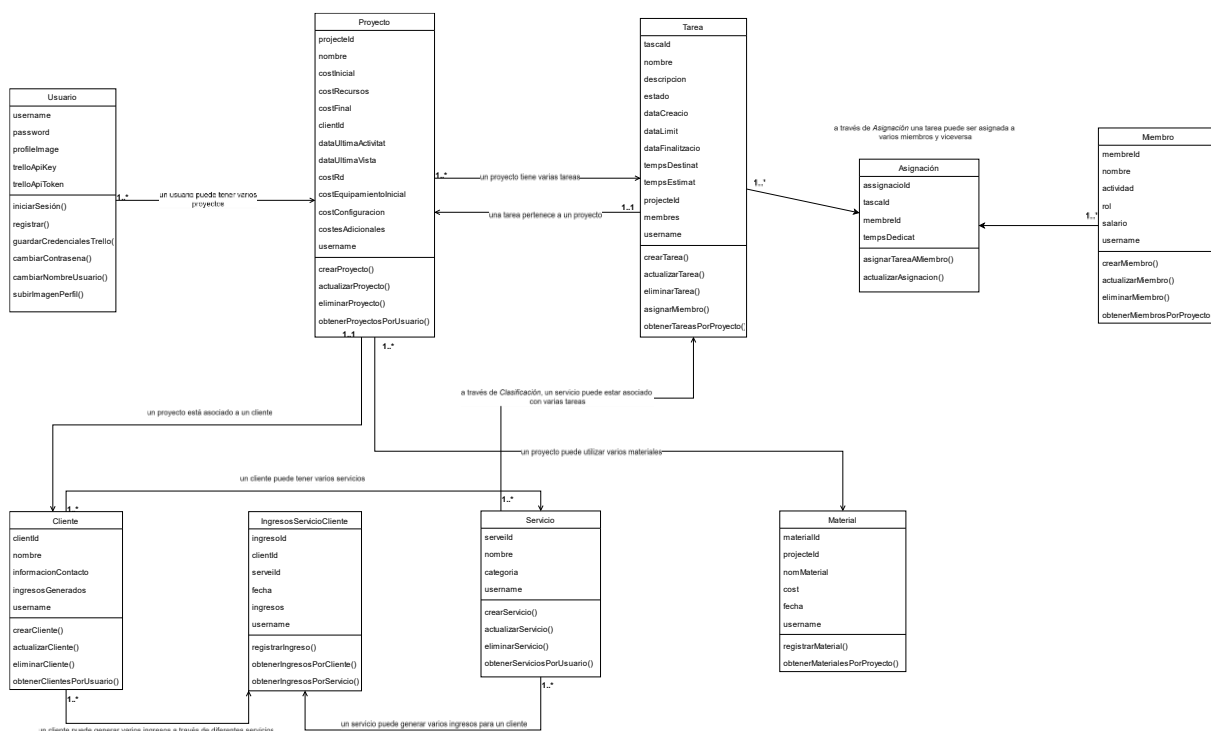


Figura 10. Diagrama de Clases

### 4.3.2 Lógica de la Organización

La organización de las clases se basa en las entidades y relaciones clave identificadas en la base de datos y el flujo de trabajo de la aplicación.

El objetivo es estructurar el sistema de manera que cada clase represente una entidad fundamental del dominio de la aplicación, y que las relaciones entre estas clases reflejen las interacciones y dependencias reales.

#### Clases Principales:

**Usuario, Proyecto, Cliente, Servicio, Tarea, Miembro:** Estas clases representan las entidades principales que son directamente gestionadas por el sistema. Cada una tiene atributos que almacenan información específica y métodos que encapsulan las operaciones que pueden realizarse sobre esa entidad.

#### Relaciones y Clases de Enlace:

**Asignación, Clasificación, IngresosServicioCliente, Material:** Estas clases sirven para modelar las relaciones entre las entidades principales. Por ejemplo, la clase Asignación se utiliza para gestionar la relación de muchos a muchos entre Tarea y Miembro, permitiendo registrar cuánto tiempo dedica cada miembro a cada tarea.

#### Encapsulamiento y Abstracción:

Cada clase está diseñada para encapsular la lógica y los datos relacionados con su entidad correspondiente. Por ejemplo, la clase Proyecto no solo almacena datos sobre un proyecto, sino que también contiene métodos para manipular esos datos, como `crearProyecto()` y `actualizarProyecto()`. Esto sigue el principio de encapsulamiento, donde los datos y las funciones que los manipulan están agrupados.

El diseño de clases propuesto está orientado a maximizar la reutilización del código y mantener la cohesión del sistema:

#### Reutilización del Código:

**Métodos Reutilizables:** Métodos como `crearProyecto()`, `asignarMiembro()` y `registrarIngreso()` están diseñados para ser reutilizados en múltiples contextos, evitando la duplicación de código. Por ejemplo, el método `asignarMiembro()` puede ser reutilizado cada vez que se asigna un miembro a una nueva tarea, sin necesidad de reescribir la lógica.

**Clases de Enlace:** Clases como `Asignación` y `IngresosServicioCliente` actúan como puentes entre otras clases, lo que permite que las relaciones complejas se manejen de manera uniforme y se puedan reutilizar en diferentes partes de la aplicación.

#### Cohesión del Sistema:

**Diseño Modular:** Al dividir el sistema en clases pequeñas y específicas, cada una responsable de una parte de su comportamiento, se asegura que el código sea más fácil de entender, mantener y expandir.

**Facilidad de Mantenimiento:** La alta cohesión de las clases significa que los cambios en un aspecto del sistema tienen un impacto limitado. Por ejemplo, si se

necesita cambiar la forma en que se calculan los ingresos de un cliente, esto se puede hacer en la clase IngresosServicioCliente sin afectar el resto del sistema.

#### **4.4 Diseño de la Interfaz de Usuario**

El **Diseño de Interfaz de Usuario (UI)** de la aplicación se ha desarrollado con un enfoque en la usabilidad, accesibilidad y diseño centrado en el usuario.

A continuación, se presenta una descripción detallada de la navegación, disposición de los elementos, y la alineación con los principios de diseño centrado en el usuario

##### ***4.4.1 Navegación y Disposición de los Elementos en Pantalla***

La aplicación cuenta con una estructura de navegación intuitiva y coherente, diseñada para facilitar la interacción del usuario con las diferentes funcionalidades disponibles.

La navegación principal se organiza en un menú lateral que incluye enlaces directos a las secciones clave de la aplicación, como el Informe de Costos por Proyecto, Análisis de la Rentabilidad, Visualización de Cargas de Trabajo, Back-Office, y otras secciones importantes.

**Menú de Navegación:** El menú se destaca por su simplicidad, permitiendo un acceso rápido a las secciones más frecuentadas de la aplicación. Cada opción del menú está claramente etiquetada y utiliza iconografía para mejorar la comprensión.

**Pantallas Principales:** Las pantallas están organizadas en secciones bien definidas, con encabezados claros y jerarquía visual adecuada para guiar al usuario en su flujo de trabajo. Los botones y elementos interactivos están dispuestos de manera lógica y accesible, con un tamaño adecuado para ser fácilmente seleccionados.

**Disposición de Contenidos:** Se ha optado por un diseño de cuadrícula (grid layout) que organiza el contenido de manera que sea fácil de interpretar. Los textos, gráficos y formularios están diseñados para ofrecer una experiencia visual limpia y sin desorden.

##### ***4.4.2 Usabilidad y Accesibilidad***

El diseño de la interfaz ha sido concebido teniendo en cuenta al usuario específico que interactuará con la aplicación (propietario de la agencia), asegurando que sea accesible y fácil de usar para él.

**Elementos de Interacción:** Los botones, menús y campos de entrada están diseñados con un tamaño y espacio suficientes para evitar errores en la interacción.

**Feedback del Usuario:** La interfaz proporciona un feedback inmediato al usuario a través de mensajes emergentes, indicadores visuales y cambios en la interfaz, como el estado de los botones y la actualización de las vistas, asegurando que el usuario siempre esté informado sobre el resultado de sus acciones.

##### ***4.4.3 Principios de Diseño Centrado en el Usuario***

El diseño de la interfaz de usuario se ha desarrollado siguiendo los principios fundamentales del diseño centrado en el usuario (UCD), lo que garantiza que la aplicación no solo sea funcional, sino también agradable y eficiente de usar.

**Empatía con el Usuario:** Se realizaron encuestas y entrevistas preliminares con usuarios potenciales para comprender sus necesidades, desafíos y expectativas. Esta

información se utilizó para guiar las decisiones de diseño, asegurando que la interfaz resuelva los problemas reales de los usuarios y mejore su productividad.

**Simplicidad y Minimalismo:** La interfaz evita el uso excesivo de elementos decorativos y se enfoca en lo esencial, lo que reduce la carga cognitiva del usuario y facilita la realización de tareas sin distracciones.

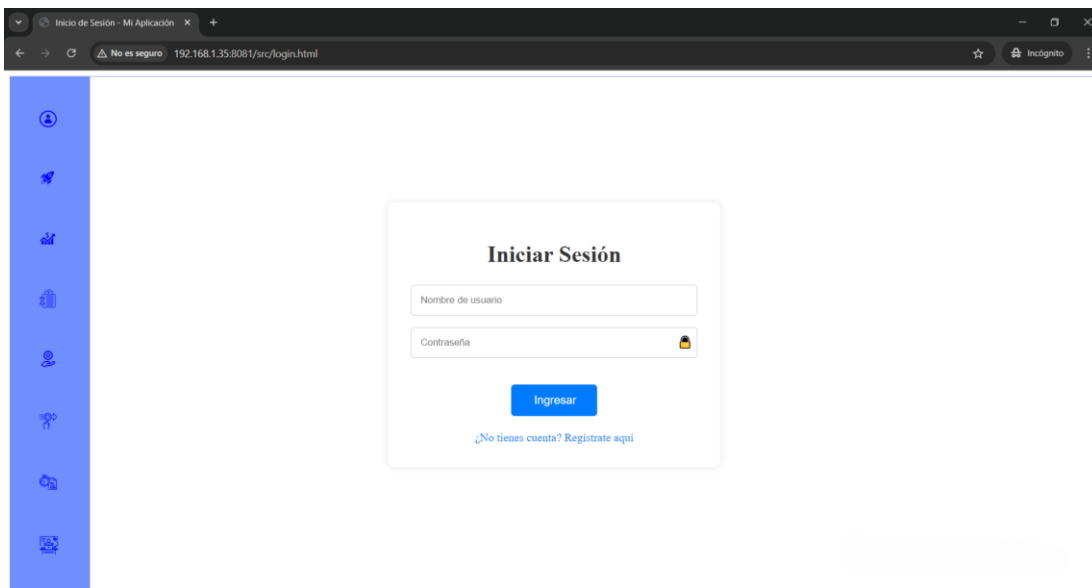
**Consistencia Visual y Funcional:** Se ha mantenido una consistencia estricta en el uso de colores, tipografías, iconos y patrones de interacción en toda la aplicación. Esto ayuda a los usuarios a aprender rápidamente cómo navegar y utilizar la aplicación, sin necesidad de una curva de aprendizaje empinada.

En resumen, el diseño de la interfaz de usuario de la aplicación ha sido cuidadosamente elaborado para proporcionar una experiencia de usuario eficiente, accesible y agradable, siguiendo los principios de usabilidad y diseño centrado en el usuario.

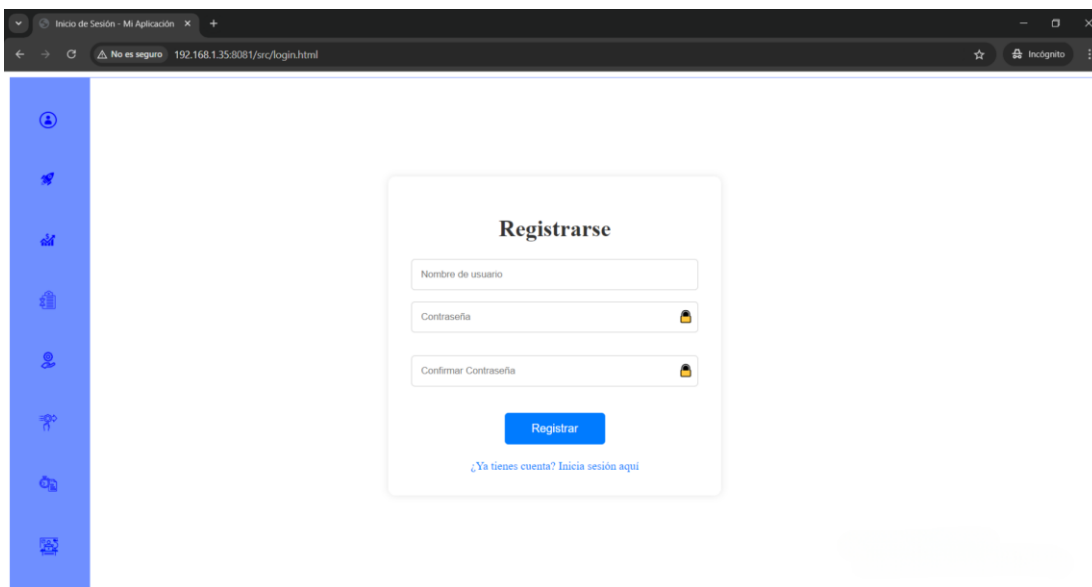
Las capturas de pantalla y maquetas que se adjuntan a continuación ilustran cómo estos principios se materializan en la práctica.



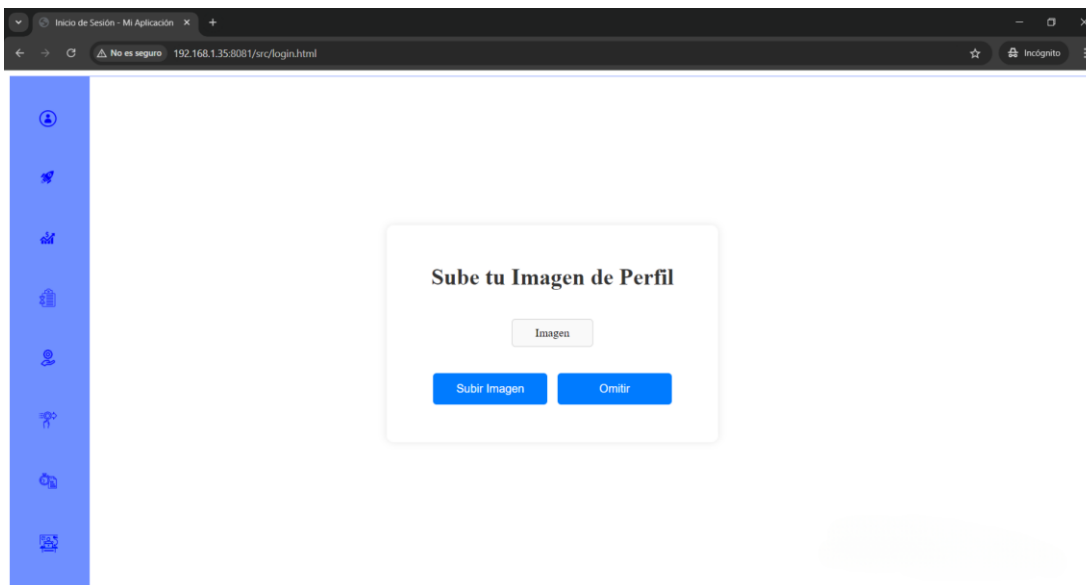
**Figura 11.** Pantalla de Bienvenida



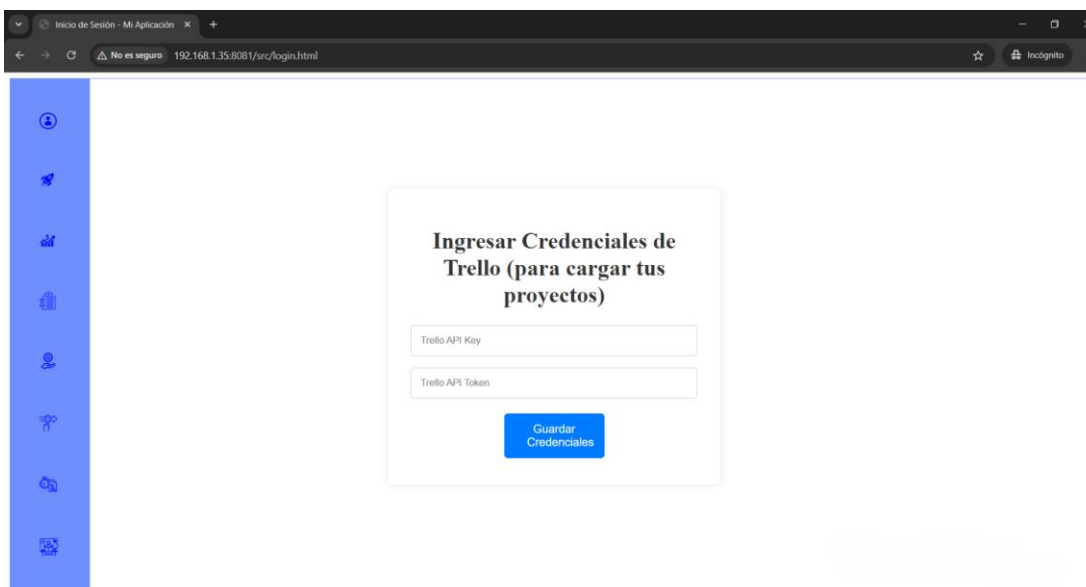
**Figura 12.** Pantalla de Inicio de Sesión



**Figura 13.** Pantalla de Registro de Sesión



**Figura 14.** Pantalla de Subir Imagen de Perfil



**Figura 15.** Pantalla de Ingresar Credenciales del Trello



Figura 16. Pantalla de Credenciales Introducidas

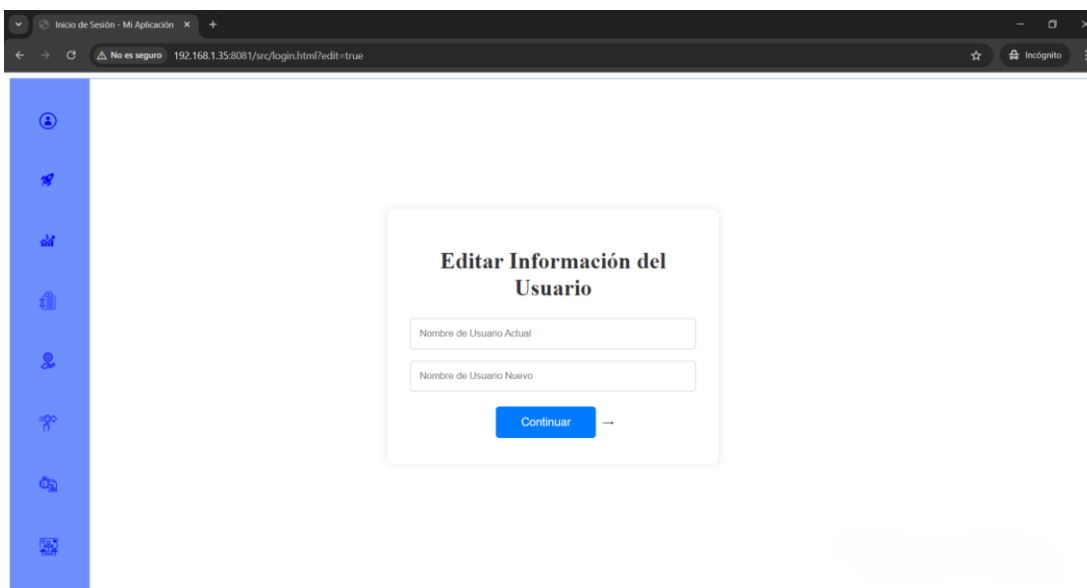


Figura 17. Pantalla de Edición de Información del Usuario

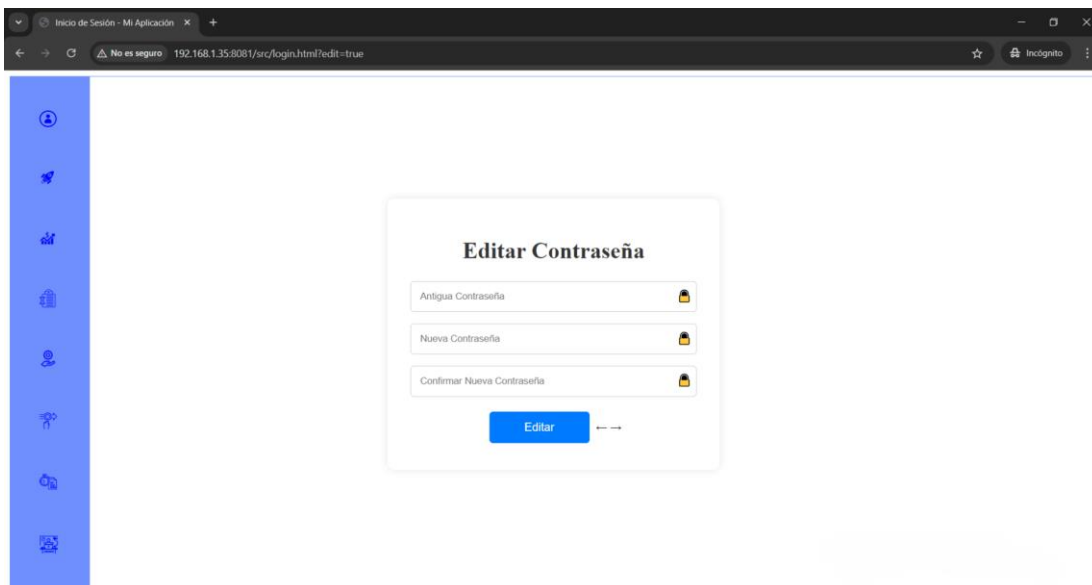


Figura 18. Pantalla de Edición de la Contraseña del Usuario

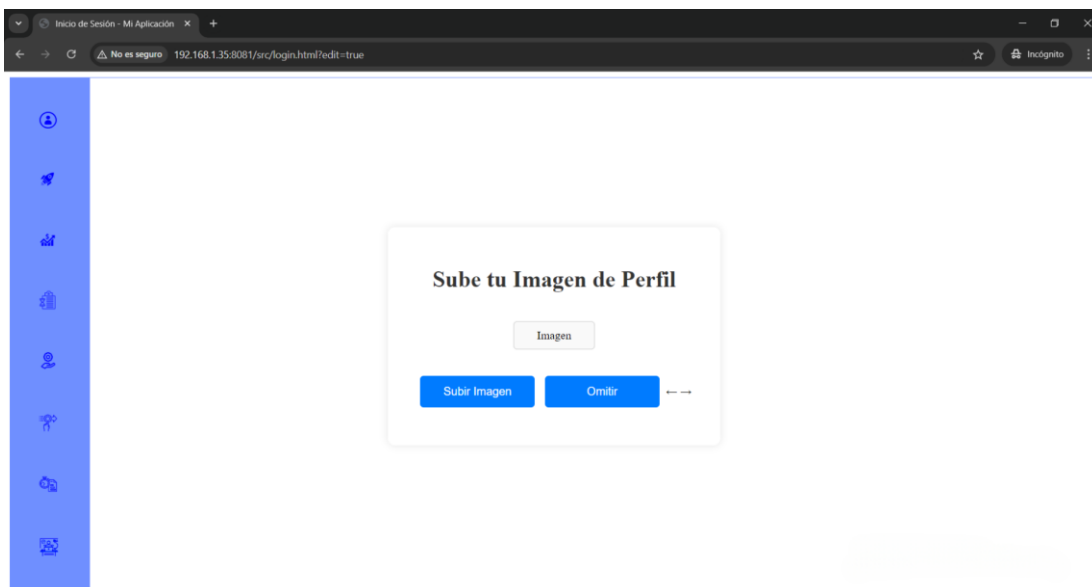


Figura 19. Pantalla de Subir Imagen de Perfil (Modo Edición)



Figura 20. Pantalla de Ingresar Credenciales del Trello (Modo Edición)

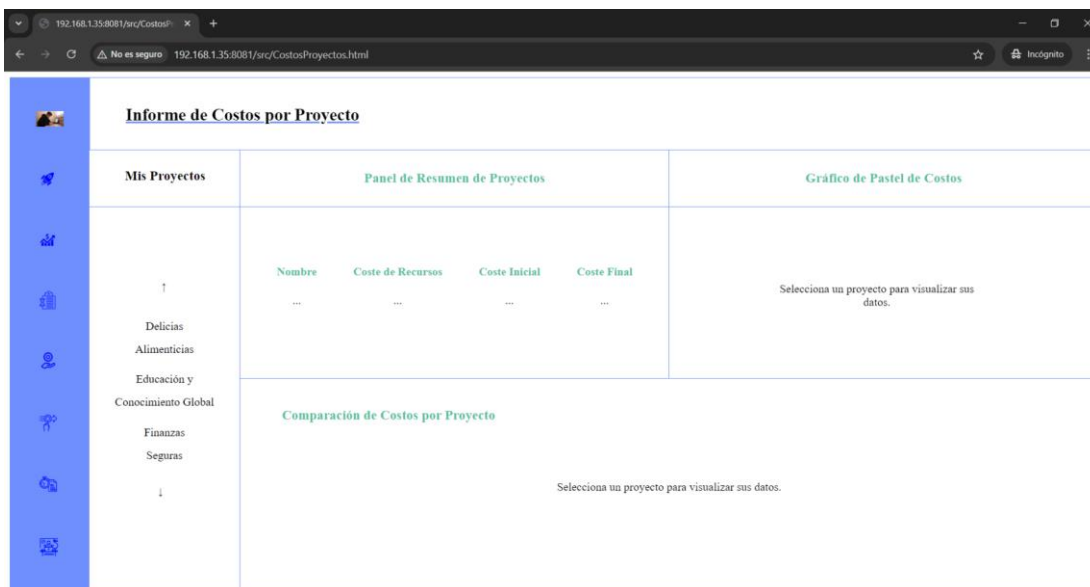


Figura 21. Pantalla de Informe de Costos por Proyecto (Carga Inicial)

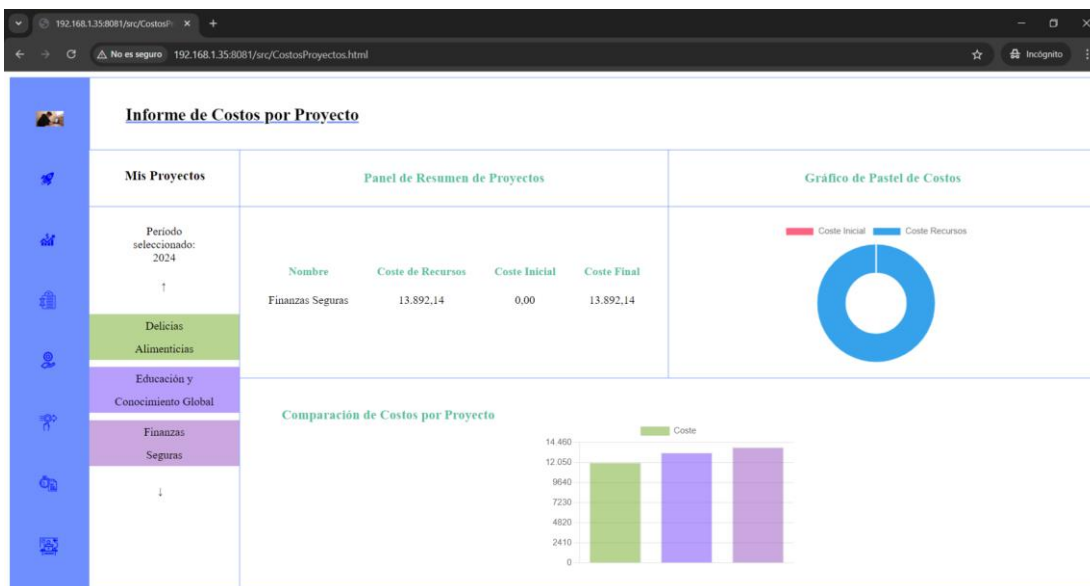


Figura 22. Pantalla de Informe de Costos por Proyecto (Información Cargada)

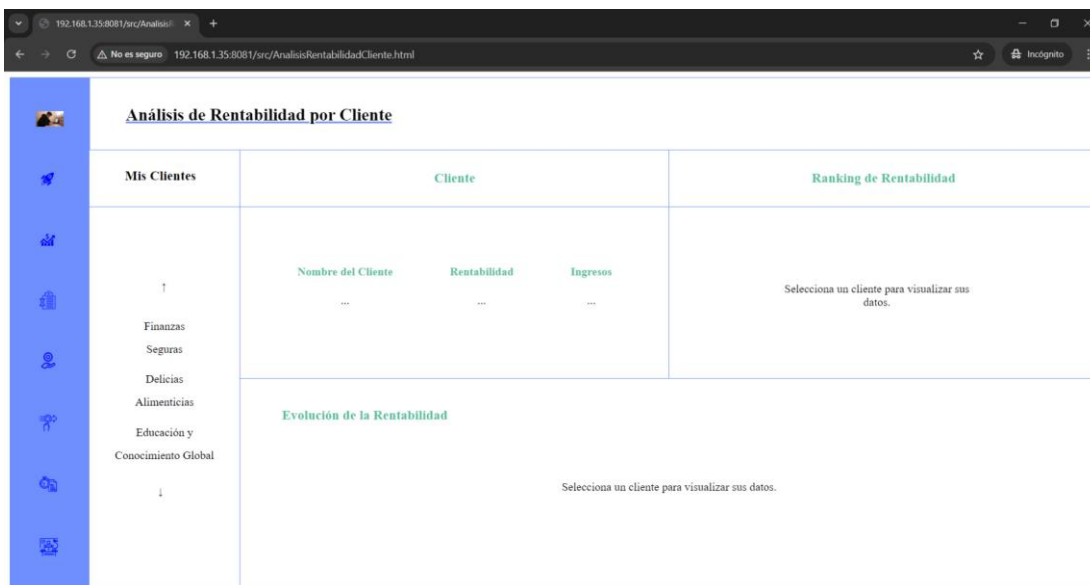


Figura 23. Pantalla de Análisis de Rentabilidad por Cliente (Carga Inicial)

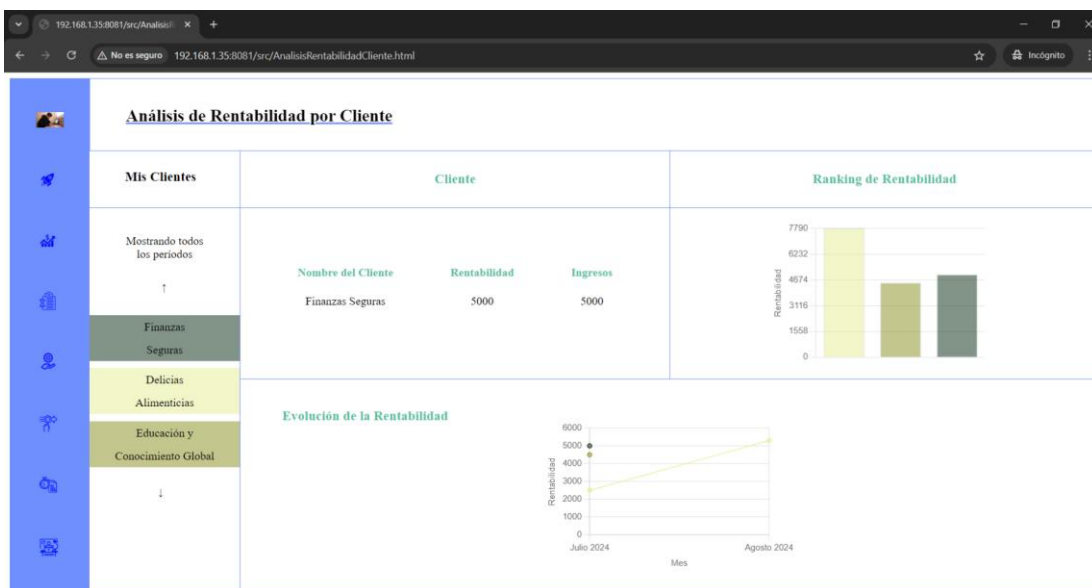


Figura 24. Pantalla de Análisis de Rentabilidad por Cliente (Información Cargada)

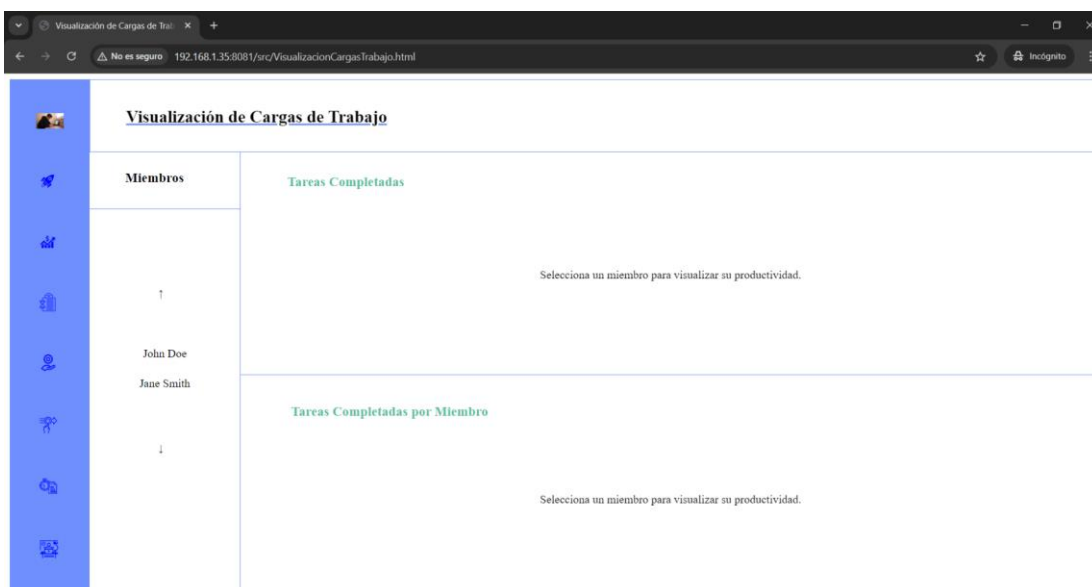


Figura 25. Pantalla de Visualización de Cargas de Trabajo (Carga Inicial)

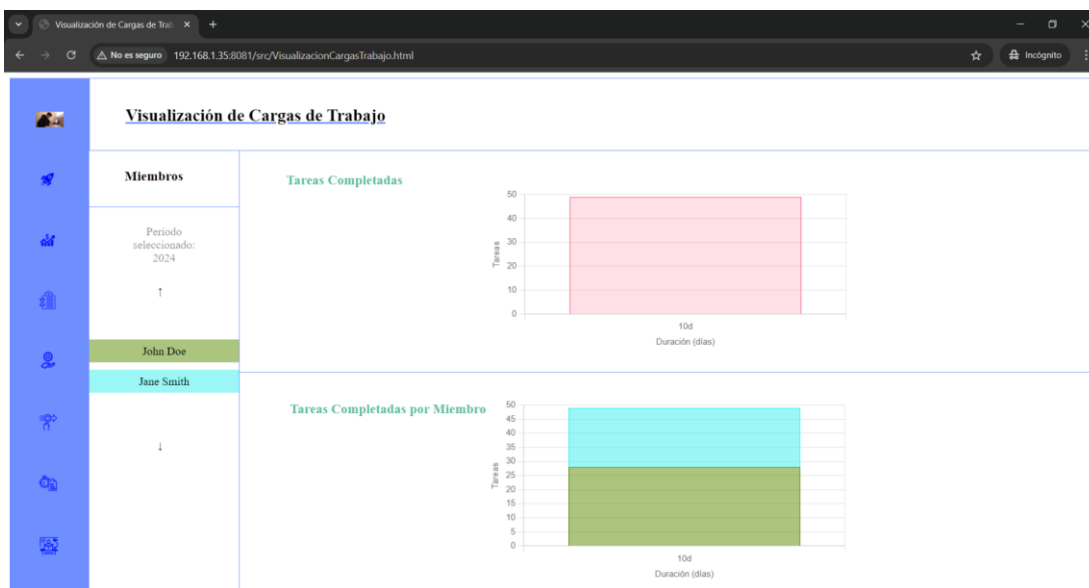


Figura 26. Pantalla de Visualización de Cargas de Trabajo (Información Cargada)

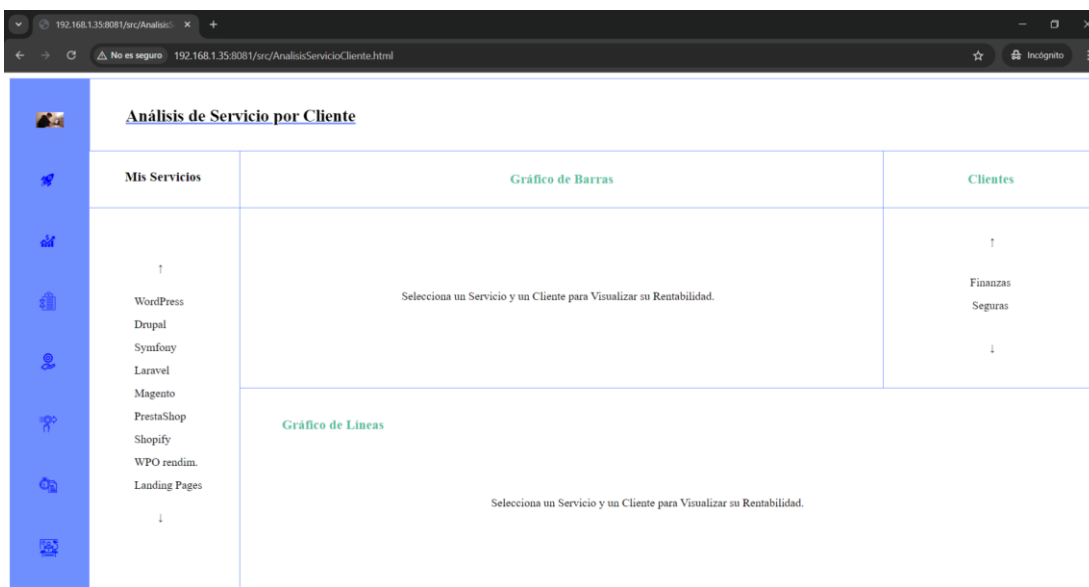


Figura 27. Pantalla de Análisis de Servicio por Cliente (Carga Inicial)

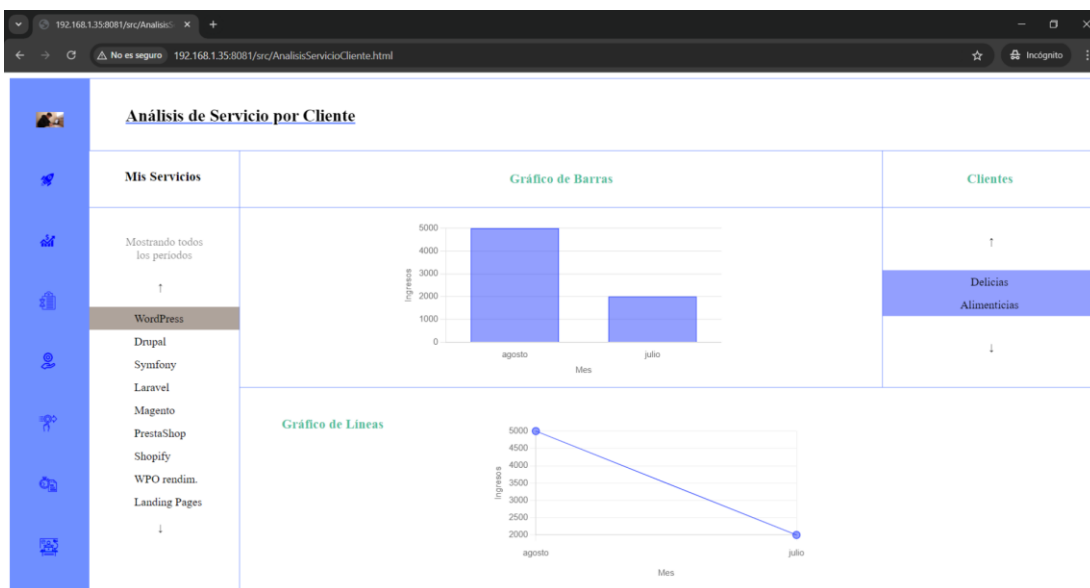


Figura 28. Pantalla de Análisis de Servicio por Cliente (Información Cargada)

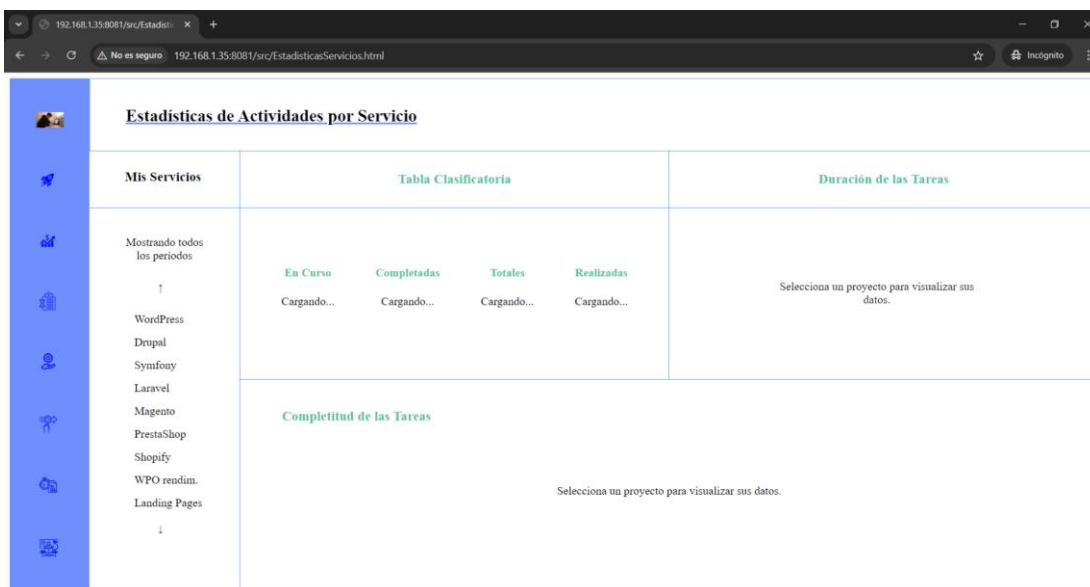


Figura 29. Pantalla de Estadísticas de Actividades por Servicio (Carga Inicial)

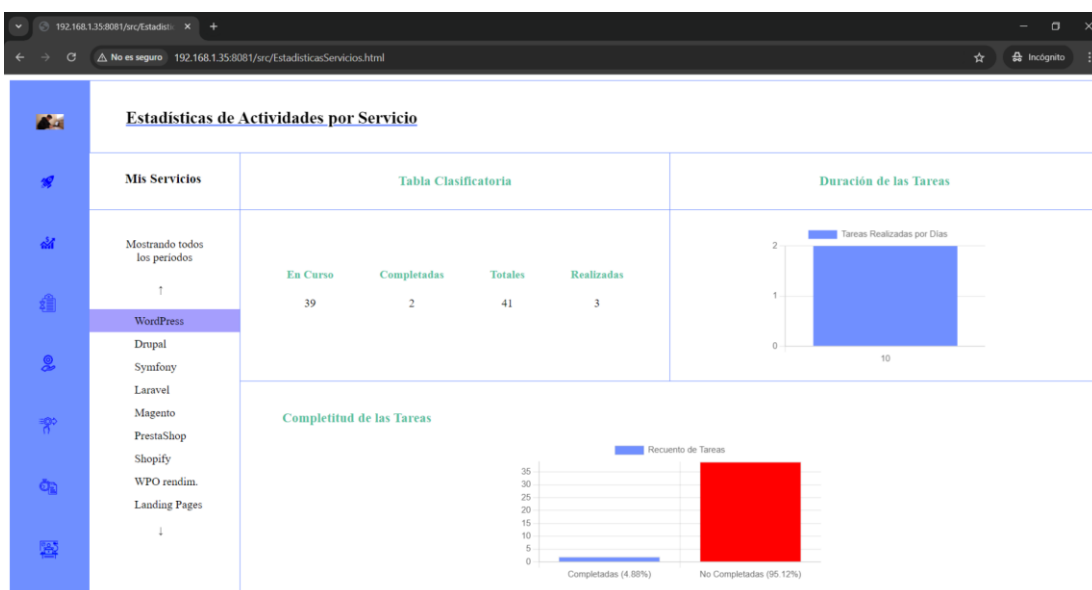


Figura 30. Pantalla de Estadísticas de Actividades por Servicio (Información Cargada)

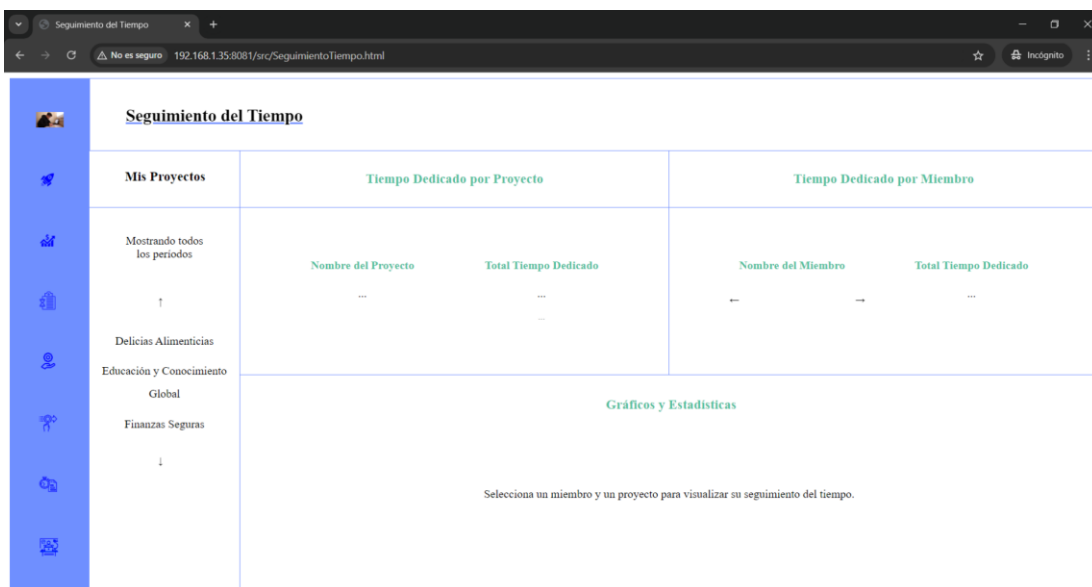


Figura 31. Pantalla de Seguimiento del Tiempo (Carga Inicial)

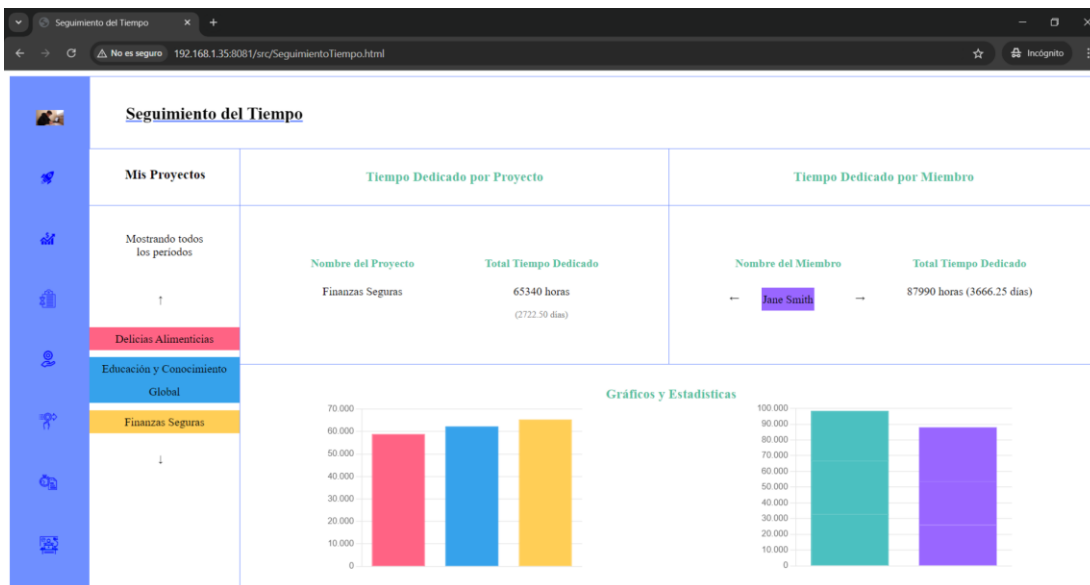


Figura 32. Pantalla de Seguimiento del Tiempo (Información Cargada)

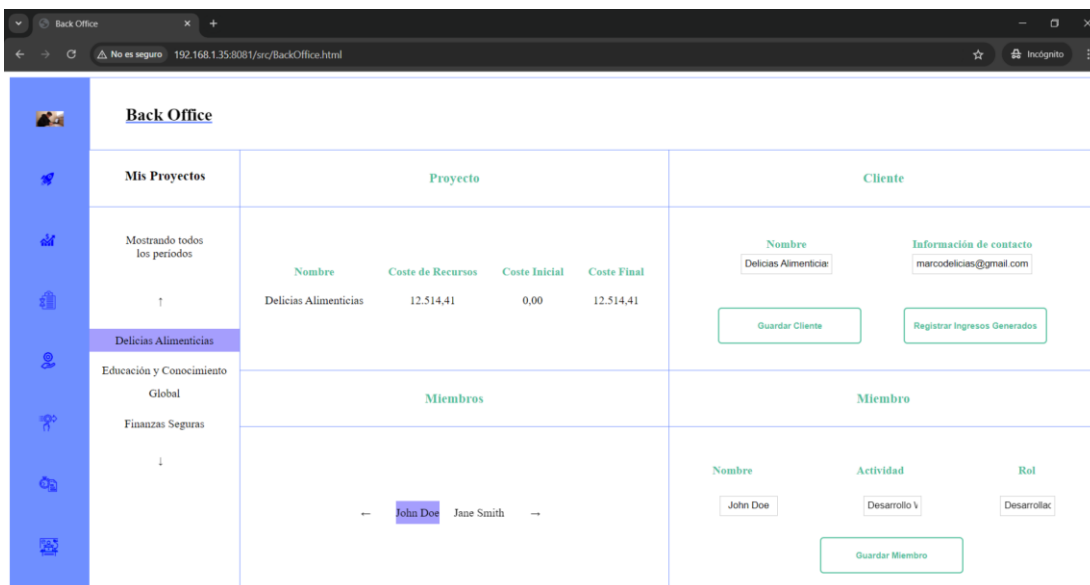


Figura 33. Pantalla de Back Office (Carga Inicial)

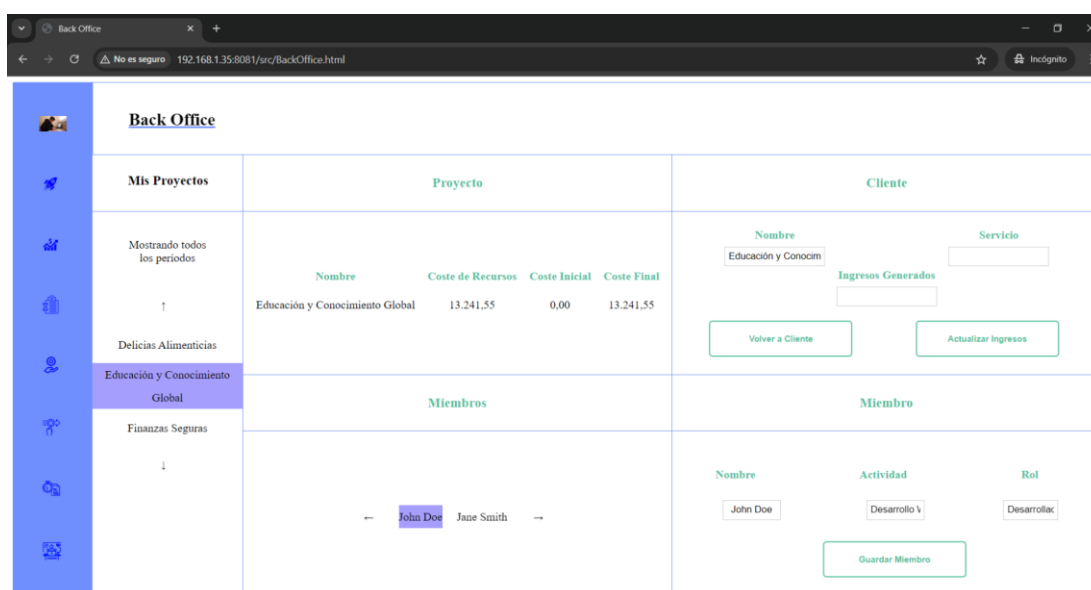


Figura 34. Pantalla de Back Office (Registro de Ingresos del Cliente de un Proyecto)

## 4.5 Diseño de la Persistencia de Datos

### 4.5.1 Estructura de la Base de Datos

La base de datos se compone de varias tablas que almacenan información relacionada con proyectos, tareas, servicios, miembros, clientes, clasificaciones, asignaciones y otros aspectos relevantes de la gestión de la aplicación.

A continuación, se describe la estructura de cada tabla, sus columnas, relaciones y restricciones clave.

#### Tabla *tasca\_nueva*:

- **Descripción:** Almacena información detallada sobre las tareas de los proyectos.
- **Columnas:**
  - **tasca\_id** (text, NOT NULL): Identificador único de la tarea.
  - **projecte\_id** (text): Identificador del proyecto al que pertenece la tarea.
  - **nom** (character varying(255)): Nombre de la tarea.
  - **descripcio** (text): Descripción de la tarea.
  - **estat** (character varying(255)): Estado de la tarea.
  - **data\_creacio** (character varying): Fecha de creación de la tarea.
  - **data\_limit** (character varying): Fecha límite de la tarea.
  - **data\_limit\_recordatori** (character varying): Fecha de recordatorio de la tarea.
  - **data\_finalitzacio** (character varying): Fecha de finalización de la tarea.
  - **temps\_destinat** (integer): Tiempo destinado a la tarea.
  - **temps\_estimat** (integer): Tiempo estimado para la tarea.

- **membres** (character varying[]): Miembros asignados a la tarea.
- **username** (character varying(255)): Nombre de usuario del propietario de la agencia.
- **Restricciones**
  - **tasca\_nueva\_tasca\_id\_key** (UNIQUE): Asegura que **tasca\_id** sea único.
  - **tasca\_nueva\_projecte\_id\_fkey** (FOREIGN KEY): Relaciona **projecte\_id** con **projecte(projecte\_id)**.

#### Tabla *servei*:

- **Descripción:** Almacena información sobre los diferentes servicios ofrecidos.
- **Columnas:**
  - **servei\_id** (integer, NOT NULL): Identificador único del servicio.
  - **nom\_servei** (character varying(255), NOT NULL): Nombre del servicio.
  - **categoria** (character varying(255), NOT NULL): Categoría del servicio.
  - **username** (character varying(255)): Nombre de usuario del propietario de la agencia.
- **Restricciones**
  - **servei\_pkey** (PRIMARY KEY): Asegura que **servei\_id** sea único.
  - **nombre\_unico** (UNIQUE): Asegura que **nom\_servei** sea único.

#### Tabla *projecte*:

- **Descripción:** Almacena información detallada sobre los proyectos.
- **Columnas:**
  - **projecte\_id** (text, NOT NULL): Identificador único del proyecto.
  - **nom** (character varying(255)): Nombre del proyecto.
  - **cost\_inicial** (numeric(10,2)): Coste inicial del proyecto.
  - **cost\_recurso** (numeric(10,2)): Coste de recursos del proyecto.
  - **cost\_final** (numeric(10,2)): Coste final del proyecto.
  - **client\_id** (text): Identificador del cliente asociado al proyecto.
  - **data\_ultima\_activitat** (character varying): Fecha de la última actividad del proyecto.
  - **data\_ultima\_vista** (character varying): Fecha de la última vista del proyecto.
  - **cost\_rd** (numeric(10,2)): Coste de I+D asociado al proyecto.
  - **cost equipamiento\_inicial** (numeric(10,2)): Coste de equipamiento inicial.
  - **cost\_configuracion** (numeric(10,2)): Coste de configuración.
  - **costes\_adicionales** (numeric(10,2) DEFAULT 0): Costes adicionales asociados al proyecto.
  - **username** (character varying(255)): Nombre de usuario del propietario de la agencia.

- **Restricciones**
  - **projecte\_pkey** (PRIMARY KEY): Asegura que **projecte\_id** sea único.
  - **projecte\_client\_id\_fkey** (FOREIGN KEY): Relaciona **client\_id** con **client(client\_id)**.

#### Tabla *membre*:

- **Descripción:** Contiene la información sobre los miembros de la agencia.
- **Columnas:**
  - **membre\_id** (text, NOT NULL): Identificador único del miembro.
  - **nom** (character varying(255)): Nombre del miembro.
  - **activitat** (text): Descripción de la actividad que realiza el miembro.
  - **rol** (text): Rol del miembro dentro de la agencia.
  - **nom\_usuari** (text): Nombre de usuario asociado al miembro.
  - **salari** (numeric(10,2)): Salario del miembro.
  - **username** (character varying(255)): Nombre de usuario del propietario de la agencia.
- **Restricciones**
  - **membre\_pkey** (PRIMARY KEY): Asegura que **membre\_id** sea único.

#### Tabla *client*:

- **Descripción:** Almacena los datos de los clientes de la agencia.
- **Columnas:**
  - **client\_id** (text, NOT NULL): Identificador único del cliente.
  - **nom** (character varying(255)): Nombre del cliente.
  - **informacio\_contacte** (text): Información de contacto del cliente.
  - **ingresos\_generados** (numeric(10,2)): Ingresos generados por el cliente.
  - **username** (character varying(255)): Nombre de usuario del propietario de la agencia.
- **Restricciones**
  - **client\_pkey** (PRIMARY KEY): Asegura que **client\_id** sea único.

#### Tabla *classificacio*:

- **Descripción:** Registra la clasificación de tareas dentro de servicios.
- **Columnas:**
  - **classificacio\_id** (integer, NOT NULL): Identificador único de la clasificación.
  - **tasca\_id** (text, NOT NULL): Identificador de la tarea.
  - **servei\_id** (integer, NOT NULL): Identificador del servicio.
  - **nom\_tasca** (character varying(255)): Nombre de la tarea.
- **Restricciones**

- `classificacio_pkey` (PRIMARY KEY): Asegura que `classificacio_id` sea único.
- `classificacio_unique_constraint` (UNIQUE): Asegura que la combinación de `tasca_id`, `servei_id` y `nom_tasca` sea única.
- `classificacio_servei_id_fkey` (FOREIGN KEY): Relaciona `servei_id` con `servei(servei_id)`.

#### Tabla *assignacions*:

- **Descripción:** Almacena las asignaciones de miembros a tareas específicas.
- **Columnas:**
  - `assignacio_id` (integer, NOT NULL): Identificador único de la asignación.
  - `tasca_id` (text): Identificador de la tarea asignada.
  - `membre_id` (text): Identificador del miembro asignado.
  - `temps_dedicat` (integer): Tiempo dedicado por el miembro a la tarea.
- **Restricciones**
  - `assignacions_pkey` (PRIMARY KEY): Asegura que `assignacio_id` sea único.
  - `assignacions_membre_id_fkey` (FOREIGN KEY): Relaciona `membre_id` con `membre(membre_id)`.
  - `assignacions_tasca_id_fkey` (FOREIGN KEY): Relaciona `tasca_id` con `tasca_nueva(tasca_id)`.

#### Tabla *materials*:

- **Descripción:** Almacena los materiales utilizados en los proyectos.
- **Columnas:**
  - `material_id` (integer, NOT NULL): Identificador único del material.
  - `projecte_id` (text, NOT NULL): Identificador del proyecto asociado al material.
  - `nom_material` (character varying(255), NOT NULL): Nombre del material.
  - `cost` (numeric(10,2), NOT NULL): Coste del material.
  - `username` (character varying(255)): Nombre de usuario del propietario de la agencia.
  - `fecha` (date): Fecha de adquisición o uso del material.
- **Restricciones**
  - `materials_pkey` (PRIMARY KEY): Asegura que `material_id` sea único.
  - `materials_projecte_id_fkey` (FOREIGN KEY): Relaciona `projecte_id` con `projecte(projecte_id)`.

**Tabla *ingresos\_servicio\_cliente*:**

- **Descripción:** Registra los ingresos generados por los servicios prestados a los clientes.
- **Columnas:**
  - **ingreso\_id** (integer, NOT NULL): Identificador único del registro de ingreso.
  - **client\_id** (text, NOT NULL): Identificador del cliente asociado al ingreso.
  - **servei\_id** (integer, NOT NULL): Identificador del servicio relacionado con el ingreso.
  - **fecha** (date, NOT NULL): Fecha en la que se registró el ingreso.
  - **ingresos** (numeric(10,2), NOT NULL): Monto del ingreso generado.
  - **username** (character varying(255)): Nombre de usuario del propietario de la agencia.
- **Restricciones**
  - **ingresos\_servicio\_cliente\_pkey** (PRIMARY KEY): Asegura que **ingreso\_id** sea único.
  - **ingresos\_servicio\_cliente\_client\_id\_fkey** (FOREIGN KEY): Relaciona **client\_id** con **client(client\_id)**.
  - **ingresos\_servicio\_cliente\_servei\_id\_fkey** (FOREIGN KEY): Relaciona **servei\_id** con **servei(servei\_id)**.

**Tabla *users*:**

- **Descripción:** Contiene la información de los usuarios del sistema.
- **Columnas:**
  - **user\_id** (integer, NOT NULL): Identificador único del usuario.
  - **username** (character varying(50), NOT NULL): Nombre de usuario único.
  - **password** (character varying(255), NOT NULL): Contraseña del usuario.
  - **profile\_image** (character varying(255)): URL o ruta de la imagen de perfil del usuario.
  - **trello\_api\_key** (character varying(255)): API Key de Trello del usuario.
  - **trello\_api\_token** (character varying(255)): API Token de Trello del usuario.
- **Restricciones**
  - **users\_pkey** (PRIMARY KEY): Asegura que **user\_id** sea único.
  - **users\_username\_key** (UNIQUE): Asegura que **username** sea único.

**Relaciones y Restricciones****Relación *tasca\_nueva* - *proyecto*:**

- Una tarea puede estar asociada a un proyecto, definido por la relación de llave foránea **tasca\_nueva\_proyecto\_id\_fkey** que referencia **proyecto(proyecto\_id)**.

**Relación projecte - client:**

- Un proyecto puede estar asociado a un cliente, definido por la relación de llave foránea projecte\_client\_id\_fkey que referencia client(client\_id).

**Relación classificacio - servei:**

- Una clasificación de tarea está relacionada con un servicio, definido por la relación de llave foránea classificacio\_servei\_id\_fkey que referencia servei(servei\_id).

**Relación assignacions - tasca\_nueva:**

- Una asignación está relacionada con una tarea, definido por la relación de llave foránea assignacions\_tasca\_id\_fkey que referencia tasca\_nueva(tasca\_id).

**Relación assignacions - membre:**

- Una asignación está relacionada con un miembro, definido por la relación de llave foránea assignacions\_membre\_id\_fkey que referencia membre(membre\_id).

**Relación materials - projecte:**

- Un material está relacionado con un proyecto, definido por la relación de llave foránea materials\_projecte\_id\_fkey que referencia projecte(projecte\_id).

**Relación ingresos\_servicio\_cliente - client y ingresos\_servicio\_cliente - servei:**

- Los ingresos registrados están asociados tanto a un cliente como a un servicio, definidos por las relaciones de llave foránea ingresos\_servicio\_cliente\_client\_id\_fkey y ingresos\_servicio\_cliente\_servei\_id\_fkey.

Esta estructura garantiza la integridad referencial y permite una gestión eficiente de los datos relacionados con proyectos, tareas, servicios, miembros, clientes, clasificaciones, asignaciones, materiales y usuarios dentro del sistema.

**4.5.2 Decisiones de Diseño detrás de la Elección del Modelo de Base de Datos**

La decisión de optar por un modelo de base de datos **relacional** para la estructura descrita se fundamenta en los requisitos específicos de la aplicación y en las características que este tipo de bases de datos ofrece en términos de integridad, consistencia y gestión de relaciones complejas entre datos.

**4.5.3 Razones para la Elección del Modelo Relacional**

**Integridad y Consistencia de los Datos:** La base de datos relacional proporciona mecanismos robustos para mantener la **integridad referencial** entre tablas, como se observa en las relaciones de llaves foráneas que garantizan que los datos vinculados entre

diferentes entidades (por ejemplo, proyectos y clientes, tareas y proyectos) permanezcan coherentes.

Esto es crucial para una aplicación que maneja información interrelacionada como la gestión de proyectos, asignación de tareas, y registro de servicios e ingresos.

**Estructura Definida y Esquemática:** Dado que la aplicación requiere una estructura de datos bien definida y esquematizada, con tipos de datos claros y restricciones (como las llaves primarias y restricciones UNIQUE), el modelo relacional es ideal.

Cada entidad de la aplicación, como tareas, proyectos, miembros y servicios, se representa claramente en tablas con relaciones establecidas entre ellas, lo que facilita tanto el diseño como el mantenimiento de la base de datos.

**Consultas Complejas y Análisis de Datos:** El uso de SQL, que es nativo en las bases de datos relacionales, permite realizar **consultas complejas** para obtener insights y análisis de datos.

La estructura relacional facilita la realización de uniones (JOINS) entre tablas para generar informes detallados, como el análisis de rentabilidad por cliente o la visualización de cargas de trabajo, donde la información se extrae de múltiples tablas interrelacionadas.

**Escalabilidad y Mantenimiento:** Aunque las bases de datos relacionales suelen ser más rígidas en términos de escalabilidad horizontal comparadas con las bases de datos NoSQL, su capacidad para manejar volúmenes de datos crecientes y realizar optimizaciones en consultas y diseño de índices sigue siendo adecuada para aplicaciones que necesitan un equilibrio entre estructura y rendimiento, como es el caso de esta aplicación.

**Seguridad y Control de Acceso:** El modelo relacional permite implementar mecanismos de **control de acceso** a nivel de base de datos, asegurando que solo los usuarios autorizados puedan realizar operaciones específicas.

En el contexto de esta aplicación, donde es fundamental proteger la información sensible de clientes, proyectos y usuarios, las bases de datos relacionales ofrecen una capa adicional de seguridad.

#### **4.5.4 Relación con los Requisitos de la Aplicación**

La estructura de la base de datos, con tablas como `tasca_nueva`, `servei`, `projecte`, `membre`, y `client`, refleja la necesidad de gestionar información que está fuertemente interrelacionada.

Los requisitos de la aplicación exigen que estas relaciones sean explícitas y que las operaciones sobre los datos aseguren la consistencia y validez en todo momento.

Por ejemplo, no sería posible eliminar un proyecto que aún tiene tareas o asignaciones activas, lo cual se asegura mediante las restricciones de integridad referencial proporcionadas por el modelo relacional.

Además, la naturaleza transaccional de la aplicación, donde las operaciones deben realizarse de manera atómica y asegurar que los cambios en los datos se completen con éxito, se gestiona eficientemente en un entorno de base de datos relacional.

En resumen, la elección de un modelo de base de datos relacional es una decisión alineada con los objetivos de la aplicación: ofrecer una gestión robusta, segura y coherente de datos complejos, con la capacidad de realizar consultas avanzadas y mantener una estructura clara y mantenible a lo largo del tiempo

#### ***4.5.5 Implementación de Operaciones de Lectura, Escritura y Consulta en la Base de Datos***

En esta sección se detalla cómo se implementan las operaciones de lectura, escritura y consulta en la base de datos dentro de la aplicación.

Estas operaciones son fundamentales para satisfacer las necesidades de la aplicación, permitiendo la manipulación eficiente de datos a través de la API.

##### ***Operación de Lectura: Obtener Información de Proyectos***

**Ruta:** /api/proyectos

**Método:** GET

**Descripción:** Esta operación permite a un usuario autenticado obtener todos los proyectos asociados a su cuenta.

La consulta SQL selecciona todos los registros de la tabla `projecte` que coinciden con el nombre de usuario autenticado.

```
app.get('/api/proyectos', authenticateJWT, async (req, res) => {
  const { username } = req.user;

  try {
    const proyectosQuery = await pool.query('SELECT * FROM
projecte WHERE username = $1', [username]);

    const proyectosData = proyectosQuery.rows;

    res.json(proyectosData);

  } catch (error) {
    console.error('Error al obtener los proyectos',
error); res.status(500).json({ error: 'Error al obtener los
proyectos' }); }
});
```

**Resultado:** Devuelve un array de objetos JSON que representan los proyectos del usuario autenticado. En caso de error, se responde con un código de estado 500.

### *Operación de Escritura: Autenticación y Registro de Usuarios*

**Ruta:** /api/login y /api/register

**Método:** POST

**Descripción:** Estas operaciones permiten la autenticación y el registro de usuarios en la aplicación.

La autenticación de usuarios valida las credenciales y genera un token JWT para sesiones autenticadas.

El registro de usuarios crea un nuevo registro en la base de datos con una contraseña encriptada.

```
/** * /api/login (POST): Autentica a un usuario y genera un token JWT para sesiones autenticadas. */
```

```
app.post('/api/login', async (req, res) => {
  const { username, password } = req.body;
  try {
    const userResult = await pool.query('SELECT * FROM users WHERE username = $1', [username]);
    if (userResult.rows.length > 0) {
      const user = userResult.rows[0];
      const match = await bcrypt.compare(password, user.password);
      if (match) {
        const token = jwt.sign({ username: user.username }, secretKey, { expiresIn: '1h' });
        res.json({
          success: true,
          profileImage: user.profile_image,
          token: token // Enviar el token en la respuesta JSON
        });
      } else {
        res.json({ success: false, error: 'Contraseña incorrecta' });
      }
    } else {
      res.json({ success: false, error: 'Usuario no encontrado' });
    }
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
});
```

```

        res.json({ success: false, error: 'Usuario no
        encontrado' });
    }
} catch (error) {
    console.error('Error al verificar las credenciales:',
    error);
    res.status(500).json({ success: false, error: 'Error
    al verificar las credenciales' });
}
});

/** * /api/register (POST): Registra un nuevo usuario en la base
de datos con una contraseña encriptada. */

app.post('/api/register', async (req, res) => {
    const { username, password } = req.body;
    try {
        const hashedPassword = await bcrypt.hash(password,
        saltRounds);
        await pool.query('INSERT INTO users (username, password)
        VALUES ($1, $2)', [username, hashedPassword]);
        res.json({ success: true });
    } catch (error) {
        console.error('Error al registrar el usuario:',
        error);
        res.status(500).json({ success: false, error:
        'Error al registrar el usuario' });
    }
});

```

**Resultado:**

- **/api/login:** Genera y devuelve un token JWT en la respuesta JSON si las credenciales son correctas. En caso de error, devuelve un mensaje descriptivo y un código de estado 500.
- **/api/register:** Registra un nuevo usuario con una contraseña encriptada y devuelve un mensaje de éxito. En caso de error, devuelve un mensaje descriptivo y un código de estado 500.

***Operación de Consulta: Obtener Tiempo Dedicado por Proyecto*****Ruta:** /api/tiempo/proyectos**Método:** GET

**Descripción:** Esta operación permite consultar el tiempo total dedicado a un proyecto específico por cada miembro del equipo. La consulta SQL suma el tiempo dedicado por cada miembro en todas las tareas asociadas al proyecto y lo devuelve en formato JSON.

```
app.get('/api/tiempo/proyectos', authenticateJWT, async (req,
res) => {
  const { nombre, year } = req.query;
  const { username } = req.user;
  try {
    if (nombre) {
      const proyecto = await pool.query('SELECT
projecte_id FROM projecte WHERE nom = $1 AND username
= $2', [nombre, username]);
      if (proyecto.rows.length === 0) {
        return res.json([]); // Devolver un array
vacío en lugar de 404
      }
      const                proyectoId                =
proyecto.rows[0].projecte_id;

      const yearFilter = year ? `AND EXTRACT(YEAR
FROM    TO_DATE(t.data_finalitzacio,    'YYYY-MM-
DD')) = ${year}` : '';

      const tareas = await pool.query(`SELECT
tasca_id FROM tasca_nueva t WHERE t.projecte_id
= $1 AND t.username = $2 ${yearFilter}`,
[proyectoId, username]);

      if (tareas.rows.length === 0) {
        return res.json([]); // Devolver un array
vacío en lugar de 404
      }

      const tareasIds = tareas.rows.map(t =>
t.tasca_id);

      const tiempos = await pool.query( `SELECT
membre.nom                AS                nombre_miembro,
SUM(assignacions.temps_dedicat) AS tiempo_total
FROM assignacions INNER JOIN membre ON
assignacions.membre_id = membre.membre_id WHERE
assignacions.tasca_id = ANY($1::text[]) GROUP
BY membre.nom`, [tareasIds] );
```

```

const resultado = tiempos.rows.map(tiempo => ({
  nombre_miembro: tiempo.nombre_miembro,
  tiempo_total: parseInt(tiempo.tiempo_total, 10)
}));

res.json(resultado);

} else {
  const proyectos = await pool.query('SELECT
projecte_id, nom FROM projecte WHERE username = $1',
[username]);
  res.json(proyectos.rows);
}
} catch (error) {
  console.error('Error al obtener el tiempo dedicado
por proyecto:', error);
  res.status(500).json({ error: 'Error interno del
servidor' });
}
});

```

**Resultado:** Devuelve un *array* de objetos JSON que representan el tiempo total dedicado por cada miembro a un proyecto específico. Si no se encuentra el proyecto o no se han registrado tiempos, devuelve un *array* vacío.

Estos ejemplos reflejan cómo mi aplicación interactúa con la base de datos para realizar operaciones fundamentales que soportan sus funcionalidades clave.

Cada operación está diseñada para manejar los datos de manera eficiente, asegurando la integridad y consistencia de la información almacenada.

## 5 Implementación

### 5.1 Tecnologías Utilizadas

Este apartado describe las tecnologías clave empleadas en el desarrollo de la aplicación, detallando las herramientas y lenguajes que forman la base técnica del sistema.

#### 5.1.1 Lenguajes de Programación

**JavaScript (ES6+):** Utilizado tanto en el backend como en el frontend debido a su versatilidad y la capacidad de compartir lógica entre ambos entornos.

En el backend, se usa con Node.js, y en el frontend, con la biblioteca Chart.js para visualizaciones gráficas y manipulación del DOM.

#### 5.1.2 Frameworks y Librerías del Backend

**Node.js:** Plataforma de desarrollo basada en JavaScript que permite construir aplicaciones de red escalables. Es la base del servidor de la aplicación.

**Express.js:** Framework minimalista y flexible para Node.js, utilizado para crear y gestionar el servidor y las rutas API de manera eficiente.

**Axios:** Librería para realizar solicitudes HTTP desde el backend, utilizada para interactuar con la API de Trello y obtener datos necesarios.

**natural:** Biblioteca de procesamiento del lenguaje natural, empleada para la tokenización y el análisis de términos clave en las descripciones y nombres de las tarjetas de Trello.

**pg:** Cliente de PostgreSQL para Node.js, utilizado para la interacción con la base de datos PostgreSQL.

#### 5.1.3 Frameworks y Librerías del Frontend

**Chart.js:** Librería de gráficos utilizada para crear visualizaciones interactivas y dinámicas de los datos del proyecto en el frontend.

**Fetch API:** Utilizada para realizar solicitudes HTTP desde el frontend al backend y obtener datos para las visualizaciones.

#### 5.1.4 Base de Datos

**PostgreSQL:** Sistema de gestión de bases de datos relacional utilizado para almacenar información de proyectos, tareas, servicios y usuarios. Seleccionada por su robustez, escalabilidad y soporte para transacciones complejas.

#### 5.1.5 Integraciones de API

**Trello API:** Utilizada para integrar la gestión de proyectos, sincronizando datos de tableros y tareas con la aplicación.

#### 5.1.6 Seguridad

**JWT (JSON Web Tokens):** Implementado para la autenticación segura de usuarios, permitiendo el acceso a rutas protegidas del servidor.

**bcrypt:** Utilizado para el hashing de contraseñas, asegurando que las credenciales de usuario estén protegidas contra accesos no autorizados.

## 5.2 Algoritmos Específicos

En este apartado se detallan los algoritmos y procesos clave implementados en la aplicación, los cuales son fundamentales para la gestión de datos, la interacción con APIs externas y la manipulación de información en la base de datos PostgreSQL.

### 5.2.1 *Procesamiento de Tareas y Proyectos desde Trello (app.js)*

Este conjunto de funciones se encarga de interactuar con la API de Trello para extraer información sobre tableros, tareas y miembros asociados, para posteriormente procesarla y almacenarla en la base de datos. Se destacan los siguientes algoritmos:

#### **Procesamiento de Tableros de Trello:**

El algoritmo **procesarTablero** recibe el ID de un tablero y obtiene toda la información relevante, como las tareas asociadas, miembros y datos del proyecto. La información se guarda o actualiza en la base de datos PostgreSQL.

#### **Clasificación de Servicios y Actividades:**

Mediante la función **procesarTarjeta**, se analizan las descripciones y nombres de las tarjetas de Trello utilizando técnicas de procesamiento del lenguaje natural (NLP) para identificar palabras clave y asociarlas a categorías predefinidas (como servicios de desarrollo web o marketing).

#### **Manejo de Tiempo y Recursos:**

Los algoritmos como **insertarActualizarTarea** y **insertarActualizarRelacionTareaMiembros** manejan la inserción y actualización de datos sobre tareas y la relación con miembros del equipo, incluyendo el tiempo dedicado a cada tarea.

### 5.2.2 *Carga y Procesamiento de Archivos CSV (loadcsv.js)*

Este script automatiza la lectura de archivos CSV que contienen información sobre tareas y proyectos, actualizando las fechas de creación y finalización de las tareas en la base de datos PostgreSQL. Se destacan los siguientes algoritmos:

#### **Lectura y Parsing de Archivos CSV:**

La función **leerCSV** utiliza la librería **csv-parser** para leer y analizar los datos del archivo CSV, transformando cada fila en un objeto que luego será procesado.

#### **Generación Aleatoria de Fechas:**

El algoritmo **generarFechaAleatoria** crea una fecha de inicio aleatoria para una tarea, asegurando que esta fecha sea anterior a la fecha de finalización. Esto es crucial para simular un rango de tiempo en la creación de tareas.

**Actualización de Fechas en la Base de Datos:**

La función **actualizarTarea** recibe la información de una tarea desde el CSV y actualiza las fechas de creación y finalización correspondientes en la base de datos, asegurando la coherencia de los datos históricos.

**5.2.3 Manejo de Autenticación y Seguridad (app.js)**

La aplicación implementa un sistema de autenticación basado en JSON Web Tokens (JWT) para proteger las rutas y garantizar que solo los usuarios autenticados puedan acceder a ciertos recursos.

**Autenticación JWT:**

El middleware **authenticateJWT** verifica la validez de los tokens JWT en las solicitudes entrantes. Si el token es válido, permite el acceso a la ruta; de lo contrario, bloquea la solicitud.

**Encriptación de Contraseñas:**

Se utiliza **bcrypt** para el hashing seguro de contraseñas antes de almacenarlas en la base de datos. Esto asegura que las contraseñas no se almacenen en texto plano, protegiendo la información del usuario.

## 6 Evaluación

### 6.1 Diseño de los Casos de Prueba

El diseño de los casos de prueba ha cubierto exhaustivamente todas las funcionalidades clave de la aplicación, desde las pantallas de bienvenida e inicio de sesión hasta las distintas secciones dedicadas al análisis de costos, rentabilidad, seguimiento del tiempo y back-office.

Cada pantalla ha sido probada en múltiples escenarios, incluyendo la selección y desección de elementos, la navegación entre secciones y la edición de datos.

Además, se han incluido pruebas para validar requisitos no funcionales críticos como el rendimiento y la seguridad, asegurando que la aplicación cumpla con los estándares de calidad establecidos.

**Nota:** Los casos de prueba detallados se encuentran disponibles en el apartado de **Anexos** de este documento

### 6.2 Resultados Detallados de las Pruebas

Las pruebas realizadas han mostrado que la aplicación se comporta conforme a lo esperado en la mayoría de los escenarios. A continuación, se detalla el desempeño de cada área:

**Pantalla de Bienvenida:** Todas las funcionalidades de la pantalla de bienvenida, incluyendo la carga inicial y la navegación a la pantalla de inicio de sesión, han funcionado sin problemas.

**Pantalla de Inicio de Sesión:** La carga, el proceso de registro, la autenticación y la gestión de la información del usuario se completan exitosamente, con una pequeña observación respecto a la actualización de datos que requiere recargar la página.

**Pantalla de Informe de Costos por Proyecto:** La pantalla se carga correctamente, y las funcionalidades de selección, comparación y desección de proyectos funcionan como se esperaba, con una adecuada actualización de los gráficos y datos.

**Pantalla de Análisis de Rentabilidad por Cliente:** La selección y desección de clientes, así como la comparación de su rentabilidad, se realizan correctamente, con una interfaz que responde de manera adecuada.

**Pantalla de Visualización de Cargas de Trabajo:** La selección y comparación de miembros de la agencia funciona adecuadamente, con gráficos que se actualizan correctamente según lo esperado.

**Pantalla de Análisis de Servicio por Cliente:** Todas las pruebas de selección y comparación de servicios y clientes muestran resultados positivos, con gráficos y datos que se actualizan según las expectativas.

**Pantalla de Estadísticas de Actividades por Servicio:** La pantalla se carga correctamente y permite una adecuada selección y filtración de tareas. Se ha observado que en algunos casos se requería deseccionar y volver a seleccionar un servicio para actualizar el período en el que se visualizan los datos.

**Pantalla de Seguimiento del Tiempo:** La selección de proyectos y miembros, así como la comparación de datos entre ellos, se realiza correctamente. La desección de elementos y la navegación a otras secciones también funcionan sin inconvenientes.

**Pantalla de Back-Office:** La carga inicial, la selección y deselección de proyectos y miembros, y la edición de información se completan con éxito, mostrando datos actualizados en la interfaz. La navegación entre secciones también funciona sin problemas.

En cuanto a los **requisitos no funcionales**, el rendimiento de la aplicación es generalmente positivo, aunque se ha observado una pequeña espera durante la carga de credenciales de Trello, que no afecta significativamente la usabilidad.

En cuanto a la **seguridad**, las contraseñas se almacenan correctamente utilizando bcrypt, cumpliendo con los estándares establecidos.

### 6.3 Evaluación del Proyecto

En general, la evaluación del proyecto es positiva, con la mayoría de las funcionalidades funcionando conforme a los requisitos definidos. El sistema ha demostrado ser funcional, seguro y eficiente en su rendimiento, cumpliendo con las expectativas del usuario final.

#### Aspectos positivos:

**Funcionalidad:** La aplicación ofrece todas las funcionalidades clave necesarias para la gestión de usuarios, análisis de costos, rentabilidad, y seguimiento de tiempo. Las pruebas confirman que estas funcionalidades operan de manera fluida y sin problemas significativos.

**Seguridad:** La implementación de medidas de seguridad, como el almacenamiento de contraseñas con bcrypt, asegura la protección de los datos de los usuarios, cumpliendo con los estándares de seguridad requeridos.

**Rendimiento:** En la mayoría de los casos, el sistema responde rápidamente, ofreciendo una experiencia de usuario fluida, incluso bajo condiciones normales de carga.

#### Áreas de mejora:

**Interfaz de Usuario:** Se ha identificado que en algunos casos, como la actualización de la información después de la edición de datos, la interfaz requiere una recarga de la página para reflejar los cambios. Mejorar este aspecto podría aumentar la eficiencia y satisfacción del usuario.

**Manejo de Errores:** Aunque el sistema maneja adecuadamente los errores comunes, podría beneficiarse de una mayor personalización en los mensajes de error para mejorar la claridad y la orientación al usuario final.

**Seguridad de Datos:** La encriptación de datos sensibles (datos financieros) en la base de datos no ha sido implementada en esta versión. Esto representa un área crítica de mejora para garantizar la protección de la información del cliente.

**Conclusión:** El proyecto es considerado exitoso en cumplir con los objetivos planteados, proporcionando una herramienta robusta y funcional para la gestión de proyectos y clientes. Con algunas áreas menores que podrían optimizarse en futuras versiones, la aplicación cumple con las expectativas y estándares de calidad establecidos.

## 7 Evaluación de Costes

Este apartado detalla los costos estimados para el desarrollo y mantenimiento de la aplicación web, desglosando tanto los costos de personal involucrado como los materiales necesarios. El objetivo es proporcionar una visión clara y precisa del presupuesto necesario para la ejecución efectiva del proyecto

### 7.1 Costes de Personal

#### 7.1.1 Personal Involucrado

El desarrollo de esta aplicación requiere un equipo multidisciplinario con roles bien definidos, cada uno de los cuales es crucial para asegurar que todas las áreas del proyecto estén adecuadamente cubiertas:

##### 1. Desarrollador Full Stack

**Descripción:** Este rol es fundamental para el desarrollo tanto del frontend como del backend de la aplicación. El desarrollador full stack es responsable de implementar la lógica de negocio, gestionar las bases de datos, y crear interfaces de usuario atractivas y funcionales.

**Salario Base Anual:** 26.534,12 €.

##### 2. Administrador de Sistemas

**Descripción:** El administrador de sistemas se encarga de la infraestructura técnica que soporta la aplicación. Esto incluye la gestión de servidores, bases de datos y la seguridad del sistema, asegurando que la aplicación sea robusta y esté siempre disponible.

**Salario Base Anual:** 26.534,12 €.

##### 3. Gerente de Proyecto

**Descripción:** El gerente de proyecto es responsable de coordinar al equipo, planificar el desarrollo, y asegurar que el proyecto se complete a tiempo y dentro del presupuesto. Este rol es clave para la gestión de los recursos y la comunicación entre los miembros del equipo y otros interesados.

**Salario Base Anual:** 28.850,23 €.

#### 7.1.2 Costo Total del Personal Anual

Para calcular el costo total de personal anual, se suman los salarios base anuales de cada miembro del equipo:

**Total Personal Anual:** 26.534,12 € (Desarrollador Full Stack) + 26.534,12 € (Administrador de Sistemas) + 28.850,23 € (Gerente de Proyecto) = **81.918,47 €**

Este total representa el gasto en recursos humanos necesarios para llevar a cabo el desarrollo y mantenimiento de la aplicación durante un año.

## 7.2 Costes de Material

### 7.2.1 Material Necesario

Además del personal, el proyecto requiere una serie de recursos materiales y servicios que son fundamentales para su implementación y operación:

#### Servidores y Hosting

**Descripción:** Estos son esenciales para alojar la aplicación web y asegurar su disponibilidad continua. Incluye tanto los servidores físicos como los servicios de hosting en la nube.

##### Ejemplo:

**Amazon Web Services (AWS):** Utilización de instancias EC2 para alojar la aplicación web, con una configuración básica que incluya balanceo de carga y copias de seguridad automáticas.

**Costo Anual Estimado:** 1.500 €.

#### Servicios en la Nube

**Descripción:** Los servicios en la nube proporcionan bases de datos, almacenamiento y otras infraestructuras necesarias para que la aplicación funcione eficientemente.

##### Ejemplo:

**Amazon RDS:** Base de datos gestionada en la nube para almacenar la información de la aplicación.

**Amazon S3:** Almacenamiento de archivos para guardar imágenes y otros recursos estáticos de la aplicación.

**Costo Anual Estimado:** 2.000 €.

#### Equipo Informático

**Descripción:** El hardware necesario para que el equipo de desarrollo pueda trabajar eficientemente incluye ordenadores, monitores, y otros equipos de oficina.

##### Ejemplo:

**Ordenadores de Desarrollo:** Equipos de alto rendimiento con procesadores Intel i7/i9, 16GB de RAM, y discos SSD de 1TB.

**Monitores:** Pantallas de 27 pulgadas con resolución 4K para cada miembro del equipo.

**Costo Inicial Estimado:** 12.000 €.

#### Gastos Administrativos

**Descripción:** Incluyen costos operativos tales como electricidad, internet, alquiler de oficinas, y otros gastos administrativos necesarios para mantener el funcionamiento del equipo.

##### Ejemplo:

**Electricidad e Internet:** Servicios de electricidad e internet de alta velocidad para la oficina.

**Alquiler de Oficina:** Espacio de trabajo físico para el equipo de desarrollo.

**Costo Anual Estimado:** 5.000 €.

### ***7.2.2 Costo Total del Material Anual***

Sumando todos los costos de material y servicios necesarios:

**Total Material Anual:** 1.500 € (Servidores y Hosting) + 2.000 € (Servicios en la Nube) + 12.000 € (Equipo Informático) + 5.000 € (Gastos Administrativos) = **20.500 €**

### **7.3 Resumen de Evaluación de Costes**

Finalmente, al combinar los costos de personal y los costos de material, se obtiene el costo total anual del proyecto:

**Total del Personal Anual:** 81.918,47 €

**Total del Material Anual:** 20.500 €

**Costo Total Anual del Proyecto:** 102.418,47 €

Este desglose proporciona una imagen clara del presupuesto necesario para la ejecución y operación del proyecto, lo que facilita la planificación financiera y la toma de decisiones estratégicas.

## 8 Legislación y Protección de Datos

En esta sección, se aborda la legislación aplicable en materia de protección de datos personales para la información manejada por la aplicación.

Se identifican los datos personales relevantes almacenados en la base de datos y se detalla la normativa de protección de datos que debe aplicarse para garantizar el cumplimiento de las obligaciones legales correspondientes.

### 8.1 Identificación de Datos Personales en la Aplicación

La aplicación gestiona varios tipos de datos que, según la legislación vigente, se consideran datos personales o sensibles, especialmente en el contexto de los propietarios de agencias de marketing digital. A continuación, se enumeran los datos relevantes:

#### Datos Personales Directos:

**Nombre de Usuario (username):** Este dato identifica directamente al propietario de la agencia, por lo que se clasifica como un dato personal.

**Contraseña del Usuario (password):** Considerado un dato crítico, su protección es esencial para garantizar la seguridad de la cuenta.

**Imagen de Perfil (profile\_image):** Aunque menos sensible, la imagen de perfil también es un dato personal, especialmente si permite identificar al usuario.

#### Datos Financieros y Empresariales Asociados a Personas:

**Ingresos Generados por Clientes:** Este dato está vinculado a clientes específicos y se considera información sensible, reflejando la situación financiera de la agencia.

**Salario de los Miembros de la Agencia (salari):** Dato financiero que, al estar asociado a un miembro específico de la agencia, debe ser protegido adecuadamente.

**Costos de los Proyectos (cost\_inicial, cost\_recurso, cost\_final, etc.):** Estos datos reflejan la inversión y gastos de la agencia en cada proyecto. Al estar vinculados al propietario y su empresa, también son considerados datos sensibles que requieren protección.

#### Datos de Acceso y Autenticación:

**Trello API Key y API Token:** Estos datos permiten acceder a cuentas y servicios externos (Trello), por lo que su protección es fundamental para evitar accesos no autorizados.

### 8.2 Normativa de Protección de Datos Aplicable

#### 8.2.1 Reglamento General de Protección de Datos (RGPD)

**Ámbito de Aplicación:** El RGPD se aplica a cualquier dato personal que pueda identificar a una persona física en la Unión Europea. Esto incluye nombres de usuario, contraseñas, costos de proyectos y cualquier otro dato identificativo o financiero que afecte al propietario o su empresa.

#### Principales Obligaciones:

**Consentimiento:** Es obligatorio obtener el consentimiento explícito de los usuarios antes de procesar sus datos personales.

**Transparencia:** Los usuarios deben ser informados de manera clara sobre el uso de sus datos, las finalidades del tratamiento y el tiempo durante el cual se conservarán.

**Derechos de los Usuarios:** La aplicación debe permitir a los usuarios ejercer sus derechos de acceso, rectificación, eliminación y portabilidad de sus datos.

### 8.2.2 *Ley Orgánica de Protección de Datos Personales y Garantía de los Derechos Digitales (LOPDGDD)*

**Ámbito de Aplicación:** Complementa el RGPD en España, ofreciendo un marco adicional para la protección de datos personales.

#### **Principales Obligaciones:**

**Medidas de Seguridad:** Se deben implementar medidas técnicas y organizativas para garantizar la seguridad de los datos personales, incluidos los datos financieros y de proyectos de la agencia.

**Registro de Actividades:** La aplicación debe mantener un registro de todas las actividades de tratamiento de datos personales.

**Responsabilidad Proactiva:** Es necesario demostrar que la aplicación cumple con los principios de protección de datos de manera proactiva.

### 8.2.3 *Leyes Sectoriales y de Confidencialidad*

**Ley de Competencia Desleal y Protección del Secreto Empresarial:** Esta ley es relevante para la protección de datos empresariales y financieros, asegurando la confidencialidad y el uso adecuado de información sensible, como los costos de los proyectos.

### 8.3 **Medidas de Protección y Cumplimiento Normativo (Futuras Implementaciones)**

En futuras versiones de la aplicación, se implementarán las siguientes medidas de protección y cumplimiento normativo para mejorar la seguridad y la conformidad legal:

#### **Cifrado de Datos:**

**Plan de Implementación:** Todos los datos sensibles, incluidos los ingresos, costos de proyectos, y otros datos financieros, serán cifrados tanto en tránsito como en reposo. Actualmente, estos datos no están cifrados, pero se espera implementar esta medida en próximas actualizaciones para garantizar una mayor seguridad.

#### **Política de Privacidad:**

**Plan de Implementación:** Se desarrollará una política de privacidad clara y accesible que explique cómo se recopilan, utilizan y protegen los datos personales de los usuarios, incluyendo detalles sobre los costos de los proyectos y otros datos financieros sensibles.

#### **Gestión del Consentimiento:**

**Plan de Implementación:** Se implementará un sistema para obtener el consentimiento explícito de los usuarios antes de cualquier tratamiento de sus datos

personales. Además, se mantendrá un registro del consentimiento otorgado por cada usuario.

**Auditorías de Seguridad:**

**Plan de Implementación:** Se programarán auditorías de seguridad periódicas para garantizar que las medidas de seguridad implementadas sean efectivas y cumplan con las normativas vigentes.

**8.4 Conclusión**

La aplicación tiene previsto el diseño para cumplir con las normativas de protección de datos personales aplicables, específicamente el RGPD y la LOPDGDD.

Aunque actualmente no se cifran los ingresos, costos de proyectos y otros datos financieros en la base de datos, se espera implementar esta y otras medidas de seguridad en futuras versiones para reforzar la protección de los datos financieros y personales de los usuarios.

La implementación de estas mejoras garantizará que la aplicación continúe cumpliendo con los estándares de seguridad y privacidad requeridos.

## 9 Conclusiones

El desarrollo de esta aplicación ha sido un proyecto que ha abarcado múltiples fases, desde la planificación y el diseño inicial hasta la implementación y evaluación final.

A través de este proceso, he alcanzado los objetivos establecidos, logrando una solución robusta y bien documentada que cubre todos los aspectos fundamentales del proyecto.

A continuación, presento una evaluación general del trabajo realizado, destacando su aportación tanto al desarrollo del proyecto como en mi formación personal.

### 9.1 Evaluación del Trabajo Realizado

**Introducción:** Desde el inicio, establecí una visión clara del proyecto, definiendo los objetivos y el alcance de la aplicación.

Esta claridad facilitó el foco en los objetivos clave, asegurando que todas las decisiones posteriores estuvieran dirigidas a cumplir con los fines del trabajo.

**Planificación:** La planificación fue un pilar fundamental para el éxito del proyecto.

Desarrollé un plan detallado que incluía un cronograma de actividades y la asignación de recursos.

Esta planificación me permitió gestionar eficazmente el tiempo y los recursos, pudiendo tener un desarrollo fluido y sin contratiempos mayores.

La metodología utilizada ha hecho más sencillo la adaptación a cambios inesperados, manteniendo el proyecto dentro de los plazos establecidos.

**Requisitos del Proyecto:** La fase de análisis de requisitos fue crítica para definir las funcionalidades y características necesarias.

Con un análisis previo, identifiqué los casos de uso y establecí los requisitos funcionales y no funcionales.

Esta base me permitió un diseño dirigido a satisfacer las necesidades específicas del usuario final, asegurando que la aplicación cubriera todas las expectativas.

**Diseño:** El diseño de la aplicación, basado en una arquitectura de tres capas, ha demostrado ser eficiente y escalable.

La separación clara entre la capa de presentación, la lógica de negocio y el acceso a datos ha facilitado el desarrollo modular, permitiendo que cada componente se desarrollara y probara de manera independiente.

Además, el diseño de la base de datos relacional ha asegurado la integridad y consistencia de los datos, lo que es esencial para la confiabilidad de la aplicación.

**Implementación:** La fase de implementación se llevó a cabo utilizando tecnologías modernas que aseguraron un rendimiento óptimo y una experiencia de usuario satisfactoria.

La integración con APIs externas, como Trello, y la implementación de sistemas de seguridad avanzados, como la autenticación con JWT y el hashing de contraseñas con bcrypt, son testimonio de un enfoque en la seguridad y la eficiencia del sistema.

Los algoritmos específicos desarrollados para el procesamiento de datos y la gestión de información han demostrado ser efectivos y bien optimizados.

**Evaluación:** Las pruebas realizadas han confirmado que la aplicación funciona conforme a los requisitos especificados, con un rendimiento adecuado y una interfaz de usuario intuitiva.

Aunque se han identificado algunas áreas de mejora, como la necesidad de optimizar la actualización de datos en tiempo real, el sistema en general cumple con las expectativas de funcionalidad, seguridad y rendimiento.

La documentación detallada de los casos de prueba asegura que cualquier problema pueda ser identificado y solucionado rápidamente.

**Evaluación de Costes:** El análisis de costes proporciona una visión clara de los recursos necesarios para el desarrollo y mantenimiento del proyecto.

Se han identificado y cuantificado tanto los costes de personal como los materiales y servicios necesarios, lo que permite una planificación financiera sólida.

Este foco en la evaluación de costes asegura que el proyecto sea sostenible y que los recursos sean utilizados de manera eficiente.

**Legislación y Protección de Datos:** La consideración de la legislación aplicable, especialmente en términos de protección de datos, ha sido otra parte esencial del proyecto.

Se han identificado los datos personales gestionados por la aplicación y se han pensado en medidas de seguridad para implementar con el fin de proteger esta información, asegurando el cumplimiento con normativas como el RGPD y la LOPDGDD.

### 9.2 Aportación a la Formación Profesional

Este proyecto ha sido una experiencia enriquecedora que me ha permitido desarrollar y consolidar habilidades en diversas áreas clave del desarrollo de software.

Desde la planificación y gestión de proyectos hasta la implementación de sistemas complejos y la consideración de aspectos legales, cada fase del proyecto ha contribuido a un aprendizaje significativo.

La capacidad de integrar múltiples tecnologías y tener en cuenta las posibilidades para asegurar la calidad y seguridad del producto final es ejemplo del crecimiento técnico y profesional que he alcanzado durante el desarrollo de esta aplicación.

### 9.3 Conclusión

El proyecto ha sido un éxito en términos de cumplimiento de los objetivos planteados y ha proporcionado una solución sólida para la gestión de proyectos y clientes.

La aplicación es un buen mínimo producto viable para futuras expansiones y mejoras, con una base técnica y funcional robusta.

A nivel personal, el proyecto ha sido una oportunidad valiosa para aplicar conocimientos teóricos en un entorno práctico, lo que ha resultado en un avance significativo en mis habilidades técnicas y de gestión.

Con algunas áreas para optimizar, la aplicación cumple con los estándares de calidad establecidos y la considero un proyecto exitoso tanto desde una perspectiva técnica como formativa.

## 10 Recursos Utilizados

### 10.1 Bibliografía y Documentación Técnica

Para garantizar el éxito en la implementación del proyecto, se recurrió a diversas fuentes de documentación técnica y bibliografía especializada, las cuales fueron cruciales para entender y aplicar las tecnologías y buenas prácticas de desarrollo de software.

**Documentación de Chart.js:** Fundamental para la implementación de gráficos interactivos en el frontend, permitiendo una visualización clara y efectiva de los datos.

**Manuales y Guías de la API de Trello:** Indispensables para la integración de Trello con la aplicación, asegurando una extracción y sincronización de datos eficiente.

**ChatGPT:** Mi asistente de desarrollo, que me ha asistido en todo el proyecto, proporcionando orientación, resolución de problemas y apoyo en la toma de decisiones técnicas.

### 10.2 Software Utilizado

El desarrollo del proyecto se apoyó en diversas herramientas de software que facilitaron tanto la programación como la gestión de proyectos.

**Visual Studio Code:** Editor de código empleado para el desarrollo del backend y frontend. Su integración con extensiones específicas para JavaScript y Node.js mejoró significativamente mi productividad.

**Node.js y NPM:** Node.js sirvió como la plataforma base para la ejecución del servidor, mientras que NPM (Node Package Manager) se utilizó para la instalación y gestión de las dependencias necesarias.

**PostgreSQL:** Sistema de gestión de bases de datos relacional utilizado para almacenar y gestionar la información de la aplicación, elegido por su robustez y capacidad para manejar grandes volúmenes de datos.

**Trello:** Herramienta de gestión de proyectos utilizada para organizar tareas, asignar responsabilidades y monitorizar el progreso del desarrollo, además de integrarse con la aplicación para la sincronización de datos.

**Chart.js:** Biblioteca de JavaScript implementada en el frontend para la creación de gráficos interactivos y dinámicos que visualizan datos clave de los proyectos.

**csv-parser y Natural (Node.js):** Bibliotecas de Node.js utilizadas para el procesamiento de archivos CSV y el análisis de lenguaje natural, fundamentales para la manipulación de datos y la clasificación automática de tareas.

### 10.3 Hardware Utilizado

**Mi ordenador personal:** Utilizado para todo el desarrollo del proyecto. Este hardware fue esencial para programar, probar y ejecutar la aplicación en un entorno local.

### 10.4 Servicios en la Nube y Hosting

**Entorno Local:** La aplicación no está actualmente alojada en un servidor externo; solo es accesible desde mi máquina local.

Esto se debe a que me enfoqué en crear un mínimo producto viable de la aplicación, priorizando la funcionalidad básica y la implementación inicial.

## 11 Anexos

### 11.1 Instalación y Requisitos

Para poner en marcha la aplicación en un entorno local, sigue los pasos y asegúrate de cumplir con los siguientes requisitos:

#### Requisitos del Sistema:

**Sistema Operativo:** Windows, macOS, o Linux.

**RAM:** Mínimo 8 GB.

**Espacio en Disco:** Al menos 500 MB para instalar las dependencias.

**Node.js:** Versión 14.x o superior.

**NPM (Node Package Manager):** Versión 6.x o superior.

**PostgreSQL:** Versión 12.x o superior, con una base de datos configurada para la aplicación.

#### Proceso de Instalación:

##### 1. Clonar el Repositorio del Proyecto:

Clona el repositorio desde GitHub a tu máquina local utilizando el siguiente enlace:

[Aplicacion-Rendimiento-Agencias](https://github.com/germanhernandezmairal/Aplicacion-Rendimiento-Agencias)

Usa el siguiente comando:

```
git clone
```

```
https://github.com/germanhernandezmairal/Aplicacion-Rendimiento-Agencias.git
```

##### 2. Instalar Dependencias:

Navega a la carpeta del proyecto en tu terminal y ejecuta el siguiente comando para instalar todas las dependencias necesarias:

```
npm install
```

##### 3. Configurar la Base de Datos:

Crea una base de datos PostgreSQL utilizando las credenciales de tu entorno local.

Importa el esquema de la base de datos proporcionado en el archivo **database/esquema.sql** que se encuentra en el repositorio.

Usa el siguiente comando:

```
psql -U tu_usuario -d tu_base_de_datos -f database/esquema.sql
```

Esto cargará la estructura de la base de datos (sin datos) necesaria para que la aplicación funcione correctamente.

#### 4. Configurar Variables de Entorno:

Abre el archivo **app.js** y revisa las credenciales del Trello proporcionadas para la cuenta de prueba del usuario "Germán Hernández Mairal".

Asegúrate de modificar también las credenciales de la base de datos en el archivo **app.js** para que coincidan con tu entorno local.

#### 5. Iniciar la Aplicación:

Una vez configurado todo, inicia la aplicación ejecutando los siguientes comandos en dos terminales diferentes:

##### Terminal 1:

```
http-server -p 8081
```

##### Terminal 2:

```
cd Server  
node app.js
```

La aplicación estará disponible en <http://192.168.1.35:8081> por defecto.

## 11.2 Documentación para el Uso y Mantenimiento

### Guía de Uso:

**Acceso:** Al iniciar la aplicación, accede desde tu navegador a la URL local <http://192.168.1.35:8081>.

**Registro de Sesión:** Puedes registrar un nuevo usuario con las credenciales proporcionadas.

**Nombre de Usuario:** 'El que elijas'

**Contraseña:** 'La que elijas'

**Credenciales del Trello:** Usa la ApiKey y el Token proporcionados en **app.js**.

**Inicio de Sesión:** Utiliza las credenciales utilizadas en registro de sesión.

**Navegación:** Explora las diferentes secciones de la aplicación, como la gestión de proyectos, análisis de rentabilidad, y seguimiento de tiempo.

### Mantenimiento:

#### Actualización de Dependencias:

Es recomendable ejecutar **npm update** periódicamente para mantener el sistema actualizado con las últimas versiones de las dependencias.

#### Respaldo de la Base de Datos:

Realiza respaldos regulares de la base de datos PostgreSQL para evitar la pérdida de datos.

#### Revisión de Logs:

Monitorea los logs de la aplicación para identificar y resolver posibles errores o problemas de rendimiento.

### 11.3 Casos de Prueba Detallados

Los casos de prueba han sido diseñados para cubrir exhaustivamente todas las funcionalidades clave de la aplicación.

A continuación, se proporcionan los detalles específicos de cada caso de prueba, donde se describen los escenarios, los pasos a seguir, y los resultados esperados.

#### 11.3.1 Casos de Prueba de la Pantalla de Bienvenida

##### Visualización de la Pantalla de Bienvenida

**Caso de Prueba:** Verificar que la pantalla de bienvenida se carga correctamente.

**Entradas:** Acceso a la URL de la aplicación sin estar autenticado.

**Resultados Esperados:** La pantalla de bienvenida se carga y muestra un mensaje de bienvenida, las características principales de la aplicación y el botón "Empezar".

**Resultado Obtenido:** La pantalla de bienvenida se carga correctamente y muestra todos los elementos esperados, incluyendo el mensaje de bienvenida, las características de la aplicación y el botón "Empezar". La funcionalidad general de esta pantalla es conforme a lo previsto.

##### Navegación a la Pantalla de Inicio de Sesión

**Caso de Prueba:** Verificar la navegación desde la pantalla de bienvenida a la pantalla de inicio de sesión.

**Entradas:** Clic en el botón "Empezar".

**Resultados Esperados:** El sistema redirige al usuario a la pantalla de inicio de sesión.

**Resultado Obtenido:** Al hacer clic en el botón "Empezar", el sistema redirige correctamente al usuario a la pantalla de inicio de sesión, tal como se esperaba.

#### 11.3.2 Casos de Prueba de la Pantalla de Inicio de Sesión

##### Cargar la Pantalla de Inicio de Sesión

**Caso de Prueba:** Verificar que la pantalla de inicio de sesión se carga correctamente.

**Entradas:** Acceso a la URL de la pantalla de inicio de sesión.

**Resultados Esperados:** Se muestra un formulario de inicio de sesión y opciones para registrarse si el usuario no tiene una cuenta.

**Resultado Obtenido:** La pantalla de inicio de sesión se carga correctamente, presentando el formulario y las opciones de registro de manera clara y accesible, cumpliendo con los requisitos establecidos.

## Registro de Usuario

**Caso de Prueba:** Verificar el proceso de registro de un nuevo usuario.

**Entradas:** Completar y enviar el formulario de registro con nombre de usuario, contraseña, confirmación de contraseña, imagen de perfil, API Key y Token de Trello.

**Resultados Esperados:** El sistema valida la información, crea la cuenta, carga los proyectos de Trello y redirige a la pantalla principal.

**Resultado Obtenido:** El sistema valida la información correctamente y procede a registrar al usuario, cargando los proyectos de Trello y redirigiendo a la pantalla principal. Además, se manejan adecuadamente los casos en que el nombre de usuario ya existe o si las credenciales de Trello son incorrectas, mostrando los errores correspondientes.

## Iniciar Sesión

**Caso de Prueba:** Verificar el inicio de sesión con credenciales correctas.

**Entradas:** Ingreso de nombre de usuario y contraseña correctos.

**Resultados Esperados:** El sistema autentica al usuario, carga su perfil y redirige a la pantalla principal.

**Resultado Obtenido:** El sistema autentica al usuario con éxito, carga la información de su perfil y lo redirige a la pantalla principal, cumpliendo con los resultados esperados.

## Gestión de la Información del Usuario

**Caso de Prueba:** Verificar la edición de la información personal del usuario.

**Entradas:** Modificación de nombre, contraseña e imagen de perfil desde el formulario de edición.

**Resultados Esperados:** El sistema actualiza los datos en la base de datos y refleja los cambios en la interfaz.

**Resultado Obtenido:** La información del usuario se actualiza correctamente en la base de datos. Sin embargo, los cambios no se reflejan inmediatamente en la interfaz de usuario; el usuario debe recargar la página para ver los cambios actualizados. Aunque esto no es ideal, el sistema cumple con la actualización de datos en la base de datos como se esperaba.

## Manejo de Errores en Inicio de Sesión

**Caso de Prueba:** Verificar el manejo de errores al ingresar credenciales incorrectas.

**Entradas:** Ingreso de un nombre de usuario o contraseña incorrectos.

**Resultados Esperados:** El sistema muestra un mensaje de error indicando que las credenciales son incorrectas.

**Resultado Obtenido:** El sistema muestra correctamente el mensaje de error cuando se ingresan credenciales incorrectas, permitiendo al usuario corregir el error y volver a intentar el inicio de sesión, cumpliendo con los resultados esperados.

### 11.3.3 Casos de Prueba de la 1ª Pantalla (Informe de Costos por Proyecto)

#### Cargar la Pantalla de Informe de Costos por Proyecto

**Caso de Prueba:** Verificar que la pantalla de informe de costos se carga correctamente.

**Entradas:** Acceso a la URL de la pantalla de informe de costos.

**Resultados Esperados:** Se muestra la lista de proyectos, con las secciones vacías esperando la selección de un proyecto.

**Resultado Obtenido:** La pantalla se carga correctamente, mostrando la lista de proyectos y dejando las secciones en estado vacío, tal como se esperaba.

#### Seleccionar un Proyecto

**Caso de Prueba:** Verificar la visualización del informe de costos al seleccionar un proyecto.

**Entradas:** Selección de un proyecto de la lista.

**Resultados Esperados:** Se muestran los datos del proyecto seleccionado en las tres secciones de la pantalla.

**Resultado Obtenido:** Al seleccionar un proyecto, el sistema primero presenta una ventana emergente solicitando al usuario que seleccione el período de tiempo para visualizar los datos. Después de que el usuario elige el período, los datos correspondientes del proyecto seleccionado se muestran correctamente en las tres secciones de la pantalla. Este comportamiento es coherente con las expectativas.

#### Deseleccionar un Proyecto

**Caso de Prueba:** Verificar que al deseleccionar un proyecto, la pantalla vuelve al estado inicial.

**Entradas:** Deselección de un proyecto previamente seleccionado.

**Resultados Esperados:** Las secciones de la pantalla vuelven a estar vacías.

**Resultado Obtenido:** La pantalla vuelve a su estado inicial con las secciones vacías después de deseleccionar un proyecto, cumpliendo con los resultados esperados.

#### Seleccionar Múltiples Proyectos

**Caso de Prueba:** Verificar la comparación de costos entre múltiples proyectos.

**Entradas:** Selección consecutiva de dos proyectos.

**Resultados Esperados:** Las secciones superiores de la pantalla muestran la información del último proyecto seleccionado y comparan ambos proyectos en el gráfico.

**Resultado Obtenido:** Al seleccionar múltiples proyectos, las dos primeras secciones muestran la información del último proyecto seleccionado, y la tercera sección compara los datos de ambos proyectos en un gráfico. El sistema funciona como se esperaba.

### **Deseleccionar un Proyecto al Comparar Múltiples Proyectos**

**Caso de Prueba:** Verificar la actualización de la visualización al deseleccionar uno de los proyectos.

**Entradas:** Deselección de uno de los proyectos en comparación.

**Resultados Esperados:** La pantalla elimina la información del proyecto deseleccionado y mantiene la del proyecto restante.

**Resultado Obtenido:** Al deseleccionar uno de los proyectos en comparación, la pantalla elimina correctamente los datos del proyecto deseleccionado y mantiene la información del proyecto que sigue seleccionado, funcionando conforme a lo esperado.

### **Seleccionar Otro Proyecto Después de Deseleccionar**

**Caso de Prueba:** Verificar que el sistema muestra la información correcta cuando se selecciona un nuevo proyecto después de deseleccionar todos los proyectos anteriores.

**Entradas:** Selección de un nuevo proyecto después de haber deseleccionado todos los proyectos previamente seleccionados.

**Resultados Esperados:** Las dos primeras secciones de la pantalla deben actualizarse para mostrar la información del nuevo proyecto seleccionado, mientras que la tercera sección debe agregar la información del nuevo proyecto al gráfico.

**Resultado Obtenido:** Funciona como se esperaba. Las dos primeras secciones se actualizan correctamente con la información del nuevo proyecto, y la tercera sección incorpora los datos en el gráfico de forma adecuada.

### **Navegar a Otra Página**

**Caso de Prueba:** Verificar que la navegación a otra sección de la aplicación funciona correctamente desde la pantalla de informe de costos.

**Entradas:** Clic en una opción de menú o botón para navegar a otra página.

**Resultados Esperados:** El sistema debe redirigir al usuario a la nueva página seleccionada sin ningún problema.

**Resultado Obtenido:** Funciona como se esperaba. El usuario es redirigido a la nueva página seleccionada de manera fluida y sin errores.

## **11.3.4 Casos de Prueba de la 2ª Pantalla (Análisis de Rentabilidad por Cliente)**

### **Cargar la Pantalla de Análisis de Rentabilidad por Cliente**

**Caso de Prueba:** Verificar que la pantalla de análisis de rentabilidad se carga correctamente.

**Entradas:** Acceso a la URL de la pantalla de análisis de rentabilidad.

**Resultados Esperados:** Se muestra la lista de clientes, con las secciones vacías esperando la selección de un cliente.

**Resultado Obtenido:** La pantalla se carga correctamente, mostrando la lista de clientes y dejando las secciones vacías, tal como se esperaba.

### Seleccionar un Cliente

**Caso de Prueba:** Verificar la visualización de la rentabilidad al seleccionar un cliente.

**Entradas:** Selección de un cliente de la lista.

**Resultados Esperados:** Se muestran los datos del cliente seleccionado en las tres secciones de la pantalla.

**Resultado Obtenido:** Antes de mostrar los datos, se presenta al usuario una ventana emergente para seleccionar el período de tiempo deseado. Una vez seleccionado el período, los datos del cliente se muestran correctamente en las tres secciones de la pantalla.

### Deseleccionar un Cliente

**Caso de Prueba:** Verificar que al deseleccionar un cliente, la pantalla vuelve al estado inicial.

**Entradas:** Deselección de un cliente previamente seleccionado.

**Resultados Esperados:** Las tres secciones de la pantalla deben restablecerse a su estado base, sin mostrar información específica.

**Resultado Obtenido:** Al deseleccionar un cliente, la pantalla vuelve correctamente a su estado inicial, sin mostrar información en las secciones.

### Seleccionar Múltiples Clientes

**Caso de Prueba:** Verificar la comparación de rentabilidad entre múltiples clientes.

**Entradas:** Selección consecutiva de dos clientes de la lista.

**Resultados Esperados:** Las secciones superiores de la pantalla deben actualizarse con la información del último cliente seleccionado, mientras que sección inferior debe mostrar la información combinada de ambos clientes, permitiendo la comparación.

**Resultado Obtenido:** La comparación de rentabilidad entre múltiples clientes se realiza correctamente, con las secciones superiores y la sección inferior mostrando la información esperada.

### Deseleccionar un Cliente al Comparar Múltiples Clientes

**Caso de Prueba:** Verificar la actualización de la visualización al deseleccionar uno de los clientes en comparación.

**Entradas:** Deselección de uno de los clientes en comparación.

**Resultados Esperados:** La pantalla debe eliminar la información del cliente deseleccionado y mantener la del cliente restante, actualizando las secciones correspondientes.

**Resultado Obtenido:** Al deseleccionar uno de los clientes, la información se actualiza correctamente, eliminando los datos del cliente deseleccionado y manteniendo los datos del cliente restante.

### Seleccionar Otro Cliente Después de Deseleccionar

**Caso de Prueba:** Verificar que el sistema muestra la información correcta cuando se selecciona un nuevo cliente después de deseleccionar todos los anteriores.

**Entradas:** Selección de un nuevo cliente después de haber deseleccionado todos los clientes previamente seleccionados.

**Resultados Esperados:** Las secciones superiores deben actualizarse con la información del nuevo cliente seleccionado, mientras que la tercera sección debe mostrar la información relevante del cliente seleccionado.

**Resultado Obtenido:** El sistema actualiza correctamente las secciones superiores y la tercera sección con la información del nuevo cliente seleccionado después de deseleccionar todos los anteriores.

### **Navegar a Otra Página**

**Caso de Prueba:** Verificar que la navegación a otra sección de la aplicación funciona correctamente desde la pantalla de análisis de rentabilidad.

**Entradas:** Clic en una opción de menú o botón para navegar a otra página.

**Resultados Esperados:** El sistema debe redirigir al usuario a la nueva página seleccionada sin ningún problema.

**Resultado Obtenido:** La redirección a otra página funciona perfectamente, sin errores ni interrupciones en la navegación.

## **11.3.5 Casos de Prueba de la 3ª Pantalla (Visualización de Cargas de Trabajo)**

### **Cargar la Pantalla de Visualización de Cargas de Trabajo**

**Caso de Prueba:** Verificar que la pantalla de cargas de trabajo se carga correctamente.

**Entradas:** Acceso a la URL de la pantalla de visualización de cargas de trabajo.

**Resultados Esperados:** Se muestra la lista de miembros de la agencia, con los gráficos vacíos esperando la selección de un miembro.

**Resultado Obtenido:** La pantalla se carga correctamente, mostrando la lista de miembros de la agencia y gráficos en estado inicial, tal como se esperaba.

### **Seleccionar un Miembro**

**Caso de Prueba:** Verificar la visualización de la productividad de un miembro seleccionado.

**Entradas:** Selección de un miembro de la lista.

**Resultados Esperados:** Los gráficos se actualizan para mostrar las tareas completadas y la duración para el miembro seleccionado.

**Resultado Obtenido:** Antes de que los gráficos se actualicen, el sistema presenta una ventana emergente solicitando al usuario que seleccione un período de tiempo. Una vez que se ingresa el período, los gráficos se actualizan correctamente con la información del miembro seleccionado.

### **Deseleccionar un Miembro**

**Caso de Prueba:** Verificar que al deseleccionar un miembro previamente seleccionado, los gráficos vuelven al estado inicial.

**Entradas:** Deselección de un miembro previamente seleccionado.

**Resultados Esperados:** Los gráficos se restablecen a su estado base, sin mostrar información específica, y se muestra un mensaje solicitando la selección de un miembro para visualizar su productividad.

**Resultado Obtenido:** Los gráficos se restablecen adecuadamente, volviendo al estado inicial y mostrando el mensaje correspondiente, cumpliendo con lo esperado.

### **Seleccionar un Segundo Miembro**

**Caso de Prueba:** Verificar la comparación de productividad entre dos miembros.

**Entradas:** Selección consecutiva de un segundo miembro mientras ya hay uno seleccionado.

**Resultados Esperados:** Los gráficos se actualizan para mostrar la productividad combinada de ambos miembros, diferenciando sus tareas y duraciones con colores distintos.

**Resultado Obtenido:** La comparación entre miembros se realiza correctamente, con los gráficos actualizándose para reflejar las tareas y duraciones de ambos miembros de manera diferenciada.

### **Deseleccionar un Miembro al Comparar Múltiples Miembros**

**Caso de Prueba:** Verificar la actualización de los gráficos al deseleccionar uno de los miembros en comparación.

**Entradas:** Deselección de uno de los miembros en comparación.

**Resultados Esperados:** El gráfico elimina la información del miembro deseleccionado, manteniendo la del miembro restante.

**Resultado Obtenido:** Al deseleccionar un miembro, los gráficos se actualizan como se esperaba, eliminando la información del miembro deseleccionado y manteniendo los datos del miembro restante.

### **Navegar a Otra Página**

**Caso de Prueba:** Verificar que la navegación a otra sección de la aplicación funciona correctamente desde la pantalla de cargas de trabajo.

**Entradas:** Clic en una opción de menú o botón para navegar a otra página.

**Resultados Esperados:** El sistema redirige al usuario a la nueva página seleccionada sin problemas.

**Resultado Obtenido:** La navegación a otra página se realiza sin inconvenientes, y el usuario es redirigido a la nueva sección tal como se esperaba.

## **11.3.6 Casos de Prueba de la 4ª Pantalla (Análisis de Servicio por Cliente)**

### **Cargar la Página de Análisis de Servicio por Cliente**

**Caso de Prueba:** Verificar que la página de análisis de servicio por cliente se carga correctamente.

**Entradas:** Acceso a la URL de la 4a pantalla de la aplicación.

**Resultados Esperados:** Se muestra la lista de servicios y clientes, con las secciones de gráficos en estado inicial.

**Resultado Obtenido:** La página se carga correctamente, mostrando la lista de servicios y clientes disponibles. Los gráficos se presentan en su estado inicial, con un mensaje que invita al usuario a seleccionar un servicio y un cliente para visualizar su rentabilidad.

### Seleccionar un Servicio

**Caso de Prueba:** Verificar la visualización de los datos al seleccionar un servicio.

**Entradas:** Selección de un servicio de la lista.

**Resultados Esperados:** El servicio seleccionado se resalta, y la interfaz se actualiza para esperar la selección de un cliente.

**Resultado Obtenido:** Al seleccionar un servicio, este se resalta correctamente en la interfaz, y los gráficos muestran un mensaje indicando que el siguiente paso es seleccionar un cliente para continuar con el análisis.

### Seleccionar un Cliente

**Caso de Prueba:** Verificar la visualización de los ingresos generados al seleccionar un cliente después de haber seleccionado un servicio.

**Entradas:** Selección de un cliente de la lista después de haber seleccionado un servicio.

**Resultados Esperados:** El sistema muestra una ventana para la selección de un período de tiempo. Tras seleccionar un período o no seleccionar ninguno, los gráficos se actualizan para mostrar los ingresos generados por el servicio seleccionado para el cliente en el período especificado.

**Resultado Obtenido:** Después de seleccionar un cliente, el sistema solicita el período de tiempo. Una vez seleccionado, los gráficos se actualizan correctamente para reflejar los ingresos generados por el cliente para el servicio seleccionado durante ese período.

### Deseleccionar el Cliente

**Caso de Prueba:** Verificar que al deseleccionar un cliente, los gráficos vuelven al estado inicial.

**Entradas:** Deselección de un cliente previamente seleccionado.

**Resultados Esperados:** Los gráficos eliminan los datos del cliente, mostrando el mensaje "Selecciona un Cliente para Visualizar su Rentabilidad".

**Resultado Obtenido:** Al deseleccionar el cliente, los gráficos vuelven a su estado inicial como se esperaba, eliminando la información del cliente y mostrando nuevamente el mensaje para seleccionar un cliente.

### Deseleccionar el Servicio

**Caso de Prueba:** Verificar que al deseleccionar un servicio, los gráficos vuelven al estado inicial.

**Entradas:** Deselección del servicio seleccionado.

**Resultados Esperados:** Al deseleccionar el servicio, los gráficos se eliminan como se esperaba, y la página muestra el mensaje inicial que solicita la selección de un servicio y un cliente.

**Resultado Obtenido:** Funciona como se esperaba

### **Usuario Ha Deseleccionado un Servicio o un Cliente, Estando Seleccionado el Otro**

**Caso de Prueba:** Verificar la reacción del sistema cuando se deselectiona un servicio o cliente mientras el otro está seleccionado.

**Entradas:** Deselección de un servicio o cliente mientras el otro elemento sigue seleccionado.

**Resultados Esperados:** Los gráficos asociados desaparecen y se muestra el mensaje "Selecciona un Servicio para Visualizar su Rentabilidad" o "Selecciona un Cliente para Visualizar su Rentabilidad".

**Resultado Obtenido:** Al deseleccionar uno de los elementos, los gráficos relacionados desaparecen, y la interfaz actualiza el mensaje según corresponda, indicando que el siguiente paso es seleccionar un servicio o un cliente, cumpliendo con las expectativas.

### **Seleccionar un Segundo Cliente**

**Caso de Prueba:** Verificar la comparación de ingresos generados al seleccionar un segundo cliente.

**Entradas:** Selección de un segundo cliente después de haber seleccionado un primer cliente y un servicio.

**Resultados Esperados:** Los gráficos se actualizan para incluir los ingresos generados por ambos clientes para el servicio seleccionado. Si los ingresos coinciden en el mismo período, se apilan en la misma barra diferenciados por color.

**Resultado Obtenido:** Al seleccionar un segundo cliente, los gráficos se actualizan correctamente para reflejar los ingresos de ambos clientes, mostrando la información combinada y utilizando colores diferentes para distinguir los datos en caso de coincidencias temporales.

### **Deseleccionar el Segundo Cliente**

**Caso de Prueba:** Verificar que al deseleccionar un segundo cliente, los gráficos eliminan su información.

**Entradas:** Deselección del segundo cliente seleccionado.

**Resultados Esperados:** Los gráficos eliminan la información del segundo cliente, manteniendo los datos del primer cliente.

**Resultado Obtenido:** Al deseleccionar el segundo cliente, los gráficos se ajustan correctamente, eliminando la información de dicho cliente mientras mantienen la del primero.

### **Navegar a Otra Página**

**Caso de Prueba:** Verificar que la navegación a otra sección de la aplicación funciona correctamente desde la página de análisis de servicio por cliente.

**Entradas:** Clic en una opción de menú o botón para navegar a otra página.

**Resultados Esperados:** El sistema redirige al usuario a la nueva página seleccionada sin problemas.

**Resultado Obtenido:** Al hacer clic para navegar a otra página, el sistema redirige correctamente al usuario, manteniendo la funcionalidad esperada.

#### **11.3.7 Casos de Prueba de la 5ª Pantalla (Estadísticas de Actividades por Servicio)**

### **Cargar la Página de Estadísticas de Actividades por Servicio**

**Caso de Prueba:** Verificar que la página de estadísticas de actividades se carga correctamente.

**Entradas:** Acceso a la URL de la 5ª pantalla de la aplicación.

**Resultados Esperados:** Se muestran los servicios del cliente para el que se trabaja.

**Resultado Obtenido:** La página se carga correctamente, mostrando los servicios del cliente según lo esperado.

### **Seleccionar un Servicio**

**Caso de Prueba:** Verificar la actualización de las tareas al seleccionar un servicio.

**Entradas:** Selección de un servicio de la lista.

**Resultados Esperados:** Las tareas se filtran según el servicio seleccionado y, si corresponde, el período de tiempo elegido. Se muestran información y gráficos relacionados con la completitud y duración de estas tareas.

**Resultado Obtenido:** El sistema filtra las tareas correctamente de acuerdo con el servicio seleccionado, mostrando la información y gráficos esperados sin ningún problema.

### **Navegar a Otra Página**

**Caso de Prueba:** Verificar que la navegación a otra sección de la aplicación funciona correctamente desde la pantalla de estadísticas de actividades por servicio.

**Entradas:** Clic en una opción de menú o botón para navegar a otra página.

**Resultados Esperados:** El sistema redirige al usuario a la nueva página seleccionada sin problemas.

**Resultado Obtenido:** La navegación a otra página se realiza sin inconvenientes, redirigiendo al usuario correctamente.

### **Usuario Selecciona un Período de Tiempo**

**Caso de Prueba:** Verificar que la selección de un período de tiempo filtra correctamente las tareas mostradas para el servicio seleccionado.

**Entradas:** Selección de un período de tiempo después de haber seleccionado un servicio.

**Resultados Esperados:** El sistema filtra y muestra solo las tareas completadas dentro del período seleccionado para el servicio.

**Resultado Obtenido:** Aunque el usuario debe deseleccionar y volver a seleccionar el servicio para actualizar la información de las tareas en el período seleccionado, el sistema funciona como se esperaba, mostrando correctamente las tareas dentro del período seleccionado.

#### **Usuario No Selecciona un Período de Tiempo**

**Caso de Prueba:** Verificar que, al no seleccionar un período de tiempo, el sistema muestra todas las tareas completadas para el servicio seleccionado sin filtrar.

**Entradas:** Opción de no seleccionar ningún período después de haber seleccionado un servicio.

**Resultados Esperados:** El sistema muestra todas las tareas completadas para el servicio seleccionado, sin aplicar ningún filtrado temporal.

**Resultado Obtenido:** El sistema muestra todas las tareas completadas para el servicio seleccionado sin aplicar ningún filtro de tiempo, cumpliendo con las expectativas.

#### **Guardar la Selección del Período de Tiempo**

**Caso de Prueba:** Verificar que el sistema recuerda la selección del período de tiempo mientras el usuario permanezca en la misma pantalla.

**Entradas:** Selección de un período de tiempo para visualizar las tareas.

**Resultados Esperados:** El sistema mantiene la selección del período para futuras interacciones dentro de la misma sesión de pantalla.

**Resultado Obtenido:** El sistema recuerda la selección del período de tiempo correctamente durante toda la sesión en la pantalla, funcionando perfectamente según lo esperado.

### **11.3.8 Casos de Prueba de la 6ª Pantalla (Seguimiento del Tiempo)**

#### **Cargar la Página de Seguimiento del Tiempo**

**Caso de Prueba:** Verificar que la página de seguimiento del tiempo se carga correctamente.

**Entradas:** Acceso a la URL de la 6ª pantalla de la aplicación.

**Resultados Esperados:** Se muestra la lista de proyectos de la agencia y los mensajes indicando que el usuario debe interactuar.

**Resultado Obtenido:** La página se carga correctamente, mostrando la lista de proyectos y los mensajes de orientación tal como se esperaba.

#### **Seleccionar un Proyecto**

**Caso de Prueba:** Verificar la visualización del tiempo dedicado al seleccionar un proyecto.

**Entradas:** Selección de un proyecto de la lista.

**Resultados Esperados:** La página se actualiza para mostrar el tiempo total dedicado al proyecto y los gráficos correspondientes.

**Resultado Obtenido:** La página responde de manera correcta, mostrando el tiempo total dedicado y actualizando los gráficos según lo previsto.

### **Deseleccionar el Proyecto**

**Caso de Prueba:** Verificar que al deseleccionar un proyecto, la pantalla vuelve al estado inicial.

**Entradas:** Deselección de un proyecto previamente seleccionado.

**Resultados Esperados:** Todas las pantallas vuelven al estado base, eliminando la información del proyecto y mostrando los mensajes de espera de interacción.

**Resultado Obtenido:** La pantalla regresa al estado inicial correctamente, eliminando toda la información del proyecto y mostrando los mensajes de interacción iniciales.

### **Seleccionar un Miembro**

**Caso de Prueba:** Verificar la visualización del tiempo dedicado por un miembro al proyecto seleccionado.

**Entradas:** Selección de un miembro de la lista que han trabajado en el proyecto.

**Resultados Esperados:** La 2a pantalla se actualiza con el tiempo total dedicado por el miembro al proyecto, y la 3a pantalla muestra el gráfico correspondiente al miembro seleccionado.

**Resultado Obtenido:** La información se muestra correctamente, con la 2a y 3a pantallas actualizándose para reflejar con precisión el tiempo dedicado por el miembro seleccionado.

### **Seleccionar un Segundo Miembro**

**Caso de Prueba:** Verificar la comparación del tiempo dedicado por dos miembros al proyecto seleccionado.

**Entradas:** Selección consecutiva de un segundo miembro de la lista.

**Resultados Esperados:** La 2a pantalla se actualiza para mostrar el tiempo total dedicado por el último miembro seleccionado, y la 3a pantalla muestra la comparación de sus tiempos en un gráfico, diferenciados por color.

**Resultado Obtenido:** La comparación entre los dos miembros se realiza de manera efectiva, con ambos tiempos representados correctamente en el gráfico, permitiendo una comparación visual clara.

### **Seleccionar un Segundo Proyecto**

**Caso de Prueba:** Verificar la comparación del tiempo dedicado por los miembros en dos proyectos diferentes.

**Entradas:** Selección de un segundo proyecto de la lista.

**Resultados Esperados:** La 1a pantalla se actualiza con el nombre del segundo proyecto y el tiempo total dedicado por todos los miembros, y la 3a pantalla muestra la comparación de los tiempos entre los dos proyectos.

**Resultado Obtenido:** La página muestra correctamente los datos del segundo proyecto y actualiza los gráficos para permitir una comparación clara entre los tiempos dedicados a los dos proyectos.

#### **Deseleccionar un Miembro**

**Caso de Prueba:** Verificar que al deseleccionar un miembro, su información se elimina de los gráficos.

**Entradas:** Deselección de un miembro previamente seleccionado.

**Resultados Esperados:** La 2a pantalla vuelve a su estado inicial, y la 3a pantalla elimina los datos del miembro deseleccionado del gráfico.

**Resultado Obtenido:** El sistema responde correctamente, eliminando la información del miembro deseleccionado y actualizando los gráficos según lo esperado.

#### **Deseleccionar un Proyecto**

**Caso de Prueba:** Verificar que al deseleccionar un proyecto, su información se elimina de las pantallas correspondientes.

**Entradas:** Deselección del proyecto seleccionado.

**Resultados Esperados:** Las pantallas vuelven a su estado inicial, eliminando la información del proyecto deseleccionado.

**Resultado Obtenido:** La página vuelve al estado inicial correctamente, eliminando todos los datos del proyecto deseleccionado y mostrando los mensajes de orientación.

#### **Navegar a Otra Página**

**Caso de Prueba:** Verificar que la navegación a otra sección de la aplicación funciona correctamente desde la pantalla de seguimiento del tiempo.

**Entradas:** Clic en una opción de menú o botón para navegar a otra página.

**Resultados Esperados:** El sistema redirige al usuario a la nueva página seleccionada sin problemas.

**Resultado Obtenido:** La navegación a otra sección de la aplicación se realiza sin problemas, llevando al usuario a la página seleccionada de manera eficiente.

### **11.3.9 Casos de Prueba de la 7ª Pantalla (Back-Office)**

#### **Cargar la Página de Back-Office**

**Caso de Prueba:** Verificar que la página de Back-Office se carga correctamente.

**Entradas:** Acceso a la URL de la 7a pantalla de la aplicación.

**Resultados Esperados:** Se muestra la lista de proyectos, con uno seleccionado por defecto y la información del proyecto, cliente y miembros mostrada en las respectivas pantallas.

**Resultado Obtenido:** La página se carga correctamente, mostrando todos los elementos según lo esperado.

### Seleccionar un Proyecto

**Caso de Prueba:** Verificar la actualización de la información al seleccionar un proyecto diferente.

**Entradas:** Selección de un proyecto de la lista.

**Resultados Esperados:** La información del proyecto, cliente y miembros se actualiza según el proyecto seleccionado.

**Resultado Obtenido:** La información se actualiza sin problemas, mostrando los datos correctos para el proyecto seleccionado.

### Seleccionar un Miembro

**Caso de Prueba:** Verificar la actualización de la información al seleccionar un miembro específico.

**Entradas:** Selección de un miembro de la lista horizontal de miembros.

**Resultados Esperados:** La pantalla específica del miembro se actualiza para mostrar la información relacionada con el miembro seleccionado.

**Resultado Obtenido:** La información del miembro seleccionado se muestra correctamente, reflejando todos los datos pertinentes.

### Deseleccionar un Proyecto

**Caso de Prueba:** Verificar que al deseleccionar un proyecto, la pantalla vuelve al estado inicial.

**Entradas:** Deselección del proyecto actualmente seleccionado.

**Resultados Esperados:** Las pantallas vuelven a su estado base, eliminando la información del proyecto y mostrando mensajes de espera para la selección de un nuevo proyecto.

**Resultado Obtenido:** La pantalla vuelve efectivamente al estado inicial, eliminando toda la información del proyecto deseleccionado.

### Deseleccionar un Miembro

**Caso de Prueba:** Verificar que al deseleccionar un miembro, la pantalla del miembro vuelve al estado inicial.

**Entradas:** Deselección de un miembro previamente seleccionado.

**Resultados Esperados:** La pantalla del miembro se restablece, eliminando la información del miembro deseleccionado y mostrando un mensaje de espera para la selección de un nuevo miembro.

**Resultado Obtenido:** La pantalla se restablece correctamente, eliminando los datos del miembro deseleccionado y mostrando el mensaje de espera.

### Editar Información del Cliente

**Caso de Prueba:** Verificar que la información del cliente se puede editar y guardar correctamente.

**Entradas:** Selección de la opción para editar la información del cliente, modificación de los datos y guardado de los cambios.

**Resultados Esperados:** El sistema permite la edición, guarda los cambios y actualiza la información del cliente en la base de datos y en la pantalla correspondiente.

**Resultado Obtenido:** La edición y el guardado de la información del cliente funcionan correctamente, con los datos actualizados reflejados en la interfaz.

### **Editar Información del Miembro**

**Caso de Prueba:** Verificar que la información de un miembro se puede editar y guardar correctamente.

**Entradas:** Selección de la opción para editar la información de un miembro, modificación de los datos y guardado de los cambios.

**Resultados Esperados:** El sistema permite la edición, guarda los cambios y actualiza la información del miembro en la base de datos y en la pantalla correspondiente.

**Resultado Obtenido:** Los cambios realizados en la información del miembro se guardan y se reflejan correctamente en la aplicación, funcionando como se esperaba.

### **Registrar Ingresos Generados por un Cliente**

**Caso de Prueba:** Verificar que los ingresos generados por un cliente se pueden registrar y guardar correctamente.

**Entradas:** Selección de la opción para registrar ingresos, entrada de los datos y guardado de la información.

**Resultados Esperados:** El sistema permite la entrada de los datos, los guarda y actualiza la información de ingresos en la base de datos y en la pantalla correspondiente.

**Resultado Obtenido:** El sistema registra y guarda los ingresos generados por un cliente correctamente, mostrando la información actualizada en la interfaz.

### **Navegar a Otra Página**

**Caso de Prueba:** Verificar que la navegación a otra sección de la aplicación funciona correctamente desde la pantalla de Back-Office.

**Entradas:** Clic en una opción de menú o botón para navegar a otra página.

**Resultados Esperados:** El sistema redirige al usuario a la nueva página seleccionada sin problemas.

**Resultado Obtenido:** La navegación entre secciones de la aplicación funciona sin inconvenientes, redirigiendo al usuario a la página seleccionada tal como se esperaba.

## **11.3.10 Casos de Prueba Para Requisitos No Funcionales**

### **Rendimiento**

**Caso de Prueba:** Verificar que la aplicación responde en menos de 2 segundos en el 95% de las interacciones.

**Entradas:** Realizar múltiples interacciones simultáneas en condiciones de carga normal.

**Resultados Esperados:** El tiempo de respuesta es inferior a 2 segundos en el 95% de los casos.

**Resultado Obtenido:** La aplicación responde correctamente dentro del tiempo esperado en la mayoría de los casos. La única excepción se presenta cuando el usuario carga las credenciales de Trello, un proceso que, aunque no se considera una respuesta directa de la interfaz, requiere un tiempo de carga adicional. Este proceso no afecta la capacidad del usuario para interactuar de manera fluida con el resto de la aplicación.

### **Seguridad - Almacenamiento de Contraseñas**

**Caso de Prueba:** Verificar que las contraseñas se almacenan utilizando bcrypt con un costo adecuado.

**Entradas:** Crear un nuevo usuario y revisar cómo se almacena la contraseña en la base de datos.

**Resultados Esperados:** La contraseña está almacenada como un hash bcrypt.

**Resultado Obtenido:** La contraseña del nuevo usuario se almacena correctamente como un hash bcrypt, cumpliendo con el estándar de seguridad esperado. Esto asegura que las contraseñas estén protegidas de manera efectiva contra accesos no autorizados