



TASK MATE

PROYECTO INTEGRADOR



Carrera

Licenciatura en Sistemas de Información

Asignatura

Ingeniería de Software II

Equipo Docente

- Gomez Solis, Laura
- Ferraro, María de los Ángeles

Integrantes

- Gauna Víctor Octavio
- Gimenez Maximiliano Daniel
- Gomez Nasich Fabio Agustín
- Guevara Gonzalez Johan Sebastian



ÍNDICE

1. INTRODUCCIÓN.....	4
1.1 Propósito.....	4
2. CICLO DE VIDA DEL PROYECTO.....	4
2.1 Planificación realizada.....	5
2.2 Historias de usuario.....	6
3. REQUERIMIENTOS FUNCIONALES.....	9
4. REQUERIMIENTOS NO FUNCIONALES.....	10
4.1 Requerimientos no Funcionales Producto.....	10
4.2 Requerimientos no Funcionales Externos.....	10
5. ANÁLISIS DE RIESGOS.....	10
5.1 Identificación de Riesgos.....	11
5.2 Análisis de Riesgos.....	12
5.3 Análisis de Riesgos.....	13
6. EVALUACIÓN DE LA ARQUITECTURA.....	14
7. HERRAMIENTAS UTILIZADAS.....	15
8. CASOS DE USO.....	16
9. CONVERSACIONES.....	17
9.1 Colaborador.....	17
9.2 Usuario.....	18
9.3 Líder de proyecto.....	18
10. DIAGRAMA DE SECUENCIAS.....	19
11. DIAGRAMA ENTIDAD - RELACIÓN.....	21
12. REFERENCIAS.....	22



ÍNDICE DE GRÁFICOS Y TABLAS

1. GRÁFICOS

Gráfico N° 1: Caso de Uso - Líder de Proyecto.....	16
Gráfico N° 2: Caso de Uso - Usuario.....	16
Gráfico N° 3: Caso de Uso - Colaborador.....	17
Gráfico N° 4: Diagrama de Secuencias - Colaborador.....	19
Gráfico N° 5: Diagrama de Secuencias - Líder de proyecto.....	19
Gráfico N° 6: Diagrama de Secuencias - Usuario.....	20
Gráfico N° 7: Diagrama Entidad - Relación.....	21

2. TABLAS

Tabla N° 1: Requerimientos no Funcionales Externos.....	10
Tabla N° 2: Requerimientos no Funcionales Externos.....	10
Tabla N° 3: Identificación de Riesgos.....	11
Tabla N°4: Análisis de Riesgos.....	12
Tabla N°5: Análisis de Riesgos.....	13
Tabla N°6: Colaborador.....	17
Tabla N°7: Usuario.....	18
Tabla N°8: Líder de proyecto.....	18



1. INTRODUCCIÓN

TaskMate busca ser una aplicación de tareas colaborativas entre usuarios. Tiene como objetivo principal permitir que un grupo de personas trabajen juntas en un mismo proyecto o tarea, de manera eficiente y coordinada, a través de una plataforma digital. Esto implica compartir información, asignar tareas, establecer plazos y trabajar en conjunto para alcanzar los objetivos establecidos. De esta manera, se fomenta el trabajo en equipo, se minimizan los errores y se maximiza la productividad. Esta herramienta ha sido diseñada para ser intuitiva y sencilla de utilizar, lo que permite a los usuarios mantenerse al día con sus proyectos y tareas. Además, TaskMate está adaptada a las necesidades individuales de los usuarios, proporcionándoles una herramienta personalizada y efectiva para mejorar su rendimiento en el trabajo y en su vida diaria.

El alcance del proyecto comprende el desarrollo de una aplicación multiplataforma de gestión de proyectos y tareas que permita la creación, organización y asignación de tareas y subtareas con la capacidad de establecer plazos y fechas límite, de igual manera permite colaborar con otros miembros del equipo en tiempo real. TaskMate cuenta con un sistema de seguimiento de tiempo para mantener a los usuarios informados sobre el progreso de sus proyectos y tareas.

1.1 Propósito

El propósito de esta aplicación es proporcionar una herramienta intuitiva y eficiente para la gestión de tareas y proyectos en equipo. La aplicación permitirá a los usuarios crear y asignar tareas a otros miembros del equipo, establecer fechas límite, hacer seguimiento de los avances y enviar notificaciones automáticas para mantener a los miembros del equipo informados sobre el progreso de las tareas. La aplicación además ofrecerá herramientas de comunicación para que los miembros del equipo puedan colaborar y resolver cualquier problema que surja durante la realización de las tareas.



2. CICLO DE VIDA DEL PROYECTO

Después de un detallado análisis, tomamos la decisión de trabajar con la metodología SCRUM. Para ello empezaremos creando una lista ordenada de todas las tareas que se harán en el desarrollo de nuestro producto, a esta lista se la llama *product backlog*, luego esta lista se subdivide en backlogs o *sprint backlogs*. El Sprint Backlog o pila del sprint es el trabajo pendiente del sprint, es decir, la serie de trabajos o productos con los que nuestro equipo se ha comprometido durante el sprint. El sprint es un periodo de tiempo que por lo general es de 2 a 4 semanas de duración.

En nuestro *product backlog* irían todos los requerimientos funcionales descritos tal cual están en este documento y luego en cada *sprint* identificamos esos requerimientos funcionales con historias de usuario para ampliar los detalles.

Entonces nuestro **ciclo de vida del proyecto** sería el siguiente:

1. **Sprint Planning:** El equipo selecciona elementos del Product Backlog para el próximo sprint, transformándolos en un Sprint Backlog.
2. **Desarrollo (Sprint):** Durante cada sprint, el equipo trabaja en los elementos del Sprint Backlog, desarrollando incrementos de software.
3. **Daily Scrum:** Reuniones diarias para sincronizar el trabajo del equipo y planificar las próximas 24 horas.
4. **Revisión del Sprint:** Al final del sprint, se presenta el incremento de software a las partes interesadas para obtener su feedback.
5. **Retrospectiva del Sprint:** El equipo analiza lo que funcionó y lo que no durante el sprint para mejorar en los sprints futuros.
6. **Refinamiento del Backlog:** Se revisa y se ajusta el Product Backlog según el feedback y el progreso del proyecto, lo cual informa la planificación del próximo sprint.

Este ciclo se repite hasta que todas las tareas del ***product backlog*** estén completas, asegurando que el equipo se adapte a los cambios y entregue software de alta calidad que satisfaga las necesidades de los usuarios finales.

2.1 Planificación realizada

En nuestro ***product backlog*** tenemos la lista de los requerimientos funcionales que son los siguientes:

- **RF#01:** El sistema debe tener un formulario para la creación de un nuevo proyecto accesible para el usuario.



- **RF#02:** El sistema debe permitir la creación de nuevas subtareas y su asignación a un proyecto existente, con campos para título, descripción y asignación de colaborador.
- **RF#03:** El sistema debe permitir la visualización de una lista de todos los proyectos existentes.
- **RF#04:** El usuario debe poder seleccionar un proyecto de la lista para ver los detalles del mismo, incluyendo título, descripción, colaboradores, roles de los colaboradores, fecha de inicio, fecha de entrega y estado.
- **RF#5:** El sistema debe permitir agregar colaboradores a un proyecto.
- **RF#6:** Se debe poder ver una lista de todos los colaboradores con sus nombres y direcciones de correo electrónico y su rol en el proyecto.
- **RF#7:** El sistema debe permitir agregar comentarios a cualquier subtask del proyecto seleccionado.
- **RF#8:** El sistema debe permitir eliminar y editar los comentarios que un mismo usuario haya agregado a una subtask.
- **RF#9:** El sistema debe permitir a los usuarios enviar y recibir mensajes de los colaboradores del proyecto y ver un historial completo de la conversación.
- **RF#10:** El sistema debe incluir un sistema de notificaciones que alerte a los usuarios cuando reciban nuevos mensajes o cuando haya mensajes sin leer, para asegurar una comunicación oportuna.

Luego definimos una cantidad de sprints y definimos qué requerimientos funcionales vamos a cumplir en cada uno de ellos. Para los sprints creamos historias de usuario para que el equipo tenga bien en claro qué criterios de aceptación tener para cada requerimiento funcional.

2.2 Historias de usuario

N°1

ID: 001

Título: Crear un nuevo proyecto

Descripción: Como usuario quiero crear un nuevo proyecto en el sistema para poder planificar y llevar a cabo tareas y objetivos específicos.

Valoración: 10 puntos

Programadores a cargo: Fabio, Maxi y Octavio.

Criterios de aceptación:

- El usuario debe tener acceso a un formulario para la creación de un nuevo proyecto.
- El formulario debe incluir los siguientes campos obligatorios:
 - Título: campo de texto para el título del proyecto.



- Descripción: campo de texto para describir el objetivo y alcance del proyecto.
- Fecha de inicio: campo de fecha para establecer la fecha de inicio del proyecto.
- Fecha de fin: campo de fecha para establecer la fecha de finalización del proyecto.
- El formulario debe incluir los siguientes campos opcionales:
 - Responsable: campo de texto para el nombre del responsable del proyecto.
 - Colaboradores: campo de texto para especificar los nombres de los colaboradores en el proyecto.
- Al enviar el formulario, el sistema debe generar automáticamente un identificador único para el proyecto y crear una lista de subtareas vacía para el mismo.
- Si se ingresan campos inválidos, el sistema debe mostrar un mensaje de error al usuario y permitir la corrección de los mismos.

N°2

ID: 002

Título: Agregar subtareas a un proyecto

Descripción: Como usuario quiero agregar subtareas a un proyecto existente.

Valoración: 2 puntos

Programador a cargo: Seba.

Criterios de aceptación:

- El usuario debe poder agregar el título y la descripción de la subtask.
- La subtask debe pertenecer al proyecto seleccionado.
- Se debe poder asignar la subtask a un colaborador que forme parte del proyecto.
- La subtask debe tener un estado "pendiente" por defecto.
- Se debe poder editar todos los campos, estado o valores de la subtask.

N°3

ID: 003

Título: Barra de progreso de tareas.

Descripción: Como usuario quiero poder ver una barra de progreso para tener noción del porcentaje de tareas que ya están completadas en el proyecto.

Valoración: 2 puntos

Programador a cargo: Fabio.

Criterios de aceptación:

- La barra de progreso se actualiza automáticamente a medida que se completan las tareas, reflejando siempre el estado actual del progreso.
- La barra de progreso es configurable, permitiendo al usuario personalizar su apariencia, tamaño o colores según sus preferencias.

N°4

ID: 004

Título: Ver lista de proyectos

Descripción: Como usuario quiero poder ver la lista y los detalles de los proyectos existentes en el sistema de administración de proyectos.



Valoración: 4 puntos

Programadores: Andrea y Pablo

Criterios de aceptación:

- El usuario debe poder ver una lista de todos los proyectos existentes.
- Cada proyecto de la lista debe mostrar su título y su descripción.
- La lista debe estar ordenada alfabéticamente por título.
- El usuario debe poder seleccionar un proyecto de la lista para ver los detalles (título, descripción, colaboradores, roles de los colaboradores, fecha de inicio, fecha de entrega y estado) del mismo.

N°5

ID: 005

Título: Agregar colaboradores a un proyecto

Descripción: Como usuario quiero poder agregar nuevos colaboradores a un proyecto existente en el sistema.

Valoración: 3 puntos

Programadores: Octavio y Seba.

Criterios de aceptación:

- Debe haber un campo para seleccionar el rol que va a tomar el colaborador al ingresar al proyecto.
- El usuario debe poder seleccionar el proyecto al que desea agregar colaboradores.
- El usuario debe poder agregar uno o varios colaboradores al proyecto seleccionado.
- Cada colaborador debe tener un nombre y una dirección de correo electrónico.
- Se debe poder ver una lista de todos los colaboradores con sus nombres y direcciones de correo electrónico y su rol.

N°6

ID: 006

Título: Agregar comentarios a una subtarea de un proyecto

Valoración: 4 puntos

Descripción: Como usuario de la aplicación, deseo tener la capacidad de agregar comentarios a cada subtarea para aclarar cosas al respecto o dar feedback al programador a cargo.

Programadores a cargo: Maxi y Sebastian.

Criterios de aceptación:

- Como usuario, puedo eliminar cualquier comentario que haya agregado a una subtarea.
- Como usuario, puedo editar cualquier comentario que haya agregado a una subtarea.
- Como usuario, puedo ver en tiempo real cualquier cambio que se realice en los comentarios de una subtarea.
- Como usuario, los comentarios están disponibles solo para los usuarios con permisos de acceso adecuados.



N°7

ID: 007

Título: Sistema de mensajería interna del proyecto.

Descripción: Como usuario quiero tener un sistema de mensajería interna en el sistema de administración de proyectos para poder comunicarme con los colaboradores del proyecto y compartir información importante.

Valoración: 9 puntos

Programadores: Fabio, Seba y Octavio

Criterios de aceptación:

- El usuario debe tener acceso a un sistema de mensajería interna dentro del proyecto al que pertenece como colaborador o responsable del mismo.
- El usuario debe poder enviar mensajes a los colaboradores del proyecto y recibir mensajes de los mismos.
- El sistema debe permitir la búsqueda de mensajes por palabra clave.
- El sistema de mensajería debe incluir un sistema de notificaciones que alerte al usuario cuando reciba un nuevo mensaje o cuando haya un mensaje sin leer.
- El sistema debe permitir la edición de los mensajes.
- El sistema debe permitir la visualización de los perfiles de los colaboradores, incluyendo información relevante como el nombre, cargo, correo electrónico y teléfono de contacto.
- El sistema debe permitir la creación de grupos de colaboradores para la Comunicación en equipo.
- El sistema debe garantizar la privacidad y seguridad de los mensajes, evitando que terceros puedan acceder a ellos

3. REQUERIMIENTOS FUNCIONALES

RF#01: El sistema debe tener un formulario para la creación de un nuevo proyecto accesible para el usuario.

RF#02: El sistema debe permitir la creación de nuevas subtareas y su asignación a un proyecto existente, con campos para título, descripción y asignación de colaborador.

RF#03: El sistema debe permitir la visualización de una lista de todos los proyectos existentes.

RF#04: El usuario debe poder seleccionar un proyecto de la lista para ver los detalles del mismo, incluyendo título, descripción, colaboradores, roles de los colaboradores, fecha de inicio, fecha de entrega y estado.

RF#5: El sistema debe permitir agregar colaboradores a un proyecto.

RF#6: Se debe poder ver una lista de todos los colaboradores con sus nombres y direcciones de correo electrónico y su rol en el proyecto.



RF#7: El sistema debe permitir a los usuarios enviar y recibir mensajes de los colaboradores del proyecto y ver un historial completo de la conversación.

RF#8: El sistema debe incluir un sistema de notificaciones que alerte a los usuarios cuando reciban nuevos mensajes o cuando haya mensajes sin leer, para asegurar una comunicación oportuna.

4. REQUERIMIENTOS NO FUNCIONALES

4.1 Requerimientos no Funcionales Producto

Tabla N° 1: Requerimientos no Funcionales Externos

RNF	Descripción del Requerimiento	Clasificación
RNF-01	La aplicación deberá ser compatible con los sistemas operativos iOS y Android	Portabilidad
RNF-02	La aplicación deberá ser fácil de usar y con una interfaz intuitiva	Usabilidad
RNF-03	La aplicación deberá tener un tiempo de respuesta rápido para las acciones del usuario-	Rendimiento
RNF-04	La aplicación deberá tener la capacidad de sincronizar tareas entre múltiples dispositivos	Portabilidad
RNF-05	La aplicación deberá estar disponible en varios idiomas	Usabilidad

Fuente: Elaboración propia (2024).

4.2 Requerimientos no Funcionales Externos

Tabla N° 2: Requerimientos no Funcionales Externos

RNF	Descripción del Requerimiento	Clasificación
RNF-01	La aplicación deberá integrarse con el calendario del sistema operativo	Interoperabilidad
RNF-02	La aplicación deberá cumplir con las regulaciones y leyes locales sobre privacidad de datos	Legislativos

Fuente: Elaboración propia (2023)



5. ANÁLISIS DE RIESGOS

Esta sección tiene como objetivo analizar los riesgos asociados al desarrollo de nuestro proyecto, tomando como punto de partida algunas de las historias de usuario que hemos realizado y otras cuestiones al respecto. En este sentido, se ha llevado a cabo una clasificación e identificación de los riesgos, considerando tres categorías fundamentales: riesgos de negocio, producto y proyecto. Además, se han evaluado las distintas sub clasificaciones de riesgos, que incluyen aspectos de tecnología, requerimientos, herramientas, personal, organizacionales y estimación.

A través de esta metodología de análisis de riesgos, se busca garantizar el éxito del proyecto, minimizando los posibles contratiempos que puedan surgir durante su desarrollo. De esta manera, estamos seguros de que podremos desarrollar nuestro proyecto de manera eficiente y cumplir con los objetivos establecidos, garantizando la satisfacción del cliente.

5.1 Identificación de Riesgos

Tabla N° 3: Identificación de Riesgos

Descripción	Tipo
Si los programadores no están disponibles o tienen dificultades técnicas, puede haber un retraso en la finalización del proyecto.	Proyecto - Estimación
Si la valoración de 3 puntos de una historia de usuario, es inexacta, puede haber un retraso en la finalización del proyecto.	Proyecto - Estimación
Si el paradigma de programación que se eligió para desarrollar el sistema no es escalable o no puede manejar grandes cantidades de datos, puede haber problemas de rendimiento y usabilidad.	Producto - Tecnología
Si los usuarios no pueden agregar subtareas a un proyecto existente, puede haber una pérdida de clientes y reputación del producto a futuro.	Negocio - Organizacionales
Si los criterios de aceptación no están claros o no son adecuados, el producto final no cumplirá con las expectativas del usuario.	Producto - Requerimientos
Si la lista de proyectos no está actualizada, los usuarios pueden tomar decisiones basadas en información desactualizada, lo que podría resultar en pérdidas financieras o de tiempo.	Negocio - Organizacionales
Si la lista de proyectos no está actualizada, los usuarios pueden tomar decisiones basadas en información desactualizada, lo que podría resultar en pérdidas financieras o de tiempo.	Negocio - Organizacionales
Si la implementación de la lista de proyectos requiere una tecnología que los programadores no conocen, puede retrasar la	Producto - Tecnología



implementación o incluso generar problemas en la funcionalidad.	
Si el lenguaje de programación utilizado para el sistema no tiene la capacidad de manejar la simultaneidad entre colaboradores en los proyectos, podría generar problemas técnicos y limitaciones en el sistema.	Producto - Tecnología
Si no se cuenta con políticas claras para la gestión de la información y la privacidad de los mensajes, podrían generarse problemas de confidencialidad y protección de datos.	Negocio - Organizacionales
Si el cliente dice que además que no solo sean mensajes de texto, sino que también se puedan mandar mensajes de audio.	Proyecto - Requerimientos
Si el sistema de mensajería no está diseñado para ser escalable, podría generar problemas técnicos al tener que soportar una cantidad elevada de usuarios y mensajes.	Producto - Tecnología

Fuente: Elaboración propia (2024)

5.2 Análisis de Riesgos

Tabla N°4: Análisis de Riesgos

Descripción	Probabilidad	Nivel de Impacto
Si los programadores no están disponibles o tienen dificultades técnicas, puede haber un retraso en la finalización del proyecto.	Alta	Serio
Si la valoración de 3 puntos de una historia de usuario, es inexacta, puede haber un retraso en la finalización del proyecto.	Moderado	Tolerable
Si el paradigma de programación que se eligió para desarrollar el sistema no es escalable o no puede manejar grandes cantidades de datos, puede haber problemas de rendimiento y usabilidad.	Bajo	Catastrófico
Si los usuarios no pueden agregar subtarear a un proyecto existente, puede haber una pérdida de clientes y reputación del producto a futuro.	Muy bajo	Serio
Si los criterios de aceptación no están claros o no son adecuados, el producto final no cumplirá con las expectativas del usuario	Moderado	Serio
Si la lista de proyectos no está actualizada, los usuarios pueden tomar decisiones basadas en	Bajo	Catastrófico



información desactualizada, lo que podría resultar en pérdidas financieras o de tiempo.		
Si la implementación de la lista de proyectos requiere una tecnología que los programadores no conocen, puede retrasar la implementación o incluso generar problemas en la funcionalidad.	Moderado	Catastrófico
Si el lenguaje de programación utilizado para el sistema no tiene la capacidad de manejar la simultaneidad entre colaboradores en los proyectos, podría generar problemas técnicos y limitaciones en el sistema.	Muy bajo	Catastrófico
Si no se cuenta con políticas claras para la gestión de la información y la privacidad de los mensajes, podrían generarse problemas de confidencialidad y protección de datos.	Bajo	Catastrófico
Si el cliente dice que además que no solo sean mensajes de texto, sino que también se puedan mandar mensajes de audio.	Moderado	Serio
Si el sistema de mensajería no está diseñado para ser escalable, podría generar problemas técnicos al tener que soportar una cantidad elevada de usuarios y mensajes	Bajo	Serio

Fuente: Elaboración propia (2024)

5.3 Análisis de Riesgos

Tabla N°5: Análisis de Riesgos

Riesgo	Gestión de Riesgo
Pérdida de Personal	Tener un equipo de respaldo de programadores, políticas claras de comunicación y colaboración y ofrecer formación continua.
Inexactitud de valoración en historias de usuario.	Establecer un proceso riguroso para la evaluación de historias de usuario que involucre a múltiples partes interesadas.
Paradigma no escalable.	Realizar un análisis detallado del paradigma de programación antes de iniciar el proyecto y establecer procesos de monitoreo y evaluación continua.



Fallos en las funcionalidades del sistema	Hacer testing de cada componente del sistema. Tanto con pruebas unitarias e integradas, también ofreciendo una versión beta para que sea probada por clientes de confianza.
Criterios de aceptación mal especificados	Trabajar en estrecha colaboración con los usuarios para definir y establecer los criterios de aceptación y recibir comentarios regulares
Lenguaje de programación mal seleccionado	Evitar la implementación de tecnologías desconocidas y asegurar que los programadores estén actualizados en las últimas tecnologías y herramientas disponibles.
Políticas de privacidad no claras	Establecer políticas claras de privacidad y seguridad, implementar medidas de seguridad sólidas y comunicarlas claramente a los usuarios.
Cambio de requerimientos	Trabajar con el cliente para comprender sus necesidades y encontrar una solución para sus cambios.
Software no escalable.	Tener un equipo senior de programadores que lideren los equipos de desarrollo y que vayan orientando a los demás a escribir un código limpio y legible

Fuente: Elaboración propia (2024)

6. EVALUACIÓN DE LA ARQUITECTURA

Para el desarrollo de **TaskMate**, es crucial elegir una arquitectura que sea escalable, flexible y permita la integración de nuevas funcionalidades de manera sencilla. En este contexto la arquitectura “**Cliente-Servidor**” es la que mejor se adapta a las necesidades de **TaskMate**.

TaskMate, por su naturaleza, adopta una arquitectura Cliente-Servidor como su principal enfoque. Esta elección estratégica divide las responsabilidades fundamentales en dos áreas críticas, cada una respaldada por una arquitectura específica diseñada para optimizar el rendimiento y la eficiencia del sistema.

Tanto en el cliente como en el servidor, se ha implementado una arquitectura **Por Capas** para gestionar las responsabilidades de manera estructurada. En el cliente, esta estructura permite una gestión eficiente de la aplicación móvil, promoviendo la modularidad y escalabilidad del código. Por su parte, en el servidor, la arquitectura por capas facilita la gestión de la lógica de negocios, el acceso a datos y la presentación de información, mejorando la mantenibilidad y escalabilidad del sistema. Esta división clara de capas promueve la reutilización de código y simplifica la integración con otros sistemas o servicios externos.



7. HERRAMIENTAS UTILIZADAS

Cliente (Aplicación Móvil):

- **React Native:** Plataforma para el desarrollo de aplicaciones móviles nativas multiplataforma, permitiendo la creación de interfaces de usuario ricas y responsivas.

Servidor (Backend):

- **Node.js:** Entorno de ejecución de JavaScript del lado del servidor, ideal para construir aplicaciones escalables y de alta performance.
- **Express.js:** Framework de aplicaciones web para Node.js, utilizado para gestionar rutas y solicitudes HTTP en el servidor.

Base de Datos:

- **PostgreSQL:** Sistema de gestión de bases de datos relacional, elegido por su robustez, flexibilidad y capacidad para manejar grandes volúmenes de datos.

Contenedorización:

- **Docker:** Plataforma para la creación, implementación y ejecución de contenedores, lo que permite encapsular las aplicaciones y sus dependencias de forma consistente y portátil.

Diseño:

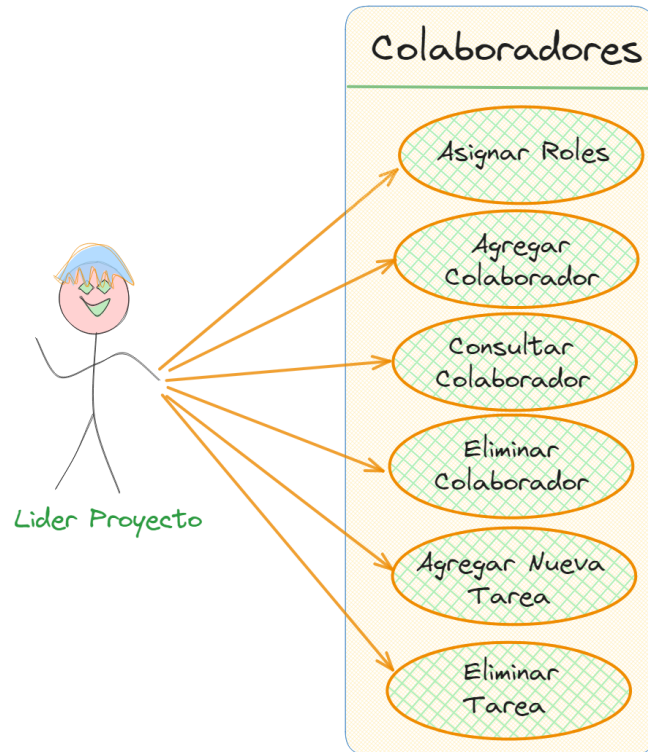
- **Figma:** Herramienta de diseño colaborativo basada en la nube, utilizada para crear y compartir prototipos de interfaces de usuario.
- **Illustrator:** Software de diseño vectorial utilizado para la creación de gráficos, ilustraciones y elementos de interfaz.
- **Photoshop:** Software de edición de imágenes utilizado para el diseño y manipulación de elementos gráficos y visuales.

Despliegue:

- **Google Play Store:** Para publicar y distribuir la aplicación móvil en dispositivos Android.
- **Railway:** Plataforma de alojamiento en la nube para desplegar y gestionar aplicaciones Node.js y bases de datos PostgreSQL de forma sencilla y escalable.

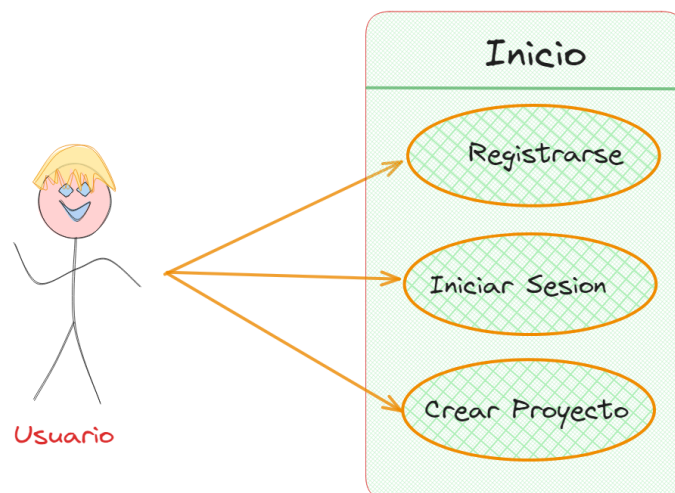
8. CASOS DE USO

Gráfico N° 1: Caso de Uso - Líder de Proyecto



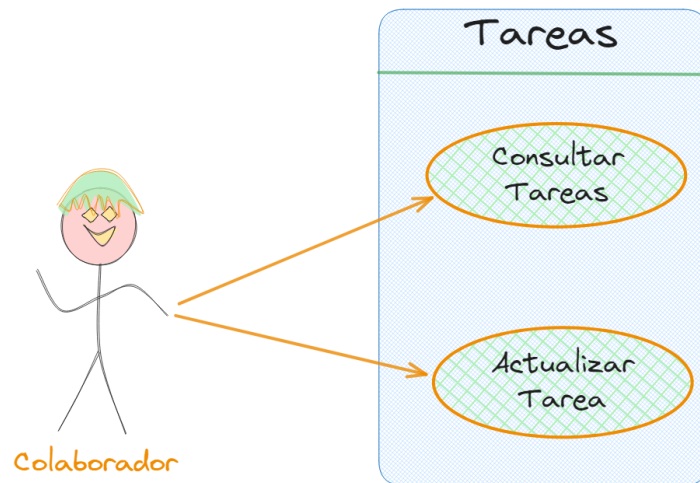
Fuente: Elaboración Propia (2024)

Gráfico N° 2: Caso de Uso - Usuario



Fuente: Elaboración Propia (2024)

Gráfico N° 3: Caso de Uso - Colaborador



Fuente: Elaboración Propia (2024)

9. CONVERSACIONES

9.1 Colaborador

Tabla N°6: Colaborador

Accion	Caso Normal	Caso Alternativo
1. Un colaborador ingresa al menu de tareas personales.		
2. Selecciona una tarea, para ver el estado de esta.	2.1 Se despliega la vista de la tarea. 2.2 Se muestran los datos basicos de esta, como otros colaboradores, fechas de presentacion y el estado de la misma.	
3. El colaborador corrobora los datos.		
4. Fin Caso de uso		

Fuente: Elaboración propia (2024)

9.2 Usuario



Tabla N°7: Usuario

Accion	Caso Normal	Caso Alternativo
1 Usuario crea Proyecto		
2. Se completan los datos del formulario, indicando nombre, descripcion, recursos y fechas.	2.1 Se verifican correctamente los datos. 2.2 Se crea el nuevo proyecto, con el usuario como Lider del Proyecto. 2.3 Se muestra el proyecto recién creado, con todos los datos y las opciones posteriores	
3. Fin del Caso de Uso		

Fuente: Elaboración propia (2024)

9.3 Líder de proyecto

Tabla N°8: Líder de proyecto

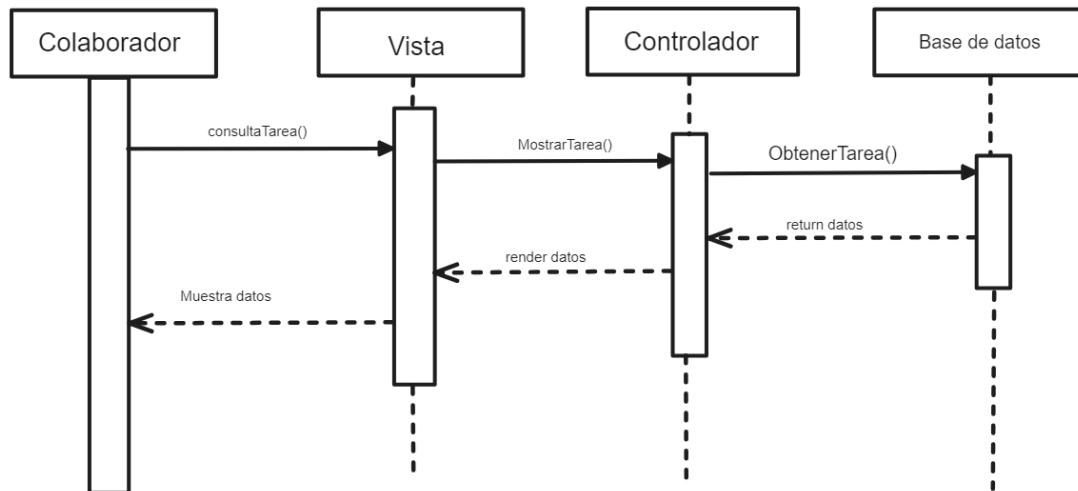
Accion	Caso Normal	Caso Alternativo
1 Lider del Proyecto agrega Colaborador		
2. Se busca un usuario por su correo electronico	2.1 Se encuentra el usuario especificado. 2.2 Se envia al usuario una notificacion de la invitacion al proyecto.	
3. Fin del Caso de Uso		

Fuente: Elaboración propia (2024)



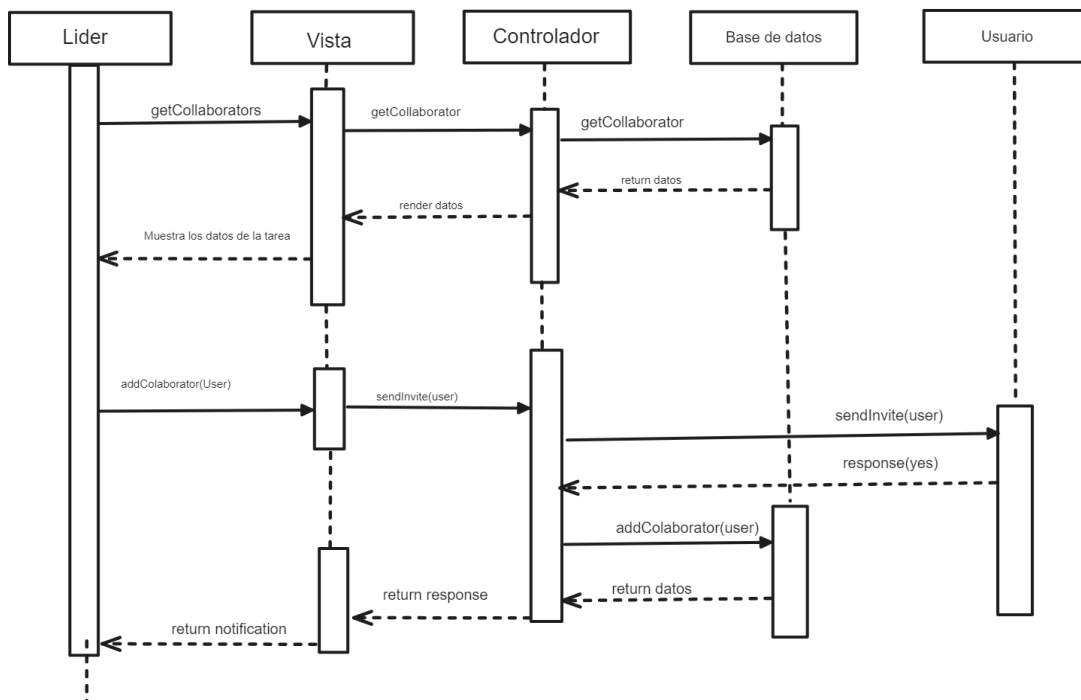
10. DIAGRAMA DE SECUENCIAS

Gráfico N° 4: Diagrama de Secuencias - Colaborador



Fuente: Elaboración propia (2024)

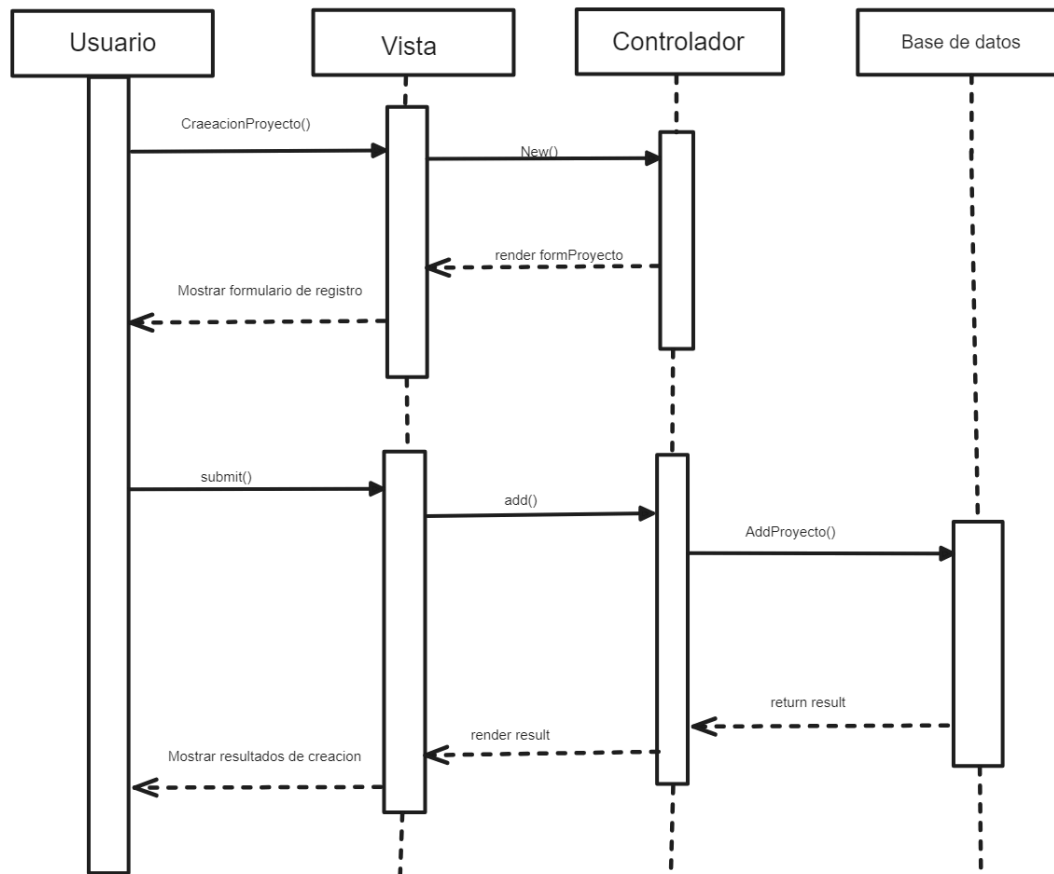
Gráfico N° 5: Diagrama de Secuencias - Líder de proyecto



Fuente: Elaboración propia (2024)



Gráfico N° 6: Diagrama de Secuencias - Usuario

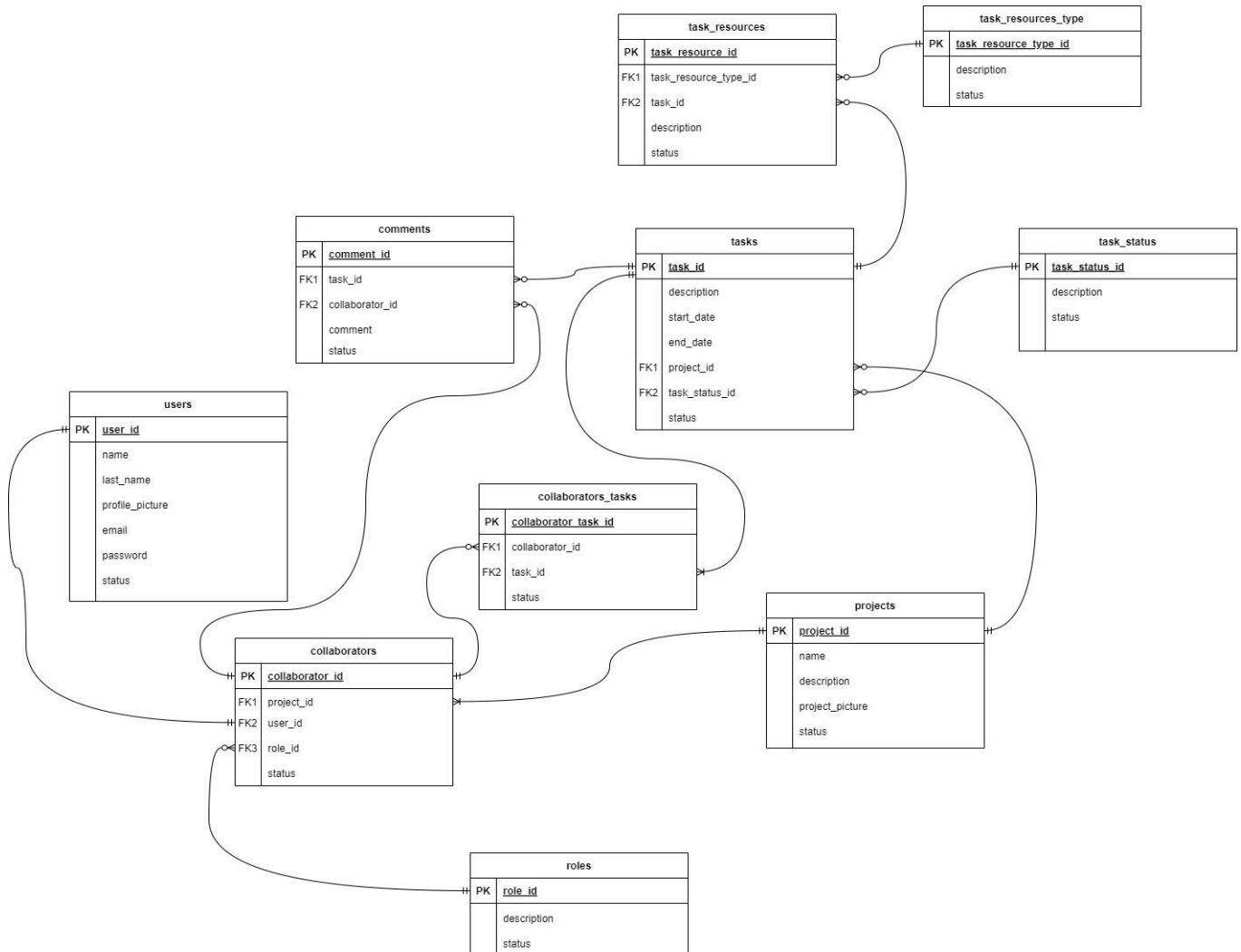


Fuente: Elaboración propia (2024)



11. DIAGRAMA ENTIDAD - RELACIÓN

Gráfico N° 7: Diagrama Entidad - Relación



Fuente: Elaboración propia (2024)



12. REFERENCIAS

- Cátedra Ingeniería de Software II. (2024). Guía de estudio para la materia de Ingeniería de Software II. Universidad Nacional del Nordeste
- Cátedra Ingeniería de Software I. (2023). Guía de estudio para la materia de Ingeniería de Software I. Universidad Nacional del Nordeste
- Intelequia. (2021).Intelequia. Recuperado de [Ciclo de vida del software](#)
- Pirani. (2022). Pirani. Recuperado de [Gestión de riesgos en proyectos de software](#)