

**UNIVERSIDAD CONTINENTAL**  
**FACULTAD DE INGENIERÍA**  
**ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA**  
**DE SISTEMAS E INFORMÁTICA**



**PROYECTO**

**“Plataforma Web de Gestión de Proyectos Colaborativos con Integración de Inteligencia Artificial, Dirigida a una Clínica”**

**PRESENTADO POR:**

APELLIDOS Y NOMBRES	CÓDIGO
Damian Espinoza Fran Sthip	61311942
Pastrana Escobar Emil Brent	73655486
Vladimir Jorge Bendezú Peña	46748916
Llantoy Balbin Maria Milagros	77428439
Condori Huarcaya Gian Piero	74022578

**ASESOR:**

**Daniel Gamarra Moreno**

**HUANCAYO – PERÚ**

# 2023

## LISTA DE CONTENIDO

PORADA	1
LISTA DE CONTENIDO	2
LISTA DE TABLAS	7
LISTA DE FIGURAS	8
CAPÍTULO 1	9
PLANTEAMIENTO DEL ESTUDIO	9
1.1. Aspectos Generales de la Empresa	9
1.1.1. Organigrama	9
1.1.2. Misión y visión	9
1.2. Diagnóstico del Problema	9
1.3. Procesos de la Empresa	9
1.4. Oportunidad Encontrada	9
1.5. Detalles del Proyecto	9
CAPÍTULO 2	10
ESTUDIO DE FACTIBILIDAD	10
2.1. Alternativas de Solución	10
2.2. Factibilidad Técnica	10
2.2.1....	10
2.2.2....	10
2.3. Factibilidad Económica	10
2.3.1. Gastos generales	10
2.4. Factibilidad Operacional	10
2.4.1. Sistemas de ventas	10
CAPÍTULO 3	11
ANÁLISIS DE REQUERIMIENTOS	11
3.1. Metas del Sistema de Información	11
3.2. Requisitos del Sistema	11
3.2.1. Requerimientos funcionales	11
3.2.2. Requerimientos no funcionales	11
3.3. Identificación de Actores del Sistema	11
3.3.1....	11
CAPÍTULO 4	12
PLANIFICACIÓN DEL PROYECTO	12
4.1. Definición de Roles de Trabajo	12
4.1.1. Product owner	12
4.1.2. Scrum master	12
4.1.3. Team member	12
4.1.4. Tester	12
4.2. Product Backlog	12
4.3. Sprint Backlog	12
4.3.1. Sprint 1	12
4.3.2. Sprint 2	12
4.3.3. Sprint 3	12

4.3.4. Sprint 4	12
4.3.5. Sprint 5	13
4.4. Planificación de Sprints	13
4.4.1. Historias de usuario	13
4.4.2. Priorización de historias de usuario	13
4.5. Cronograma de Actividades	13
4.6. Gestión de Riesgos	13
<b>CAPÍTULO 5</b>	<b>14</b>
<b>DISEÑO DEL SISTEMA DE INFORMACIÓN</b>	<b>14</b>
5.1. Diseño de Diagramas UML	14
5.1.1. Diagramas de casos de uso	14
5.1.2. Diagramas de secuencia	14
5.1.3. Diagramas de colaboración	14
5.1.4. Diagramas de clases	14
5.2. Diseño de Base de Datos	14
5.2.1. Diseño conceptual (E/R)	14
5.2.2. Diseño lógico	14
5.2.3. Diseño físico	14
5.2.4. Modelado de base de datos	14
5.3. Diseño de Interfaces Básicas	14
5.3.1. Acceso login	15
5.3.2. Interfaz ...	15
<b>CAPÍTULO 6</b>	<b>16</b>
<b>CODIFICACIÓN DEL SOFTWARE</b>	<b>16</b>
6.1. Desarrollo del Sprint 1	16
6.1.1. Sprint planning	16
6.1.2. Sprint backlog	16
6.1.3. Historias de usuarios	16
6.1.4. Taskboard	16
6.1.5. Daily scrum	16
6.1.6. Sprint review	16
6.1.7. Criterios de aceptación	16
6.1.8. Resultados del sprint	16
6.1.8.1. Evidencias.	16
6.1.8.2. Prueba de desarrollo.	16
6.1.8.3. ....	17
6.1.9. Sprint retrospective	17
6.2. Desarrollo del Sprint 2	17
6.2.1. Sprint planning	17
6.2.2. Sprint backlog	17
6.2.3. Historias de usuarios	17
6.2.4. Taskboard	17
6.2.5. Daily scrum	17
6.2.6. Sprint review	17
6.2.7. Criterios de aceptación	17

6.2.8. Resultados del sprint	17
6.2.8.1. Evidencias.	17
6.2.8.2. Prueba de desarrollo.	18
6.2.8.3. ....	18
6.2.9. Sprint retrospective	18
6.3. Desarrollo del Sprint 3	18
6.3.1. Sprint planning	18
6.3.2. Sprint backlog	18
6.3.3. Historias de usuarios	18
6.3.4. Taskboard	18
6.3.5. Daily scrum	18
6.3.6. Sprint review	18
6.3.7. Criterios de aceptación	18
6.3.8. Resultados del sprint	18
6.3.8.1. Evidencias.	19
6.3.8.2. Prueba de desarrollo.	19
6.3.8.3. ....	19
6.3.9. Sprint retrospective	19
6.4. Desarrollo del Sprint 4	19
6.4.1. Sprint planning	19
6.4.2. Sprint backlog	19
6.4.3. Historias de usuarios	19
6.4.4. Taskboard	19
6.4.5. Daily scrum	19
6.4.6. Sprint review	19
6.4.7. Criterios de aceptación	19
6.4.8. Resultados del sprint	20
6.4.8.1. Evidencias.	20
6.4.8.2. Prueba de desarrollo.	20
6.4.8.3. ....	20
6.4.9. Sprint retrospective	20
CAPÍTULO 7	21
PRUEBAS DE SOFTWARE	21
7.1. Plan de Pruebas	21
CONCLUSIONES	22
RECOMENDACIONES	23
ANEXOS	24
Anexo 01. Manual Técnico	25
Anexo 02. Manual de Usuario	26

## **LISTA DE TABLAS**

<b>Gastos Generales .....</b>	<b>12</b>
<b>Definición de Roles de Trabajo.....</b>	<b>18</b>
<b>Historias de Usuario .....</b>	<b>20</b>
<b>Priorización de Historias de Usuario .....</b>	<b>21</b>
<b>Gestión de Riesgos .....</b>	<b>23</b>
<b>Categorías de Riesgos .....</b>	<b>24</b>
<b>Heurísticas de Nielsen .....</b>	<b>41</b>
<b>SUS .....</b>	<b>42</b>

## LISTA DE FIGURAS

<b>Figura 1: Organigrama .....</b>	<b>7</b>
<b>Figura 2: Cronograma de Actividades.....</b>	<b>24</b>
<b>Figura 3: Diagramas de Caso de Uso.....</b>	<b>27</b>
<b>Figura 4: Diagrama de Secuencia.....</b>	<b>28</b>
<b>Figura 5: Diagrama de Colaboración.....</b>	<b>29</b>
<b>Figura 6: Diagrama de Clases.....</b>	<b>30</b>
<b>Figura 7: Diseño de Base de Datos.....</b>	<b>31</b>
<b>Figura 8: Modelado de Base de Datos.....</b>	<b>37</b>
<b>Figura 9: Diseño de las Interfaces.....</b>	<b>38</b>
<b>Figura 10: Resultados del sprint 1.....</b>	<b>46</b>
<b>Figura 11: Resultados del sprint 2.....</b>	<b>52</b>
<b>Figura 10: Resultados del sprint 3.....</b>	<b>56</b>
<b>Figura 12: Anexos.....</b>	<b>46</b>

# CAPÍTULO 1

## PLANTEAMIENTO DEL ESTUDIO

### 1.1. Aspectos Generales de la Empresa

#### 1.1.1. Organigramas

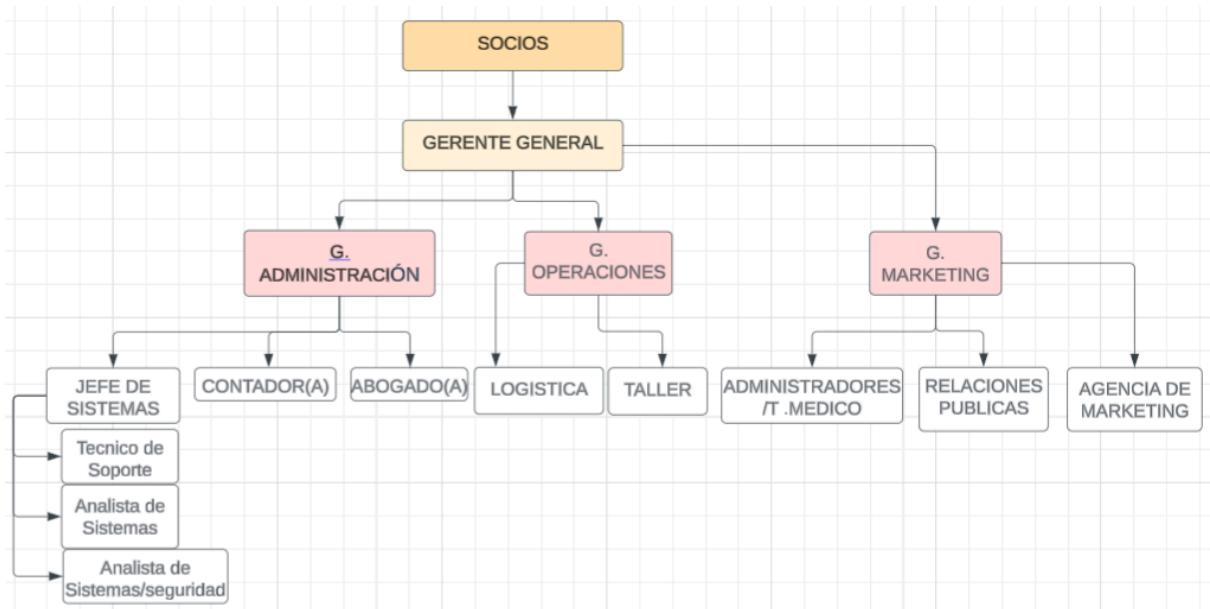


Figura 1

#### 1.1.2. Misión y visión

**Misión:** Brindar atención médica integral y especializada, combinando excelencia profesional con innovación tecnológica, para garantizar servicios de salud oportunos, eficientes y centrados en el paciente en Huancayo. Nos comprometemos a optimizar continuamente nuestros procesos internos para una gestión efectiva y coordinada.

**Visión:** Ser una clínica referente en la región centro del país por su capacidad de integrar tecnología, eficiencia organizacional y calidad médica, superando desafíos operativos mediante soluciones innovadoras que aseguren el bienestar de nuestros pacientes y la excelencia en cada servicio prestado.

### 1.2. Diagnóstico del Problema

Actualmente la clínica enfrenta problemas críticos de gestión de proyectos: uso de hojas de cálculo y comunicaciones dispersas que provocan pérdidas de información,

retrasos y falta de visibilidad del avance. Estudios señalan que las prácticas ineficientes de gestión conlleva demoras en las entregas y sobrecostos. La falta de automatización en asignación de tareas y seguimiento hace difícil coordinar el trabajo de múltiples desarrolladores, perjudicando la productividad.

### **1.3. Procesos de la Empresa**

En el estado actual, el flujo de trabajo de la clínica incluye reuniones semanales de seguimiento, reportes por correo electrónico y hojas de cálculo compartidas para asignar tareas. No existe una herramienta única que integre la comunicación, la asignación de tareas y la generación de reportes. Este esquema genera cuellos de botella: por ejemplo, las actualizaciones de estado dependen de informes manuales y carecen de trazabilidad. Además, los cambios frecuentes en requerimientos no quedan registrados de forma ágil.

### **1.4. Oportunidad Encontrada**

La oportunidad radica en desarrollar una aplicación web colaborativa que centralice la gestión de proyectos y mejore la comunicación. Al integrar inteligencia artificial, se pueden automatizar tareas rutinarias (como clasificación de urgencia o sugerencias de priorización) y ofrecer asistentes inteligentes. La IA puede ayudar a analizar patrones en los datos de proyectos y generar predicciones útiles.

Esto no solo ahorrará tiempo al equipo, sino que también alinearía a la empresa con tendencias actuales de gestión de proyectos, donde estudios destacan que la IA mejora la toma de decisiones en planificación y asignación de recursos.

### **1.5. Detalles del Proyecto**

Se implementó una aplicación web usando la pila tecnológica MERN (MongoDB, Express.js, React.js, Node.js).

Esta elección permite desarrollar rápidamente una solución full-stack en JavaScript. El sistema permitirá crear y administrar proyectos, tareas y miembros de equipo, todo con una interfaz web interactiva. El control de versiones del código se gestionará en un repositorio GitHub, aplicando flujo de trabajo GitHub Flow con ramas feature y pull requests.

Para el despliegue y pruebas se garantizara que la aplicación se ejecute de manera consistente en desarrollo, prueba y producción.

Además, se incorporarán al menos dos funcionalidades de IA: un chatbot integrado y gestión de tareas con IA. Se planifica el desarrollo ágil del proyecto, dividiendo el trabajo en sprints mensuales bajo la metodología Scrum con roles claramente asignados.

## CAPÍTULO 2

### ESTUDIO DE FACTIBILIDAD

#### **2.1. Alternativas de Solución**

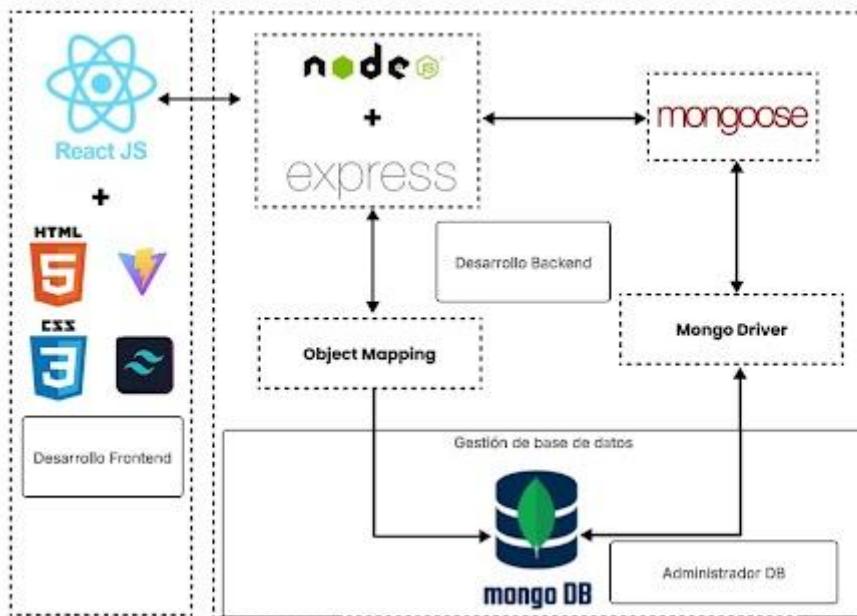
**2.1.1. Desarrollo a medida (Stack MERN):** Crear la plataforma desde cero usando MongoDB, Express, React y Node. Esto permite total personalización y uso de tecnologías gratuitas de código abierto. Como ventaja, el equipo puede controlar todas las funcionalidades según las necesidades específicas. Como desventaja, requiere mayor tiempo inicial de desarrollo y recursos de programación.

**2.1.2. Plataforma SaaS existente:** Utilizaremos Jira como herramienta de gestión de proyectos. Esta plataforma ofrece una implementación rápida y un soporte confiable, lo que facilita el monitoreo del progreso de los proyectos. Además, Jira permite la gestión de tareas, la asignación de responsabilidades y la visualización del estado de los proyectos en tiempo real, lo que contribuye a una mejor colaboración entre los miembros del equipo.

#### **2.2. Factibilidad Técnica**

La factibilidad técnica es alta gracias a que se utilizan tecnologías maduras y probadas. MongoDB es una base de datos orientada a documentos, multiplataforma y “source-available”. Express.js es un framework web rápido y minimalista para Node.js, y React es la biblioteca JavaScript oficial de Meta para construir interfaces de usuario. Node.js, por su parte, es un entorno de ejecución JavaScript de código abierto y multiplataforma que permite crear servidores y aplicaciones web. Esta combinación (MERN) facilita la construcción de APIs RESTful y aplicaciones escalables. Además, la arquitectura cliente-servidor propuesta se basa en JSON para la comunicación, algo natural en este stack. El uso de Docker mejora la portabilidad: al empaquetar cada componente en un contenedor, se asegura que la app corra igual en distintos entornos.

Figura 1



### 2.2.1. Requerimientos de hardware y software

- Hardware:
  - Computadoras personales/laptops para el equipo de desarrollo.
  - Servidor local para despliegue de pruebas.
  - Acceso a internet estable.
- Software:
  - Stack MERN:
    - Node.js y Express.js (para el servidor backend)
    - React.js (frontend)
    - MongoDB (base de datos NoSQL)
  - Docker y Docker Compose para contenerización
  - GitHub para control de versiones y colaboración

- Visual Studio Code como editor de código
- Postman para pruebas de API
- Librerías de IA: aprovechando servicios tipo LLM (Large Language Models) disponibles vía web y eso mediante una API REST
- Sistema operativo compatible: Windows

### **2.2.2. Capacidades del equipo de trabajo**

- Conocimientos:
  - El equipo tiene experiencia en desarrollo web en JavaScript.
  - Manejo intermedio-avanzado de herramientas como React, Node y MongoDB.
  - Familiarización con metodologías ágiles (Scrum).
  - Conocimientos básicos/intermedios en inteligencia artificial (procesamiento de lenguaje natural, clasificación, regresión).
  - Uso de Docker para pruebas y despliegue.
  - Pruebas unitarias (Jest) y E2E (Cypress).
- Recursos humanos:
  - Scrum Master, Product Owner, desarrolladores front-end y back-end con roles definidos.
  - Tiempo disponible suficiente para cumplir con los tres sprints planificados

### **2.3. Factibilidad Económica**

La factibilidad económica analiza la viabilidad del proyecto en términos de costos, recursos financieros y beneficios esperados, se determinó que su implementación es altamente factible desde el punto de vista económico.

### **2.3.1. Gastos generales**

Categoría	Costo Estimado (S/.)	Descripción resumida
Honorarios de programadores	S/ 6,000.00	Asumiendo una tarifa mensual estimada de S/ 2,000 por programador junior a tiempo parcial.
Servicios en la nube / hosting	S/ 150.00	Hosting para pruebas, base de datos remota, certificados SSL.
Internet	S/ 240.00	Conectividad de cada integrante (S/ 80 mensual x 3 personas).
Energía eléctrica compartida	S/ 180.00	Estimación de consumo mensual asociado al trabajo remoto.
Materiales de presentación	S/ 60.00	Impresiones, USBs, presentación final.
Otros gastos operativos	S/ 100.00	Copias, respaldo en la nube, imprevistos menores.
<b>Total estimado del proyecto</b>	<b>S/ 6,730.00</b>	Presupuesto basado en costos reales de un equipo pequeño sin gastos de oficina.

Tabla 1. Gastos generales

### **2.4. Factibilidad Operacional**

Evaluar si el sistema propuesto podrá ser utilizado de forma efectiva y sostenida por los usuarios finales, considerando su facilidad de uso, adaptabilidad a las necesidades del entorno y alineación con los procesos de trabajo existentes, se determinó que la solución es operacionalmente viable, como se expone a continuación

#### **2.4.1. Viabilidad de uso por los usuarios finales**

- **Accesibilidad y facilidad de uso:** La plataforma se accede desde un navegador web moderno, sin necesidad de instalaciones adicionales. Su diseño responsivo permite el acceso desde laptops, tablets o smartphones..

- **Curva de aprendizaje baja:** Los usuarios no requieren conocimientos técnicos especializados para utilizar el sistema. La navegación es clara y las funciones están agrupadas de manera lógica. Se incluyen etiquetas y mensajes de ayuda contextual en formularios y paneles.
- **Flujo de trabajo integrado:** Las funcionalidades están alineadas con el flujo natural de trabajo de equipos colaborativos. Los usuarios pueden crear y gestionar proyectos, asignar tareas, recibir notificaciones, comunicarse mediante chat y obtener recomendaciones mediante el asistente virtual, todo desde un solo entorno.
- **Soporte a la toma de decisiones:** Gracias a la integración de IA, el sistema sugiere prioridades, estima tiempos de tareas y ayuda a los usuarios mediante el asistente virtual, reduciendo carga cognitiva y mejorando la productividad.
- **Manual de usuario y soporte:** Se entrega manual de usuario y guía rápida que explica paso a paso el uso del sistema. Además, el asistente virtual está disponible como canal de soporte básico en tiempo real.
- **Escalabilidad y continuidad operativa:** La solución está desarrollada sobre tecnologías modernas stack MERN y Docker, permiten su fácil despliegue en servidores propios o en la nube.
- **Integración con la cultura organizacional:** El enfoque colaborativo, la comunicación interna y el control de tareas se ajustan a las prácticas comunes en equipos de desarrollo, investigación u organizaciones académicas.

## CAPÍTULO 3

### ANÁLISIS DE REQUERIMIENTOS

#### **3.1 Metas del Sistema de Información**

El sistema tiene como meta principal proporcionar una plataforma digital integral para gestionar proyectos colaborativos de forma eficiente, facilitando la asignación de tareas, la comunicación entre miembros del equipo y la automatización de procesos clave mediante la integración de Inteligencia Artificial (IA)

#### **3.2. Requisitos del Sistema**

##### **3.2.1 Requerimientos funcionales**

Los requerimientos funcionales corresponden a las funcionalidades que el sistema debe ofrecer a los usuarios:

##### **3.2.2 Gestión de Proyectos**

- RF01: Crear, editar y eliminar proyectos.
- RF02: Gestionar tareas: creación, edición, asignación y seguimiento.
- RF03: Visualizar tareas y avances en tableros tipo Kanban o listas.
- RF04: Ver estadísticas visuales sobre el progreso del proyecto (gráficos, porcentajes, tiempos).

##### **3.2.3. Gestión de Usuarios y Permisos**

- RF05: Registrar y autenticar usuarios.
- RF06: Asignar roles con diferentes niveles de acceso.
- RF07: Controlar permisos para acceso a tareas, proyectos y herramientas de IA.

##### **3.2.4. Comunicación y Colaboración**

- RF08: Chat integrado para cada proyecto o equipo.
- RF09: Enviar y recibir notificaciones en tiempo real.
- RF10: Mencionar miembros del equipo en comentarios o tareas.

### **3.2.5. Integración de Inteligencia Artificial**

- RF11: Gestión de tareas mediante IA: sugerencias automáticas de asignación y prioridades.
- RF12: Clasificación automática de tareas según su tipo (bug, mejora, administrativa, etc.)

## **3.3 Requerimientos no funcionales**

Estos requisitos describen atributos de calidad que debe cumplir el sistema:

- **RNF01:** El sistema debe tener tiempos de respuesta inferiores a 2 segundos en la mayoría de las operaciones.
- **RNF02:** La interfaz debe ser responsive y funcional en dispositivos móviles, tablets y PC.
- **RNF03:** El sistema debe asegurar los datos de los usuarios mediante cifrado de contraseñas y tokens JWT.
- **RNF04:** La arquitectura debe permitir escalar horizontalmente en caso de aumento de usuarios.
- **RNF05:** El sistema debe contar con pruebas unitarias y de integración automatizadas para validar su funcionamiento.
- **RNF06:** El sistema debe permitir auditoría básica mediante logs de operaciones (crear, modificar, eliminar).

### **•.1. Identificación de Actores del Sistema**

#### **•.1.1.Administrador del sistema (Product Owner)**

- Usuario con acceso total al sistema, responsable de la creación y gestión de proyectos, asignación de tareas, monitoreo general del desempeño de la plataforma.
- Funciones clave: crear proyectos, gestionar usuarios, generar reportes.

### **3.3.2. Usuario colaborador (Miembro del equipo de proyecto)**

- Usuario que participa activamente en la ejecución de tareas asignadas. Puede visualizar proyectos a los que está asignado, gestionar sus tareas, interactuar por chat y recibir notificaciones.
- Funciones clave: ver tareas asignadas, cambiar estado de tareas, consultar al asistente virtual, comunicarse con otros miembros.

### **3.3.3. Asistente virtual (Componente IA)**

- Agente inteligente integrado en el sistema que interpreta preguntas en lenguaje natural, sugiere tiempos de ejecución, clasifica tareas y proporciona asistencia básica.
- Funciones clave: responder preguntas, predecir duración de tareas, clasificar tareas por prioridad, ayudar con comandos comunes del sistema.

## CAPÍTULO 4

### PLANIFICACIÓN DEL PROYECTO

#### 4.1. Definición de Roles de Trabajo

Rol	Nombre(s)
<b>Product Owner</b>	Llantoy Balbin Maria Milagros
<b>Scrum Master</b>	Pastrana Escobar Emil Brent
<b>Team Members</b>	Condori Huarcaya Gian Piero Vladimir Jorge Bendezú Peña
<b>Tester (QA)</b>	Damian Espinoza Fran Sthip

##### 4.1.1. Product owner :

- Representa al cliente final y vela por el valor del producto.
- Define y prioriza los elementos del **Product Backlog**.
- Asegura que cada funcionalidad entrega valor al usuario.
- Participa en revisiones de sprint y toma decisiones sobre el producto.

##### 4.1.2. Scrum master

- Facilita la metodología Scrum.
- Coordina y guía las ceremonias: Sprint Planning, Daily Scrum, Sprint Review y Retrospective.
- Elimina obstáculos para que el equipo avance sin impedimentos.
- Promueve la mejora continua y buenas prácticas ágiles.

##### 4.1.3. Team member

- Conformado por desarrolladores frontend, backend.
- Desarrolla funcionalidades según prioridades.
- Ejecuta pruebas unitarias y de integración.

- Participa en la estimación de historias y entrega de incrementos funcionales.

#### 4.1.4. Tester

- Diseña y ejecuta pruebas funcionales, unitarias y de integración.
- Verifica los criterios de aceptación.
- Documenta errores encontrados y colabora con el equipo para solucionarlos.

### 4.1. Product Backlog

Nº	Código	Descripción	Prioridad	Detalle	Complejidad
1	HU01	Módulo de creación y edición de proyectos	Alta (1)	Permite al usuario crear, editar y guardar proyectos.	25
2	HU02	Tablero Kanban con movimiento de épicas	Alta (1)	Permite organizar visualmente las tareas arrastrándolas entre columnas.	25
3	HU03	Visualización de prioridades	Media (2)	Muestra colores visuales según la prioridad (rojo, amarillo, verde).	20
4	HU04	Asignación de épicas a miembros	Alta (1)	Permite al Project Manager asignar tareas y ver responsables.	25
5	HU05	Generación de historias de usuario	Media (2)	Permite generar historias de usuario manualmente.	20
6	HU06	Chat interno	Media (2)	Comunicación en tiempo real dentro del proyecto.	25
7	HU07	Generación automática de historias de usuario	Alta (1)	Genera historias automáticamente según la épica y el proyecto.	25

8	HU08	Traducción automática del chat	Media (2)	Traduce los mensajes del chat al idioma nativo del usuario.	20
9	HU09	Visualización del dashboard	Alta (1)	Muestra resumen del desempeño y avance del equipo.	25
10	HU10	Creación y edición de versiones	Media (2)	Permite crear y editar versiones del proyecto.	25
11	HU11	Control de versiones	Media (2)	Permite controlar las versiones de los proyectos creados.	20

## 4.2. Sprint Backlog

### 4.2.1. Sprint 1

Duración: 28 días

En este sprint se desarrollaron las funcionalidades base del sistema, enfocadas en la gestión de proyectos y usuarios.

- Épica: PDGPCCII-10 – Gestión de Proyectos
  - HU: PDGPCCII-1 – Crear proyectos
  - HU: PDGPCCII-2 – Gestión de tareas
  - HU: PDGPCCII-3 – Visualizar tarea
  - HU: PDGPCCII-14 – Estadísticas visuales sobre el progreso del proyecto
- Épica: PDGPCCII-13 – Gestionar usuarios y permisos
  - HU: PDGPCCII-13 – Gestión de usuarios y control de acceso

### 4.2.2. Sprint 2

Duración: 28 días

Durante este sprint se implementaron funcionalidades de comunicación en tiempo real para mejorar la colaboración entre miembros del equipo.

- Épica: PDGPCCII-11 – Comunicación y Colaboración
  - HU: PDGPCCII-4 – Chat
  - HU: PDGPCCII-5 – Recibir notificaciones
  - HU: PDGPCCII-6 – Mencionar miembros

### 4.2.3. Sprint 3

Duración: 28 días

Este sprint estuvo dedicado a la integración de capacidades de inteligencia artificial para automatizar y mejorar la gestión de tareas.

- Épica: PDGPCCII-12 – Inteligencia Artificial
  - HU: PDGPCCII-7 – Gestión de tareas con IA
  - HU: PDGPCCII-9 – Clasificación de tareas

## **4.2. Planificación de Sprints**

Para el desarrollo del sistema se utilizó la metodología ágil Scrum, la cual permitió dividir el trabajo en iteraciones cortas y funcionales llamadas sprints, facilitando la entrega continua de valor y la adaptación a cambios.

### **4.2.1. Historias de usuario**

ID	Descripción	Responsable	Sprint
HU01	Crear proyectos	Equipo de Desarrollo	Sprint 1
HU02	Gestión de tareas	Equipo de Desarrollo	Sprint 1
HU03	Visualizar tarea	Equipo de Desarrollo	Sprint 1
HU04	Estadísticas visuales sobre el progreso del proyecto	Equipo de Desarrollo	Sprint 1
HU05	Gestionar usuarios y permisos	Equipo de Desarrollo	Sprint 1
HU06	Chat	Equipo de Desarrollo	Sprint 2
HU07	Recibir notificaciones	Equipo de Desarrollo	Sprint 2
HU08	Mencionar miembros	Equipo de Desarrollo	Sprint 2
HU09	Gestión de tareas con IA	Equipo de Desarrollo	Sprint 3
HU10	Clasificación de tareas	Equipo de Desarrollo	Sprint 3

#### 4.2.2. Priorización de historias de usuario

ID	Historia resumida	Prioridad	Complejidad	Método
HU01	Crear proyectos	Alta	Media	MoSCoW
HU02	Gestión de tareas	Alta	Alta	MoSCoW
HU03	Visualizar tarea (Kanban)	Alta	Media	MoSCoW
HU04	Estadísticas visuales del progreso del proyecto	Media	Alta	MoSCoW
HU05	Gestionar usuarios y permisos	Alta	Media	MoSCoW
HU06	Chat interno	Media	Alta	MoSCoW
HU07	Recibir notificaciones	Media	Media	MoSCoW
HU08	Mencionar miembros	Media	Media	MoSCoW
HU09	Gestión de tareas con IA	Alta	Alta	MoSCoW
HU10	Clasificación automática de tareas mediante IA	Alta	Alta	MoSCoW

### 4.3. Cronograma de Actividades

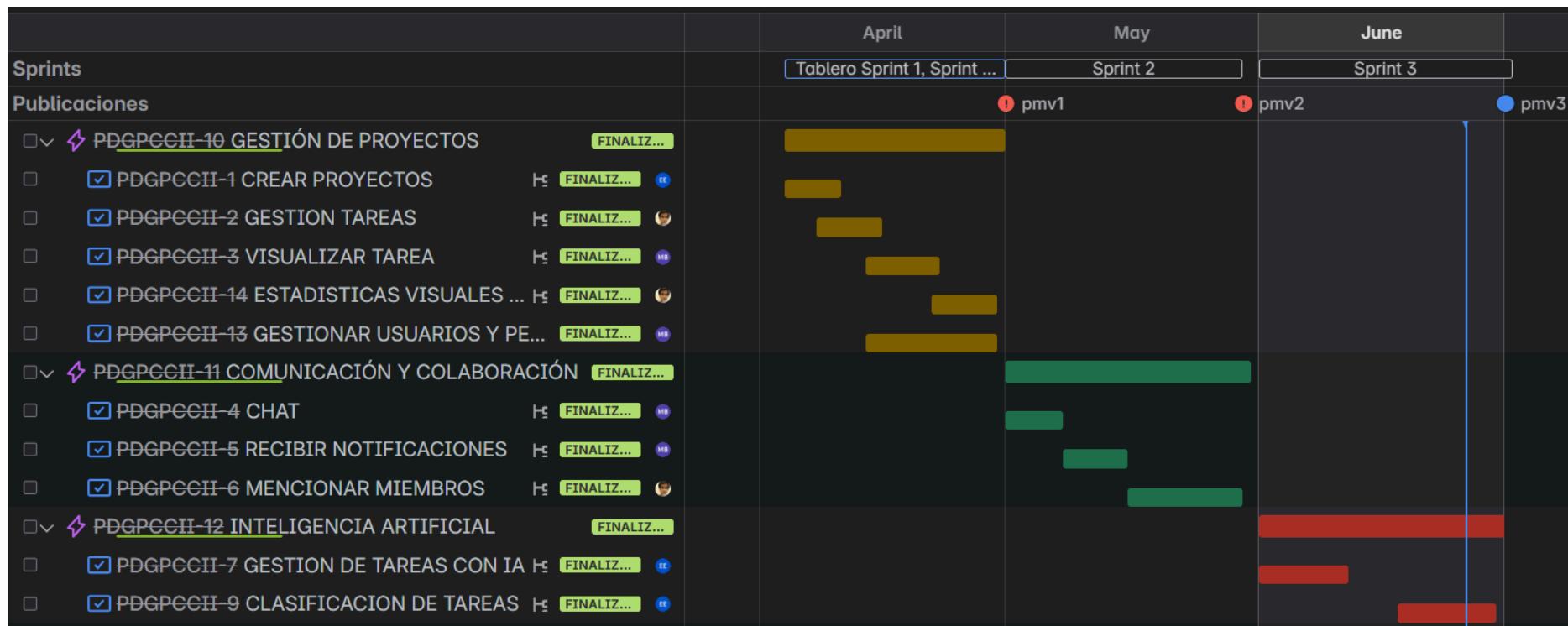


Figura 2

#### 4.4. Gestión de Riesgos

ID	Riesgo	Descripción	Impacto	Probabilidad	Estrategia de Mitigación
R1	Caídas del servidor	Fallas en el entorno cloud durante pruebas o despliegues.	Alto	Media	Uso de autoescalado, pruebas de carga, monitoreo constante.
R2	Accesos indebidos	Configuración incorrecta de roles compromete seguridad.	Alto	Media	Auditoría de seguridad y validación periódica de permisos.
R3	Bajo rendimiento del sistema	El backend no escala bien con múltiples usuarios simultáneos.	Alto	Alta	Optimización de backend, uso de caché y escalabilidad horizontal.
R4	Fallo en APIs externas	Las integraciones con servicios externos no siempre son estables.	Medio	Alta	Fallbacks locales y monitoreo activo de servicios.
R5	Competencia reduce tiempo disponible	Otros lanzan productos similares antes del nuestro.	Medio	Media	Liberación incremental (MVP), foco en diferenciación.
R6	Tecnologías nuevas para el equipo	El equipo tiene poca experiencia en IA y microservicios.	Alto	Alta	Capacitaciones internas y soporte técnico.
R7	Alta rotación de personal	Cambios frecuentes en el equipo afectan la continuidad del desarrollo.	Medio	Media	Cross-training, documentación técnica actualizada.

R8	Estimaciones incorrectas	Tareas IA o backend tienen incertidumbre alta.	Alto	Media	Añadir buffers y utilizar métricas históricas.
R9	Cronograma inflexible	No se adapta a cambios o bloqueos no previstos.	Alto	Media	Revisión del roadmap en cada sprint, adaptabilidad del backlog.
R10	Falta de seguimiento	No se hace seguimiento claro del progreso del equipo.	Medio	Media	Uso obligatorio de herramientas ágiles (Kanban, Trello).
R11	Falta de comunicación con stakeholders	Desalineación por falta de reuniones o visibilidad de avances.	Alto	Media	Reuniones quincenales, minutos y seguimiento de acuerdos.

## CAPÍTULO 5

### DISEÑO DEL SISTEMA DE INFORMACIÓN

#### 5.1. Diseño de Diagramas UML

##### 5.1.1. Diagramas de casos de uso

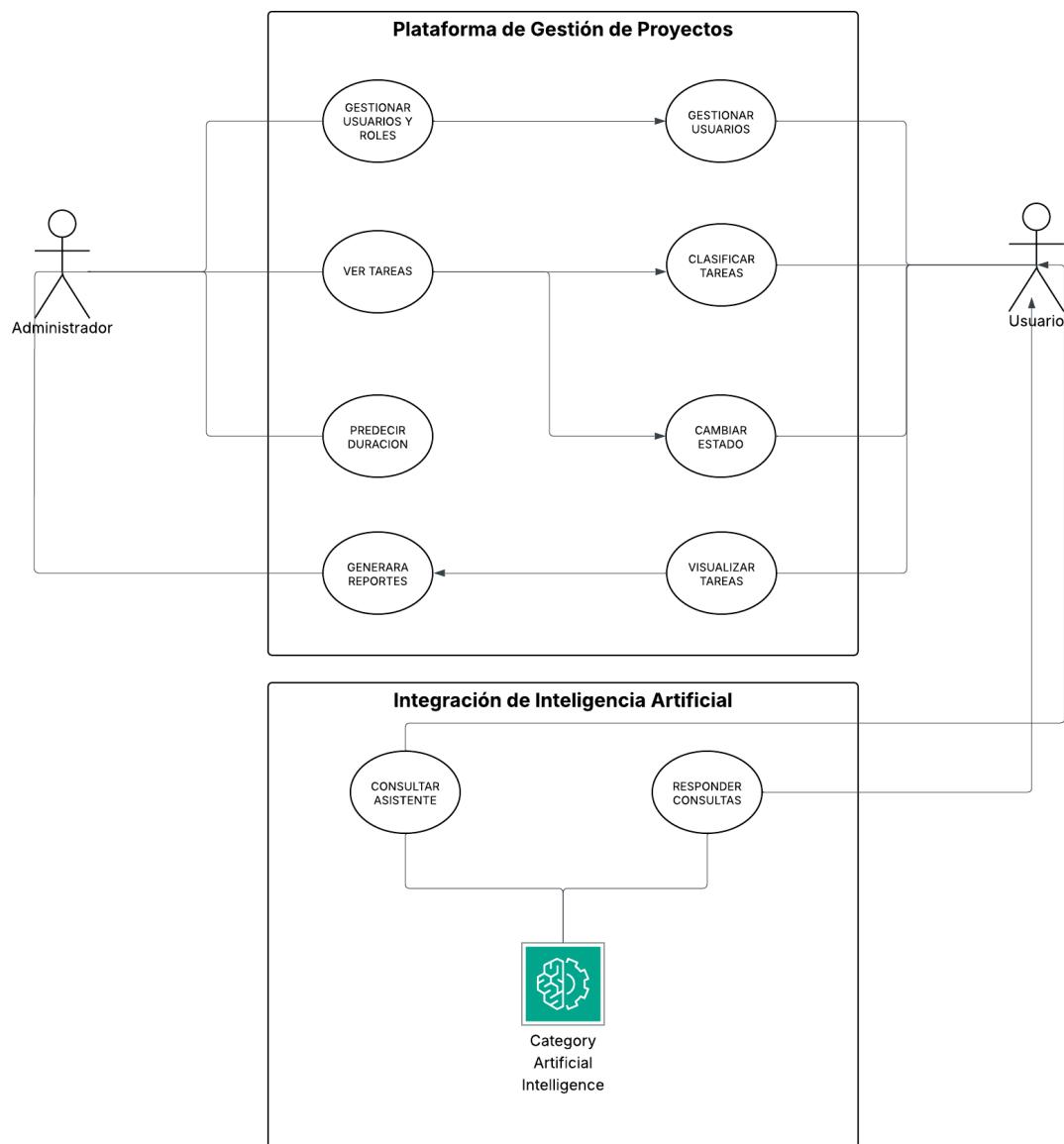
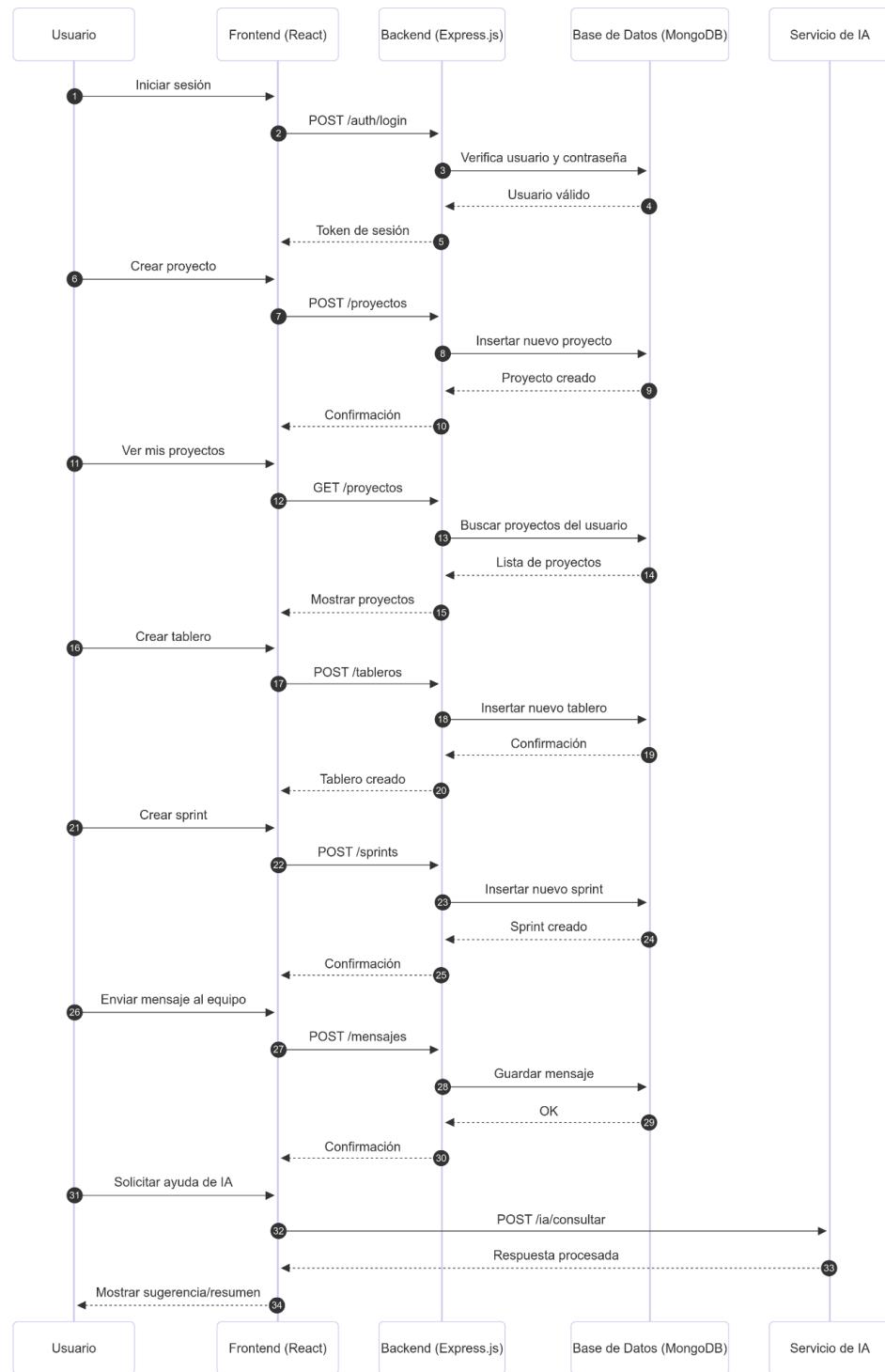
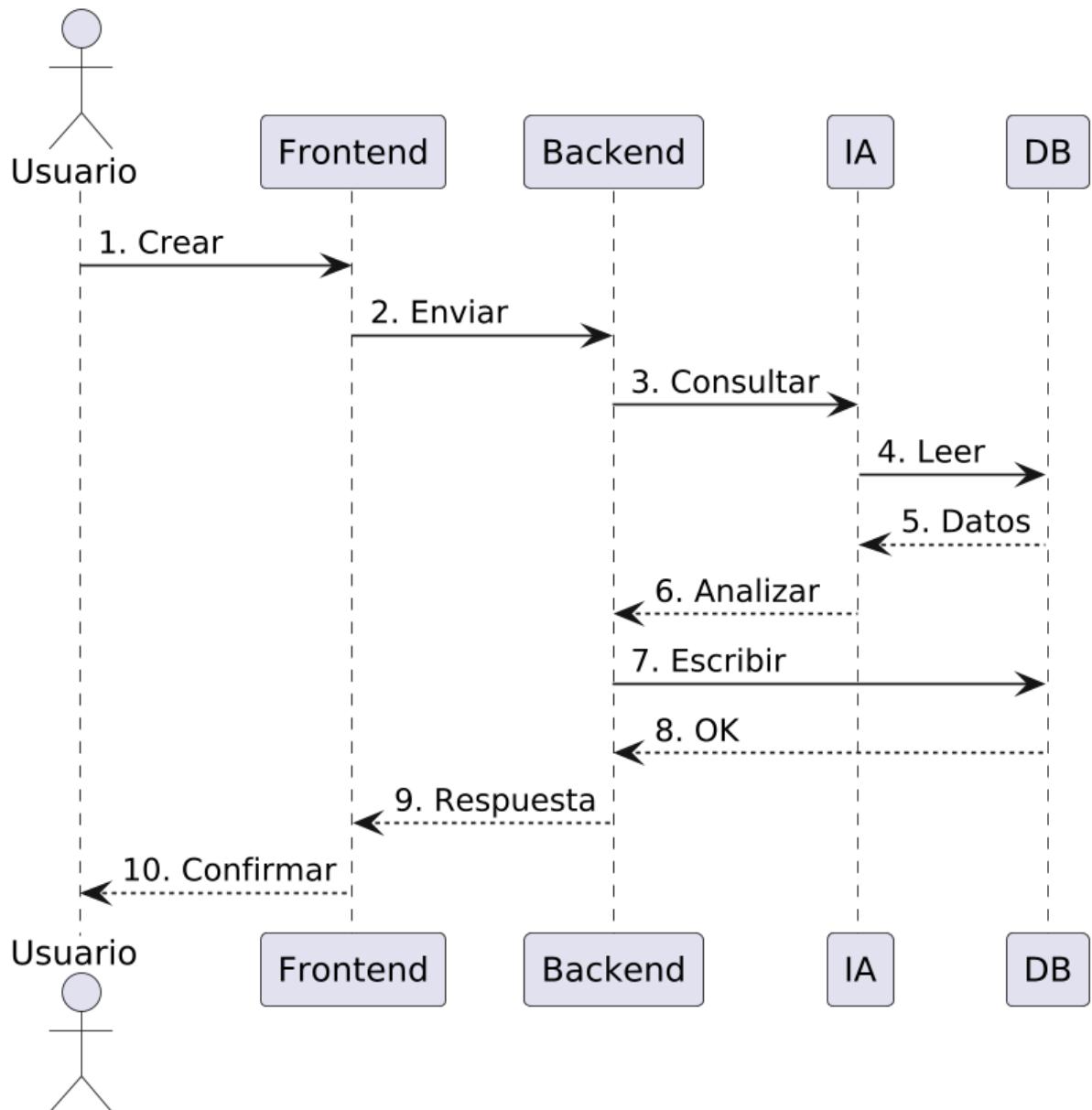


Figura 3

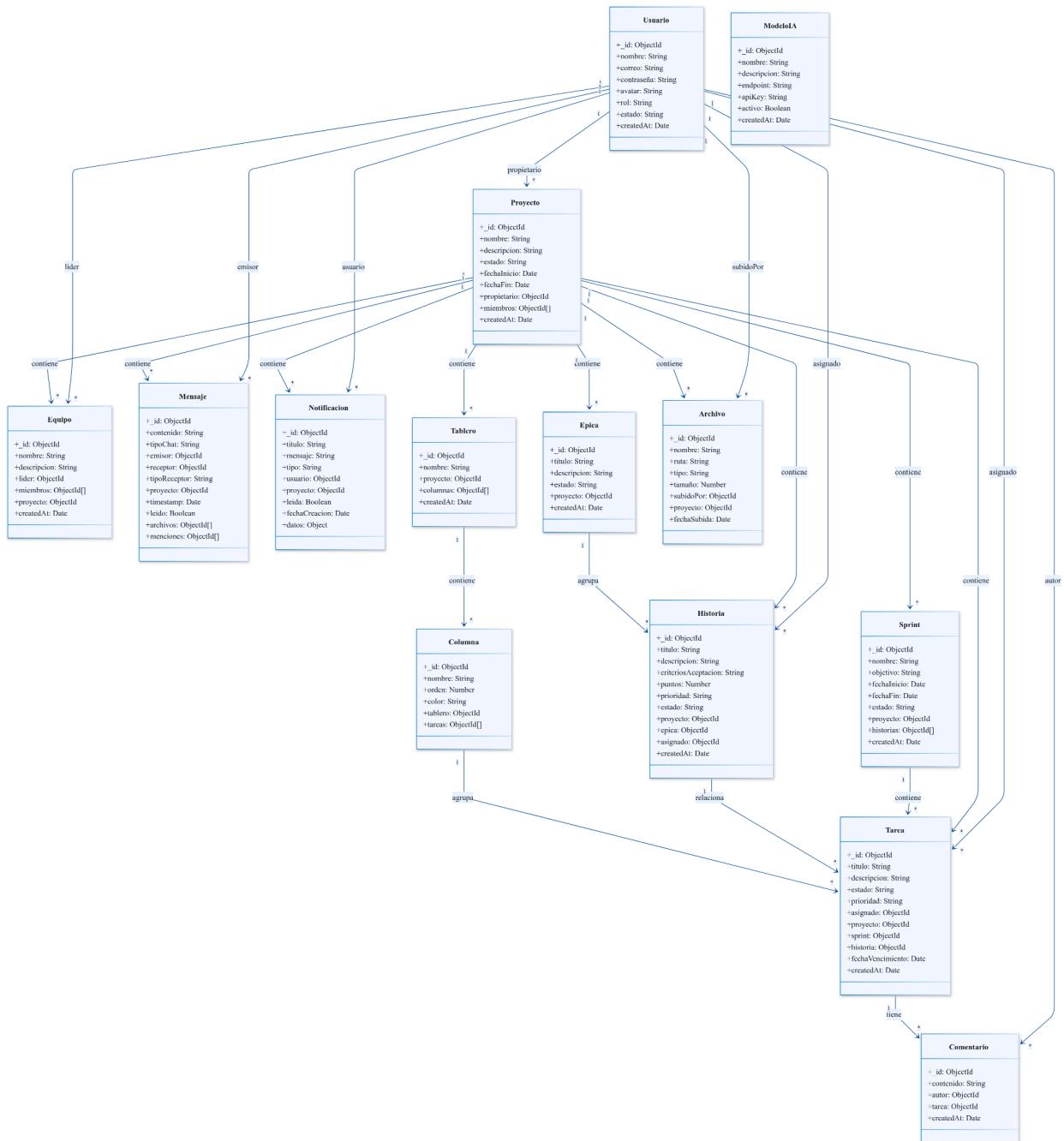
### 5.1.2. Diagramas de secuencia



### 5.1.3. Diagramas de colaboración

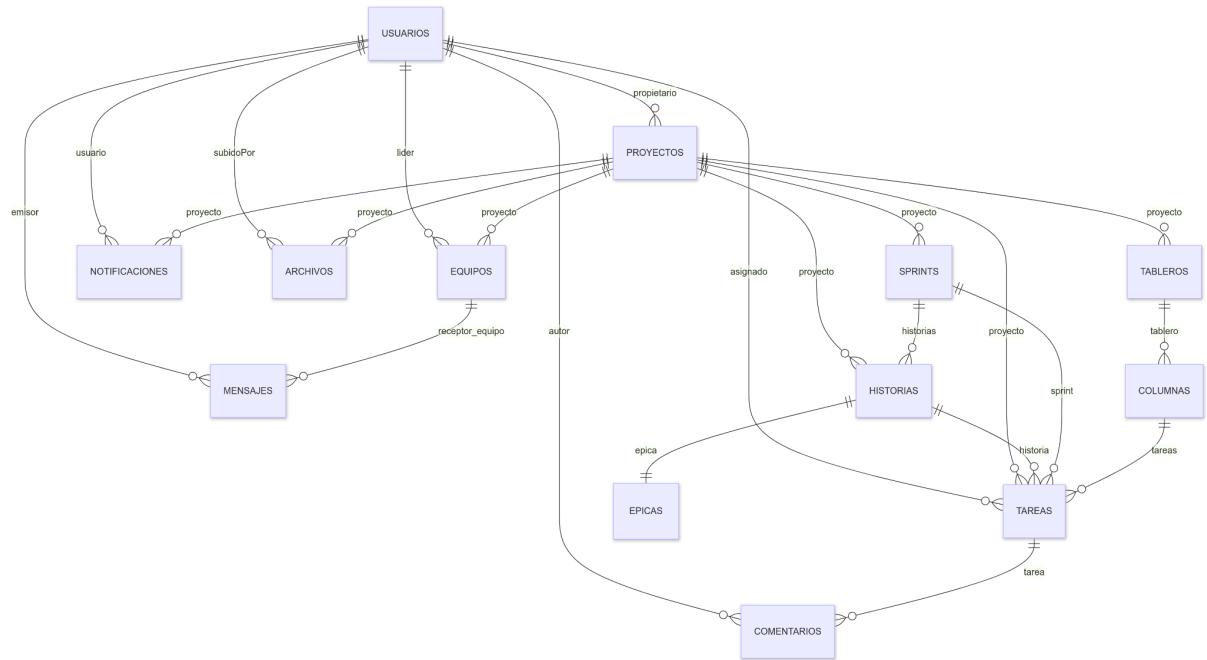


## 5.1.4. Diagramas de clases

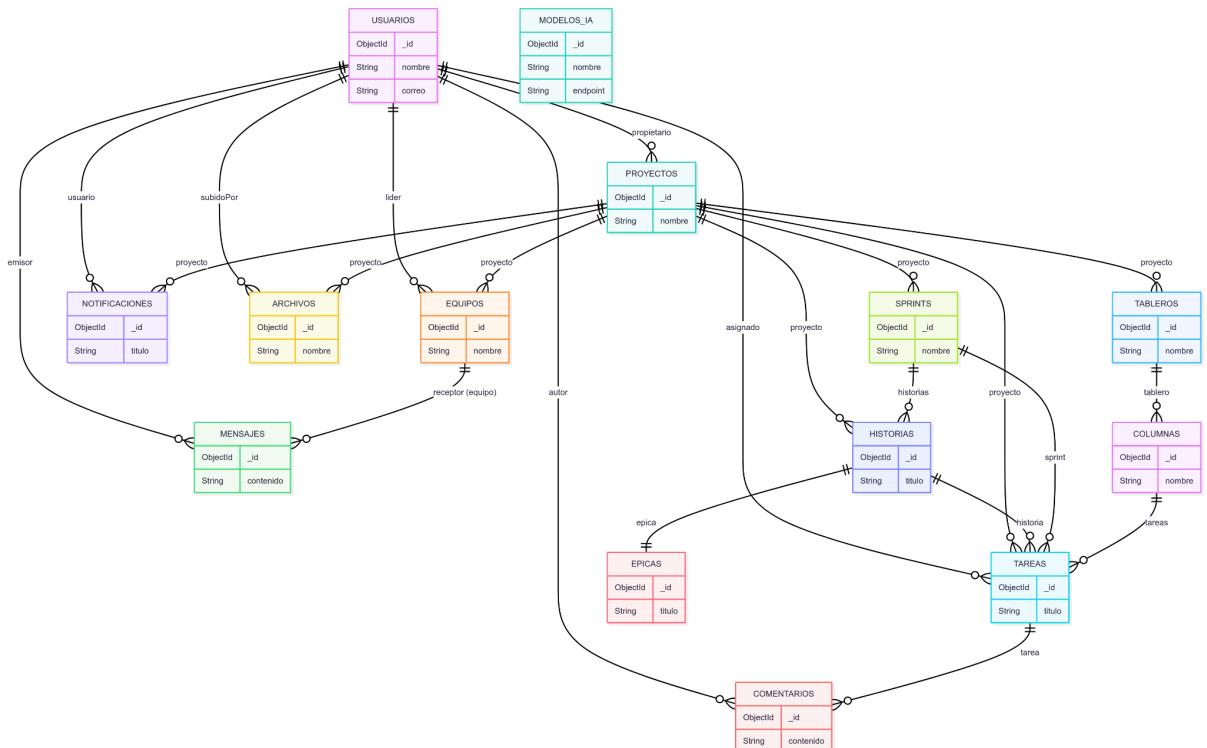


## 5.2. Diseño de Base de Datos

### 5.2.1. Diseño conceptual (E/R)



### 5.2.2. Diseño lógico



### 5.2.3. Diseño físico

```
// Tabla: USUARIOS
usuarios {
    _id:      ObjectId, // PK
    nombre:   String,   // NOT NULL
    correo:   String,   // UNIQUE, NOT NULL
    contraseña: String, // NOT NULL
    avatar:   String,   // DEFAULT ""
    rol:      String,   // ENUM("admin", "usuario", "líder"), DEFAULT "usuario"
    estado:   String,   // ENUM("activo", "inactivo", "bloqueado"), DEFAULT "activo"
    createdAt: Date,    // DEFAULT NOW()
}

// Tabla: PROYECTOS
proyectos {
    _id:      ObjectId, // PK
    nombre:   String,   // NOT NULL
    descripcion: String,
    estado:   String,   // ENUM("planificación", "activo", "pausado", "completado")
    fechalinicio: Date,
    fechaFin:  Date,
    propietario: ObjectId, // FK -> usuarios._id
    miembros:  Array,    // [ObjectId] -> usuarios._id
    createdAt:  Date,    // DEFAULT NOW()
}

// Tabla: EQUIPOS
equipos {
    _id:      ObjectId, // PK
    nombre:   String,   // NOT NULL
    descripcion: String,
    líder:    ObjectId, // FK -> usuarios._id
    miembros: Array,   // [ObjectId] -> usuarios._id
```

```

    proyecto: ObjectId, // FK -> proyectos._id
    createdAt: Date,   // DEFAULT NOW()
}

// Tabla: MENSAJES
mensajes {
    _id: ObjectId, // PK
    contenido: String, // NOT NULL
    tipoChat: String, // ENUM("directo", "equipo", "proyecto"), NOT NULL
    emisor: ObjectId, // FK -> usuarios._id, NOT NULL
    receptor: ObjectId, // NOT NULL
    tipoReceptor: String, // ENUM("usuario", "equipo", "proyecto"), NOT NULL
    proyecto: ObjectId, // FK -> proyectos._id, NULLABLE
    timestamp: Date, // DEFAULT NOW()
    leido: Boolean, // DEFAULT false
    archivos: Array, // [ObjectId] -> archivos._id
    menciones: Array, // [ObjectId] -> usuarios._id
}

// Tabla: NOTIFICACIONES
notificaciones {
    _id: ObjectId, // PK
    titulo: String, // NOT NULL
    mensaje: String, // NOT NULL
    tipo: String, // ENUM("info", "warning", "error", "success")
    usuario: ObjectId, // FK -> usuarios._id, NOT NULL
    proyecto: ObjectId, // FK -> proyectos._id, NULLABLE
    leida: Boolean, // DEFAULT false
    fechaCreacion: Date, // DEFAULT NOW()
    datos: Object, // JSON adicional
}

// Tabla: TAREAS
tareas {

```

```

_id: ObjectId, // PK
titulo: String, // NOT NULL
descripcion: String,
estado: String, // ENUM("pendiente", "en_progreso", "completada")
prioridad: String, // ENUM("baja", "media", "alta", "critica")
asignado: ObjectId, // FK -> usuarios._id
proyecto: ObjectId, // FK -> proyectos._id, NOT NULL
sprint: ObjectId, // FK -> sprints._id
historia: ObjectId, // FK -> historias._id
fechaVencimiento: Date,
createdAt: Date, // DEFAULT NOW()
}

```

```

// Tabla: SPRINTS
sprints {
    _id: ObjectId, // PK
    nombre: String, // NOT NULL
    objetivo: String,
    fechalinicio: Date, // NOT NULL
    fechaFin: Date, // NOT NULL
    estado: String, // ENUM("planificado", "activo", "completado")
    proyecto: ObjectId, // FK -> proyectos._id, NOT NULL
    historias: Array, // [ObjectId] -> historias._id
    createdAt: Date, // DEFAULT NOW()
}

```

```

// Tabla: HISTORIAS
historias {
    _id: ObjectId, // PK
    titulo: String, // NOT NULL
    descripcion: String,
    criteriosAceptacion: String,
    puntos: Number, // Story points
    prioridad: String, // ENUM("baja", "media", "alta")
}

```

```

estado:      String, // ENUM("pendiente", "en_progreso", "completada")
proyecto:    ObjectId, // FK -> proyectos._id, NOT NULL
epica:       ObjectId, // FK -> epicas._id
asignado:    ObjectId, // FK -> usuarios._id
createdAt:   Date,   // DEFAULT NOW()
}

// Tabla: EPICAS
epicas {
  _id:      ObjectId, // PK
  titulo:   String,  // NOT NULL
  descripcion: String,
  estado:   String, // ENUM("pendiente", "en_progreso", "completada")
  proyecto: ObjectId, // FK -> proyectos._id, NOT NULL
  createdAt: Date,   // DEFAULT NOW()
}

// Tabla: TABLEROS
tableros {
  _id:      ObjectId, // PK
  nombre:   String,  // NOT NULL
  proyecto: ObjectId, // FK -> proyectos._id, NOT NULL
  columnas: Array,   // [ObjectId] -> columnas._id
  createdAt: Date,   // DEFAULT NOW()
}

// Tabla: COLUMNAS
columnas {
  _id:      ObjectId, // PK
  nombre:   String,  // NOT NULL
  orden:    Number,  // Para ordenamiento
  color:    String,  // Color hex
  tablero:  ObjectId, // FK -> tableros._id, NOT NULL
  tareas:   Array,   // [ObjectId] -> tareas._id
}

```

```

}

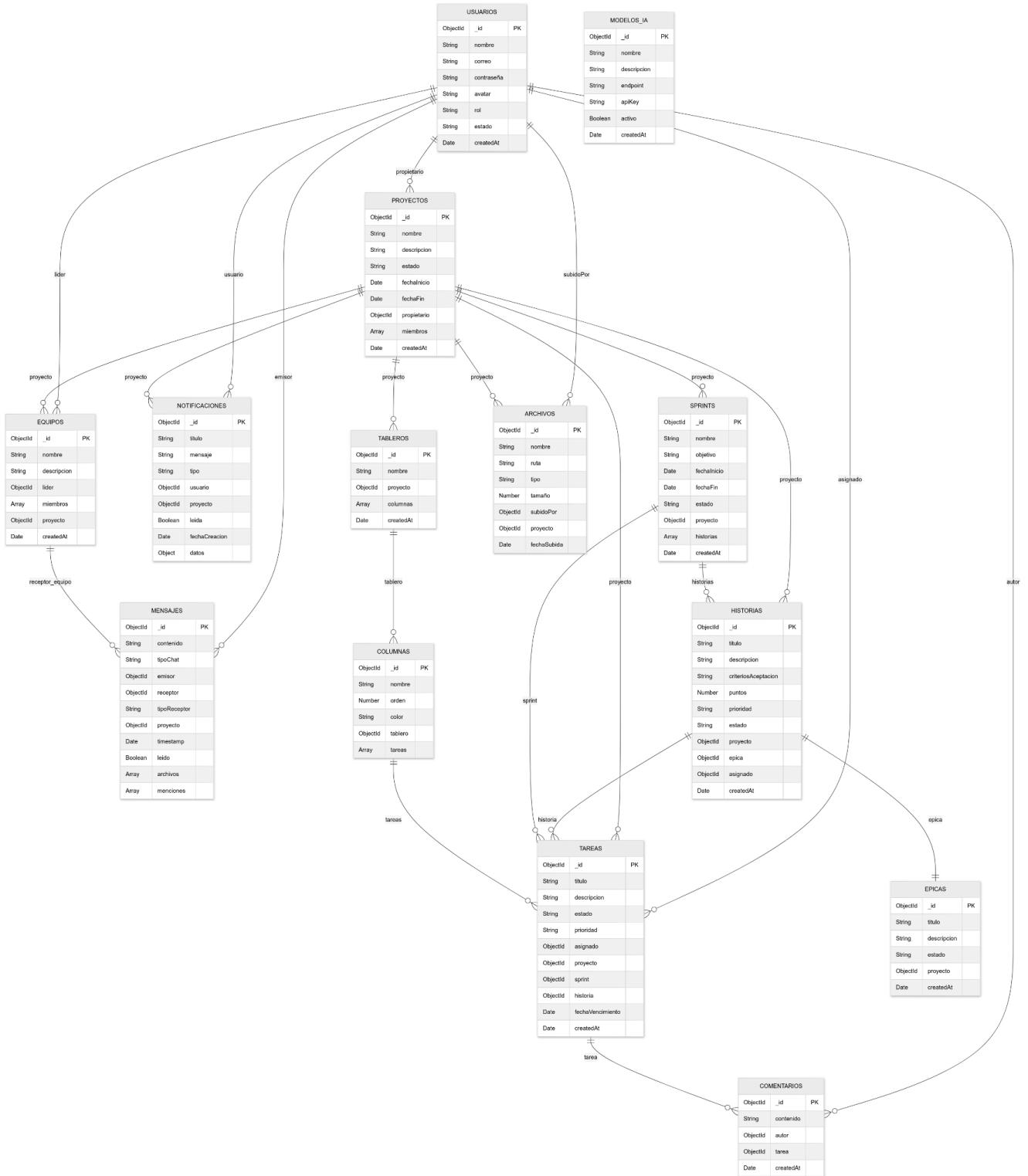
// Tabla: COMENTARIOS
comentarios {
    _id: ObjectId, // PK
    contenido: String, // NOT NULL
    autor: ObjectId, // FK -> usuarios._id, NOT NULL
    tarea: ObjectId, // FK -> tareas._id, NOT NULL
    createdAt: Date, // DEFAULT NOW()
}

// Tabla: ARCHIVOS
archivos {
    _id: ObjectId, // PK
    nombre: String, // NOT NULL
    ruta: String, // NOT NULL
    tipo: String, // MIME type
    tamaño: Number, // En bytes
    subidoPor: ObjectId, // FK -> usuarios._id, NOT NULL
    proyecto: ObjectId, // FK -> proyectos._id
    fechaSubida: Date, // DEFAULT NOW()
}

// Tabla: MODELOS_IA
modelos_ia {
    _id: ObjectId, // PK
    nombre: String, // NOT NULL
    descripcion: String,
    endpoint: String, // URL del API
    apiKey: String, // Clave de API (encriptada)
    activo: Boolean, // DEFAULT true
    createdAt: Date, // DEFAULT NOW()
}

```

#### 5.2.4. Modelado de base de datos



## 5.3. Diseño de Interfaces Básicas

### 5.3.1. Acceso login

**Iniciar Sesión**

Ingresa tus credenciales para acceder a tu cuenta

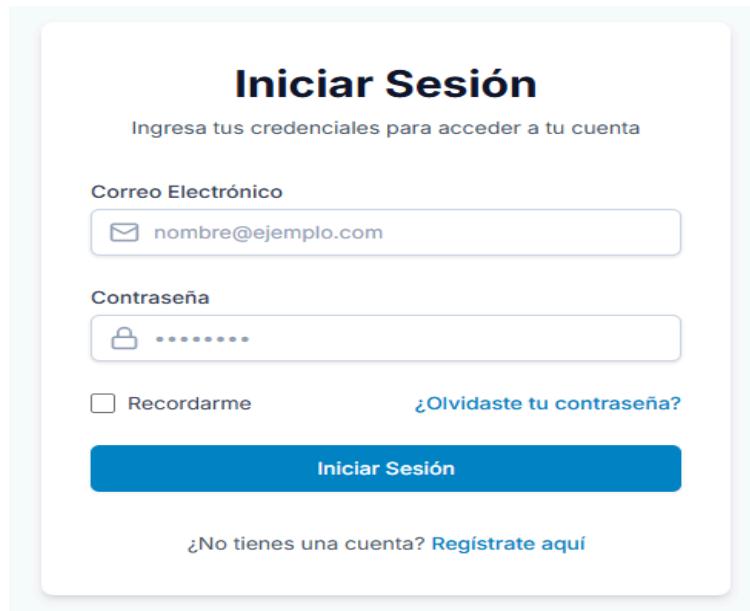
Correo Electrónico

Contraseña

Recordarme      [¿Olvidaste tu contraseña?](#)

**Iniciar Sesión**

¿No tienes una cuenta? [Regístrate aquí](#)



**Crear Cuenta**

Regístrate para comenzar a gestionar tus proyectos

Nombre Completo

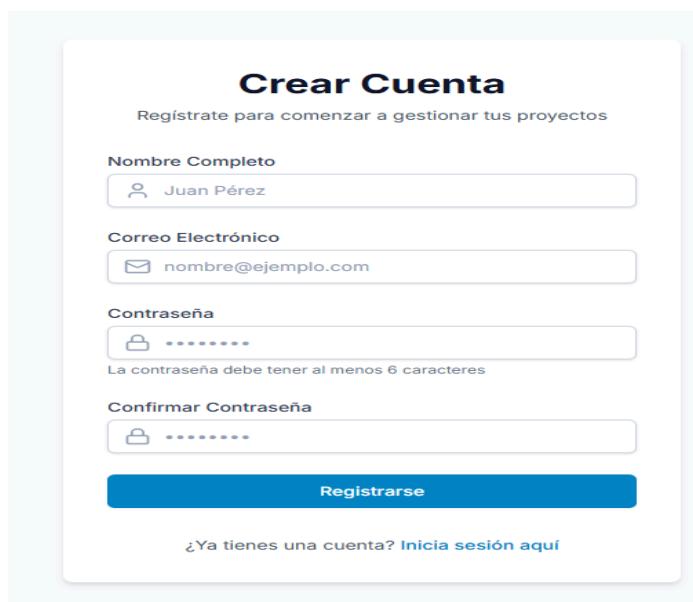
Correo Electrónico

Contraseña  
  
La contraseña debe tener al menos 6 caracteres

Confirmar Contraseña

**Registrarse**

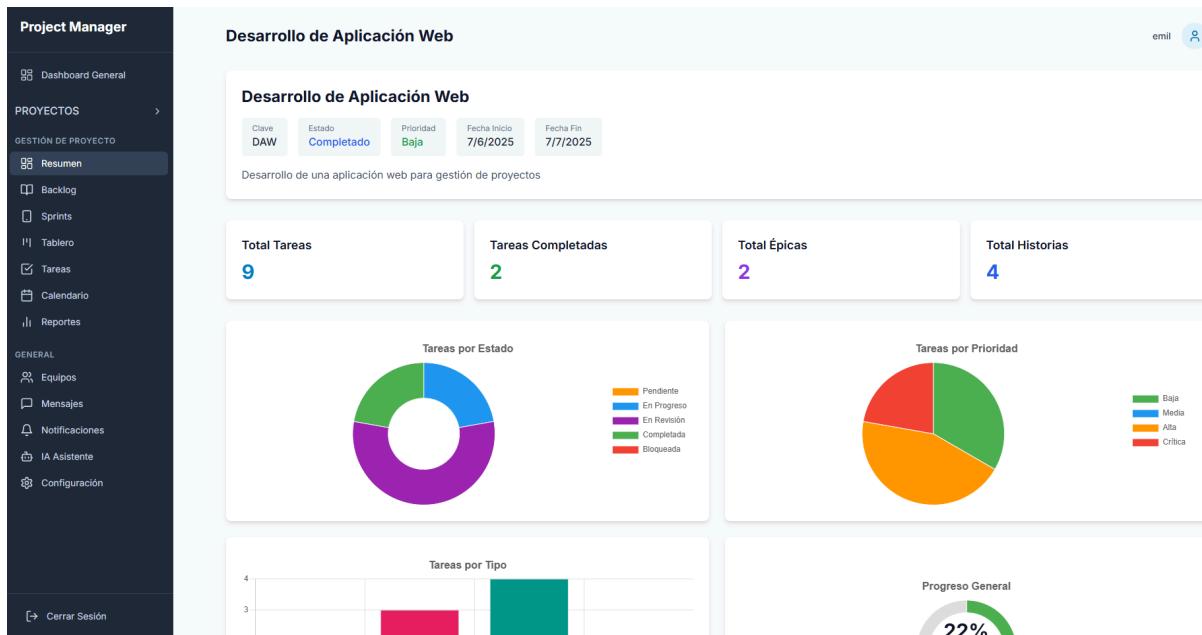
¿Ya tienes una cuenta? [Inicia sesión aquí](#)



### 5.3.2. Interfaces

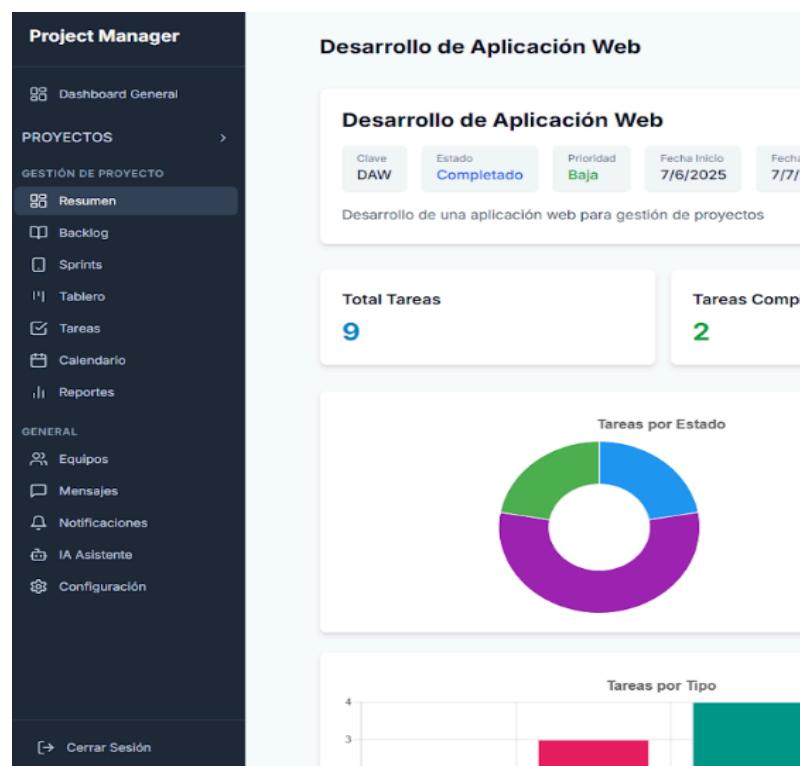
### 5.3.3. Panel de Dashboard:

Vista principal con resumen del proyecto, tareas y gráficas.



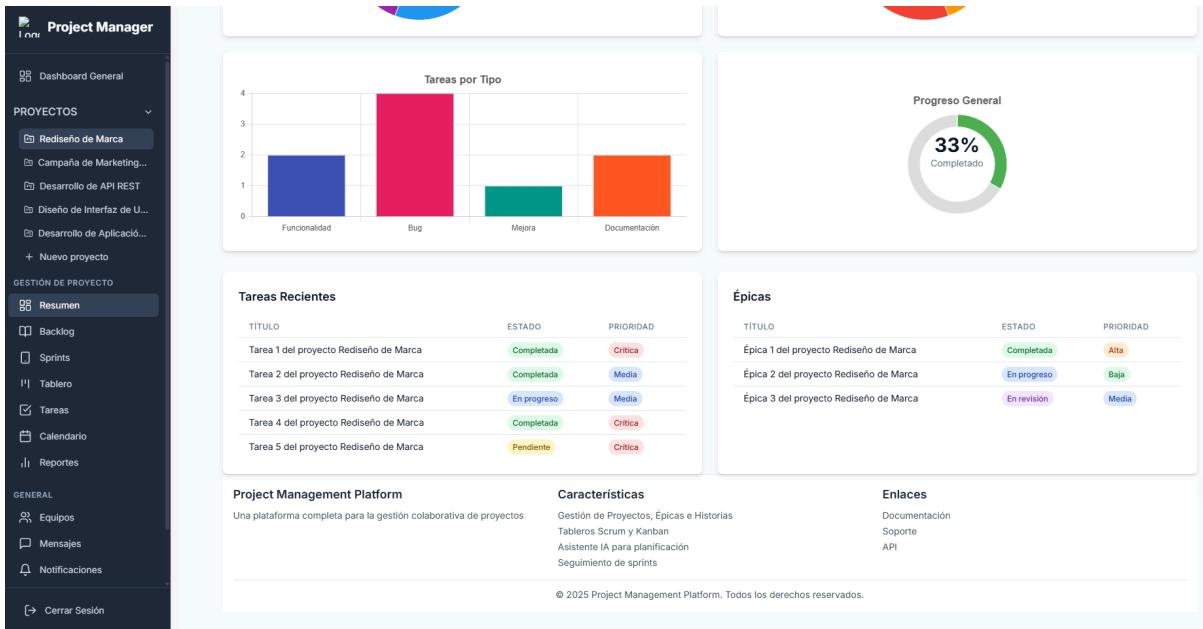
### 5.3.4. Menú de Navegación Lateral

Contiene accesos a módulos: Backlog, Sprints, Tablero, Tareas, Calendario y Reportes.



### 5.3.5. Vista de Resumen de Proyecto

Muestra información detallada del proyecto como: clave, estado, prioridad, fechas de inicio/fin, descripción.



### 5.3.6. Gráficas de Progreso

- Tareas por Estado (gráfico circular)
- Tareas por Tipo (gráfico de barras)



### 5.3.7. Módulo de Tareas

Visualización y gestión de todas las tareas del proyecto.

The screenshot shows the 'Tareas' (Tasks) module of the Project Manager application. On the left, there's a sidebar with navigation links for 'Dashboard General', 'PROYECTOS' (with items like 'Rediseño de Marca', 'Campaña de Marketing...', etc.), 'GESTIÓN DE PROYECTO' (with 'Resumen', 'Backlog', 'Sprints', 'Tablero', 'Tareas' - which is selected and highlighted in blue), 'Reportes', 'GENERAL' (with 'Equipos', 'Mensajes', 'Notificaciones'), and 'Cerrar Sesión'. The main content area has a header 'Tareas' with a 'Nueva Tarea' button. Below it is a table with columns: Título (Title), Proyecto (Project), Estado (Status), Prioridad (Priority), Asignado a (Assigned to), Fecha Límite (Deadline), and Acciones (Actions). The table contains five tasks:

Título	Proyecto	Estado	Prioridad	Asignado a	Fecha Límite	Acciones
Diseñar interfaz de usuario	DAW	en progreso	alta	Ana Martínez	1 jul 2025	
Implementar autenticación	DAW	pendiente	alta	Carlos Gómez	3 jul 2025	
Crear documentación API	API	completada	media	Juan Pérez	24 jun 2025	
Optimizar rendimiento	DAW	en revisión	critica	Maria López	29 jun 2025	
Corregir bug en formulario	RDM	pendiente	critica	Ana Martínez	27 jun 2025	

Below the table, there's a section for 'Project Management Platform' with a brief description and links to 'Características' (Features) and 'Enlaces' (Links). At the bottom, a copyright notice reads '© 2025 Project Management Platform. Todos los derechos reservados.'

### 5.3.8. Módulo de Sprints

Administración de sprints para desarrollo ágil.

The screenshot shows the 'Sprints' module of the Project Manager application. The sidebar is identical to the previous one. The main content area has a header 'Sprints' with a 'Proyecto: Rediseño de Marca' and a '+ Nuevo Sprint' button. A modal window titled 'Crear Nuevo Sprint' is open, prompting for 'Nombre del Sprint \*' (Ej: Sprint 1 - Funcionalidades básicas), 'Objetivo del Sprint' (Describe el objetivo principal de este sprint...), 'Fecha de Inicio \*' (26/06/2025), and 'Fecha de Fin \*' (10/07/2025). To the right of the modal, there's a list of 'Historias del Sprint' (Stories of the Sprint) with three items: 'Historia 2 de la épica Épica 1 del proyecto...' (media priority), 'Historia 1 de la épica Épica 2 del proyecto...' (baja priority), and 'Historia 2 de la épica Épica 2 del proyecto...' (critica priority). At the bottom of the modal are 'Cancelar' and 'Crear Sprint' buttons. The footer of the page includes a section for 'Project Management Platform' with its features and links, and a copyright notice.

### 5.3.9. Módulo de Tablero (Kanban)

Vista tipo Kanban para el estado de tareas

**Tablero de Rediseño de Marca**

● Sprint Activo: botame (#1)

Estado	Contenido
Por hacer	Historia 2 de la épica Épica 1 Descripción de la historia 2 de la épica Épica 1 del proyecto Rediseño de Marca Épica 1 del proyecto Rediseño de Marca Sprint: botame
En progreso	Historia 1 de la épica Épica 2 Descripción de la historia 1 de la épica Épica 2 del proyecto Rediseño de Marca Épica 2 del proyecto Rediseño de Marca Sprint: botame
Completado	

**Project Management Platform**  
Una plataforma completa para la gestión colaborativa de proyectos

**Características**  
Gestión de Proyectos, Épicas e Historias  
Tableros Scrum y Kanban  
Asistente IA para planificación  
Seguimiento de sprints

**Enlaces**  
Documentación  
Soporte  
API

© 2025 Project Management Platform. Todos los derechos reservados.

### 5.3.10. Gestión de Equipos

Asignación de miembros del equipo al proyecto.

**Equipos**

Gestiona los equipos de trabajo de tu organización

**+ Nuevo Equipo**

Equipo	Miembros
Equipo de Diseño	3 miembros
Equipo de Marketing	3 miembros
Equipo de Desarrollo	5 miembros

**Project Management Platform**  
Una plataforma completa para la gestión colaborativa de proyectos

**Características**  
Gestión de Proyectos, Épicas e Historias  
Tableros Scrum y Kanban  
Asistente IA para planificación  
Seguimiento de sprints

**Enlaces**  
Documentación  
Soporte  
API

© 2025 Project Management Platform. Todos los derechos reservados.

### 5.3.11. Asistente IA

The screenshot displays the Project Manager software interface. On the left, a dark sidebar contains navigation links for 'Dashboard General', 'PROYECTOS' (with items like 'Rediseño de Marca', 'Campaña de Marketing...', 'Desarrollo de API REST', 'Diseño de Interfaz de U...', 'Desarrollo de Aplicació...', '+ Nuevo proyecto'), 'GESTIÓN DE PROYECTO' (with 'Resumen', 'Backlog', 'Sprints', 'Tablero', 'Tareas', 'Calendario', 'Reportes'), 'GENERAL' (with 'Equipos', 'Mensajes', 'Notificaciones'), and a 'Cerrar Sesión' button.

The main area is titled 'Asistente IA' and shows a conversation with the AI. The conversation history includes:

- Conversación actual: 'Hey'
- Análisis de proyecto DAW (Ayer)
- Planificación sprint (12/05/2023)

The AI response is: '¡Hola! Soy tu asistente IA especializado en gestión de proyectos. Puedo ayudarte con:' followed by a list of services: Análisis de progreso y métricas, Generación de reportes detallados, Optimización de procesos, Gestión de equipos, Planificación de sprints, and '¿En qué puedo asistirte hoy?'

Suggestions include: Analizar progreso del proyecto, Generar reporte de sprint, Revisar métricas del equipo, and Optimizar planificación.

Quick actions (Acciones Rápidas) include: ¿Cuál es el estado de mis proyectos?, Muestra mis tareas pendientes, Analiza el rendimiento de mi equipo, Genera un informe de progreso, ¿Cómo puedo mejorar la productividad?, Ayúdame a priorizar mis tareas, Identifica riesgos en el proyecto, and Planifica el próximo sprint.

A search bar at the bottom says 'Escribe tu pregunta o solicitud...'.

Footer information includes: Project Management Platform (Una plataforma completa para la gestión colaborativa de proyectos), Características (Gestión de Proyectos, Épicas e Historias, Tablero, Scrum y Kanban), and Enlaces (Documentación, Soporte).

## CAPÍTULO 6

### CODIFICACIÓN DEL SOFTWARE

#### 6.1. Desarrollo del Sprint 1

##### 6.1.1. Sprint planning

En la etapa de planificación del Sprint 1 se definió como prioridad construir la base funcional del sistema de gestión de proyectos. El objetivo principal fue permitir la creación, visualización y gestión de tareas, además de establecer la arquitectura principal del backend y frontend usando el stack MERN. Se acordó un sprint de 28 días, con foco en entregar funcionalidades mínimas viables que permitieran probar la integración entre los módulos iniciales.

##### 6.1.2. Sprint backlog

ID	Historia/Tarea
PDGPCCII-1	Crear proyectos
PDGPCCII-2	Gestión de tareas
PDGPCCII-3	Visualización de tareas
PDGPCCII-10	Finalización de proyectos
PDGPCCII-13	Gestión de usuarios y permisos
PDGPCCII-14	Estadísticas visuales del progreso del proyecto

##### 6.1.3. Historias de usuarios

- **HU01:** Como usuario, quiero crear proyectos. (*PDGPCCII-1*)
- **HU02:** Como usuario, quiero gestionar tareas. (*PDGPCCII-2*)
- **HU03:** Como usuario, quiero visualizar tareas por estado. (*PDGPCCII-3*)
- **HU07:** Como usuario, quiero finalizar proyectos. (*PDGPCCII-10*)
- **HU08:** Como administrador, quiero gestionar usuarios y permisos. (*PDGPCCII-13*)

- **HU09:** Como usuario, quiero ver estadísticas visuales del progreso. (*PDGPCCII-14*)

#### **6.1.4. Taskboard**

Historias	To Do	In Progress	Done
Dashboard			X
Crear proyectos			X
Gestión de tareas			X
Visualizar tarea			X
Estadísticas visuales			X
Gestionar usuarios y permisos			X
Comunicación y colaboración			X

#### **6.1.5. Daily scrum**

Las reuniones diarias se llevaron a cabo a las 9:00 p.m., permitiendo al equipo compartir su progreso, identificar bloqueos y sincronizar actividades. Como desarrollador principiante en el stack MERN, el seguimiento diario fue clave para consolidar aprendizajes. Al inicio del sprint se presentaron desafíos con la configuración del entorno y conexión a MongoDB, pero con el paso de los días se logró avanzar en la implementación de rutas API y componentes React.

#### **6.1.6. Sprint review**

Se demostró:

- Interfaz de creación de proyectos

- CRUD de tareas
  - Vista tipo Kanban
  - Progreso visual (estadísticas)
- Administración de usuarios

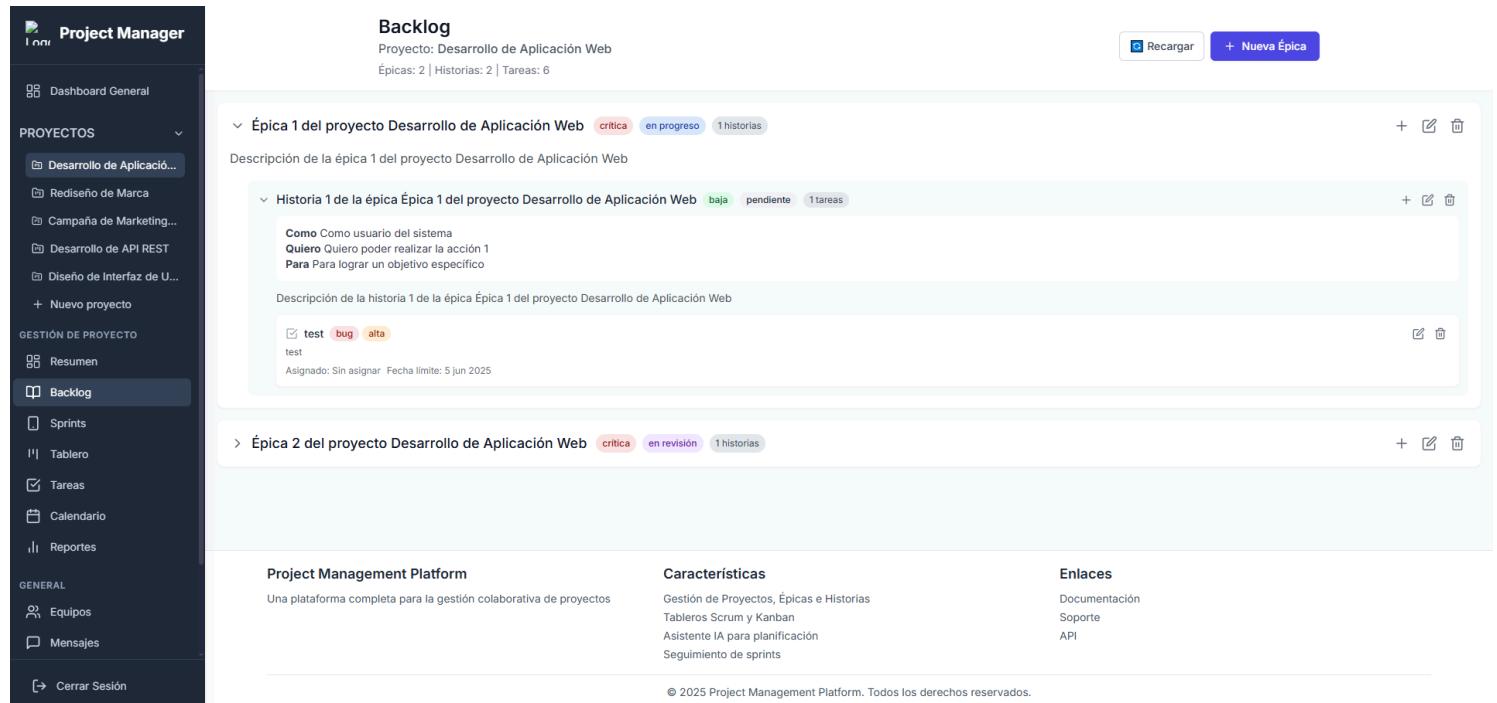
Feedback del PO: funcionalidades operativas, propuesta de mejoras visuales en tablero

#### 6.1.7. Criterios de aceptación

- Creación y edición de proyectos desde la interfaz web.
- CRUD de tareas asociadas a cada proyecto
- Visualización de tareas por estado en el tablero de kanban
- Autenticación funcional con control de acceso por rol .

#### 6.1.8. Resultados del sprint

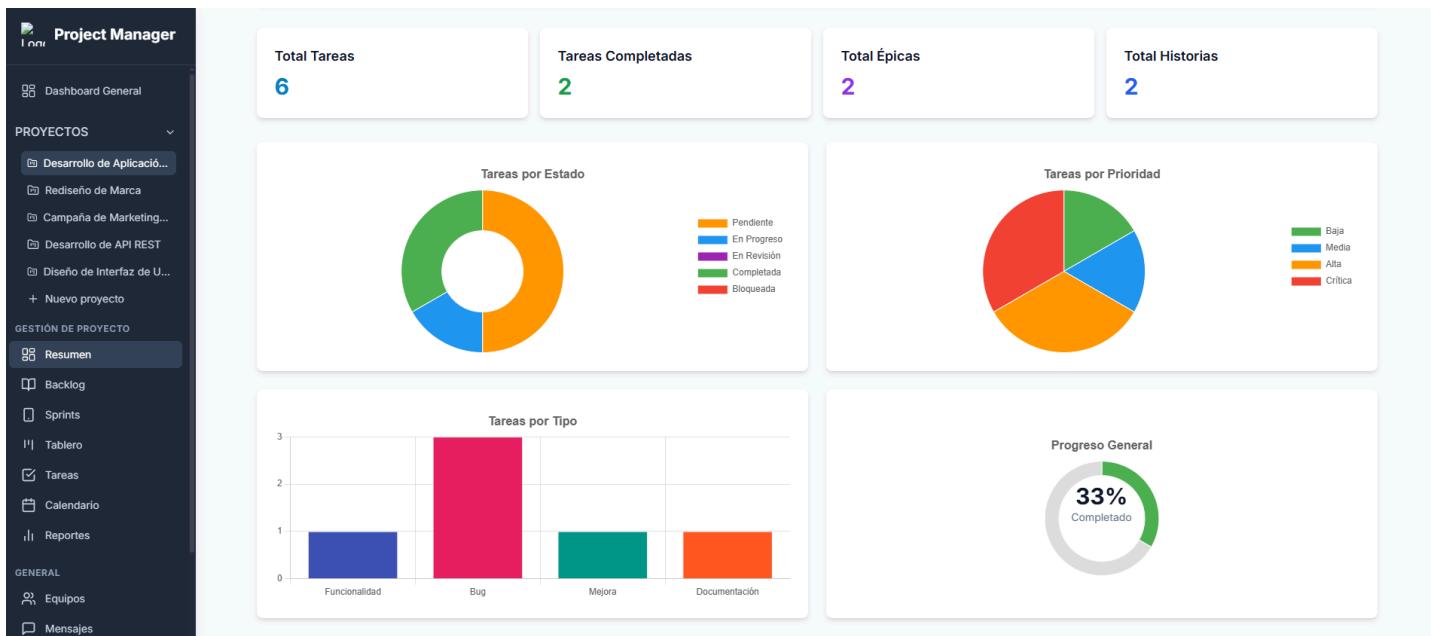
##### 6.1.8.1. Evidencias.



The screenshot shows the Project Manager application's interface. On the left, there is a sidebar with navigation links: Dashboard General, PROYECTOS (with Desarrollo de Aplicación selected), GESTIÓN DE PROYECTO (with Resumen and Backlog selected), GENERAL (with Equipos and Mensajes), and a Cerrar Sesión button. The main content area is titled "Backlog" and shows details for the project "Desarrollo de Aplicación Web". It lists 2 Épicas, 2 Historias, and 6 Tareas. A specific Epic, "Épica 1 del proyecto Desarrollo de Aplicación Web", is expanded, showing its description and a detailed history item. Another Epic, "Épica 2 del proyecto Desarrollo de Aplicación Web", is also listed. At the bottom, there is a footer with sections for "Project Management Platform", "Características", and "Enlaces", along with copyright information.

Project Management Platform		Características	Enlaces
Una plataforma completa para la gestión colaborativa de proyectos	Gestión de Proyectos, Épicas e Historias Tableros Scrum y Kanban Asistente IA para planificación Seguimiento de sprints	Documentación Soporte API	

© 2025 Project Management Platform. Todos los derechos reservados.



**Project Manager**

**PROYECTOS**

- Desarrollo de Aplicación...
- Rediseño de Marca
- Campaña de Marketing...
- Desarrollo de API REST
- Diseño de Interfaz de U...
- + Nuevo proyecto

**GESTIÓN DE PROYECTO**

- Resumen
- Backlog
- Sprints
- Tablero
- Tareas
- Calendario
- Reportes

**GENERAL**

- Equipos
- Mensajes

**Proyectos**

**Desarrollo de Aplicación Web** alta  
DAV/pausado  
Desarrollo de una aplicación web para gestión de proyectos  
Inicio: 26 jun 2025  
Fin: 26 jul 2025

**Ver Tablero** **Editar** **Eliminar**

**Diseño de Interfaz de Usuario** media  
DIU/activo  
Diseño de la interfaz de usuario para la aplicación móvil  
Inicio: 26 jun 2025  
Fin: 6 jul 2025

**Ver Tablero** **Editar** **Eliminar**

**Nuevo Proyecto**

**Nombre del Proyecto \***

**Clave del Proyecto \***   
La clave debe tener entre 2 y 10 caracteres (ej. PRO1, DEV, MKT).

**Descripción**

**Equipo \***  Selecciona un equipo

**Estado** **Actividad** **Prioridad** **Media**

**Fecha de Inicio** 26/06/2025 **Fecha de Fin (opcional)** dd/mm/aaaa

**Ayuda** Asistente IA para planificación, Optimización de recursos, Seguimiento de sprints.

**Nuevo Proyecto**

**Desarrollo de API REST** alta  
API/activo  
Desarrollo de una API REST para integración con sistemas externos  
Inicio: 26 jun 2025  
Fin: 16 jul 2025

**Ver Tablero** **Editar** **Eliminar**

**Sprints**

Proyecto: Desarrollo de Aplicación Web

+ Nuevo Sprint

**test2 #1**  
Pendiente

dedede  
25 jun 2025 - 9 jul 2025  
14 días

2 Historias 0 Puntos

> Iniciar Detalles

**Project Management Platform**  
Una plataforma completa para la gestión colaborativa de proyectos

**Características**  
Gestión de Proyectos, Épicas e Historias  
Tableros Scrum y Kanban  
Asistente IA para planificación  
Seguimiento de sprints

**Enlaces**  
Documentación  
Soporte  
API

© 2025 Project Management Platform. Todos los derechos reservados.

**Tablero de Desarrollo de Aplicación Web**

Sprint Activo: test2 (#1)

Por hacer: 2 En progreso: 0 Completado: 0

Historia 1 de la épica Épica 1 del proyecto Desarrollo de...  
Descripción de la historia 1 de la épica Épica 1 del proyecto Desarrollo de Aplicación Web  
Épica 1 del proyecto Desarrollo de Aplicación Web pendiente  
Sprint: test2

Historia 1 de la épica Épica 2 del proyecto Desarrollo de...  
Descripción de la historia 1 de la épica Épica 2 del proyecto Desarrollo de Aplicación Web  
Épica 2 del proyecto Desarrollo de Aplicación Web pendiente  
Sprint: test2

**Project Management Platform**  
Una plataforma completa para la gestión colaborativa de proyectos

**Características**  
Gestión de Proyectos, Épicas e Historias  
Tableros Scrum y Kanban  
Asistente IA para planificación  
Seguimiento de sprints

**Enlaces**  
Documentación  
Soporte  
API

© 2025 Project Management Platform. Todos los derechos reservados.

### 6.1.8.2. Prueba de desarrollo.

```

PRUEBASAPP
> client
> server
> config
> controllers
> controllers_pure
> middleware
> models
> routes
> seeders
> tests
  > epicasfunctional.test.js
  > equiposfunctional.test.js
  > historiasfunctional.test.js
  > proyectosfunctional.test.js
  > runAllFunctionalTests.js
  > tareasfunctional.test.js
  > usuariosfunctional.test.js
> jest.config.js
> package-lock.json
> package.json
> server.js
> components.json
> next.config.js
> package-lock.json
> package.json
> pnpm-locky.yaml
> postcss.config.js
> tailwind.config.js
tsconfig.json

server > tests > proyectosfunctional.test.js > createProyectoModel > updateOne
1 import assert from 'assert'
2
3 // Simulación de un modelo Proyecto en memoria
4 function createProyectoModel() {
5   let data = []
6   let idCounter = 1
7   return {
8     async find() { return data },
9     async findById(id) { return data.find(p => p._id === id) || null },
10    async findOne(query) { return data.find(p => p.clave === query.clave) || null },
11    async create(obj) {
12      const nuevo = { ...obj, _id: String(idCounter++) }
13      data.push(nuevo)
14      return nuevo
15    },
16    async updateOne(query, update) {
17      const idx = data.findIndex(p => p._id === query._id)
18      if (idx === -1) return { nModified: 0 }
19      data[idx] = { ...data[idx], ...update }
20      return { nModified: 1 }
21    },
22    async deleteOne(query) {
23      const idx = data.findIndex(p => p._id === query._id)
24      if (idx === -1) return { deletedCount: 0 }
25      data.splice(idx, 1)
26      return { deletedCount: 1 }
27    },
28    _reset() { data = []; idCounter = 1 },
29  }
30}
31 // Controladores funcionales CRUD
32 const Proyecto = createProyectoModel()
33
34 async function crearProyecto(req, res) {
35   const { nombre, clave } = req.body
36   if (!nombre || !clave) {
37     return res.status(400).json({ success: false, message: 'Faltan campos' })
38   }
39   if (await Proyecto.findOne({ clave })) {
40     return res.status(400).json({ success: false, message: 'Clave duplicada' })
41   }
42   const nuevo = await Proyecto.create(req.body)
43   res.status(201).json({ success: true, proyecto: nuevo })
44 }
45

```

### 6.1.9. Sprint retrospective

Fortalezas: colaboración constante, cumplimiento de entregables

Mejoras propuestas: optimizar tiempos de revisión de código y documentar mejor los cambios.

## 6.2. Desarrollo del Sprint 2

### 6.2.1. Sprint planning

En la etapa de planificación del Sprint 2, se definió como prioridad fortalecer la comunicación y colaboración entre los usuarios dentro del sistema de gestión de proyectos.

Se acordó un sprint de 28 días, enfocado en implementar tres funcionalidades clave:

- Mención de miembros
- Notificaciones en tiempo real

- Chat de comunicación colaborativa

El objetivo fue extender la experiencia del usuario más allá de la gestión de tareas, permitiendo una interacción fluida entre los participantes del proyecto.

### 6.2.2. Sprint backlog

Código	Historia
PDGPCCII-15	HU10: Como usuario, quiero mencionar miembros en comentarios.
PDGPCCII-16	HU11: Como usuario, quiero recibir notificaciones en tiempo real.
PDGPCCII-17	HU12: Como usuario, quiero tener un chat para comunicación y colaboración.

### 6.2.3. Historias de usuarios

- Chat en tiempo real
- recibir notificaciones
- mención de miembros

### 6.2.4. Taskboard

Historia	To Do	In Progress	Done
Mención de miembros			X
Notificaciones en tiempo real			X
Comunicación y colaboración (chat)			X

#### **6.2.5. Daily scrum**

Las reuniones diarias continuaron a las 9:00 p.m., sirviendo como espacio de organización, resolución de bloqueos y aprendizaje colaborativo.

Durante la primera semana del sprint se trabajó en la estructura base del sistema de notificaciones y en el autocompletado para menciones de usuarios.

#### **6.2.6. Sprint review**

**Se demostró:**

- Menciones a miembros en comentarios con autocompletado por nombre de usuario.
- Sistema de notificaciones visuales cuando un usuario es mencionado o asignado.
- Versión inicial del chat con envío y recepción de mensajes en tiempo real.

**Feedback del PO:**

Solicitud de mejorar la interfaz del chat para que se sienta más “moderna”.

Sugerencia de incluir historial de notificaciones y no solo alertas instantáneas.

#### **6.2.7. Criterios de aceptación**

- Menciones de miembros funcionales en campos de comentarios o descripciones.
- Notificaciones en tiempo real con identificación clara del tipo de alerta.
- Chat visible y funcional entre usuarios logueados, con actualización automática.

- Uso correcto del contexto de autenticación para limitar interacciones por rol.

## 6.2.8. Resultados del sprint

### 6.2.8.1. Evidencias.

The image contains two screenshots of a Project Management Platform interface, likely demonstrating the use of context-aware authentication.

**Screenshot 1: Mensajes (Messages)**

This screenshot shows a messaging interface. The sidebar includes 'Dashboard General', 'PROYECTOS' (with items like 'Diseño de Interfaz de...', 'Desarrollo de Aplicaci...', etc.), 'GESTIÓN DE PROYECTO' (with items like 'Resumen', 'Backlog', 'Sprints', etc.), 'GENERAL' (with items like 'Equipos', 'Mensajes'), and 'Cerrar Sesión'. The main area is titled 'Mensajes' with a search bar 'Buscar chat...'. It lists messages from 'Juan Pérez' and 'Equipo de Desarrollo'. A message from 'Juan Pérez' at 03:57 p.m. says: 'Hola, ¿cómo va el desarrollo de la API? Estamos avanzando bien. Ya tenemos implementados los endpoints principales.' Another message from 'Juan Pérez' at 03:58 p.m. says: '¿Cuándo crees que estará lista para pruebas?' A message from 'Equipo de Desarrollo' at 03:59 p.m. says: 'Probablemente para el viernes. Hoy necesitamos implementar la autenticación y los tests.' A message from 'Juan Pérez' at 04:00 p.m. says: 'Perfecto. ¿Necesitas ayuda con algo?'. A message from 'Equipo de Desarrollo' at 04:01 p.m. says: '¿Cómo va el desarrollo de la API? Nueva versión del logo disponible p...'. At the bottom, there's a text input field 'Escribe un mensaje...' and some icons.

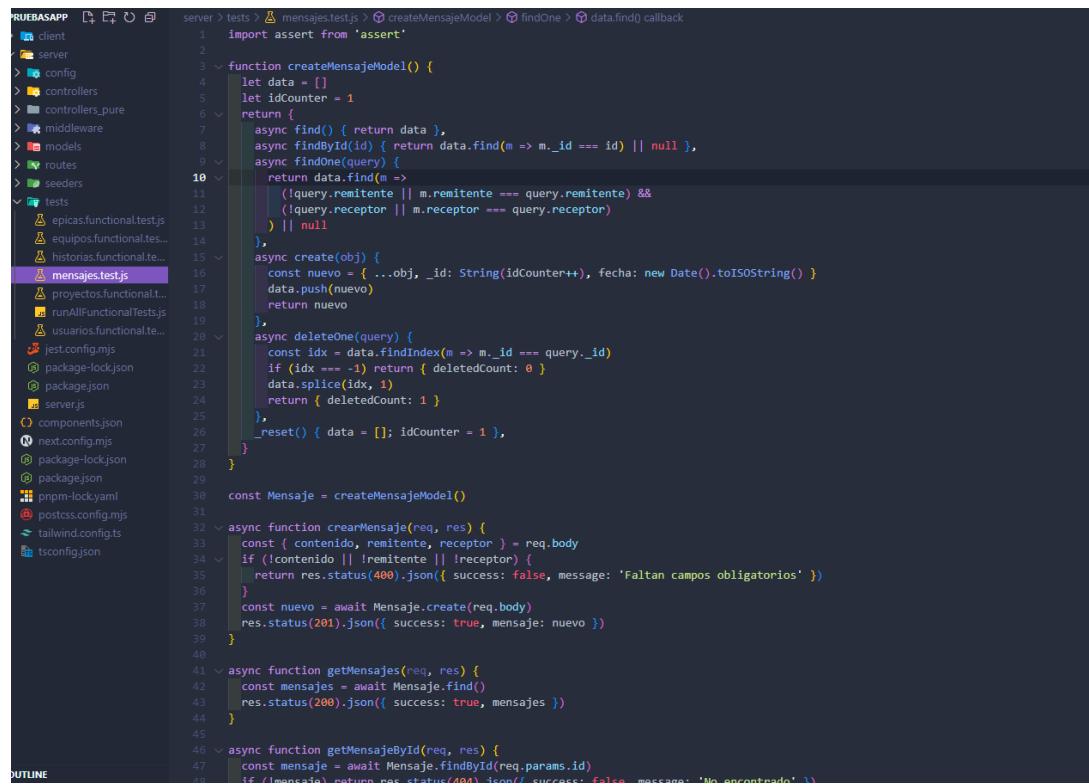
**Screenshot 2: Notificaciones (Notifications)**

This screenshot shows a notifications interface. The sidebar is identical to the first one. The main area is titled 'Notificaciones' with a button 'Marcar todas como leídas'. It lists several notifications:

- TodasNo leídasTareasProyectosEquiposMencionesSistema
- Nueva tarea asignada hace 30 minutos: Se te ha asignado la tarea 'Diseñar interfaz de usuario'. Ver detalles.
- Proyecto actualizado hace 2 horas: El proyecto 'Desarrollo de Aplicación Web' ha sido actualizado. Ver detalles.
- Te han mencionado en un comentario hace 1 día: Juan Pérez te ha mencionado en un comentario: '@admin ¿Puedes revisar esto?'. Ver detalles.
- Nuevo miembro en el equipo hace 3 días: Ana Martínez se ha unido al equipo 'Equipo de Desarrollo'. Ver detalles.
- Mantenimiento programado hace 5 días: El sistema estará en mantenimiento el próximo domingo de 2:00 AM a 4:00 AM.

At the bottom, there's a footer with 'Project Management Platform' (descripción: Una plataforma completa para la gestión colaborativa de proyectos), 'Características' (Gestión de Proyectos, Épicas e Historias), and 'Enlaces' (Documentación, Soporte, API).

### 6.2.8.2. Prueba de desarrollo.



```

RUEBASAPP
  ↴ client
  ↴ server
  > config
  > controllers
  > middleware
  > models
  > routes
  > seeders
  ↵ tests
    ↳ epicasfunctional.test.js
    ↳ equiposfunctional.test...
    ↳ historiasfunctional.te...
    ↳ mensajesfunctional.te...
    ↳ proyectosfunctional.t...
    ↳ runAllFunctionalTests.js
    ↳ usuariosfunctional.te...
  ↵ jest.config.mjs
  ↵ package-lock.json
  ↵ package.json
  ↵ server.js
  ↵ components.json
  ↵ next.config.mjs
  ↵ package-lock.json
  ↵ package.json
  ↵ pnpm-lockyaml
  ↵ postcss.config.mjs
  ↵ tailwind.config.js
  ↵ tsconfig.json

server > tests > mensajesfunctional.test.js > createMensajeModel() > findOne() > data.findOne() callback
  1   import assert from 'assert'
  2
  3   function createMensajeModel() {
  4     let data = []
  5     let idCounter = 1
  6   return {
  7     async find() { return data },
  8     async findById(id) { return data.find(m => m._id === id) || null },
  9     async findOne(query) {
 10       return data.find(m =>
 11         (!query.remitente || m.remitente === query.remitente) &&
 12         (!query.receptor || m.receptor === query.receptor)
 13       ) || null
 14     },
 15     async create(obj) {
 16       const nuevo = { ...obj, _id: String(idCounter++), fecha: new Date().toISOString() }
 17       data.push(nuevo)
 18       return nuevo
 19     },
 20     async deleteOne(query) {
 21       const idx = data.findIndex(m => m._id === query._id)
 22       if (idx === -1) return { deletedCount: 0 }
 23       data.splice(idx, 1)
 24       return { deletedCount: 1 }
 25     },
 26     _reset() { data = []; idCounter = 1 },
 27   }
 28 }
 29
 30 const Mensaje = createMensajeModel()
 31
 32 async function crearMensaje(req, res) {
 33   const { contenido, remitente, receptor } = req.body
 34   if (!contenido || !remitente || !receptor) {
 35     return res.status(400).json({ success: false, message: 'Faltan campos obligatorios' })
 36   }
 37   const nuevo = await Mensaje.create(req.body)
 38   res.status(201).json({ success: true, mensaje: nuevo })
 39 }
 40
 41 async function getMensajes(req, res) {
 42   const mensajes = await Mensaje.find()
 43   res.status(200).json({ success: true, mensajes })
 44 }
 45
 46 async function getMensajeById(req, res) {
 47   const mensaje = await Mensaje.findById(req.params.id)
 48   if (!mensaje) return res.status(404).json({ success: false, message: 'No encontrado' })
 49 }

```

### 6.2.9. Sprint retrospective

#### Fortalezas:

- Trabajo colaborativo para superar dificultades técnicas con tiempo real.
- Buena organización del equipo durante los daily scrums.
- Se cumplieron los objetivos principales del sprint.

#### Mejoras propuestas:

Dedicar más tiempo a pruebas UI antes de dar por finalizadas las tareas.

Usar herramientas de diseño para definir vistas antes del desarrollo.

## 6.3. Desarrollo del Sprint 3

### 6.3.1. Sprint planning

Durante la planificación del Sprint 3, se definió como prioridad la incorporación de herramientas que automatizaran la asistencia al usuario y ofrecieran mayor visibilidad del avance del proyecto.

El sprint tiene una duración de 28 días, con enfoque en:

- Implementar un chatbot funcional que pueda responder preguntas frecuentes y guiar al usuario dentro de la plataforma.
- Diseñar e integrar un módulo de análisis del estado del proyecto, con métricas visuales y resúmenes automáticos del progreso.

### 6.3.2. Sprint backlog

Código	Historia
PDGPCCII-18	HU13: Como usuario, quiero interactuar con un chatbot que me asista.
PDGPCCII-19	HU14: Como usuario, quiero ver un análisis automatizado del estado del proyecto.

### 6.3.3. Historias de usuarios

- Asistente virtual
- Análisis del proyecto

### 6.3.4. Taskboard

Historia	To Do	In Progress	Done
Chatbot de asistencia			X
Análisis del estado del proyecto			

### **6.3.5. Daily scrum**

Las reuniones diarias se continuaron realizando a las **9:00 p.m.**, con foco en seguir el avance, resolver bloqueos técnicos y compartir aprendizajes.

Durante la primera semana, se priorizó el análisis de posibles tecnologías para el chatbot, mientras que para el análisis del proyecto se inició la planificación de **indicadores clave (KPIs)** como tareas completadas, tiempos de entrega y avance por usuario.

### **6.3.6. Sprint review**

**Se demostró:**

- Versión inicial del chatbot capaz de responder preguntas frecuentes y redirigir al usuario a secciones específicas del sistema.
- Tablero de análisis con métricas básicas: porcentaje de tareas completadas, tareas por usuario y gráficos de progreso.

**Feedback del PO:**

- Solicitud de mejorar el lenguaje del chatbot para hacerlo más conversacional.
- Sugerencia de incluir alertas sobre proyectos atrasados en el panel de análisis.

### **6.3.7. Criterios de aceptación**

- Chatbot funcional accesible desde cualquier vista de usuario.
- Capacidad de responder a consultas predefinidas y guiar al usuario.
- Módulo de análisis con gráficos claros (barras, líneas o circulares).
- Visibilidad del estado global y por proyecto.
- Datos actualizados automáticamente a partir del backend.

## 6.3.8. Resultados del sprint

### 6.3.8.1. Evidencias.

The screenshot shows a Project Management Platform interface. On the left, there's a sidebar with sections like 'Dashboard General', 'PROYECTOS' (with items like 'Rediseño de Marca', 'Campaña de Marketing...', 'Desarrollo de API REST', etc.), 'GESTIÓN DE PROYECTO' (with 'Resumen', 'Backlog', 'Sprints', 'Tablero', 'Tareas', 'Calendario', 'Reportes'), and 'GENERAL' (with 'Equipos', 'Mensajes', 'Notificaciones'). The main area has a header 'Asistente IA' with a sub-header 'Conversación actual'. It shows a message from 'Asistente IA' saying '¡Hola! Soy tu asistente IA especializado en gestión de proyectos. Puedo ayudarte con: Análisis de progreso y métricas, Generación de reportes detallados, Optimización de procesos, Gestión de equipos, Planificación de sprints, ¿En qué puedo asistirte hoy?'. Below this are sections for 'Acciones Rápidas' (with buttons for '¿Cuál es el estado de mis proyectos?', 'Muestra mis tareas pendientes', 'Analiza el rendimiento de mi equipo', 'Genera un informe de progreso', '¿Cómo puedo mejorar la productividad?', 'Ayúdame a priorizar mis tareas', 'Identifica riesgos en el proyecto', 'Planifica el próximo sprint') and a search bar 'Escribe tu pregunta o solicitud...'. At the bottom, there's a footer with 'Project Management Platform', 'Características' (Gestión de Proyectos, Épicas e Historias, Tácticas Scrum u Kanban), and 'Enlaces' (Documentación, Soporte).

### 6.3.8.2. Prueba de desarrollo.

The screenshot shows a code editor with a file named 'chatia.test.js' open. The file contains Jest test cases for a 'Chatia' model. The code includes imports, a test suite for creating a model, methods for finding, finding one, updating, and deleting documents, and a function for creating a new document. The code uses MongoDB's ObjectId and Date types. The file is part of a project structure with other files like 'client', 'config', 'controllers', 'models', 'routes', 'seeders', and 'tests'.

```
server > tests > chatia.test.js > run > check('Obtener chatia por ID válido') callback
1 import assert from 'assert'
2
3 function createChatiaModel() {
4   let data = []
5   let idCounter = 1
6   return {
7     async find() { return data },
8     async findById(id) { return data.find(c => c._id === id) || null },
9     async findOne(query) {
10       return data.find(c =>
11         (query.participantes || JSON.stringify(c.participantes.sort()) === JSON.stringify(query.participantes.sort())))
12       ) || null
13     },
14     async create(obj) {
15       const nuevo = { ...obj, _id: String(idCounter++), creadoEn: new Date().toISOString() }
16       data.push(nuevo)
17       return nuevo
18     },
19     async updateOne(query, update) {
20       const idx = data.findIndex(c => c._id === query._id)
21       if (idx === -1) return { nModified: 0 }
22       data[idx] = { ...data[idx], ...update }
23       return { nModified: 1 }
24     },
25     async deleteOne(query) {
26       const idx = data.findIndex(c => c._id === query._id)
27       if (idx === -1) return { deletedCount: 0 }
28       data.splice(idx, 1)
29       return { deletedCount: 1 }
30     },
31     _reset() { data = []; idCounter = 1 },
32   }
33 }
34
35 const Chatia = createChatiaModel()
36
37 async function crearChatia(req, res) {
38   const { participantes } = req.body
39   if (!participantes || !Array.isArray(participantes) || participantes.length < 2) {
40     return res.status(400).json({ success: false, message: 'Se requieren al menos 2 participantes' })
41   }
42   if (await Chatia.findOne({ participantes })) {
43     return res.status(400).json({ success: false, message: 'Chat ya existe entre estos participantes' })
44   }
45   const nuevo = await Chatia.create(req.body)
46   res.status(201).json({ success: true, chatia: nuevo })
47 }
```

### **6.3.9. Sprint retrospective**

Puntos positivos:

- integración del componente de Inteligencia artificial
- el sistema ofrece una mejor experiencia al usuario gracias a la asistencia automática

Áreas de mejora

- Optimizar la precisión del análisis
- Incluir mayor personalización de respuestas del asistente según el perfil del usuario

## CAPÍTULO 7

### PRUEBAS DE SOFTWARE

#### **7.1. Plan de Pruebas**

La validación del correcto funcionamiento del sistema se ha llevado a cabo mediante un plan integral de pruebas, que incluye pruebas unitarias, de integración y pruebas end-to-end (E2E). Estas pruebas tienen como objetivo garantizar la calidad, confiabilidad y robustez del sistema en cada una de sus capas, así como verificar la integración efectiva de las funcionalidades basadas en inteligencia artificial.

##### **1. Plan de Pruebas (unitarias, integración y E2E)**

###### **a. Pruebas Unitarias**

Se realizaron pruebas unitarias sobre funciones críticas del backend, especialmente en los controladores y servicios de tareas, autenticación y gestión de usuarios. Se utilizó **Jest** para validar que cada módulo individual ejecutara correctamente su lógica de negocio.

###### **Objetivo:**

- Validar funciones puras y controladores del backend.
- Comprobar respuestas esperadas a entradas válidas e inválidas

###### **b. Pruebas de Integración**

Se ejecutaron pruebas utilizando **Postman** para verificar el correcto funcionamiento de las rutas y APIs RESTful expuestas por el backend (Express.js). Estas pruebas garantizaron que los distintos módulos del sistema interactúen adecuadamente entre sí.

###### **Casos evaluados:**

- Creación y actualización de proyectos.
- Asignación de tareas y cambios de estado.
- Flujo de autenticación (login, registro).
- Comunicación entre frontend y backend vía API.

### c. Pruebas End-to-End (E2E)

Se utilizaron pruebas E2E automatizadas con **Cypress**, simulando el comportamiento real de un usuario en la interfaz. Esto permitió validar los flujos completos del sistema desde la perspectiva del cliente final.

#### **Flujos probados con Cypress:**

- Registro e inicio de sesión de usuario.
- Creación de proyectos épicas, historias y tareas.
- Movimiento de tareas en el tablero Kanban.
- Visualización de métricas en el dashboard.

## 7.2. Instrumentos de usabilidad

### **Heurísticas del Nielsen**

## Evaluacion 1

Resultados encuesta heurística 1:

Nº	Principio	Descripción	Escala (1-5)	Observaciones
1	Visibilidad del estado del sistema	El sistema mantiene informado al usuario	5	Siempre supe en qué parte del proceso me encontraba. Excelente feedback visual.
2	Correspondencia con el mundo real	Lenguaje y conceptos conocidos por el usuario	4	La mayoría de términos son comprensibles, aunque algunos botones podrían renombrarse.
3	Control y libertad del usuario	Posibilidad de cancelar, deshacer acciones	3	Me costó encontrar cómo deshacer un cambio en el tablero.
4	Consistencia y estándares	Elementos coherentes en toda la aplicación	4	Buen nivel de consistencia visual, aunque algunos íconos cambian entre módulos.
5	Prevención de errores	Diseño que evita errores antes que ocurran	3	En general bien, pero faltan confirmaciones al editar o eliminar tareas.
6	Reconocimiento antes que recuerdo	Opciones visibles sin necesidad de memorizar	5	Muy fácil de usar, todos los accesos están bien ubicados y claros.
7	Flexibilidad y eficiencia de uso	Accesos rápidos y funciones para usuarios expertos	3	Me hubiera gustado tener filtros avanzados o atajos de teclado.
8	Estética y diseño minimalista	Interfaz limpia, sin información irrelevante	5	Excelente diseño visual, sin elementos innecesarios.
9	Manejo de errores	Mensajes claros que indiquen causas y soluciones	3	Algunos mensajes de error son poco detallados.
10	Ayuda y documentación	Información accesible para asistir al usuario	2	No encontré un botón de ayuda ni instrucciones iniciales.
		TOTAL	37	

## Evaluación 2

Resultados encuesta heurística 2:

Nº	Principio	Descripción	Escala (1-5)	Observaciones
1	Visibilidad del estado del sistema	El sistema mantiene informado al usuario	4	La mayoría de acciones muestran retroalimentación clara, aunque no todas.
2	Correspondencia con el mundo real	Lenguaje y conceptos conocidos por el usuario	3	Algunos términos técnicos podrían ser más amigables o explicados.
3	Control y libertad del usuario	Posibilidad de cancelar, deshacer acciones	4	Se puede cancelar formularios fácilmente, pero no siempre es intuitivo.
4	Consistencia y estándares	Elementos coherentes en toda la aplicación	3	Algunos botones y menús cambian de estilo en diferentes secciones.
5	Prevención de errores	Diseño que evita errores antes que ocurran	2	Pude eliminar una tarea sin confirmación; faltan validaciones claras.
6	Reconocimiento antes que recuerdo	Opciones visibles sin necesidad de memorizar	4	Los iconos están bien diseñados y es fácil identificar funciones.
7	Flexibilidad y eficiencia de uso	Accesos rápidos y funciones para usuarios expertos	2	No encontré atajos o configuraciones avanzadas para acelerar mi trabajo.
8	Estética y diseño minimalista	Interfaz limpia, sin información irrelevante	4	El diseño es moderno y se entiende bien, aunque algunas secciones se ven más cargadas.
9	Manejo de errores	Mensajes claros que indiquen causas y soluciones	3	Algunos errores no explican claramente qué pasó ni cómo solucionarlo.
10	Ayuda y documentación	Información accesible para asistir al usuario	2	No encontré una sección de ayuda ni tutorial al ingresar por primera vez.
		TOTAL	31	

s.

Resultados encuesta de usabilidad 1:

Ítem	Enunciado	Puntaje (1-5)
1	El proceso de registro e inicio de sesión fue rápido, claro y seguro.	4
2	Crear un nuevo proyecto con título, descripción y miembros fue intuitivo.	5
3	Comprendí fácilmente cómo navegar por las diferentes secciones.	4
4	La interfaz gráfica es agradable y coherente.	5
5	Pude gestionar mis tareas eficientemente.	4
6	El tablero Kanban fue claro y funcional.	5
7	El sistema respondió rápidamente	4
	a mis acciones.	
8	El panel de control muestra el progreso del proyecto claramente.	5
9	El sistema evitó errores mediante validaciones o confirmaciones.	3
10	Usaría esta plataforma frecuentemente para trabajar en proyectos.	5

Suma de pares: 25 Suma de impares: 19

Resultados encuesta de usabilidad 2:

Ítem	Enunciado	Puntaje (1-5)
1	El proceso de registro e inicio de sesión fue rápido, claro y seguro.	3
2	Crear un nuevo proyecto con título, descripción y miembros fue intuitivo.	4
3	Comprendí fácilmente cómo navegar por las diferentes secciones.	3
4	La interfaz gráfica es agradable y coherente.	4
5	Pude gestionar mis tareas eficientemente.	3
6	El tablero Kanban fue claro y funcional.	4
7	El sistema respondió rápidamente a mis acciones.	3
8	El panel de control muestra el progreso del proyecto claramente.	4
9	El sistema evitó errores mediante validaciones o confirmaciones.	3
10	Usaría esta plataforma frecuentemente para trabajar en proyectos.	4

Suma de pares: 20

Suma de impares: 15

### 3. Interpretación de Resultados

Usuario	Promedio Heurísticas (0-5)	Total SUS (1-50)	SUS Escalado (0-100)	Interpretación Heurísticas	Interpretación SUS
Usuario 1	3.7	44	88.0	Buen nivel	Excelente usabilidad
Usuario 2	3.1	35	70.0	Necesita mejoras en varios principios	Buena usabilidad

Los resultados reflejan una buena percepción global del sistema evaluado, con niveles de usabilidad que varían entre "buena" y "excelente". Las heurísticas muestran que el diseño de la plataforma es visualmente efectivo, aunque requiere ajustes en aspectos críticos como validación de acciones, asistencia contextual y funciones avanzadas. Implementar las mejoras señaladas por ambos usuarios no sólo consolida la experiencia positiva actual, sino que elevará aún más el nivel de eficiencia, accesibilidad y confianza del sistema.

## **CONCLUSIONES**

1. Optimización de la gestión de proyectos: La implementación de la plataforma web basada en el stack MERN permitió centralizar y automatizar tareas clave como la asignación de actividades, seguimiento del progreso y comunicación entre los equipos, superando las limitaciones del sistema anterior basado en hojas de cálculo y correos electrónicos.
2. Valor agregado mediante IA: La integración de funcionalidades de inteligencia artificial, como el asistente virtual y la clasificación automática de tareas, mejoró significativamente la toma de decisiones, la priorización del trabajo y la productividad del equipo. Esto demuestra el potencial transformador de la IA aplicada a la gestión organizacional.
3. Alta viabilidad técnica, económica y operativa: El proyecto demostró ser viable en todos sus aspectos. Se utilizaron tecnologías libres y de código abierto, lo que redujo los costos sin comprometer la calidad. Además, la arquitectura del sistema, junto con su diseño centrado en el usuario, garantiza facilidad de uso, adaptabilidad y escalabilidad.
4. Desarrollo ágil exitoso: La metodología Scrum facilitó una planificación estructurada y una entrega incremental de valor en cada sprint. El equipo logró integrar correctamente los módulos previstos y responder de manera efectiva a los retos

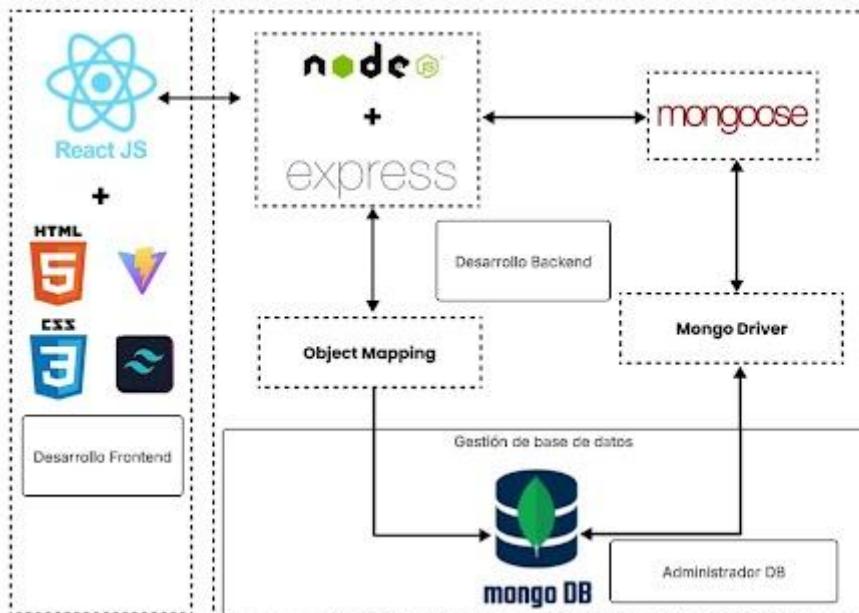
técnicos y de coordinación, evidenciando un proceso de mejora continua en cada iteración.

## RECOMENDACIONES

1. **Mejorar la precisión del asistente virtual:** Se recomienda continuar con el entrenamiento del modelo de procesamiento de lenguaje natural (NLP), ampliando su base de conocimiento y ajustando las respuestas a diferentes perfiles de usuario para brindar una asistencia más personalizada y eficiente.
2. **Incorporar métricas de desempeño avanzadas:** Es aconsejable añadir indicadores clave de rendimiento (KPIs) que permitan evaluar en tiempo real la productividad del equipo, los tiempos de entrega y el cumplimiento de objetivos, lo cual reforzará la toma de decisiones basada en datos.
3. **Desarrollar una versión móvil optimizada:** Dado que muchos usuarios acceden desde dispositivos móviles, se sugiere implementar una versión específica para smartphones que ofrezca una experiencia más fluida, rápida y adaptada al uso en movimiento.
4. **Fortalecer la seguridad del sistema:** Aunque se implementaron medidas básicas de seguridad, es recomendable realizar auditorías periódicas, implementar autenticación multifactor y políticas de acceso más granulares para proteger los datos y prevenir vulnerabilidades futuras.

## ANEXOS

### Anexo 01. Manual Técnico



### Anexo 02. Enlace GitHub

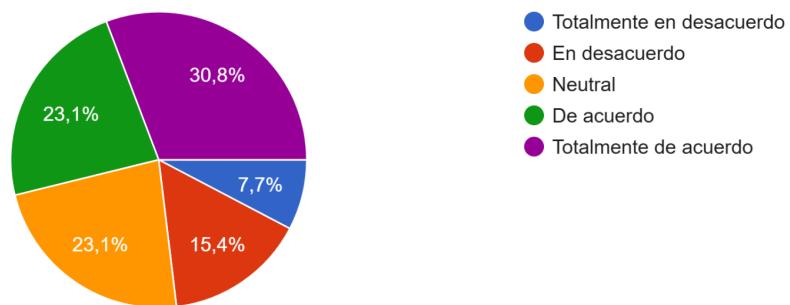
<https://github.com/emilpastrana/TallerProyectosII>

### Anexo 03. Manual de Usuario

1.

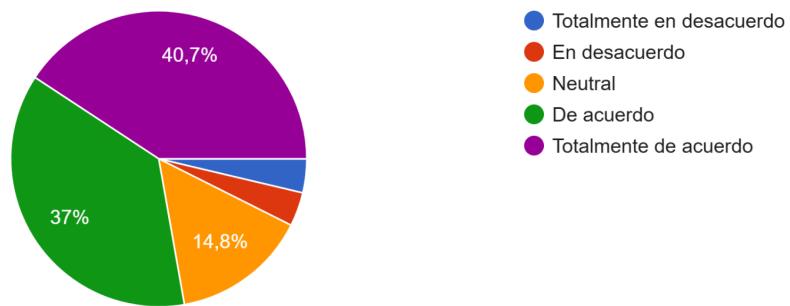
1. El proceso de registro e inicio de sesión fue rápido, claro y seguro.

13 respuestas

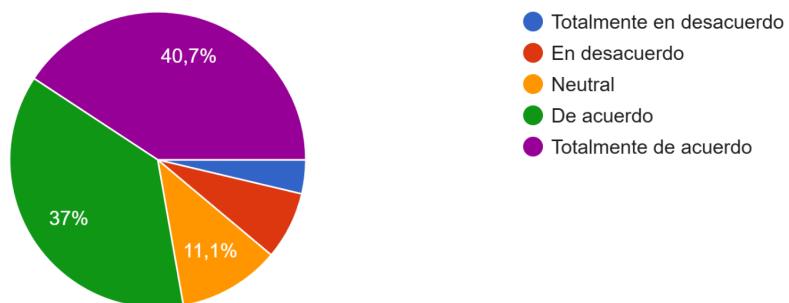


2. Crear un nuevo proyecto con título, descripción y miembros fue intuitivo.

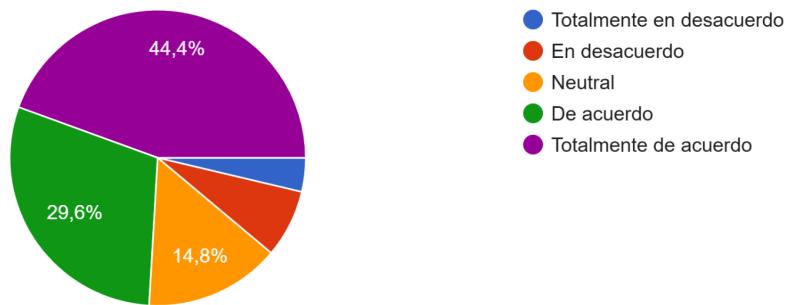
27 respuestas



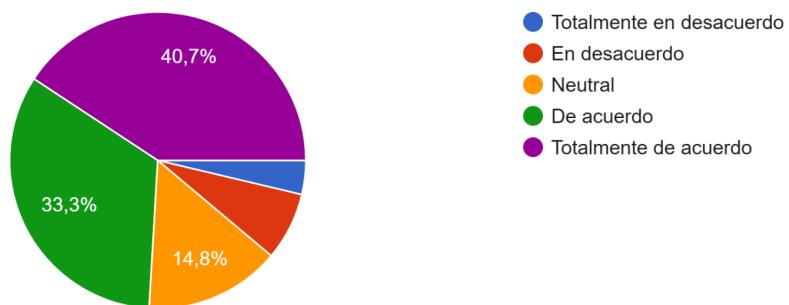
3. Comprendí fácilmente cómo navegar por las diferentes secciones de la plataforma  
27 respuestas



4. La interfaz gráfica de la plataforma es agradable y coherente.  
27 respuestas

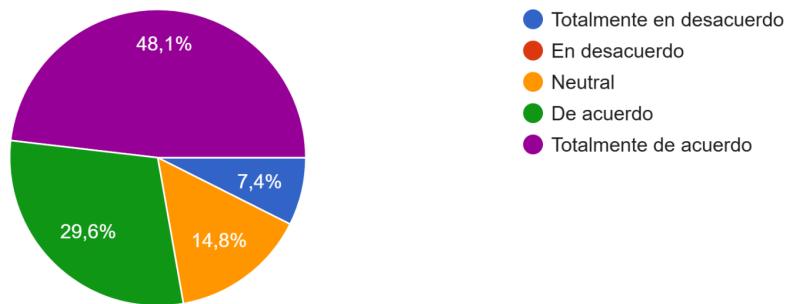


5. La plataforma me permitió gestionar mis tareas de forma eficiente  
27 respuestas



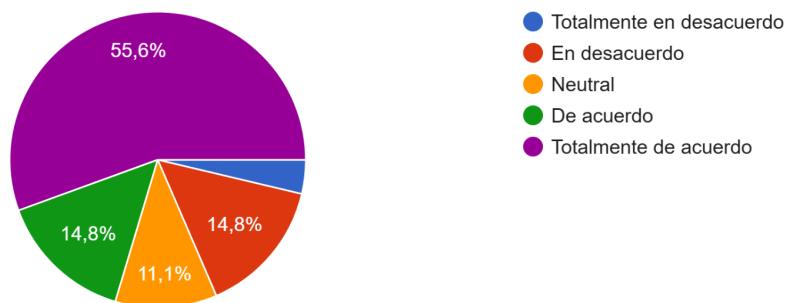
6. El tablero Kanban permite visualizar y mover tareas (pendiente, en proceso, finalizado) de forma clara.

27 respuestas



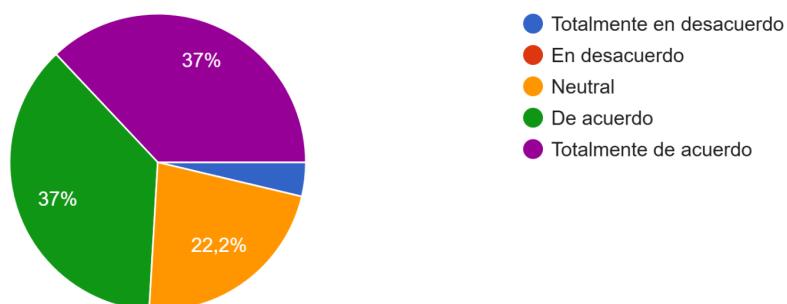
7. El sistema responde rápidamente a las acciones que realizo.

27 respuestas



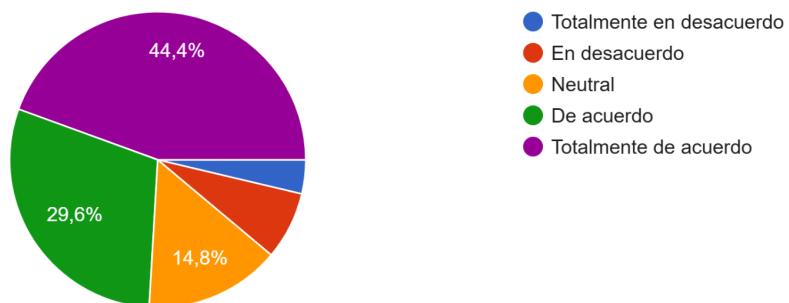
8. El panel de control muestra de manera comprensible el progreso general del proyecto (estadísticas y avances).

27 respuestas



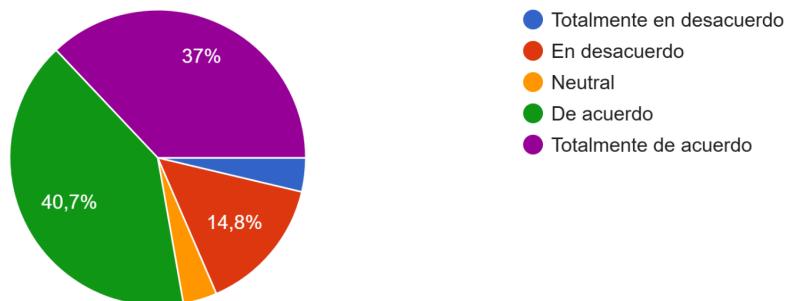
9. Pude utilizar las funciones principales sin necesidad de ayuda externa.

27 respuestas



10. El sistema evitó errores o acciones equivocadas con validaciones o confirmaciones claras.

27 respuestas



11. Usaría esta plataforma frecuentemente para trabajar en proyectos colaborativos.

26 respuestas

