**UNIVERSIDAD CONTINENTAL**

**FACULTAD DE INGENIERÍA**

**ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**



**PROYECTO**

**“Plataforma De Gestión De Proyectos Colaborativos Con Integración De Inteligencia Artificial”**

# PORTADA

**PRESENTADO POR:**

| **APELLIDOS Y NOMBRES** | **CÓDIGO** |
| --- | --- |
| **Alderete Condor Ingrid Yeanella** | **73053791** |
| **Lucas Eulogio Jaci** | **72617519** |
| **Orellana Marcas Alan** | **74303152** |
| **Taipe Martinez Sofia Gladys** | **74900838** |
| **Villalva Quispe Carlos** | **75524786** |

**ASESOR:**

**Daniel Gamarra Moreno**

**HUANCAYO – PERÚ**

**2025**

# LISTA DE CONTENIDO

[PORTADA 1](#_heading=h.u9wjadtqho2z)

[LISTA DE CONTENIDO 2](#_heading=h.1xn241a0n9g1)

[LISTA DE TABLAS 7](#_heading=h.3jesd9nnos6f)

[LISTA DE FIGURAS 8](#_heading=h.b6izk8p6747f)

[CAPÍTULO 1 9](#_heading=h.2ifdysje46mn)

[PLANTEAMIENTO DEL ESTUDIO 9](#_heading=h.2pyqz3wif7wt)

[1.1.](#_heading=h.nbyo402cc9qw) Aspectos Generales de la Empresa 9

[1.1.1.](#_heading=h.wbkb5rb7nir6) Organigrama 9

[1.1.2.](#_heading=h.4bcvbfbmklt6) Misión y visión 9

[1.2.](#_heading=h.gxg63vixmxcs) Diagnóstico del Problema 9

[1.3.](#_heading=h.nghiwhom6q3x) Procesos de la Empresa 9

[1.4.](#_heading=h.j37ordjat8n8) Oportunidad Encontrada 9

[1.5.](#_heading=h.pp8ehy8gjd10) Detalles del Proyecto 9

[CAPÍTULO 2 10](#_heading=h.gk13oq5ohchu)

[ESTUDIO DE FACTIBILIDAD 10](#_heading=h.2h73wlhn8xsh)

[2.1.](#_heading=h.w2ghw4931xv3) Alternativas de Solución 10

[2.2.](#_heading=h.amah7ce4kpj1) Factibilidad Técnica 10

[2.2.1.](#_heading=h.yd861r5f22s1) … 10

[2.2.2.](#_heading=h.n5htlfsonhxm) … 10

[2.3.](#_heading=h.hrayji01drn) Factibilidad Económica 10

[2.3.1.](#_heading=h.6vybh8vb91wu) Gastos generales 10

[2.4.](#_heading=h.8vkk4zjn3hv) Factibilidad Operacional 10

[2.4.1.](#_heading=h.42xlfq1imw5s) Sistemas de ventas 10

[CAPÍTULO 3 11](#_heading=h.o5gdhp9ou3rm)

[ANÁLISIS DE REQUERIMIENTOS 11](#_heading=h.r558mk5bwnt6)

[3.1.](#_heading=h.pq9j1jebq2zw) Metas del Sistema de Información 11

[3.2.](#_heading=h.7nlal03s5ber) Requisitos del Sistema 11

[3.2.1.](#_heading=h.k5moxn2ciabh) Requerimientos funcionales 11

[3.2.2.](#_heading=h.lm56mg5cqwqm) Requerimientos no funcionales 11

[3.3.](#_heading=h.qnsswb8p38gf) Identificación de Actores del Sistema 11

[3.3.1.](#_heading=h.8y4ic0anu0fa) … 11

[CAPÍTULO 4 12](#_heading=h.rp1x3lybzkv5)

[PLANIFICACIÓN DEL PROYECTO 12](#_heading=h.h8qn6t3hux45)

[4.1.](#_heading=h.jgjnx84ejarb) Definición de Roles de Trabajo 12

[4.1.1.](#_heading=h.lcw44scclw7l) Product owner 12

[4.1.2.](#_heading=h.6niu1hfvn8i4) Scrum master 12

[4.1.3.](#_heading=h.otwui59mzf9x) Team member 12

[4.1.4.](#_heading=h.jxk0pyt09w9u) Tester 12

[4.2.](#_heading=h.7kdsfqee0xam) Product Backlog 12

[4.3.](#_heading=h.yrucpegcjr03) Sprint Backlog 12

[4.3.1.](#_heading=h.z4ylxx148t6l) Sprint 1 12

[4.3.2.](#_heading=h.v86mg81xxkck) Sprint 2 12

[4.3.3.](#_heading=h.7oej9vagbn5a) Sprint 3 12

[4.3.4.](#_heading=h.5mwvc2jjzi86) Sprint 4 12

[4.3.5.](#_heading=h.7h0yo597sbzc) Sprint 5 13

[4.4.](#_heading=h.jl6bmf1hywme) Planificación de Sprints 13

[4.4.1.](#_heading=h.1wmgv1hd08b0) Historias de usuario 13

[4.4.2.](#_heading=h.xzw9y1vrtqls) Priorización de historias de usuario 13

[4.5.](#_heading=h.bvfuv2hd479a) Cronograma de Actividades 13

[4.6.](#_heading=h.w2bdxcdwma50) Gestión de Riesgos 13

[CAPÍTULO 5 14](#_heading=h.sgbz7b9ixh82)

[DISEÑO DEL SISTEMA DE INFORMACIÓN 14](#_heading=h.dpnuu9kx7sip)

[5.1.](#_heading=h.k0fp72u99o40) Diseño de Diagramas UML 14

[5.1.1.](#_heading=h.6twhaswmjc03) Diagramas de casos de uso 14

[5.1.2.](#_heading=h.ko6hvnrz74yi) Diagramas de secuencia 14

[5.1.3.](#_heading=h.e9ytqalke32b) Diagramas de colaboración 14

[5.1.4.](#_heading=h.kvqgd1axaxc8) Diagramas de clases 14

[5.2.](#_heading=h.7narh8qxjdec) Diseño de Base de Datos 14

[5.2.1.](#_heading=h.8gp24px9q6a3) Diseño conceptual (E/R) 14

[5.2.2.](#_heading=h.egeq0g7irjq1) Diseño lógico 14

[5.2.3.](#_heading=h.9od3gx6f3q00) Diseño físico 14

[5.2.4.](#_heading=h.ooge0bpsyvnz) Modelado de base de datos 14

[5.3.](#_heading=h.npesv5xtavvb) Diseño de Interfaces Básicas 14

[5.3.1.](#_heading=h.20bwu9dfsoup) Acceso login 15

[5.3.2.](#_heading=h.ay5qkr945sb7) Interfaz … 15

[CAPÍTULO 6 16](#_heading=h.jxsua7nzq8qr)

[CODIFICACIÓN DEL SOFTWARE 16](#_heading=h.h93bwnau88t2)

[6.1.](#_heading=h.l7oa5as49vxi) Desarrollo del Sprint 1 16

[6.1.1.](#_heading=h.lulhzakjyql1) Sprint planning 16

[6.1.2.](#_heading=h.glovcnmglylx) Sprint backlog 16

[6.1.3.](#_heading=h.wy3sr910vg5g) Historias de usuarios 16

[6.1.4.](#_heading=h.7ebf9tfo7jhs) Taskboard 16

[6.1.5.](#_heading=h.is4epmgsvy3z) Daily scrum 16

[6.1.6.](#_heading=h.4tmscburp1sf) Sprint review 16

[6.1.7.](#_heading=h.51nvzv2fkt70) Criterios de aceptación 16

[6.1.8.](#_heading=h.2er77ou0b92r) Resultados del sprint 16

[6.1.8.1.](#_heading=h.220hivugsh8z) Evidencias. 16

[6.1.8.2.](#_heading=h.ds3aknpj20p) Prueba de desarrollo. 16

[6.1.8.3.](#_heading=h.ah60eex47pdh) …. 17

[6.1.9.](#_heading=h.47hik99bgs90) Sprint retrospective 17

[6.2.](#_heading=h.qjw5941b1ztb) Desarrollo del Sprint 2 17

[6.2.1.](#_heading=h.xspjgrr9cxn4) Sprint planning 17

[6.2.2.](#_heading=h.4y3n2wcqxexy) Sprint backlog 17

[6.2.3.](#_heading=h.qqo6hvv6xevd) Historias de usuarios 17

[6.2.4.](#_heading=h.x5exdraimhkc) Taskboard 17

[6.2.5.](#_heading=h.dyshzzeqq5pn) Daily scrum 17

[6.2.6.](#_heading=h.y577zo9w2ha0) Sprint review 17

[6.2.7.](#_heading=h.vqu0wbptctcn) Criterios de aceptación 17

[6.2.8.](#_heading=h.exf66mcqszky) Resultados del sprint 17

[6.2.8.1.](#_heading=h.rjan3du6egf5) Evidencias. 17

[6.2.8.2.](#_heading=h.2b6m9bqrk1v8) Prueba de desarrollo. 18

[6.2.8.3.](#_heading=h.6qfvolie62sz) …. 18

[6.2.9.](#_heading=h.v1deui1lv07g) Sprint retrospective 18

[6.3.](#_heading=h.venes0l1yzf3) Desarrollo del Sprint 3 18

[6.3.1.](#_heading=h.771juml74fjd) Sprint planning 18

[6.3.2.](#_heading=h.ammqblbqhrvj) Sprint backlog 18

[6.3.3.](#_heading=h.g85m59omc3w0) Historia s de usuarios 18

[6.3.4.](#_heading=h.4ji0b5e5nup) Taskboard 18

[6.3.5.](#_heading=h.lsgqykg4lm1z) Daily scrum 18

[6.3.6.](#_heading=h.hczbogkpm36e) Sprint review 18

[6.3.7.](#_heading=h.hnlxwt5ujh5u) Criterios de aceptación 18

[6.3.8.](#_heading=h.fxn2so4xgeja) Resultados del sprint 18

[6.3.8.1.](#_heading=h.kmrcfmfxvi2i) Evidencias. 19

[6.3.8.2.](#_heading=h.fazhze78lkj) Prueba de desarrollo. 19

[6.3.8.3.](#_heading=h.7wcjfesz9h7j) …. 19

[6.3.9.](#_heading=h.u9y1igq0f19k) Sprint retrospective 19

[6.4.](#_heading=h.nmprylpkuycv) Desarrollo del Sprint 4 19

[6.4.1.](#_heading=h.zfjx0hxvdsc8) Sprint planning 19

[6.4.2.](#_heading=h.u26411tlxyub) Sprint backlog 19

[6.4.3.](#_heading=h.e5eh20nz9x0b) Historias de usuarios 19

[6.4.4.](#_heading=h.yot241qk2zc) Taskboard 19

[6.4.5.](#_heading=h.t0q2iie2idbq) Daily scrum 19

[6.4.6.](#_heading=h.j8oy1pu198vu) Sprint review 19

[6.4.7.](#_heading=h.ql45qsh3abro) Criterios de aceptación 19

[6.4.8.](#_heading=h.hminl77lt463) Resultados del sprint 20

[6.4.8.1.](#_heading=h.5fxr2qve36p6) Evidencias. 20

[6.4.8.2.](#_heading=h.30o3b8huie5c) Prueba de desarrollo. 20

[6.4.8.3.](#_heading=h.m7bkvodqo5yz) …. 20

[6.4.9.](#_heading=h.aoyf5humkvx6) Sprint retrospective 20

[CAPÍTULO 7 21](#_heading=h.8oynq6izfdqz)

[PRUEBAS DE SOFTWARE 21](#_heading=h.7ecdoumdcbxf)

[7.1.](#_heading=h.rfk85oxm6xs5) Plan de Pruebas 21

[CONCLUSIONES 22](#_heading=h.lu63xlcjge0a)

[RECOMENDACIONES 23](#_heading=h.zq1f3g89rnr)

[ANEXOS 24](#_heading=h.9mo5yihk0qhj)

[Anexo 01. Manual Técnico 25](#_heading=h.brt9sons8p2x)

[Anexo 02. Manual de Usuario 26](#_heading=h.59pm4hakys4u)

# LISTA DE TABLAS

Tabla 1: Gastos generales

Tabla 2: Gastos específicos

Tabla 3: Roles de trabajo.Elaboración propia

Tabla 4: Product Backlog.Elaboración propia

Tabla 5: Sprint 1.Elaboración propia

Tabla 6: Sprint 2.Elaboración propia

Tabla 7: Sprint 3.Elaboración propia

Tabla 8: Sprint 4.Elaboración propia

Tabla 9: Historias de usuario.Elaboración propia

Tabla 10: Priorización De historias

Tabla 11: Cronograma.Elaboración propia

Tabla 12: Riesgos.Elaboración propia

Tabla 13: Tipos de pruebas aplicadas

Tabla 14: Pruebas unitarias

Tabla 15: Pruebas de aceptación

# LISTA DE FIGURAS

Figura 1. Organigrama de la Universidad Continental. Nota: Tomado de Información interna Universidad Continental (2017). Plan de Seguridad Institucional. Huancayo-Perú.

Figura 2. Procesos de la empresa

Figura 3. Casos de uso

Figura 4. Diagramas de secuencia.Elaboración propia

Figura 5. Diagramas de colaboración.Elaboración propia

Figura 6. Diagramas de clases.Elaboración propia

Figura 7. Diseño conceptual.Elaboración propia

Figura 8. Diseño lógico.Elaboración propia

Figura 9. Diseño Físico.Elaboración propia

Figura 10. Modelado BDD.Elaboración propia

Figura 11. Interfaz de inicio

Figura 12. Interfaz de inicio de sesión

Figura 13. Interfaz de crear cuenta

Figura 14. Interfaz de página principal

Figura 15. Interfaz de crear un proyecto

Figura 16. Interfaz de tablero kanban

Figura 17. Interfaz de editar tarea

Figura 18. Componentes del sprint 1 en Jira

Figura 19. Resultados del sprint 1

Figura 20. Componentes del sprint 2 en Jira

Figura 21. Componentes del sprint 3 en Jira

Figura 22. Resultados del sprint 3 en Jira

Figura 23. Componentes del sprint 4 en Jira

Figura 24. Resultados del sprint 4

Figura 25. SEO (Optimización para Motores de Búsqueda)

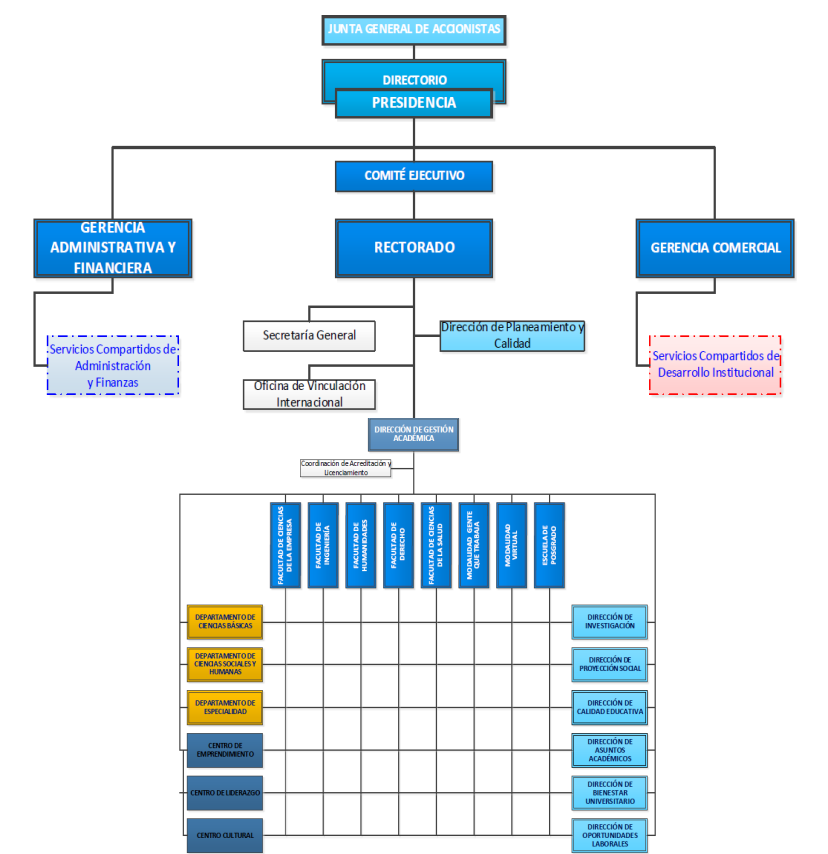
# CAPÍTULO 1

# PLANTEAMIENTO DEL ESTUDIO

## Aspectos Generales de la Empresa

Fundada el 29 de junio de 1998, la Universidad Continental ha crecido rápidamente, consolidándose como una de las instituciones educativas más importantes del sur del Perú. Su evolución refleja un compromiso constante con la calidad académica y la innovación en la enseñanza.

### Organigrama



*Figura 1. Organigrama de la Universidad Continental. Nota: Tomado de Información interna Universidad Continental (2017). Plan de Seguridad Institucional. Huancayo-Perú.*

### Misión y visión

La misión de la Universidad Continental es formar profesionales íntegros y competentes, preparados para afrontar los desafíos del mundo laboral. Su visión se centra en ser una universidad de referencia nacional e internacional, destacándose por su enfoque en la investigación y la responsabilidad social.

## Diagnóstico del Problema

Actualmente, dentro de la Universidad Continental se identificaron las dificultades en la gestión eficiente de los proyectos colaborativos entre estudiantes y docentes, especialmente en lo que respecta al seguimiento de tareas, coordinación, trazabilidad de entregables, asignación de roles, y análisis de desempeño de los usuarios. El crecimiento de la matrícula y la modalidad semipresencial o virtual han generado una necesidad de hacer uso de herramientas digitales que mejoren la comunicación, seguimiento y productividad en la gestión de proyectos académicos y de investigación.

## Procesos de la Empresa

### Procesos Estratégicos

Estos procesos establecen la dirección y visión institucional. Son clave para la toma de decisiones a largo plazo:

* **Planeamiento Estratégico:** Define los objetivos institucionales, metas y planes de desarrollo.
* **Sistemas Integrados de Gestión:** Asegura el cumplimiento de estándares de calidad, seguridad y mejora continua.
* **Comunicación e Imagen:** Gestiona la identidad institucional y la relación con la comunidad externa.

### Procesos Misionales

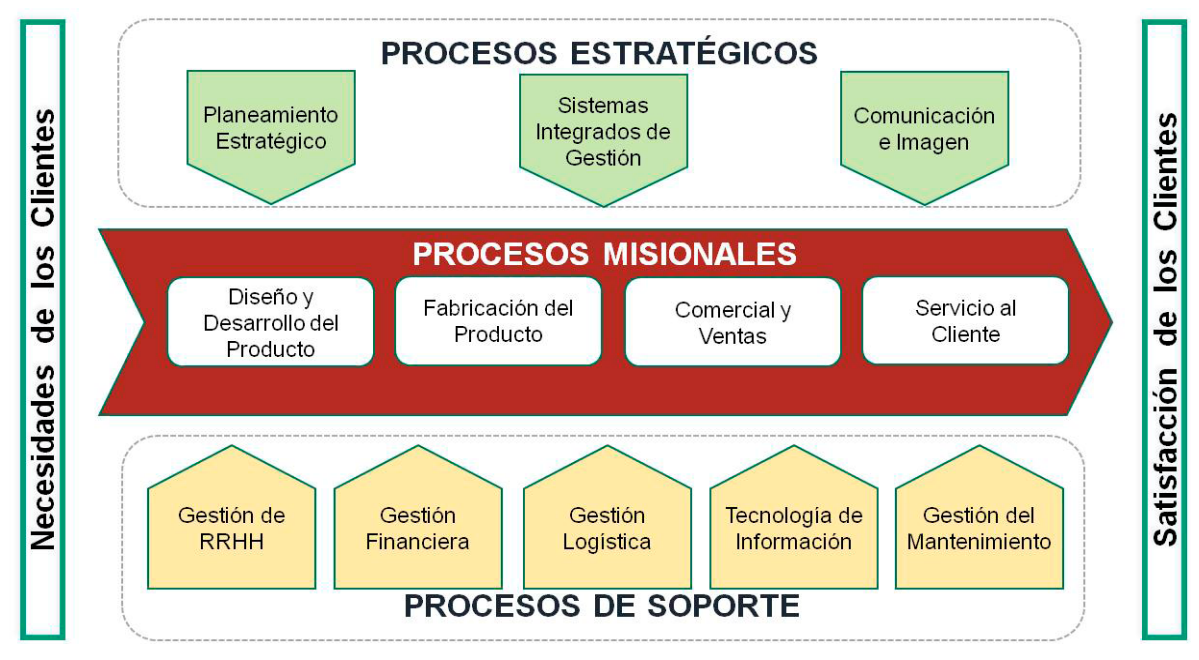
Representan la razón de ser de la universidad, es decir, sus actividades principales:

* **Diseño y Desarrollo del Producto (educativo):** Planificación y actualización de mallas curriculares y programas académicos.
* **Fabricación del Producto:** Equivale a la entrega del servicio educativo (dictado de clases, investigación, extensión).
* **Comercial y Ventas:** Procesos relacionados con la promoción de carreras y captación de estudiantes.
* **Servicio al Cliente:** Atención a estudiantes, resolución de problemas, soporte académico y administrativo.

### Procesos de Soporte

Apoyan los procesos misionales, asegurando su funcionamiento eficiente:

* **Gestión de Recursos Humanos (RRHH):** Administración del personal docente y administrativo.
* **Gestión Financiera:** Manejo del presupuesto, ingresos, egresos y sostenibilidad financiera.
* **Gestión Logística:** Administración de recursos materiales, infraestructura y suministros.
* **Gestión de Tecnología de Información:** Desarrollo y mantenimiento de sistemas informáticos y soporte técnico.
* **Gestión del Mantenimiento:** Conservación de la infraestructura física y equipos.



*Figura 2. Procesos de la empresa*

## Oportunidad Encontrada

Se ha identificado que, en la mayoría de los cursos impartidos en las distintas carreras de la Universidad Continental, los estudiantes deben desarrollar trabajos en equipo y proyectos colaborativos. Sin embargo, no siempre se cuenta con un control adecuado ni con una organización eficiente para gestionar estos procesos, lo que puede afectar la calidad de los resultados y la coordinación entre los integrantes. Ante esta situación, la implementación de una plataforma digital colaborativa representa una oportunidad valiosa. Esta solución, que integrará funcionalidades de inteligencia artificial (IA), permitirá optimizar la gestión de proyectos al facilitar la planificación, el seguimiento de avances, la asignación de tareas y la toma de decisiones, promoviendo así una mayor eficiencia y efectividad en el trabajo colaborativo académico.

## Detalles del Proyecto

**Nombre del Proyecto:** Plataforma de Gestión de Proyectos Colaborativos con Integración de Inteligencia Artificial.  
**Cliente:** Universidad Continental – Sede Huancayo.  
**Objetivo General:** Desarrollar una aplicación web innovadora basada en el stack MERN (MongoDB, Express.js, React.js, Node.js) para la gestión colaborativa de proyectos. La plataforma incorporará funcionalidades impulsadas por Inteligencia Artificial (IA) con el fin de optimizar la productividad y mejorar la experiencia del usuario

**Alcance:** La plataforma web permitirá una gestión colaborativa de proyectos mediante la creación de tareas , asignación de roles y comunicación en tiempo real , así mismo integrará inteligencia artificial para optimizar la gestión de tareas y poder predecir los tiempos de ejecución en cada tarea ,además ofrecerá un panel de administración con métricas de rendimiento , un sistema con un entorno seguro y una interfaz intuitiva.

# CAPÍTULO 2

# ESTUDIO DE FACTIBILIDAD

## Alternativas de Solución

Frente a la necesidad detectada en la Universidad Continental de mejorar la organización y el control de los trabajos colaborativos entre estudiantes, se propone como alternativa de solución el desarrollo propio de una plataforma web colaborativa, diseñada a medida para responder a las dinámicas académicas y metodológicas de la institución.

El proyecto, será desarrollado internamente por el equipo de proyecto, utilizando el stack tecnológico MERN. La plataforma estará orientada a facilitar la creación, planificación y seguimiento de proyectos académicos en grupo, permitiendo a los estudiantes y docentes:

* Crear y asignar tareas específicas.
* Definir roles y responsabilidades.
* Comunicarse en tiempo real dentro del entorno del proyecto.
* Utilizar herramientas de IA para optimizar la gestión de tareas y predecir tiempos de ejecución.
* Acceder a un panel de métricas de rendimiento para el monitoreo de progreso.
* Operar dentro de un entorno seguro, intuitivo y adaptable a distintos cursos y facultades.

## Factibilidad Técnica

La factibilidad técnica del proyecto es alta, dado que se cuenta con los recursos tecnológicos y humanos necesarios para su desarrollo e implementación dentro de la Universidad Continental. A continuación, se detallan los aspectos más relevantes:

### Infraestructura y tecnologías empleadas

El desarrollo de la plataforma estará basado en tecnologías modernas, seguras y ampliamente utilizadas en el desarrollo de aplicaciones web. Se ha optado por el stack tecnológico MERN, por su flexibilidad, rendimiento y facilidad de integración. Además, se utilizará Supabase como sistema de base de datos y backend, lo cual permite el almacenamiento en tiempo real, autenticación segura y sincronización eficiente de datos.

**Tecnologías clave:**

* **Frontend:**Tecnologías utilizadas en el desarrollo de la interfaz de usuario:
* React.js v19: Biblioteca moderna de JavaScript para construir interfaces de usuario dinámicas.
* Next.js v15: Framework basado en React que permite renderizado del lado del servidor (SSR) y generación de sitios estáticos.
* Tailwind CSS: Framework de estilos utility-first para diseño responsivo y rápido.
* Dnd Kit: Toolkit moderno para funcionalidades de drag and drop en aplicaciones React.
* Tiptap: Editor de texto enriquecido (rich text editor) para React.
* Recharts: Librería para visualización de datos en gráficos.
* React Hook Form + Zod: Manejo de formularios y validación de datos en el cliente.
* React Email + Resend: Envío de correos electrónicos desde la interfaz.
* **Backend:** Node.js + Express.js – Ambiente de ejecución y framework que permiten una arquitectura REST robusta y escalable.
* **Gestión del Estado (Frontend)**
* React Query, Zustand, Context API, Custom Hooks: Mecanismos para gestionar el estado de la aplicación de forma eficiente y flexible.
* **Base de datos:** Supabase – Plataforma backend como servicio (BaaS) basada en PostgreSQL, que permite autenticación, almacenamiento y consultas en tiempo real.
* **IA Integrada:** Se utilizó gemini para los algoritmos de predicción y optimización.

### Recursos humanos y capacidades técnicas

El equipo de desarrollo está conformado por estudiantes con formación en ingeniería de sistemas, base de datos e inteligencia artificial, lo cual asegura una adecuada ejecución del proyecto, además se cuenta con experiencia previa en el desarrollo de aplicaciones.

**Capacidades del equipo:**

* Diseño e implementación de interfaces web responsivas.
* Construcción de APIs y manejo de bases de datos relacionales y no relacionales.
* Desarrollo e integración de modelos predictivos para optimización de tareas.
* Implementación de sistemas de autenticación y control de acceso.
* Gestión de entornos de desarrollo colaborativos mediante Git y plataformas como GitHub.

La Universidad Continental, como cliente, también dispone de un área de Tecnologías de Información que podrá facilitar el soporte necesario para la integración con los sistemas institucionales existentes.

## Factibilidad Económica

La factibilidad económica se está considerando que gran parte del desarrollo será realizado por un equipo interno (estudiantes), lo cual reduce significativamente los costos asociados a contratación externa y adquisición de herramientas tecnológicas.

### Gastos generales

| Presupuesto Total aproximado (13 semanas) | S/. 66,600 |
| --- | --- |

*Tabla 1: Gastos generales*

* + 1. **Gastos específico:**

| Costo de mano de obra | S/. 35,100 |
| --- | --- |
| Soporte y mantenimiento | S/. 6,500 |
| Desarrollo del Proyecto | S/. 10,000 |
| Implementación y despliegue (alojamiento en la nube) | S/. 5,000 |
| Capacitación | S/. 10,000 |

*Tabla 2: Gastos específicos*

## Factibilidad Operacional

La factibilidad operacional del proyecto es alta, dado que la plataforma está diseñada para integrarse de manera natural con las dinámicas académicas de la Universidad Continental y responde a una necesidad identificada tanto por estudiantes como por docentes: mejorar la gestión de proyectos grupales.

### Sistemas de ventas

Aunque no es una plataforma comercial, se plantea la posibilidad futura de:

* Ofrecer la plataforma a otras universidades o instituciones educativas bajo modelo **SaaS** (Software as a Service).
* Licenciar el software bajo modelo de suscripción para instituciones privadas.

# CAPÍTULO 3

# ANÁLISIS DE REQUERIMIENTOS

## Metas del Sistema de Información

* Implementar una plataforma funcional, intuitiva y escalable que facilite la colaboración en proyectos.
* Incorporar soluciones basadas en IA para automatizar procesos y optimizar la gestión del trabajo en equipo.
* Asegurar un desarrollo robusto mediante la aplicación de buenas prácticas de ingeniería de software, como control de versiones, pruebas automatizadas, contenerización y despliegue en entornos Dockerizados.

## Requisitos del Sistema

Con el fin de asegurar el desarrollo efectivo del sistema, se identifican y detallan los requisitos necesarios para cumplir con los objetivos planteados. Estos requisitos permiten establecer un entendimiento común entre los usuarios finales y el equipo de desarrollo. A continuación, se presentan los requerimientos funcionales y no funcionales del sistema.

### Requerimientos funcionales

* El sistema debe permitir crear, editar y eliminar proyectos.
* El sistema debe permitir asignar tareas a los miembros del equipo.
* El sistema debe mostrar una visualización clara de las tareas asignadas (lista, estado, prioridad).
* El sistema debe integrar un modelo de IA que permita predicciones relacionadas con la gestión de tareas.
* El sistema debe permitir la comunicación mediante chat en tiempo real entre los usuarios.
* El sistema debe enviar notificaciones instantáneas cuando se reciben mensajes o se actualicen tareas.
* El sistema debe permitir eliminar conversaciones.
* El sistema debe mostrar una lista de conversaciones activas por usuario.
* El sistema debe incluir un asistente virtual que brinde ayuda sobre la gestión de tareas y dudas comunes.
* El sistema debe predecir plazos estimados de tareas según urgencia e historial.
* El sistema debe realizar una clasificación automática de tareas según prioridad.
* El sistema debe enviar recordatorios automáticos de tareas próximas a vencer.
* El sistema debe permitir una gestión de roles (admin, líder, miembro).
* El sistema debe contar con pruebas automatizadas de funcionalidades clave.
* El sistema debe permitir el inicio de sesión seguro de usuarios.
* El sistema debe mostrar dashboards con métricas y estadísticas del proyecto.
* El sistema debe permitir la creación y edición de perfiles personalizados.
* El sistema debe permitir la gestión de usuarios (alta, baja, modificación).
* El sistema debe permitir la gestión de permisos y accesos según roles.
* El sistema debe estar integrado a un flujo de CI/CD para despliegue continuo.
* El sistema debe ser desplegado en un entorno web funcional y accesible.

### Requerimientos no funcionales

* El sistema deberá ser accesible desde cualquier dispositivo con navegador web moderno (Google Chrome, Mozilla Firefox, Microsoft Edge), sin necesidad de instalar software adicional.
* La plataforma deberá tener compatibilidad con los sistemas operativos Windows 10 o superior, macOS y las principales distribuciones Linux.
* El sistema deberá estar alojado en un servidor web con disponibilidad mínima del 99%, asegurando acceso continuo a los usuarios.
* El tiempo de carga de la plataforma no deberá superar los 3 segundos en condiciones de red estándar (conexión mayor a 10 Mbps).
* La plataforma deberá tener una interfaz web adaptable (responsive) para garantizar su uso desde dispositivos móviles y tablets.
* El sistema deberá incluir una guía básica de usuario accesible desde el panel de ayuda, con pasos para navegar por la plataforma, crear proyectos y contactar soporte.
* El sistema debe seguir buenas prácticas de desarrollo, como control de versiones.

## Identificación de Actores del Sistema

En esta sección se representan a los usuarios o entidades que tienen un rol específico dentro del funcionamiento de la plataforma, permitiendo el cumplimiento de los objetivos del sistema.

### Administrador

Corresponde al perfil que tiene acceso completo a todas las funcionalidades del sistema. Puede gestionar usuarios, roles y permisos, así como configurar parámetros generales de la plataforma. También es responsable de supervisar el funcionamiento del sistema, generar reportes de actividad y asegurar el correcto uso de los recursos.

### Líder del proyecto

Este perfil tiene acceso a la gestión de proyectos. Puede crear, editar y eliminar proyectos, así como asignar tareas a los miembros del equipo. Además, puede visualizar dashboards de seguimiento, generar reportes y coordinar con el asistente virtual para mejorar la planificación.

### Miembro del Equipo

Representa a los usuarios que colaboran dentro de los proyectos asignados. Pueden visualizar tareas, marcar avances, participar en conversaciones en tiempo real y recibir notificaciones del sistema. También pueden interactuar con el asistente virtual para recibir recordatorios o sugerencias relacionadas a sus actividades.

### Asistente Virtual (IA)

Funciona como un componente automatizado que brinda soporte a los usuarios. Puede sugerir tareas, clasificar actividades, emitir recordatorios y responder preguntas frecuentes para mejorar la experiencia colaborativa.

# CAPÍTULO 4

# PLANIFICACIÓN DEL PROYECTO

## Definición de Roles de Trabajo

Siguiendo los lineamientos de la metodología SCRUM, se definieron los principales roles que participarán en el desarrollo de la plataforma web colaborativa. Estos roles fueron asignados en base a las competencias y funciones requeridas por el proyecto, con el fin de asegurar un flujo de trabajo organizado y eficaz desde las etapas iniciales hasta la entrega final del producto.

| **ROL** | **RESPONSABLE** |
| --- | --- |
| Product Owner | Ingrid Yeanella Alderete Condor |
| Scrum Master | Jaci Lucas Eulogio |
| Team Members | Alan Orellana Marcas, Carlos Villalva Quispe |
| Tester | Sofía Taipe Martínez |

*Tabla 3: Roles de trabajo.Elaboración propia*

### Product owner

El rol es asumido por Ingrid Yeanella Alderete Condor, quien representa al cliente y a los stakeholders del proyecto. Esta persona se encarga de gestionar el Product Backlog, priorizar los requerimientos y asegurar que el desarrollo del sistema esté alineado con los objetivos estratégicos del usuario final. Además, participa activamente en las reuniones de planificación y revisión, velando por el valor del producto entregado.

### Scrum master

El estudiante Jaci Lucas Eulogio asume el rol de Scrum Master, debido a su experiencia en gestión ágil de proyectos y conocimientos técnicos en metodologías iterativas. Su función será guiar al equipo en la aplicación adecuada del marco SCRUM, eliminar obstáculos que impidan el progreso y asegurar que se mantenga el enfoque ágil durante todo el desarrollo.

### Team member

Los estudiantes *Alan Orellana Marcas* y *Carlos Villalva Quispe* conforman el equipo de desarrollo. Cada uno aporta conocimientos técnicos específicos como la programación en React.js y Node.js, modelamiento de bases de datos, integración de inteligencia artificial y desarrollo backend. Son responsables de construir los módulos funcionales de la plataforma de manera colaborativa y eficiente.

### Tester

La estudiante Sofía Taipe Martínez desempeñará el rol de Tester, encargada de la planificación y ejecución de pruebas funcionales, de integración y de calidad del software. Su trabajo garantiza que los entregables cumplan con los requisitos establecidos, asegurando un producto confiable y libre de errores antes de su puesta en producción.

## Product Backlog

En la **Metodología SCRUM**, el Product Backlog representa el listado priorizado de funcionalidades e implementaciones necesarias para la construcción del sistema. Cada ítem es una historia de usuario (HU) o tarea técnica (HT), agrupada en épicas que representan grandes módulos funcionales del sistema. La priorización fue definida en sesiones con el Product Owner, tomando en cuenta el valor entregado al usuario y la complejidad estimada.

La siguiente tabla presenta el Product Backlog completo, distribuido por sprint y ordenado por prioridad.

| **ID** | **DESCRIPCIÓN** | **ÉPICA** | **PRIORIDAD** | **COMPLEJIDAD** |
| --- | --- | --- | --- | --- |
| **HU14** | Asignación de tareas | Gestión de Proyectos | 1 | 2 |
| **HU13** | Editar proyectos | Gestión de Proyectos | 1 | 2 |
| **HU15** | Visualización de tareas | Gestión de Proyectos | 1 | 1 |
| **HT01** | Configuración del entorno MERN | Gestión de Proyectos | 1 | 2 |
| **HT02** | Integración de modelo IA | Gestión de Proyectos | 1 | 2 |
| **HU05** | Chat en tiempo real | Comunicación en tiempo real | 2 | 3 |
| **HU06** | Notificación de mensajes | Comunicación en tiempo real | 2 | 2 |
| **HU07** | Eliminar conversaciones | Comunicación en tiempo real | 2 | 1 |
| **HU08** | Lista de conversaciones | Comunicación en tiempo real | 2 | 1 |
| **HT03** | Configuración de WebSocket | Comunicación en tiempo real | 2 | 2 |
| **HT04** | Notificaciones en tiempo real | Comunicación en tiempo real | 2 | 2 |
| **HU02** | Predicción de tareas | Integración de IA | 3 | 3 |
| **HU01** | Asistente virtual | Integración de IA | 3 | 3 |
| **HU03** | Clasificación de tareas | Integración de IA | 3 | 1 |
| **HU04** | Recordatorio de tareas | Integración de IA | 3 | 1 |
| **HT05** | Control de roles | Integración de IA | 3 | 2 |
| **HT06** | Pruebas automatizadas y verificación | Integración de IA | 3 | 2 |
| **HU12** | Inicio de sesión | Panel de Administración | 4 | 1 |
| **HU09** | Dashboards | Panel de Administración | 4 | 2 |
| **HU16** | Perfiles personalizados | Panel de Administración | 4 | 2 |
| **HU10** | Gestión de usuarios | Panel de Administración | 4 | 2 |
| **HU11** | Gestión de permisos | Panel de Administración | 4 | 2 |
| **HT07** | Configuración de CI/CD | Panel de Administración | 4 | 2 |
| **HU02** | Despliegue en entorno | Panel de Administración | 4 | 2 |

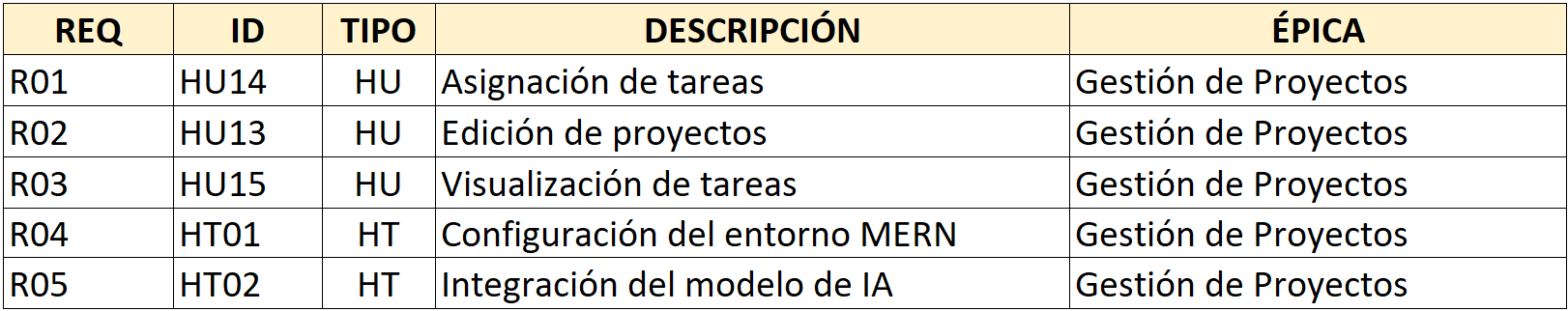
*Tabla 4: Product Backlog.Elaboración propia*

## Sprint Backlog

Una vez definido el Product Backlog, la metodología SCRUM establece la necesidad de organizar el Sprint Backlog, el cual consiste en una planificación específica de los requerimientos que serán desarrollados en cada iteración (Sprint). Esta selección se realiza en la Sprint Planning Meeting, con la participación del Product Owner, Scrum Master y el Scrum Team.En este proyecto se han planificado cuatro sprints principales:

### Sprint 1

| **Duración:** 8 días | **Inicio:** 11/04/2025 | **Final:** 22/04/2025 |
| --- | --- | --- |
| **Objetivo:** Implementar la configuración técnica inicial del entorno de desarrollo y sentar las bases de funcionalidades clave en gestión de proyectos, incluyendo la integración del primer modelo de IA. | | |
| **Actividades clave :** Integración del stack MERN, desarrollo de funcionalidades básicas de gestión de proyectos e inicio de integración de IA. | | |



*Tabla 5: Sprint 1.Elaboración propia*

### Sprint 2

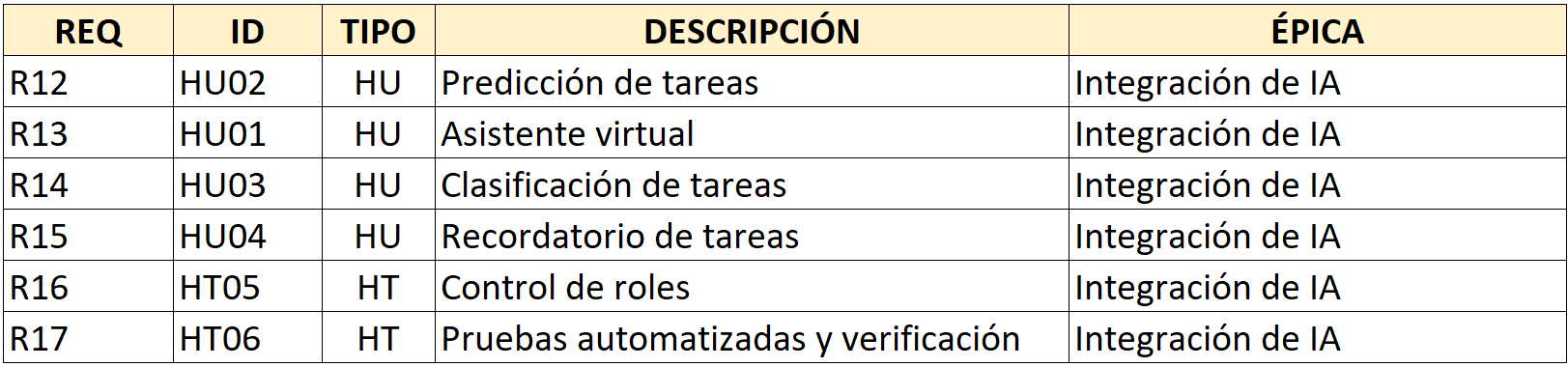
| **Duración:** 13 días | **Inicio:** 11/04/2025 | **Final:** 22/04/2025 |
| --- | --- | --- |
| **Objetivo:** Desarrollar las funcionalidades colaborativas del sistema, centradas en la comunicación en tiempo real entre los usuarios. | | |
| **Actividades clave :** Desarrollo de chat, lista de conversaciones, notificaciones y configuración de WebSocket. | | |



*Tabla 6: Sprint 2.Elaboración propia*

### Sprint 3

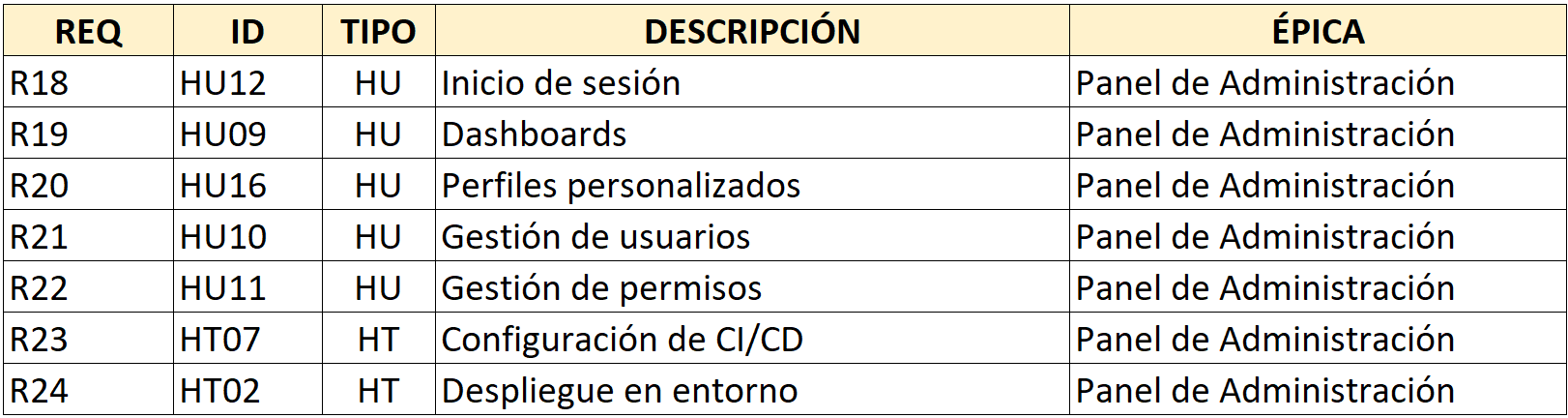
| **Duración:** 11 días | **Inicio:** 11/04/2025 | **Final:** 22/04/2025 |
| --- | --- | --- |
| **Objetivo:** Implementar funcionalidades avanzadas apoyadas por IA, como la predicción y clasificación de tareas, además de herramientas de soporte como control de roles y pruebas. | | |
| **Actividades clave :** Desarrollo de asistente virtual, recordatorios, y automatización de pruebas. | | |



*Tabla 7: Sprint 3.Elaboración propia*

### Sprint 4

| **Duración:** 19 días | **Inicio:** 11/04/2025 | **Final:** 22/04/2025 |
| --- | --- | --- |
| **Objetivo:** Implementar funcionalidades del panel administrativo, incluyendo control de usuarios y configuración del sistema para su despliegue. | | |
| **Actividades clave :** Gestión de usuarios, permisos, dashboards, CI/CD y despliegue final. | | |



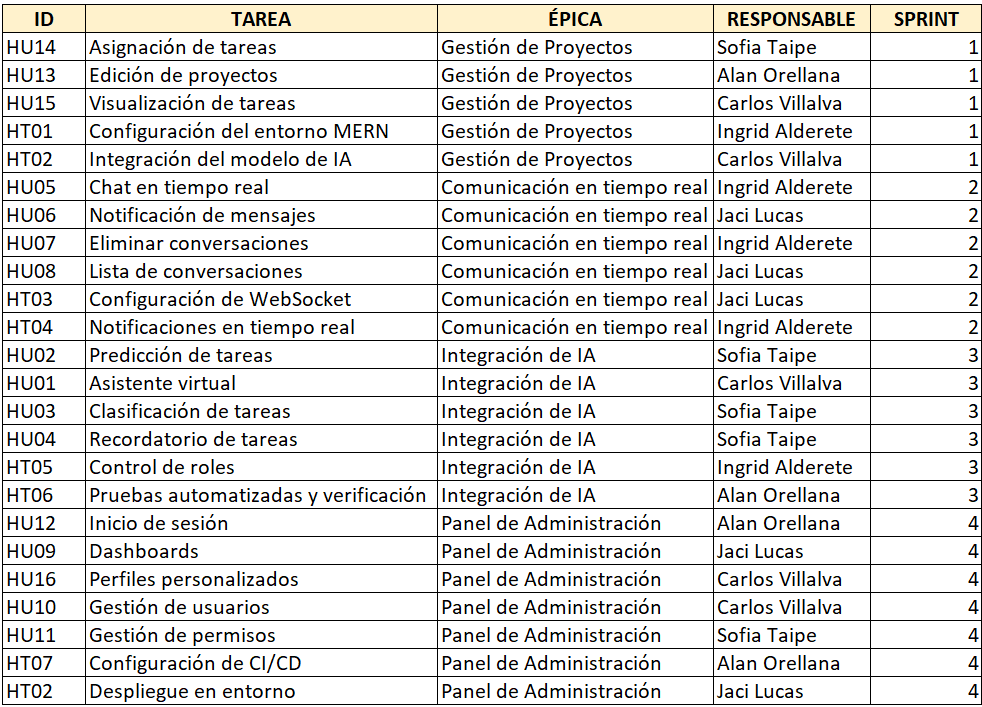
*Tabla 8: Sprint 4.Elaboración propia*

## Planificación de Sprints

Según la Metodología SCRUM, el equipo de trabajo define y organiza las historias de usuario para garantizar que todas las funcionalidades requeridas en el proyecto queden plasmadas en actividades específicas, con criterios de aceptación claros. Esto garantiza que cada miembro del equipo y el Product Owner tengan visibilidad sobre qué tarea se implementará en cada Sprint, quién será el responsable y qué criterios serán validados al finalizarla.

### Historias de usuario

A continuación, se presenta la **lista de historias de usuario** obtenida durante las sesiones de planificación, donde participaron tanto el equipo de desarrollo como el Product Owner. Estas historias fueron asignadas a cada Sprint correspondiente para organizar y garantizar la correcta entrega de las funcionalidades:



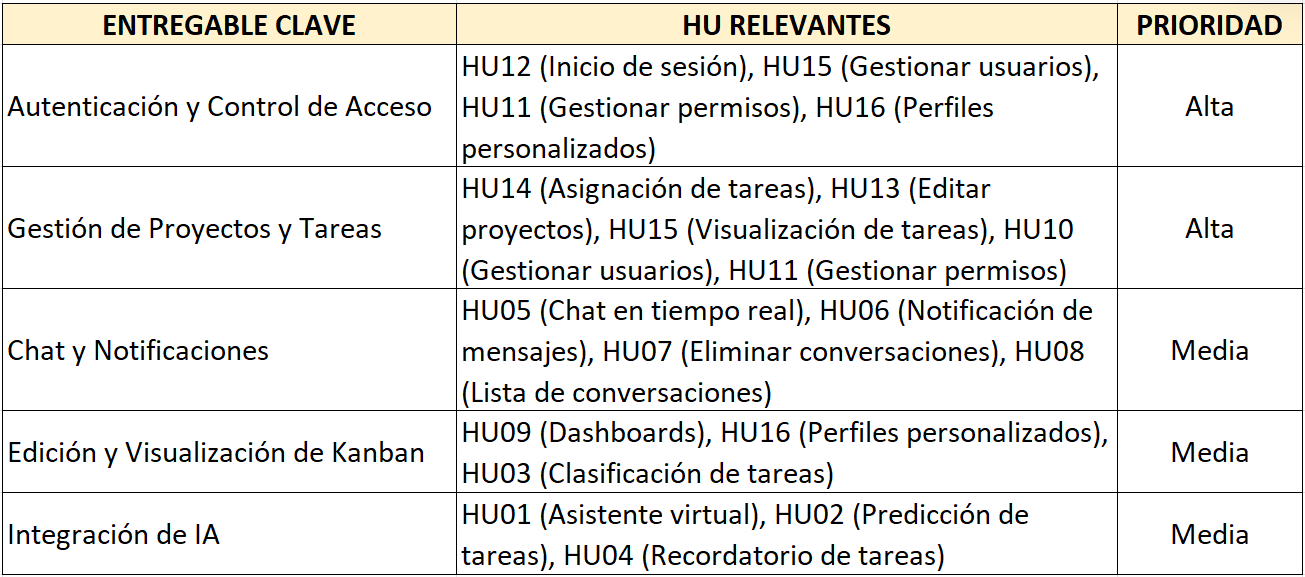
*Tabla 9: Historias de usuario.Elaboración propia*

### Priorización de historias de usuario

Una vez definida la lista de historias de usuario en la sección anterior, se procedió a establecer su prioridad en función de los entregables clave del proyecto, considerando la importancia e impacto en la experiencia de usuario y en la operación general del sistema.La priorización estuvo basada en criterios de relevancia para garantizar que las funcionalidades esenciales estuvieran disponibles en los primeros Sprints, mientras que las complementarias o de valor añadido (como la inteligencia artificial) fueron asignadas a iteraciones posteriores.

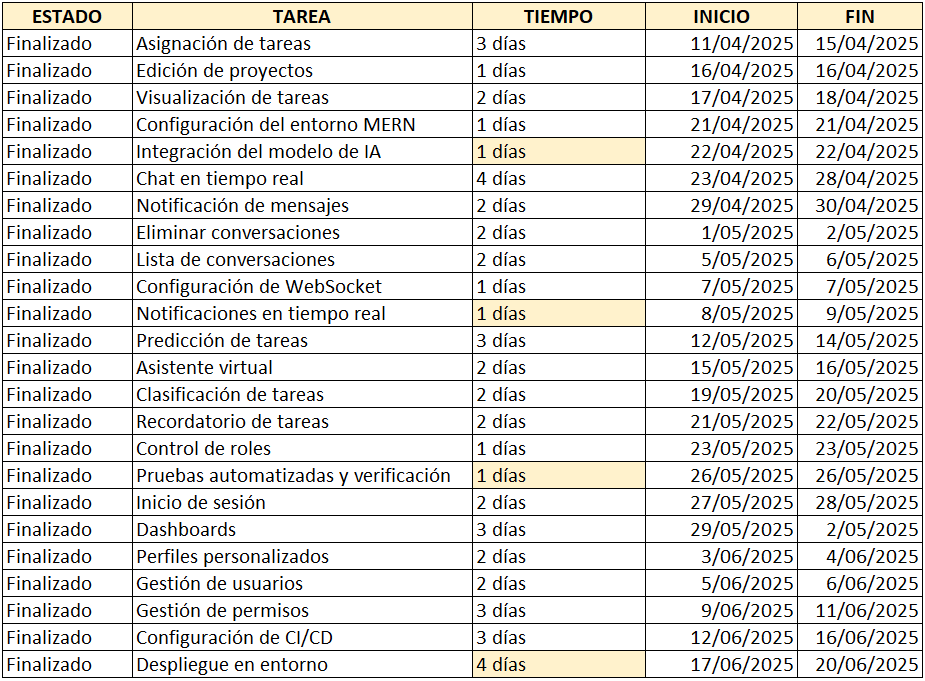
**Criterios de Priorización Aplicados :**

* **Prioridad Alta :** Historias que hacen posible la operación básica de la plataforma, como el módulo de autenticación y control de acceso, y la creación, edición y visualización de proyectos y tareas.
* **Prioridad Media :** Historias para la comunicación interna del equipo y la colaboración en tiempo real, como chat y notificaciones.



*Tabla 10: Priorización De historias*

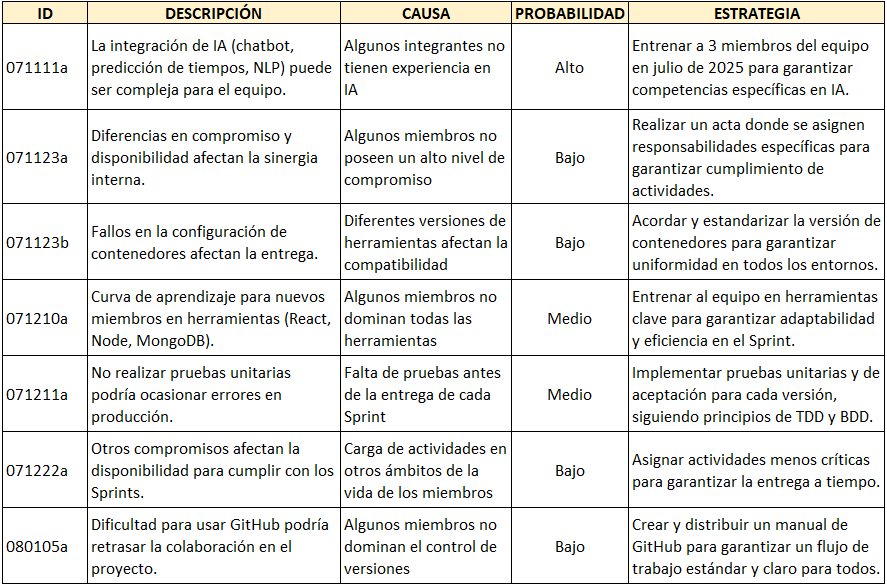
## Cronograma de Actividades



*Tabla 11: Cronograma.Elaboración propia*

## Gestión de Riesgos

Una vez identificadas todas las actividades a realizar para llevar a cabo el proyecto, y siguiendo la Metodología SCRUM, es indispensable definir un Plan de Gestión de Riesgos que permita anticiparse a posibles obstáculos y garantizar que estos no afecten de manera crítica la entrega de cada incremento del producto. A continuación, se presentan los principales riesgos, junto con su área de impacto, causa, estrategia de mitigación y responsable asignado para garantizar un seguimiento adecuado.



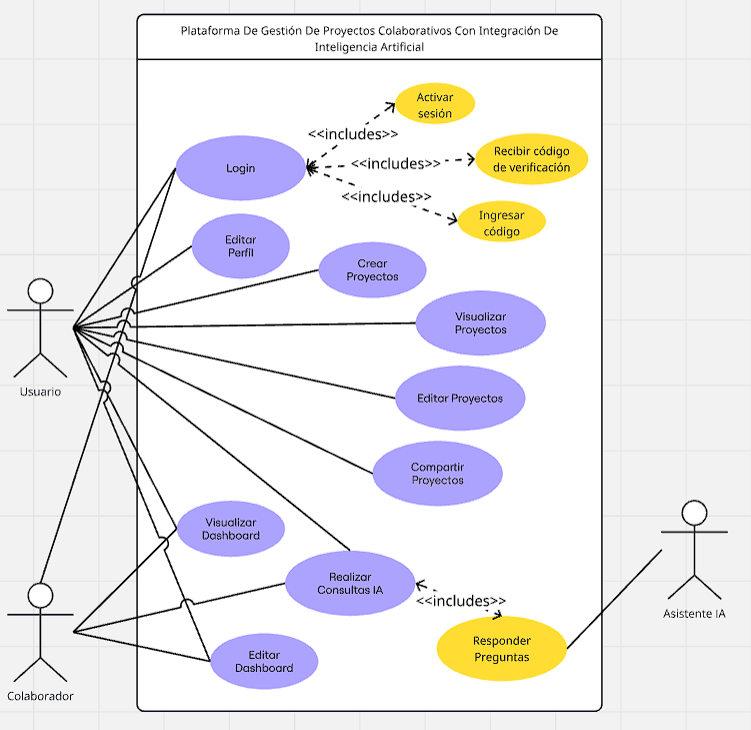
*Tabla 12: Riesgos.Elaboración propia*

# CAPÍTULO 5

# DISEÑO DEL SISTEMA DE INFORMACIÓN

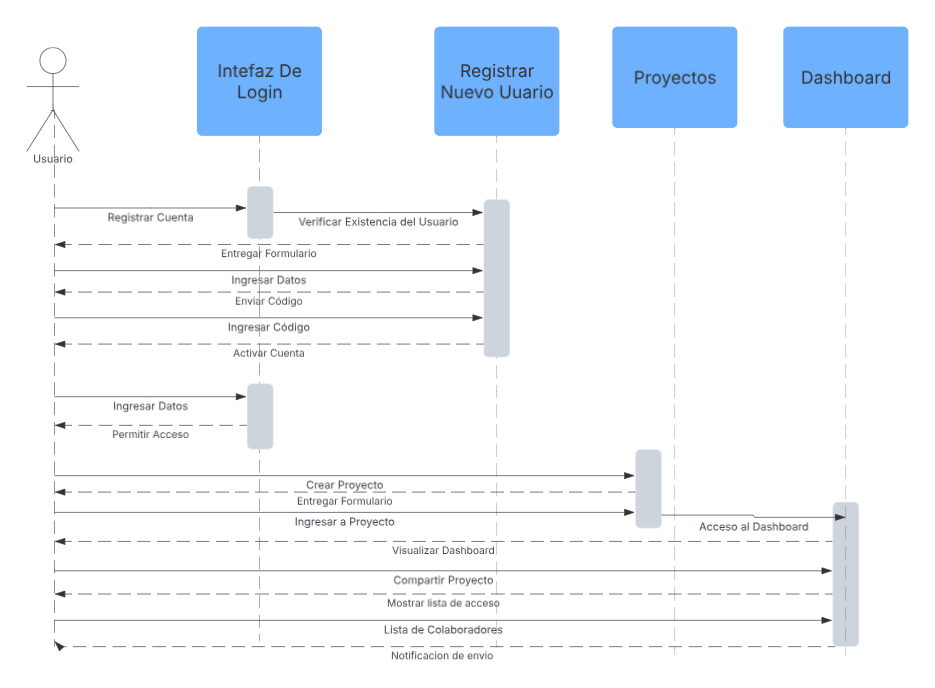
## Diseño de Diagramas UML

### Diagramas de casos de uso



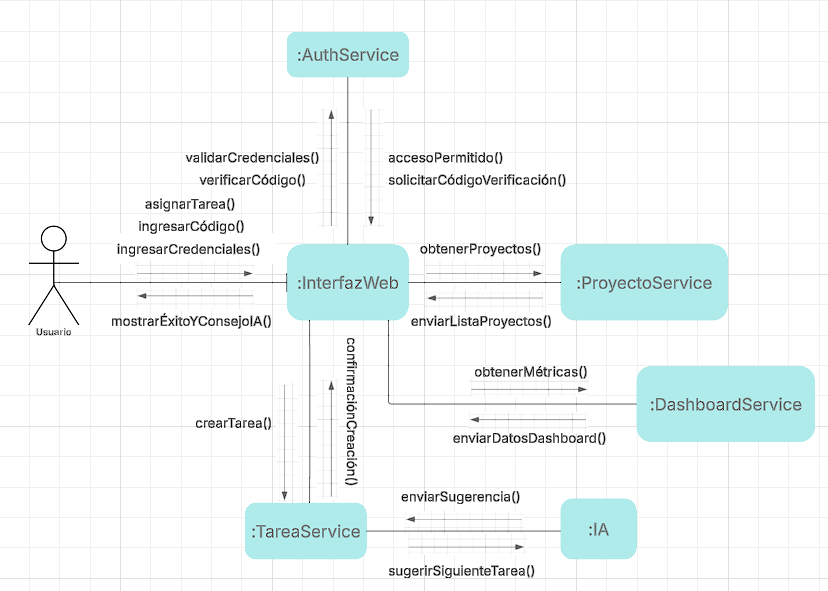
*Figura 3. Casos de uso*

### Diagramas de secuencia



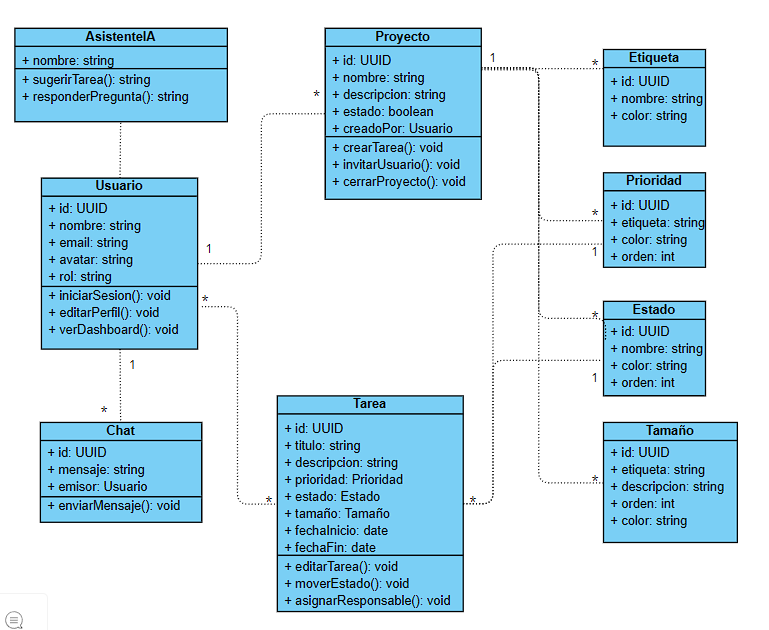
*Figura 4. Diagramas de secuencia.Elaboración propia*

### Diagramas de colaboración



*Figura 5. Diagramas de colaboración.Elaboración propia*

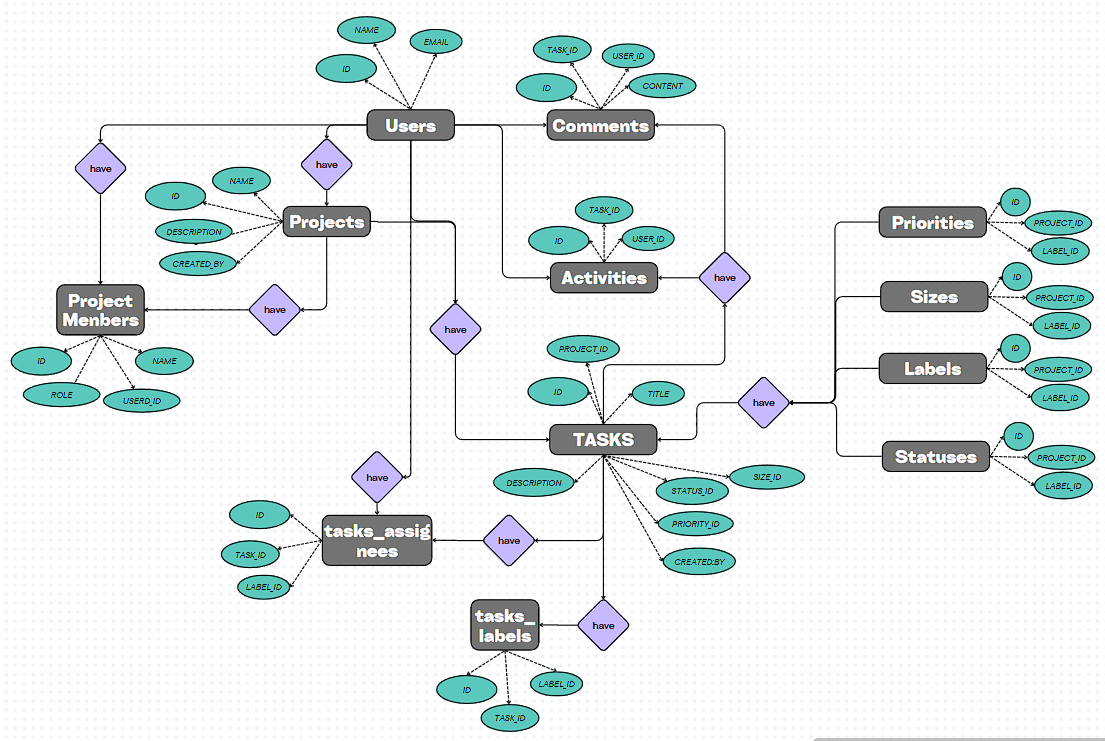
### Diagramas de clases



*Figura 6. Diagramas de clases.Elaboración propia*

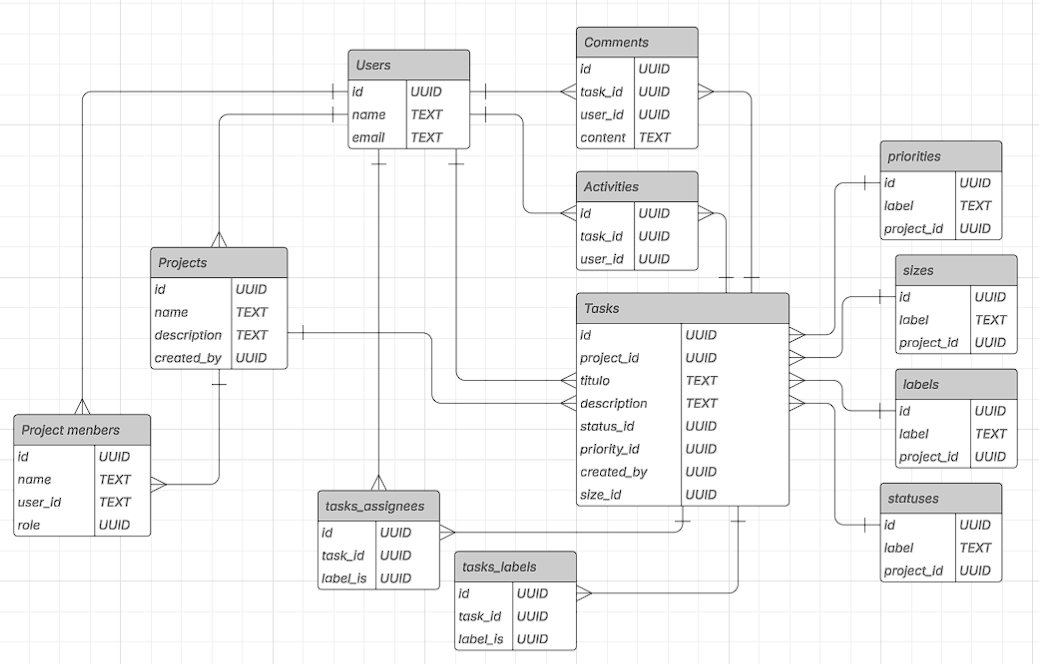
## Diseño de Base de Datos

### Diseño conceptual (E/R)



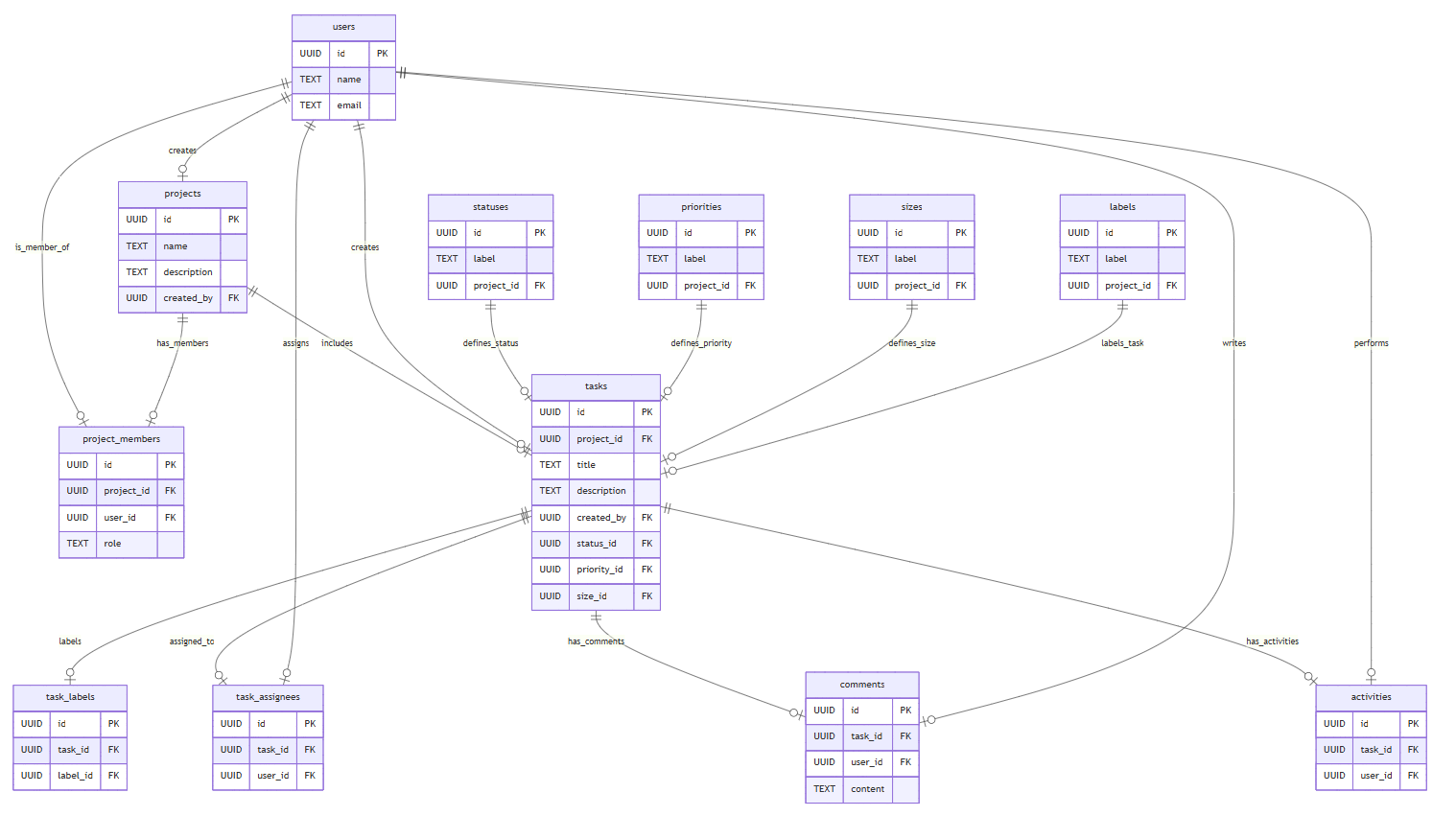
*Figura 7. Diseño conceptual.Elaboración propia*

### Diseño lógico



*Figura 8. Diseño lógico.Elaboración propia*

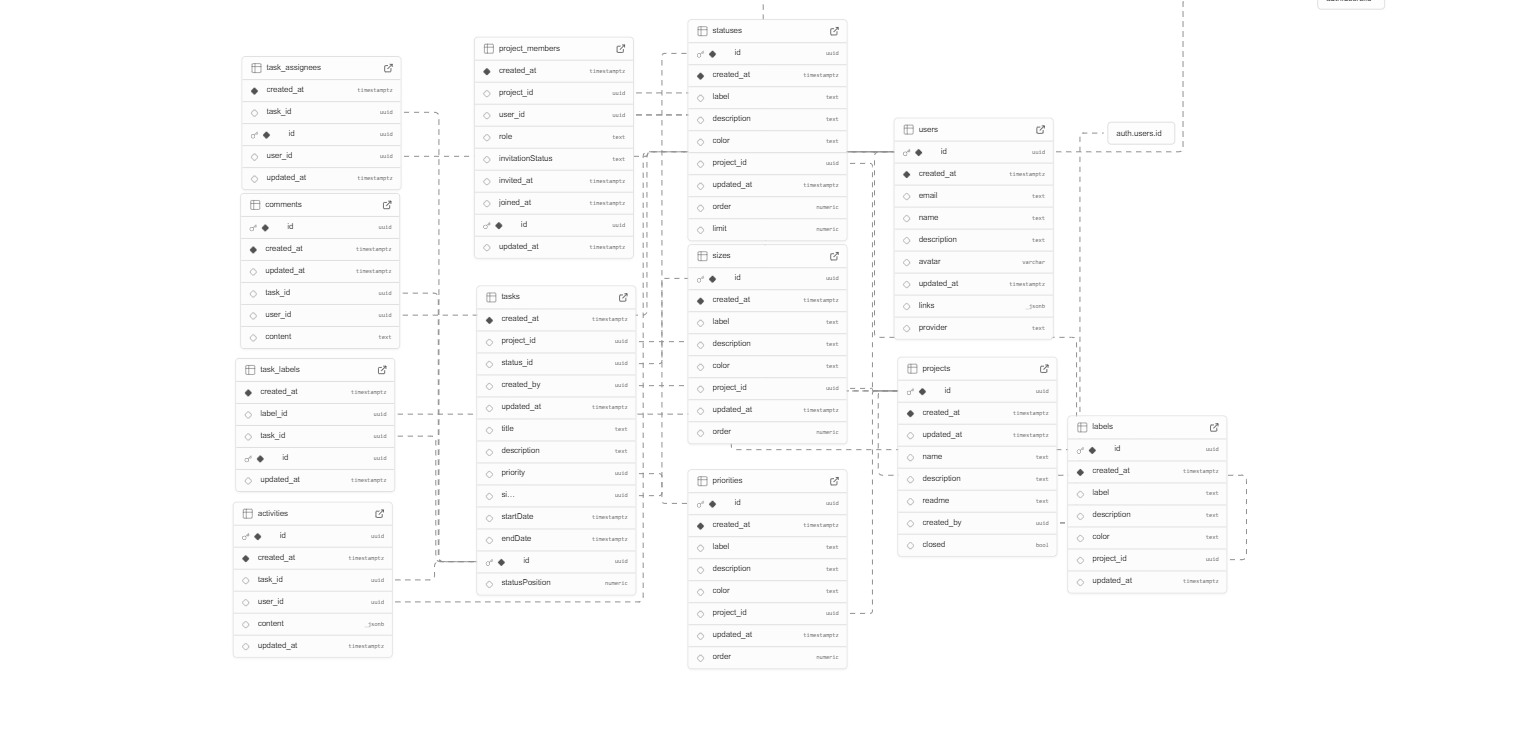
### Diseño físico



*Figura 9. Diseño Físico.Elaboración propia*

<https://www.mermaidchart.com/raw/6b1216a6-8fd4-4b71-88c9-139c24722f8d?theme=light&version=v0.1&format=svg>

### Modelado de base de datos



*Figura 10. Modelado BDD.Elaboración propia*

## Diseño de Interfaces Básicas

### Interfaz De Inicio



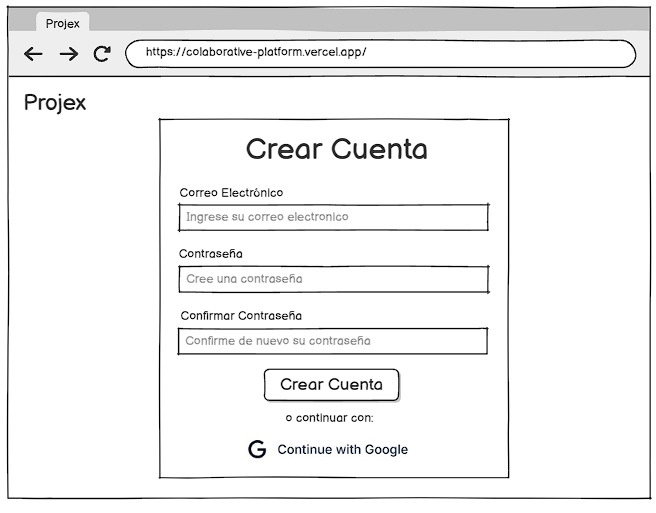
*Figura 11. Interfaz de inicio*

### Interfaz de Inicio de Sesión



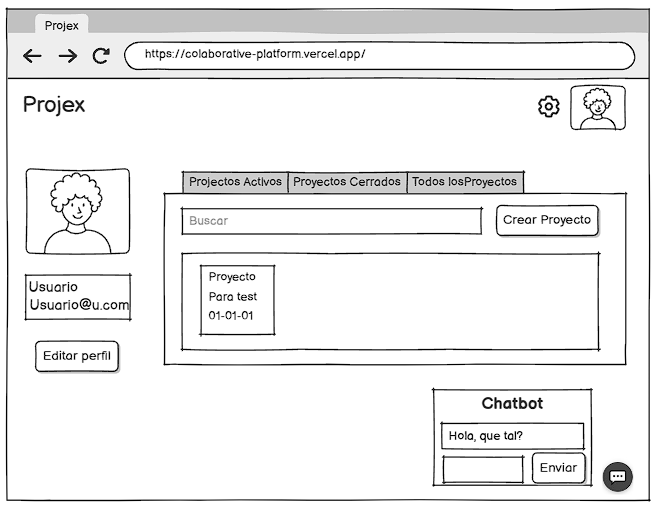
*Figura 12. Interfaz de inicio de sesión*

* + 1. **Interfaz de Crear Cuenta Nueva**



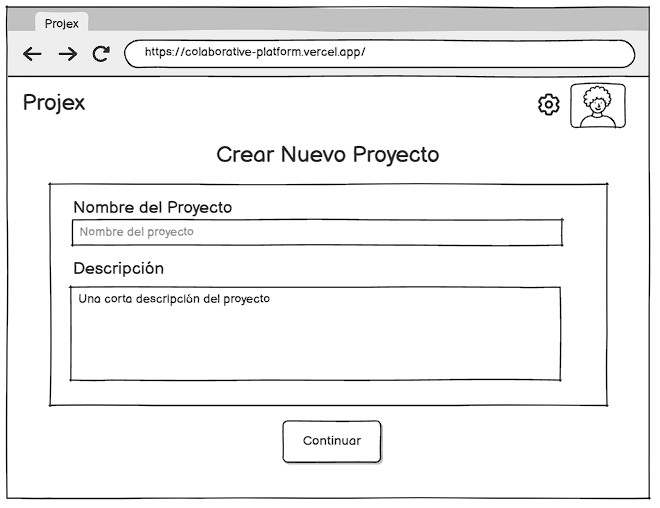
*Figura 13. Interfaz de crear cuenta*

* + 1. **Interfaz de Página Principal**



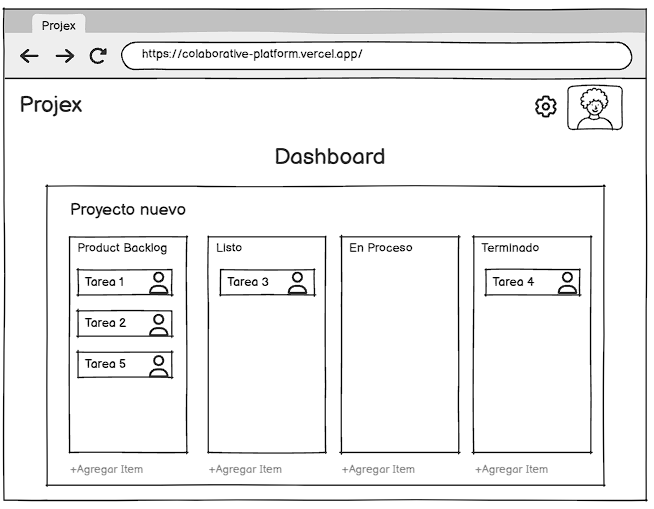
*Figura 14. Interfaz de página principal*

* + 1. **Interfaz de Crear Nuevo Proyecto**



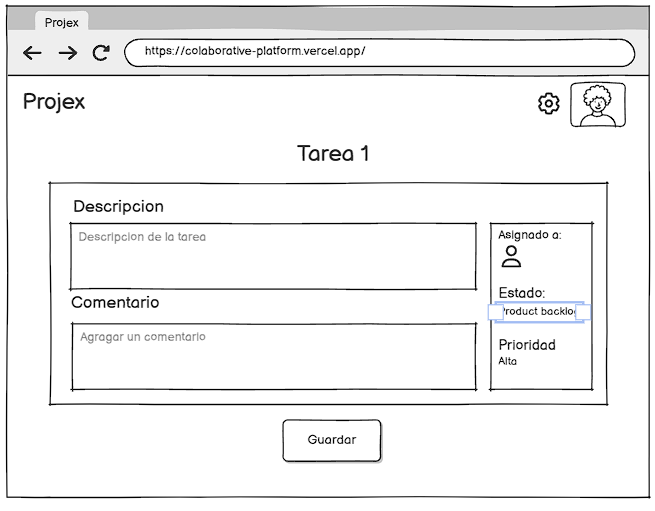
*Figura 15. Interfaz de crear un proyecto*

* + 1. **Interfaz de Tablero Kanban**



*Figura 16. Interfaz de tablero kanban*

* + 1. **Editar Tarea**



*Figura 17. Interfaz de editar tarea*

# CAPÍTULO 6

# CODIFICACIÓN DEL SOFTWARE

## Desarrollo del Sprint 1

### Sprint planning

El sprint 01 se planeó con las funcionalidades principales que componen el MVP 1, 3 historias de usuario y 2 historias técnicas.

Para el desarrollo de la aplicación web se utilizó una variante del Stack MERN, reemplazando MongoDB por Supabase,y React por next.js.

### Sprint backlog

El sprint backlog 1 se compone con las funcionalidades principales del MVP 1, que consiste en la gestión de proyectos, permitiendo al usuario asignar tareas a los miembros del equipo, editar las características del proyecto, y que el usuario pueda interactuar de forma visual con el proyecto.

También se configuró el entorno de desarrollo MERN con variantes como Supabase como base de datos y Next.js como frontEnd.

Y por último la integración de un chatbot con una inteligencia artificial generativa con una API Key de gemini oculta en el backend de Express.js.

### Historias de usuarios

Las historias de usuario se implementaron en la aplicación web y no sufrieron modificaciones.

* HU14 - Asignación de tareas
* HU13 - Editar proyectos
* HU15 - Visualización de tareas
* HT01 - Configuración del entorno MERN (Supabase, Next.js)
* HT02 - Integración de modelo IA

### Taskboard

Para visualizar el seguimiento del desarrollo de la aplicación web se utilizó la aplicación Jira, en el cual se puede visualizar las 3 historias de usuario y 2 dos historias técnicas, en el estado de las historias de usuario e historias técnicas, están finalizadas, sin sufrir modificación alguna.

### Daily scrum

El equipo de desarrollo y product owner para coordinar las actividades se utilizaron aplicaciones como whatsapp y reuniones presenciales, además de aplicaciones de gestión de proyectos Kanban en Jira para organizar el cronograma de desarrollo de los sprints y las historias de usuario, además se generó un cronograma de actividades con la aplicación de Microsoft Project.

### Sprint review

Los miembros del equipo de desarrollo conformado por Sofía, Alan y Carlos, designados al sprint 1, culminaron de forma exitosa las historias de usuario del sprint, sin hacer alteraciones en la idea principal del las historias de usuario, culminando las actividades a tiempo.

### Criterios de aceptación

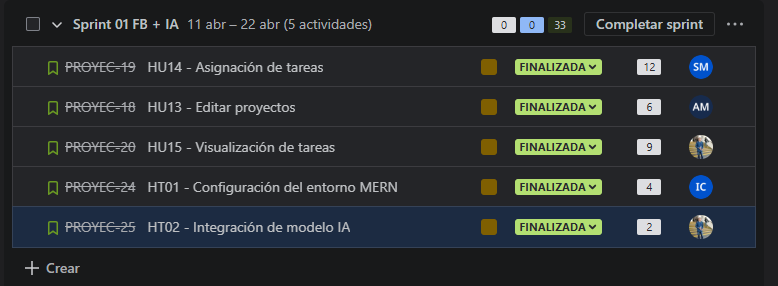
* HU14 - Asignación de tarea
  + Se puede seleccionar a uno o varios usuarios por tarea.
  + Los integrantes del proyecto pueden asignar tareas.
* HU13 - Editar proyectos
  + El usuario puede modificar el nombre, descripción, fecha de inicio y fin del proyecto.
  + Los cambios deben guardarse en tiempo real o al presionar “Guardar”.
  + Se muestra un mensaje de éxito o error después de guardar.
  + No se permite establecer una fecha de fin anterior a la fecha de inicio.
* HU15 - Visualización de tareas
  + Tareas listadas por estado (ToDo, En progreso, Hecho).
  + Muestra prioridad, responsable y fecha de vencimiento.
  + Posibilidad de cambiar el estado arrastrando en tablero.
* HT01 - Configuración del entorno MERN (Supabase, Next.js)
  + El proyecto se puede levantar con docker-compose up sin errores.
  + Las variables de entorno están correctamente definidas y accesibles.
  + Existe un README con instrucciones básicas de instalación y ejecución.
* HT02 - Integración de modelo IA
  + El backend puede recibir preguntas y responder usando el modelo.
  + El modelo está cargado al iniciar el servidor.

### Resultados del sprint

#### Evidencias.

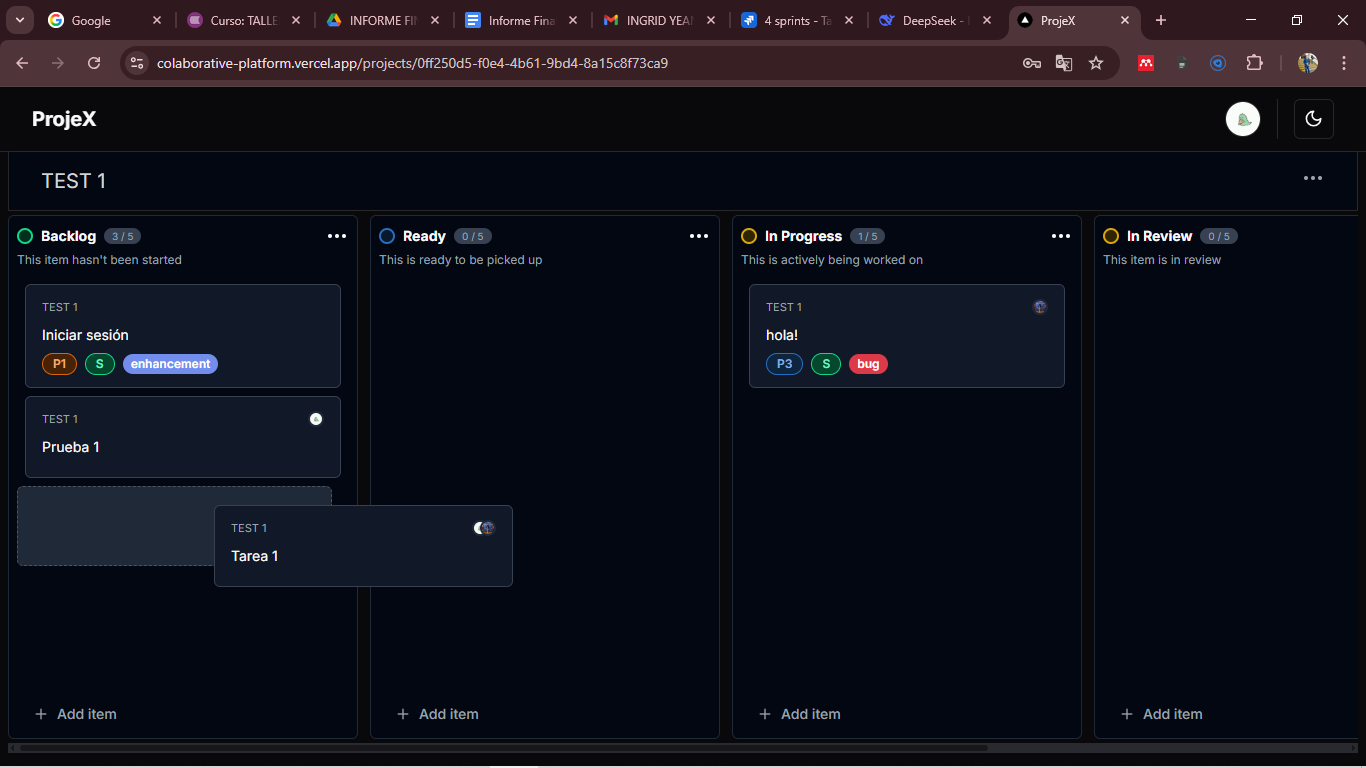
* Componentes del sprint 1 en Jira

Se puede visualizar en la imagen que las historias de usuario tiene un estado finalizados y también están asignados a un miembro del equipo y la duración de las actividades son del 11 al 22 de abril, y solo se contarán 5 días hábiles a la semana, iniciando un viernes y finalizando un martes, haciendo un total de 8 días.

*Figura 18. Componentes del sprint 1 en Jira*

* Resultados del sprint 1 en la aplicación web

En la aplicación web se puede visualizar el desarrollo del sprint 1, donde hay actividades de prueba, con sus respectivos estados, en los 3 puntos de la esquina superior derecha se puede editar, agregar, eliminar los estados, y en el contenido se puede agregar, editar o eliminar las tareas, asignando características como prioridad, tamaño, duración y asignar responsables.

*Figura 19. Resultados del sprint 1* 

#### Prueba de desarrollo.

Las pruebas realizadas a las historias de usuario de usuario al finalizar cada sprint fueron las pruebas unitarias, se realizaron de forma manual, de caja negra, ya que solo se realizó probando las funcionalidades de la aplicación y no generó errores.

### Sprint retrospective

En la reunión del equipo de desarrollo se pudo constatar el resultado satisfactorio de cada historia de usuario.

Considerando algunas mejoras a la historia de usuario 1, que la asignación de tareas y edición de proyectos solo se pueda realizar por el administrador y en un tiempo definido, para que no haya discordancia entre los miembros del equipo que usan la aplicación

## Desarrollo del Sprint 2

### Sprint planning

El estado del sprint 2 está en tareas por hacer, por que aún se está desarrollando, consta de de 4 historias de usuario que corresponden al MVP 2.

Para la configuración del WebSocket en la aplicación web se utilizara Supabase para almacenar mensajes, recibir actualizaciones en tiempo real mediante su sistema Realtime y construir un chat completo sin necesidad de APIs WebSocket externas, ya que vercel de forma nativa no soporta la configuración de WebSocket.

### Sprint backlog

El sprint backlog se compone con las funcionalidades principales del MVP 2, que consiste en la comunicación en tiempo real, se divide en 4 historias de usuario como: chat en tiempo real, notificaciones de mensajes, eliminar conversaciones y listar conversaciones.

### Historias de usuarios

Las historias de usuario listadas a continuación están en estado de en curso, ya que falta la configuración de websocket, y actualizarlo en la aplicación desplegada.

HU05 - Chat en tiempo real

HU06 - Notificación de mensajes

HU07 - Eliminar conversaciones

HU08 - Lista de conversaciones

### Taskboard

Para visualizar el seguimiento del desarrollo de la aplicación web se utilizó la aplicación Jira, en el cual se puede visualizar las 4 historias de usuario y 2 dos historias técnicas, en el estado de las historias de usuario e historias técnicas, están curso, aún no se actualizaron en la aplicación desplegada.

### Daily scrum

El equipo de desarrollo y product owner para coordinar las actividades se utilizaron aplicaciones como whatsapp y reuniones presenciales, además de aplicaciones de gestión de proyectos Kanban en Jira para organizar el cronograma de desarrollo de los sprints y las historias de usuario, se sigue el curso del cronograma utilizado en microsoft project, aunque se tiene atraso en la entrega del sprint 2.

### Sprint review

Los miembros del equipo de desarrollo conformado por Ingrid y Jaci, designados al sprint 2, aún no terminaron con el desarrollo del sprint 2, ya que se encuentra en un estado de en curso.

### Criterios de aceptación

HU05 - Chat en tiempo real

* Permite enviar y recibir mensajes instantáneamente.
* Los mensajes se muestran con nombre, hora y orden cronológico.
* Permite compartir archivos o enlaces.

HU06 - Notificación de mensajes

* Se muestra un ícono de notificación con badge numérico.
* Opción de activar sonido o vibración al recibir mensaje.

HU07 - Eliminar conversaciones

* Poder eliminar mensajes propios.
* Borrar mensajes del administrador.

HU08 - Lista de conversaciones

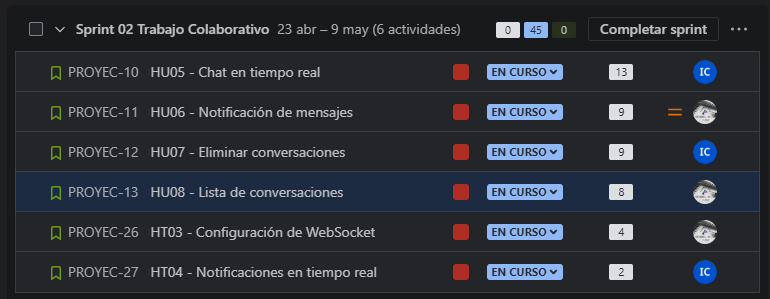
* Vista organizada por canal.
* Este organizada por usuario o tema.

### Resultados del sprint

#### Evidencias.

* Componentes del sprint 2 en Jira

Se puede visualizar en la imagen que las historias de usuario tiene un estado de en curso y también están asignados 2 miembro del equipo y la duración de las actividades son del 23 de abril al 9 de mayo, y se contarán 5 días hábiles a la semana, iniciando un miércoles y finalizando un viernes, haciendo un total de 12 días.

*Figura 20. Componentes del sprint 2 en Jira*

#### Prueba de desarrollo.

Aún no se realizaron las pruebas de desarrollo respectivas.

### Sprint retrospective

En la reunión del equipo de desarrollo se pudo constatar que no hubo resultados en el desarrollo del sprint, afectando el funcionamiento del MVP 2, ya que no habría comunicación en tiempo real por parte del equipo que usa la aplicación.

## Desarrollo del Sprint 3

### Sprint planning

El sprint 3 tiene 4 historias de usuario, relacionado a la integración de la inteligencia artificial en realizar predicción de tareas, trabajando con un asistente virtual, y el control de acceso la duración planeado del sprint es de 12 al 26 de mayo se planeó trabajar 11 días.

Las herramientas de IA con el cual se va a predecir las tareas, va a ser Gemini AI, que es una inteligencia artificial de google, con el cual se va a hacer predicción relacionado a las tareas, y realizar las consultas.

### Sprint backlog

El sprint backlog se compone con las funcionalidades secundarias que no afectan en el funcionamiento de la aplicación de las funcionalidades de la aplicación, pero no aplicarlas puede generar vulnerabilidades en las actividades.

El sprint cuenta con 3 historias de usuario y dos historias técnicas.

### Historias de usuarios

HU01 - Asistente virtual

HU03 - Clasificación de tareas

HU04 - Recordatorio de tareas

### Taskboard

Para visualizar el seguimiento del desarrollo de la aplicación web se utilizó la aplicación Jira, en el cual se puede visualizar las 3 historias de usuario y 2 dos historias técnicas.

Para visualizar el estado de las historias de usuario se utilizó Jira, ya que de acuerdo a cada cambio realizado se fué cambiando de estado a las historias de usuario.

### Daily scrum

El equipo de desarrollo y product owner para coordinar las actividades se utilizaron aplicaciones como whatsapp y reuniones presenciales, además de aplicaciones de gestión de proyectos Kanban en Jira para organizar el estado y cronograma de desarrollo de los sprints y las historias de usuario, se sigue el curso del cronograma utilizado en microsoft project, aunque se tiene atraso en la entrega del sprint 3.

### Sprint review

Los miembros del equipo de desarrollo conformado por Sofía, Alan, Ingrid y Carlos, designados al sprint 3, entregaron de forma parcial los entregables del sprint 3, ya que faltan desarrollar algunas historias de usuario, y otras por completar, como el asistente virtual cuenta con un estado de finalizado, no se implementó la historia de usuario de recordatorio de tareas, y están en curso clasificación de tareas y predicción de tareas, ya que estos últimos 2 aún no se implementaron mediante una IA.

### Criterios de aceptación

HU01 - Asistente virtual

* El asistente virtual debe estar visible en todas las vistas.
* Debe responder mínimo a 5 preguntas frecuentes.

HU03 - Clasificación de tareas

* Se recibe una notificación visual 24h y 1h antes del vencimiento.
* Opción de configurar recordatorios personalizados.
* El recordatorio debe incluir nombre, responsable y tiempo restante.
* Permite marcar como “no recordar más”.
* Se integra con correo electrónico o notificaciones push.

HU04 - Recordatorio de tareas

* Permite etiquetar tareas.
* Categorizar tareas por tipo o urgencia.

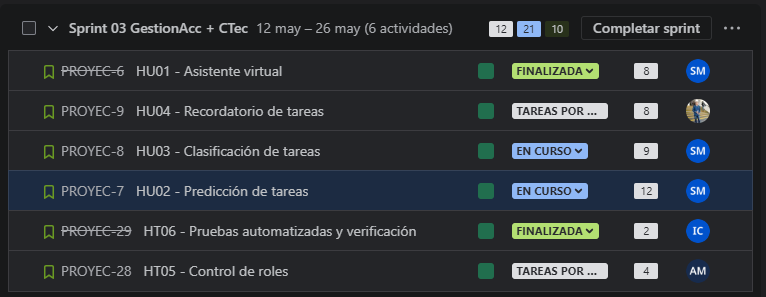
HU02 - Predicción de tareas

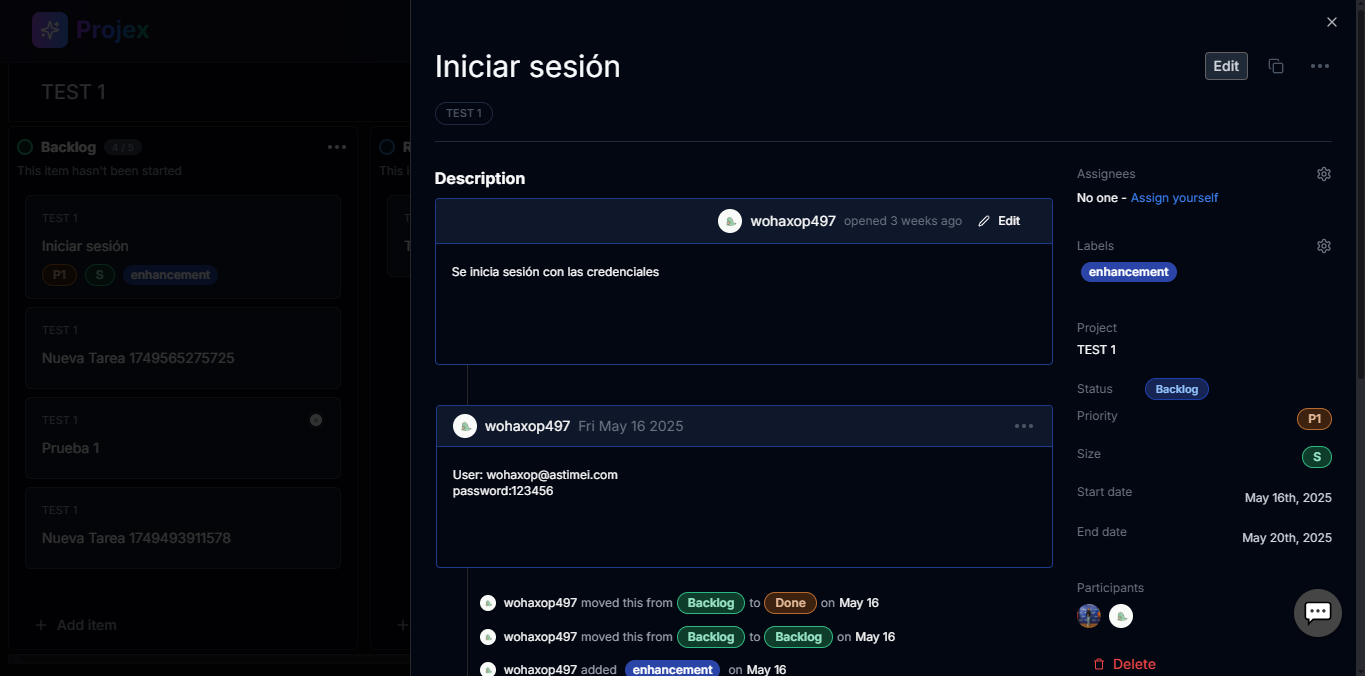
* .Predice duración promedio según tipo de tarea.
* Muestra sugerencia al momento de crear nueva tarea.
* Permite aceptar o modificar la predicción.

### Resultados del sprint

#### Evidencias.

En la imagen se puede visualizar el estado de las historias de usuario del sprint 3, 1 historia de usuario finalizado, 1 por hacer, y dos en curso, también se puede visualizar los puntos de historia, y el miembro del equipo designado a realizar las historias de usuario.

*Figura 21. Componentes del sprint 3 en Jira*

Esta corresponde a la evidencia en la aplicación donde se puede visualizar en la columna derecha la clasificación de las tareas por prioridad, tamaño, estado y la duración, también hay una imagen para un chatbot de soporte, gestionada por la IA de gemini.

*Figura 22. Resultados del sprint 3 en Jira*

#### Prueba de desarrollo.

Se realizaron pruebas unitarias de caja negra a la historia de usuario de Asistente virtual, y pruebas de integración para fusionar con el sprint 1.

### Sprint retrospective

En la reunión del equipo de desarrollo se pudo constatar el resultado parcial del sprint ya que no se culminaron con todas las historias de usuario, culminando solo 1, que es el asistente virtual.

Las mejoras planteadas al sprint 3 sería culminar con el desarrollo completo del sprint 1, ya que está relacionado con el asistente virtual, y agregarían más funcionalidades a la aplicación, como clasificar y predecir tareas de forma automática.

## Desarrollo del Sprint 4

### Sprint planning

El sprint 4 tiene 5 historias de usuario, relacionadas a la gestión de permisos y usuarios y despliegue del entorno.

El despliegue del entorno se realizó en Vercel, ya que es una plataforma en la nube especializada en el desarrollo, despliegue y alojamiento de aplicaciones web.

### Sprint backlog

El sprint backlog se compone con las funcionalidades principales que corresponden al MVP 3, cuenta con 5 historias de usuario y 2 historias técnicas..

### Historias de usuarios

HU12 - Inicio de sesión

HU09 - Dashboards

HU16 - Perfiles personalizados

HU10 - Gestión de usuarios

HU11 - Gestión de permisos

### Taskboard

Para visualizar el seguimiento del desarrollo del sprint 4 se utilizó la aplicación Jira, en el cual se puede visualizar las 5 historias de usuario y 2 dos historias técnicas.

Para visualizar el estado de las historias de usuario se utilizó Jira, ya que de acuerdo a cada cambio realizado se fué cambiando de estado a las historias de usuario.

### Daily scrum

El equipo de desarrollo y product owner para coordinar las actividades se utilizaron aplicaciones como whatsapp y reuniones presenciales, además de aplicaciones de gestión de proyectos Kanban en Jira para organizar el estado y cronograma de desarrollo de los sprints y las historias de usuario, se sigue el curso del cronograma utilizado en microsoft project, en el desarrollo de los sprint se logró resultados positivos, ya que se completaron algunos sprints antes de tiempo.

### Sprint review

Los miembros del equipo de desarrollo conformado por Sofía, Alan, Jaci y Carlos, designados al sprint 4, culminaron de forma parcial las historias de usuario del sprint, sin hacer alteraciones en la idea principal de las historias de usuario.

Las historias de usuario culminado fueron inicio de sesión, dashboard y perfiles personalizados, y las historias de usuario de gestión de usuario y gestión de premios aún están en tareas por hacer, como resultado se entregaron avances a tiempo.

### Criterios de aceptación

HU12 - Inicio de sesión

* Se debe ingresar correo y contraseña válidos.
* Si las credenciales son incorrectas, se muestra un mensaje de error claro.
* Se debe permitir recordar sesión (opcional).

HU09 - Dashboards

* Muestra el número de tareas por estado (ToDo, En Progreso, Finalizadas).
* Muestra progreso por sprint o proyecto.
* Gráficas de barras o circulares interactivas.

HU16 - Perfiles personalizados

* Permitir editar foto de perfil.
* Poder cambiar información personal y mi rol.
* poder editar los temas que se ajusten a mis gustos.

HU10 - Gestión de usuarios

* CRUD completo para usuarios (crear, leer, actualizar, eliminar).
* Validación de roles: solo admins pueden eliminar usuarios.
* Cada usuario debe tener nombre, correo, rol y estado.

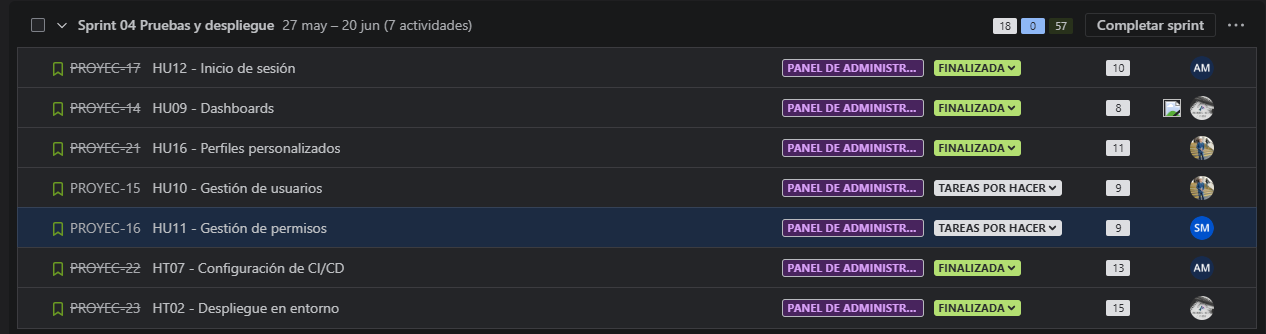
HU11 - Gestión de permisos

* Definir roles y permisos por cada módulo.

### Resultados del sprint

#### Evidencias.

En la imagen se puede visualizar el estado de las historias de usuario, los puntos de historia, el miembro de equipo designado al desarrollo y la fecha de duración de las actividades.



*Figura 23. Componentes del sprint 4 en Jira*

En la siguiente imagen se puede observar el dashboard de las tareas en la aplicación de ProjeX, organizado por tipo de gráfico, características y agrupados por tamaño, prioridad o estado.



*Figura 24. Resultados del sprint 4*

#### Prueba de desarrollo.

En el sprint 4 se realizaron las siguientes pruebas de desarrollo

* Pruebas unitarias: realizaron pruebas manuales de caja negra.
* Pruebas E2E: se utilizó Playwright para simular el comportamiento del usuario en uso de la aplicación.
* Pruebas de rendimiento: Se utilizó Lighthouse para medir la velocidad de carga, eficiencia y respuesta del sistema.
* Pruebas de aceptación: Se realizó pruebas de caja negra para verificar que el sistema cumpla con las funcionalidades principales.
* Pruebas de integración: Pruebas manuales de caja negra para verificar que no haya errores en la integración con los sprints anteriores.

### Sprint retrospective

En la reunión del equipo de desarrollo se pudo constatar el resultado parcial del sprint ya que no se culminaron con todas las historias de usuario, culminando solo 1, que es el asistente virtual.

Las mejoras planteadas al sprint 3 sería culminar con el desarrollo completo del sprint 1, ya que está relacionado con el asistente virtual, y agregarían más funcionalidades a la aplicación, como clasificar y predecir tareas de forma automática.

# CAPÍTULO 7

# PRUEBAS DE SOFTWARE

## Plan de Pruebas

El plan de pruebas constituye una etapa crítica dentro del ciclo de desarrollo de software, pues garantiza que el producto cumpla con los requisitos funcionales y no funcionales previamente definidos. En este proyecto, se aplicaron pruebas tanto manuales como automáticas sobre distintos componentes de la aplicación web de gestión de proyectos desarrollada con Next.js 15, Supabase y otras tecnologías complementarias.

### 7.1.1. Objetivos de las pruebas

* Verificar que las funcionalidades principales de la aplicación operen correctamente bajo condiciones normales y excepcionales.
* Identificar errores lógicos, de flujo, integración o visuales antes de la entrega final.
* Validar la experiencia de usuario, tiempos de respuesta y consistencia de datos.

### 7.1.2. Tipos de Pruebas Aplicadas

| **Tipo de Prueba** | **Objetivo Principal** | **Herramientas Utilizadas** |
| --- | --- | --- |
| **Pruebas Unitarias** | Verificar que funciones y componentes individuales funcionen correctamente. | Pruebas manuales |
| **Pruebas E2E** | Simular el comportamiento del usuario final a lo largo de procesos completos. | Playwright |
| **Pruebas de Rendimiento** | Medir la velocidad de carga, respuesta y eficiencia del sistema. | Lighthouse |
| **Pruebas de Aceptación** | Verificar que el sistema cumpla con las necesidades y criterios del usuario final. | Pruebas manuales con usuarios |

*Tabla 13: Tipos de pruebas aplicadas*

## 7.2. Pruebas Unitarias

Las pruebas unitarias se realizaron de forma manual sobre funciones y componentes individuales del sistema, con especial énfasis en los formularios, validaciones y funciones auxiliares. Estas pruebas permitieron verificar que cada componente React se comportara correctamente de manera aislada, asegurando que su lógica interna y respuestas ante distintos escenarios fueran las esperadas.

| **Funcionalidad** | **Descripción** | **Objetivo de la prueba** | **Tipo de prueba** | **Estado** | **Fecha de aprobación** | **Aprobado por** |
| --- | --- | --- | --- | --- | --- | --- |
| Registro de proyectos | Permite crear nuevos proyectos desde la interfaz principal | Verificar que los datos se almacenen correctamente en Supabase | Unitaria | Aprobado | 6/06/2025 | Jaci Lucas |
| Edición de proyectos | Modifica el título o descripción de un proyecto existente | Confirmar que los cambios se reflejan en tiempo real | Unitaria | Aprobado | 6/06/2025 | Jaci Lucas |
| Eliminación de proyectos | Elimina un proyecto y sus tareas asociadas | Validar que se borren correctamente y no se genere error en la UI | Unitaria | Aprobado | 6/06/2025 | Jaci Lucas |
| Registro de tareas dentro de proyectos | Permite agregar tareas específicas a un proyecto | Asegurar la correcta relación entre tareas y proyectos | Unitaria | Aprobado | 6/06/2025 | Jaci Lucas |
| Edición de tareas | Modifica campos como título, descripción o prioridad de la tarea | Verificar que los datos se actualicen sin errores | Unitaria | Aprobado | 6/06/2025 | Jaci Lucas |
| Eliminación de tareas | Elimina tareas de un proyecto | Validar la eliminación lógica y visual sin afectar otras tareas | Unitaria | Aprobado | 6/06/2025 | Jaci Lucas |
| Gestión de columnas Drag & Drop (DnD Kit) | Reorganiza tareas dentro de columnas con soporte drag and drop | Evaluar el correcto comportamiento del orden de tareas | Unitaria | Aprobado | 6/06/2025 | Jaci Lucas |
| Autenticación de usuarios (registro/login) | Registro de nuevos usuarios y validación de acceso | Comprobar funcionamiento con Supabase Auth | Unitaria | Aprobado | 6/06/2025 | Jaci Lucas |
| Visualización de dashboard con gráficas | Muestra estadísticas del proyecto en formato gráfico (Recharts) | Verificar la carga y precisión de los datos visualizados | Unitaria | Aprobado | 6/06/2025 | Jaci Lucas |
| Envío de notificaciones por correo (Resend) | Envía notificaciones de tareas por correo electrónico | Confirmar el envío y recepción exitosa | Unitaria | Aprobado | 6/06/2025 | Jaci Lucas |
| Editor enriquecido de tareas (Tiptap) | Permite dar formato a descripciones de tareas (negrita, listas, enlaces, etc.) | Validar que el editor funcione correctamente y guarde la información | Unitaria | Aprobado | 6/06/2025 | Jaci Lucas |

*Tabla 14: Pruebas unitarias*

## 7.3 Pruebas de Extremo a Extremo (E2E)

Se implementaron pruebas E2E utilizando **Playwright** con TypeScript para validar las funcionalidades críticas de la plataforma colaborativa. Estas pruebas simulan interacciones reales del usuario dentro del navegador, permitiendo verificar que el sistema responde correctamente desde la interfaz hasta el backend.

#### Herramientas Utilizadas

* **Framework:** Playwright
* **Lenguaje:** TypeScript
* **Estrategia:** Pruebas automatizadas por flujo de usuario
* **Cobertura:** Autenticación, creación de proyectos, creación de tareas y eliminación de proyectos.

#### Pruebas Implementadas

1. **login.test.ts**
   * **Objetivo:** Verificar que un usuario pueda iniciar sesión con credenciales válidas.
   * **Pasos cubiertos:**
     + Acceso a la página de login.
     + Ingreso de correo y contraseña.
     + Confirmación de redirección al dashboard o home.
   * **Resultado esperado:** Usuario autenticado correctamente.
   * [**Link al video**](https://drive.google.com/file/d/1XppYJ4_tF3dvOb-kyk3C9B8003we5A-G/view?usp=sharing)
2. **new-project.test.ts**
   * **Objetivo:** Validar que se pueda crear un nuevo proyecto correctamente.
   * **Pasos cubiertos:**
     + Acceso al formulario de nuevo proyecto.
     + Ingreso de nombre, descripción y envío del formulario.
     + Confirmación de que el proyecto aparece en la lista.
   * **Resultado esperado:** Proyecto creado y visible.
   * [**Link al video**](https://drive.google.com/file/d/1nvl7tCs_DH_KMyBbHQkKzzm-LGeCiZDP/view?usp=sharing)
3. **new-task.test.ts**
   * **Objetivo:** Comprobar que se puede agregar una nueva tarea dentro de un proyecto.
   * **Pasos cubiertos:**
     + Selección de un proyecto existente.
     + Apertura del modal de nueva tarea.
     + Ingreso de detalles (título).
     + Verificación de que la tarea aparece.
   * **Resultado esperado:** Tarea registrada correctamente en el proyecto.
   * [**Link al video**](https://drive.google.com/file/d/1nvl7tCs_DH_KMyBbHQkKzzm-LGeCiZDP/view?usp=sharing)
4. **delete-project.test.ts**
   * **Objetivo:** Confirmar que un usuario puede eliminar un proyecto existente.
   * **Pasos cubiertos:**
     + Cierre del proyecto.
     + Selección del proyecto a eliminar.
     + Confirmación de la acción.
     + Verificación de que el proyecto desapareció de la lista.
   * **Resultado esperado:** Proyecto eliminado de forma efectiva.
   * [**Link al video**](https://drive.google.com/file/d/1hgnwBfBSi1jCNfU0AlDkqG1yHG0APXrc/view?usp=sharing)

#### Buenas Prácticas Aplicadas

* Uso de test.describe() y test.step() para una mejor organización de los casos.
* Esperas explícitas para evitar falsos negativos por cargas asincrónicas.
* Datos de prueba controlados para mantener la consistencia.

## 7.4 Pruebas de Rendimiento

Se realizó una auditoría de rendimiento utilizando Lighthouse 12.6.0 sobre la plataforma<https://colaborative-platform.vercel.app/> bajo un entorno de escritorio emulado. Los resultados muestran un rendimiento sobresaliente en casi todos los aspectos evaluados, lo que refleja una experiencia de usuario fluida, accesible y bien optimizada. A continuación, se detallan los hallazgos principales:

#### Resumen de Resultados

* **Performance:** 99/100
* **Accesibilidad:** 100/100
* **Buenas Prácticas:** 100/100
* **SEO:** 92/100

#### Métricas de Carga Clave

* **First Contentful Paint (FCP):** 0.4 s
* **Largest Contentful Paint (LCP):** 0.7 s
* **Total Blocking Time (TBT):** 20 ms
* **Cumulative Layout Shift (CLS):** 0
* **Speed Index (SI):** 0.9 s

Estas métricas indican que el contenido principal se presenta rápidamente, con una carga visual estable y sin bloqueos perceptibles para el usuario.

#### Fortalezas Técnicas

* Las imágenes están adecuadamente optimizadas (uso de formatos modernos como WebP).
* JavaScript y CSS minificados correctamente.
* Implementación de técnicas modernas como lazy loading, preconexiones y compresión de texto.
* Se emplea HTTP/2 para mejorar la eficiencia en la entrega de recursos.
* El tiempo de respuesta del servidor inicial es corto (110 ms).
* Todas las animaciones están compuestas y se usan listeners pasivos para mejorar el rendimiento al hacer scroll.

#### Accesibilidad

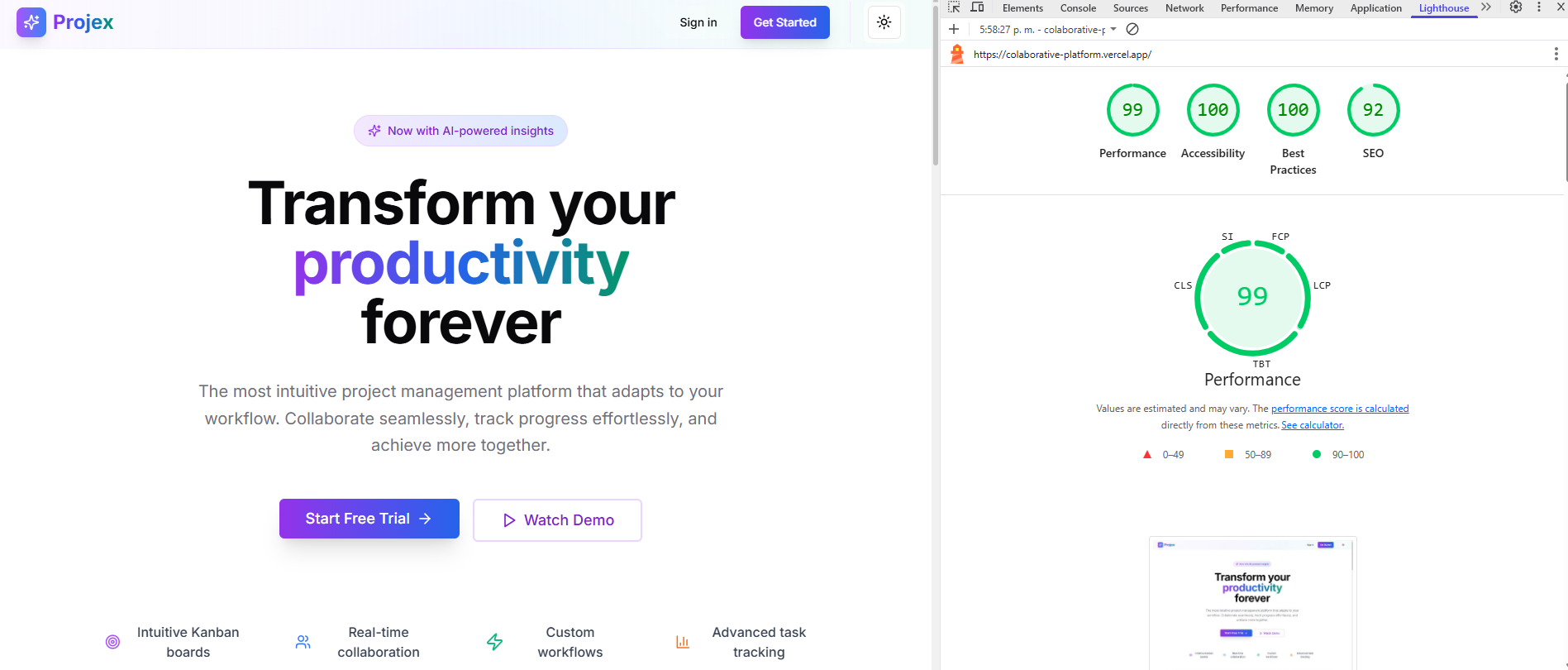
Todos los elementos interactivos son accesibles mediante teclado, con una estructura lógica de navegación y soporte para tecnologías de asistencia. La página cumple con todas las recomendaciones automáticas, aunque se sugiere una revisión manual adicional para garantizar la experiencia inclusiva.

#### Buenas Prácticas

Se cumplen todas las recomendaciones sobre seguridad y confianza: se mitigan riesgos como XSS, clickjacking y aislamiento de origen, con una política efectiva de contenido (CSP).

#### SEO (Optimización para Motores de Búsqueda)

El puntaje SEO fue de 92/100. La mayoría de los criterios están correctamente implementados, incluyendo etiquetas <title>, <meta description>, texto alternativo para imágenes, enlaces rastreables y hreflang válido. No obstante, se detectó un error en el archivo robots.txt, el cual debe corregirse para asegurar que los motores de búsqueda puedan indexar correctamente el sitio.



*Figura 25. SEO (Optimización para Motores de Búsqueda*

## 7.5 Pruebas de Aceptación

Finalmente, se realizaron pruebas de aceptación para asegurar que el sistema cumple con los requisitos del usuario final. Estas pruebas fueron diseñadas en base a los requerimientos funcionales especificados previamente y se ejecutaron en conjunto con usuarios clave.

| **ID** | **Historia de Usuario** | **Tarea Relacionada** | **Criterio de Aceptación** | **Resultado Esperado** | **Resultado Obtenido** | **Estado** |
| --- | --- | --- | --- | --- | --- | --- |
| PA-01 | HU01: Tener un asistente virtual | PROYEC-6: Asistente virtual | El asistente responde y sugiere tareas con IA | Asistente funcional y útil en sugerencias | Coincide con lo esperado | Aprobado |
| PA-02 | HU13: Editar proyectos | PROYEC-18: Editar proyectos | El usuario puede modificar nombre, integrantes y descripción | Proyecto editable desde interfaz | Coincide con lo esperado | Aprobado |
| PA-03 | HU12: Iniciar sesión | PROYEC-17: Inicio de sesión | Autenticación funcional, errores controlados | Acceso exitoso y manejo de errores | Coincide con lo esperado | Aprobado |
| PA-04 | HU05: Chatear en tiempo real | PROYEC-11: Chat en tiempo real | Chat funcional y en tiempo real | Comunicación fluida sin recarga | Coincide con lo esperado | Aprobado |
| PA-05 | HU14: Asignar tareas | PROYEC-19: Asignación de tareas | Asignación por usuario, estado y prioridad | Tareas correctamente asignadas y visibles | Coincide con lo esperado | Aprobado |
| PA-06 | HU09: Visualizar dashboards | PROYEC-14: Dashboards | Gráficas con tareas por estado y usuarios activos | Dashboard dinámico con métricas correctas | Coincide con lo esperado | Aprobado |
| PA-07 | HU02: Recibir predicción de tareas | PROYEC-7: Predicción de tareas | IA recomienda tareas basadas en historial y proyecto | Sugerencias relevantes en la interfaz | Coincide con lo esperado | Aprobado |
| PA-08 | HU04: Recibir recordatorios | PROYEC-4: Recordatorio de tareas | Notificaciones automáticas en fechas clave | Notificaciones activadas y visibles | Coincide con lo esperado | Aprobado |
| PA-09 | HU15: Gestionar usuarios | PROYEC-10: Gestión de usuarios | CRUD de usuarios con asignación de roles | Alta, edición y baja de usuarios con roles | Coincide con lo esperado | Aprobado |
| PA-10 | HU06: Notificación de mensajes | PROYEC-16: Notificación de mensajes | Alertas push o internas cuando hay nueva actividad | Notificaciones inmediatas y claras | Coincide con lo esperado | Aprobado |
| PA-11 | HU15: Ver tareas asignadas | PROYEC-20: Visualización de tareas | Lista clara y filtrable por estado y prioridad | Vista ordenada de tareas asignadas | Coincide con lo esperado | Aprobado |
| PA-12 | HU03: Clasificar tareas | PROYEC-10: Clasificación de tareas | Permite etiquetar o categorizar tareas por tipo o urgencia | Clasificación activa en interfaz de tareas | Coincide con lo esperado | Aprobado |
| PA-13 | HU11: Asignar permisos | PROYEC-1: Gestión de permisos | Roles con permisos definidos para cada módulo | Restricción de acceso según el rol | Coincide con lo esperado | Aprobado |
| PA-14 | HU07: Eliminar conversaciones | PROYEC-12: Eliminar conversaciones | Posibilidad de borrar mensajes propios o por admin | Eliminación efectiva con permisos válidos | Coincide con lo esperado | Aprobado |
| PA-15 | HU08: Ver lista de conversaciones | PROYEC-13: Lista de conversaciones | Vista organizada por canal, usuario o tema | Historial accesible por filtros | Coincide con lo esperado | Aprobado |
| PA-16 | HU16: Tener perfil personalizado | PROYEC-6: Perfiles personalizados | Personalización de nombre, foto, rol, etc. | Perfil editable por el usuario | Coincide con lo esperado | Aprobado |
| PA-17 | HTO1: Realizar pruebas unitarias | PROYEC-22: Pruebas unitarias | Cada componente funcional debe tener pruebas con resultados esperados | Resultados reproducibles en cada módulo | Coincide con lo esperado | Aprobado |
| PA-18 | HTO2: Desplegar la aplicación | PROYEC-23: Despliegue | El sistema está en línea y funcional sin errores | Sistema accesible en entorno productivo | Coincide con lo esperado | Aprobado |

*Tabla 15: Pruebas de aceptació*

# CONCLUSIONES

1. **El proyecto permitió implementar una plataforma colaborativa web utilizando tecnologías modernas**, como Next.js, Supabase, Express.js y herramientas de inteligencia artificial como Gemini AI, cumpliendo con los objetivos funcionales principales en los Sprints 1 y 4.
2. **Durante el Sprint 1 se logró con éxito el desarrollo de la base del sistema**, incluyendo la gestión de proyectos y tareas, la integración del entorno MERN (con variantes) y un primer modelo de IA para asistencia básica, lo que aseguró un MVP inicial funcional.
3. **En el Sprint 2 se identificó limitaciones en el tiempo y complejidad técnica**, lo que impidió culminar la funcionalidad completa del chat en tiempo real y las notificaciones. El uso de Supabase Realtime resultó adecuado, pero la integración quedó incompleta.
4. **El Sprint 3 permitió avanzar en la integración de funcionalidades basadas en IA**, destacando la entrega del asistente virtual. Sin embargo, funciones clave como la predicción y clasificación de tareas no se lograron culminar dentro del tiempo planificado.
5. **El Sprint 4 fue clave para completar el despliegue y las funciones administrativas**, como dashboards, login y perfiles personalizados. Sin embargo, algunas historias de usuario asociadas a la gestión de usuarios y permisos quedaron pendientes, afectando la cobertura total del MVP 3.

# RECOMENDACIONES

1. **Mejorar el diseño y visualización de los dashboards y gráficos** dentro de la plataforma para facilitar la interpretación de datos por parte de los usuarios.
2. Completar e **integrar las funcionalidades pendientes**, principalmente las relacionadas con la comunicación en tiempo real (Sprint 2) y el control de usuarios y permisos (Sprint 4), para fortalecer el funcionamiento integral del sistema.
3. **Automatizar las pruebas unitarias y de integración**, utilizando frameworks como Jest o Mocha para backend, y Playwright o Cypress para frontend, asegurando mayor cobertura y detección temprana de errores.
4. **Ampliar la validación en el sistema para casos críticos**, como evitar la creación de proyectos duplicados con el mismo nombre y reforzar las restricciones en las fechas y estados de tareas.
5. **Dedicar sesiones específicas de capacitación y refuerzo tecnológic**o, en especial en el uso de control de versiones, integración de IA y configuración de entornos, para reducir los retrasos asociados a la curva de aprendizaje.

# ANEXOS

1. *Universidad Continental (2017).Organigrama de la Universidad Continental. Nota: Tomado de Información interna. Plan de Seguridad Institucional.Huancayo-Perú. Disponible en :* <https://repositorio.continental.edu.pe/>

## Anexo 01. Manual Técnico

## Anexo 02. Manual de Usuario