

"Año de la recuperación y consolidación de la economía peruana"



**Universidad
Continental**

Facultad de Ingeniería de Sistemas E Informática

"CONSIGNA ELABORACIÓN DE LA ITERACIÓN CERO"

ASIGNATURA: TALLER DE PROYECTOS 2 - INGENIERÍA DE
SISTEMAS E INFORMÁTICA

DOCENTE: GAMARRA MORENO, JOB DANIEL

NRC: 62102

ESTUDIANTE:

Apellidos y Nombres	CODIGO
BELITO RAMIREZ MORI OCTAVIO	74902137
CORONEL BURGOS JAVIER DANIEL	71997263
CRESPO SOTO FRED JHOSSEPH	70776761
GUEVARA QUINTANA JENKINS JHOSEPH	74804104
ORE JIMENEZ JEISON JUAN	70345982
TACUNAN PALOMINO JEAN POOL	71549818
TORRES AGUILAR CIELO ALESSANDRA	73760659

HUANCAYO - 2025

Contenido


1. Objetivo.....	¡Error! Marcador no definido.
2. Consigna	¡Error! Marcador no definido.
2.1. Project Charter (Acta de Proyecto)	3
2.1.1. Propósito y justificación del proyecto.....	3
2.1.2. Objetivos generales.....	4
2.1.3. Alcance Inicial	5
2.1.4. Supuestos y restricciones.....	6
2.1.5. Identificación de los interesados (stakeholders).....	10
2.1.6. Roles y responsabilidades del equipos.....	8
2.2. Backlog inicial.....	11
2.2.1. Identificación del Epic central y descomposición en historias.	11
2.2.2. Lista de 6 a 8 historias de usuario priorizadas con INVEST.	12
2.3. Infraestructura y entorno técnico	15
2.3.1. Creación de repositorio en GitHub (un integrante crea, el resto clona).....	15
2.3.2. Configuración inicial de ramas (main, develop, ramas por funcionalidad).....	16
2.3.3. Estructura base del proyecto MERN (carpetas, dependencias mínimas).	16
2.4. Procesos de equipo	16
2.4.1. Definición de canales de comunicación (ej. Slack, Discord).....	16
2.4.2. Acuerdos de trabajo: horarios de reunión, Definition of Done, roles iniciales.) ...	17
2.5. Arquitectura mínima (Walking Skeleton).....	18
2.5.1. Configuración básica: conexión Express–MongoDB, Hello World en React.	18
2.5.2. Evidencia de despliegue inicial en un servicio gratuito (ej. Render, Vercel, Railway).19	
2.6. Entregables.....	19
2.6.1. Documento (Markdown, PDF o Google Docs) con:	19
2.6.2. URL del repositorio GitHub con:.....	19
2.6.3. Evidencia del despliegue inicial (screenshot o URL):.....	20

CONSIGNA ELABORACIÓN DE LA ITERACIÓN CERO

1.1. Project Charter (Acta de Proyecto)

PROJECT CHARTER

Título del Proyecto: Asistente de investigación académica con IA y automatización (AI Research Assistant MERN)

Patrocinador del Proyecto: Facultad de Ingeniería de Sistemas e Informática – Universidad Continental	Fecha de Preparación: 25/08/2025
Gerente del Proyecto: Coronel Burgos Javier Daniel	Cliente del Proyecto: Universidad Continental
Logo: 	Nodo Central <ul style="list-style-type: none">• Un círculo central (la app) conectado a varios nodos más pequeños (investigadores).• Minimalista, tipo diagrama de red, con colores azules y verdes.

Visión del proyecto:

Para investigadores y estudiantes universitarios, identifica al cliente interesado en optimizar sus proyectos académicos.

Para quienes necesitan reducir el tiempo dedicado a búsquedas bibliográficas, organización y redacción, ofrece más detalles.

El Asistente de Investigación Académica (IA) con IA **describe el producto** como una aplicación web colaborativa basada en IA.

Permite sugerir artículos relevantes, generar resúmenes automáticos y recibir asistencia metodológica. Identifica las funciones del producto.

A diferencia de los gestores de referencia tradicionales o las bibliotecas digitales, que describen productos de la competencia que solo almacenan documentos, **nuestro producto** integra inteligencia artificial, automatización del flujo de trabajo y métricas de productividad, lo que proporciona factores diferenciadores para mejorar la calidad del trabajo en equipo.

1.1.1. Propósito y justificación del proyecto

El propósito del proyecto es desarrollar una aplicación web full-stack basada en el stack MERN (MongoDB, Express.js, React.js y Node.js) que integre

funcionalidades de inteligencia artificial (IA) y automatización de procesos, con el fin de apoyar a investigadores, estudiantes y equipos académicos en la gestión colaborativa de proyectos de investigación.

La solución permitirá optimizar el ciclo de investigación al ofrecer herramientas innovadoras como:

- Sugerencias automáticas de artículos relevantes mediante integración con APIs académicas (Google Scholar, PubMed).
- Chatbot con IA que brinde orientación en metodología de investigación y resolución de dudas.
- Resúmenes automáticos de papers con técnicas de procesamiento de lenguaje natural (NLP).

Flujos de automatización con n8n, que reduzcan el tiempo invertido en búsquedas, descargas y notificaciones.

La justificación se justifica con la necesidad de reducir la sobrecarga de información y el tiempo que los investigadores destinan a tareas repetitivas, permitiéndoles enfocarse en el análisis y la generación de nuevo conocimiento. Asimismo, fomenta el trabajo colaborativo al permitir compartir bibliotecas académicas y acceder a métricas de productividad a través de un dashboard administrativo. Además, la aplicación promueve la sostenibilidad digital, al minimizar el uso de papel y aprovechar la contenerización con Docker para optimizar recursos computacionales, garantizando escalabilidad, seguridad y accesibilidad desde diferentes dispositivos. Con ello, se contribuye a elevar la calidad de los proyectos de investigación y a impulsar la innovación académica en un entorno digital eficiente.

Descripción del Proyecto de alto Nivel:

El sistema permitirá a investigadores y estudiantes:

- Gestionar proyectos académicos en equipo.
- Automatizar búsquedas bibliográficas en bases de datos académicas.
- Resumir papers con IA y sugerir fuentes relevantes.
- Usar un chatbot especializado en metodología y redacción científica.
- Recibir notificaciones automatizadas mediante flujos de n8n.

1.1.2. Objetivos generales.

Objetivos	Fecha de Vencimiento
Desarrollar una aplicación web full-stack con el stack MERN (MongoDB, Express.js, React.js, Node.js) que permita la gestión colaborativa de proyectos académicos, entregando un MVP funcional al finalizar la semana 15.	Semana 1 – 15

Implementar un chatbot con IA (OpenAI API / Hugging Face) capaz de responder al menos 80% de las consultas metodológicas de los usuarios en menos de 10 segundos por interacción.	Semana 8
Integrar un módulo de resúmenes automáticos de papers con modelos NLP (Transformers) que genere resúmenes de máximo 300 palabras con un nivel de precisión aceptado por al menos el 70% de usuarios piloto.	Semana 8
Configurar y desplegar flujos de automatización con n8n que realicen búsquedas diarias en APIs académicas y notifiquen a los usuarios con una tasa de éxito $\geq 90\%$ en las ejecuciones.	Semana 4
Contenerizar la solución con Docker y docker-compose para garantizar que el 100% de los servicios (frontend, backend, base de datos e IA) puedan levantarse correctamente en entornos locales y en la nube.	Semana 12
Desarrollar y ejecutar pruebas unitarias y E2E (Jest y Cypress) con al menos un 70% de cobertura de código validada en reportes de CI/CD.	Semana 14
Documentar el sistema en manuales de usuario, informes técnicos y métricas de sostenibilidad digital, entregando un compendio final validado por el equipo al cierre del proyecto.	Semana 1 – 15

1.1.3. Alcance Inicial

Incluye: <ul style="list-style-type: none"> • Desarrollo frontend (React.js, responsive, Redux/Context API). • Desarrollo backend (Node.js + Express.js). • Base de datos en MongoDB Atlas. • IA (chatbot + resúmenes automáticos con NLP). • Automatización con n8n (búsquedas → descarga → resumen → notificación). • Pruebas unitarias y E2E con Jest y Cypress. • Documentación técnica y manual de usuario. 	Excluye: <ul style="list-style-type: none"> • Desarrollo de hardware propio. • Integración con sistemas de pago o transacciones financieras. • Garantía de acceso a todas las bases académicas privadas (se limita a APIs abiertas).
--	--

1.1.4. Supuestos y restricciones.

Supuestos

- Los investigadores y estudiantes cuentan con acceso a internet estable para usar la aplicación.
- Las APIs académicas (Google Scholar, PubMed, etc.) estarán disponibles y permitirán la integración sin cambios drásticos durante el desarrollo.
- El equipo de proyecto tendrá la dedicación necesaria (aprox. 40% de su tiempo) para cumplir con las actividades en los plazos previstos.
- Las herramientas de IA seleccionadas (OpenAI API, Hugging Face, Transformers) funcionarán con los niveles de precisión esperados.
- Los usuarios finales (investigadores/estudiantes) participarán activamente en pruebas piloto y retroalimentación.
- El presupuesto estimado cubrirá los costos de infraestructura (servidores cloud, licencias y APIs).
- La Universidad Continental aportará recursos propios para el financiamiento total del proyecto.

Restricciones

- Tiempo: El proyecto debe completarse en un plazo máximo de 15 semanas
- Alcance: El producto mínimo viable (MVP) debe incluir obligatoriamente:
 - Chatbot con IA.
 - Resúmenes automáticos de papers.
 - Búsquedas automatizadas en APIs académicas.
- Presupuesto: El costo total del proyecto no podrá superar los S/ 193,861.25 (incluyendo reservas).
- Tecnología: La aplicación debe desarrollarse en el stack MERN y contenerizarse con Docker, sin sustitución por otros frameworks principales.
- Seguridad: El sistema debe implementar autenticación y manejo de roles para garantizar la protección de datos de los usuarios.
- Recursos Humanos: El equipo está limitado a los 7 integrantes ya definidos (Project Manager, Scrum Master/Programador, Programadores, Tester, Informes/Auditoría).
- Regulación: Se deben cumplir las normas de protección de datos personales aplicables (ej. GDPR/LPDP Perú).

Límites del Proyecto:

Incluye: gestión de proyectos, integración de IA, automatización con n8n, contenedores Docker, control de versiones en GitHub.

No incluye: publicación directa en revistas indexadas ni acceso ilimitado a bases de datos de pago.

Resultados Clave:

- Aplicación web full-stack MERN.
- Integración de IA (resúmenes y chatbot).
- Flujos automatizados con n8n.
- Documentación técnica y manual de usuario.
- Pruebas unitarias y E2E (cobertura mínima 70%).

Requerimientos de alto Nivel:

- Frontend en React con diseño responsive.
- Backend en Node.js + Express.
- Base de datos MongoDB.
- IA con OpenAI API / Transformers.
- Automatización con n8n.
- Contenerización con Docker.
- Git Flow para control de versiones.

Riesgos Iniciales del Proyecto:

- Riesgo de dependencia en APIs externas (Google Scholar, OpenAI).
- Complejidad en la integración MERN + IA + n8n.
- Riesgo de retrasos por curva de aprendizaje en nuevas tecnologías.

Presupuesto:

- El presupuesto total estimado del Proyecto 2 asciende a **S/ 25,000.25**, incluyendo la línea base de costos, la reserva de contingencia (15%) y la reserva de gestión (10%).
- <https://docs.google.com/spreadsheets/d/10QvCZsCAc933BNbQxletTLaPMmVKdlQd/edit?usp=sharing&oid=110211996957209144380&rtpof=true&sd=true>

Plazos Tentativos

Resumen de hitos	Criterios de éxito
Semana 1-2: Acta de constitución y diseño arquitectónico.	<ul style="list-style-type: none">• Sistema funcional en entorno Docker.
Semana 3-5: Desarrollo backend y base de datos.	<ul style="list-style-type: none">• Chatbot y resúmenes automáticos funcionando.

Semana 6-8: Desarrollo frontend y conexión con backend.

Semana 9-10: Integración de IA y automatización.

Semana 11: Pruebas automatizadas y documentación.

Semana 12: Presentación final.

- Automatización de flujos académicos implementada.
- Pruebas superadas con >70% cobertura.
- Aceptación por parte de docentes y estudiantes.

Entregables Principales:

Documento de arquitectura y diseño técnico.

Repositorio en GitHub con documentación (README, Wiki, Issues).

Prototipo de frontend con login, dashboard y biblioteca compartida.

API backend funcional con CRUD de proyectos y usuarios.

Chatbot académico (IA).

Módulo de resúmenes automáticos de papers.

Flujo de automatización con n8n integrado.

Contenerización con Docker + docker-compose.yml.

Pruebas automatizadas (70% mínimo de cobertura).

Informe final con métricas de uso y sostenibilidad digital.

Recursos Requeridos:

- Laptops
- Servidor en la Nube
- Licencias de APIs IA (ej. OpenAI API).
- Contenedores Docker y repositorio GitHub.
- Herramientas de gestión de proyectos (Trello/Jira).

1.1.5. Roles y responsabilidades del equipos.

Project Manager:

Coronel Burgos Javier Daniel

- ✓ Responsable de la planificación, ejecución y control del proyecto.
- ✓ Coordina la comunicación con el patrocinador y stakeholders principales.
- ✓ Administra recursos, tiempos, riesgos y costos.

- ✓ Toma decisiones estratégicas y aprueba cambios de alcance.
- ✓ Asegura que el proyecto cumpla con los objetivos planteados y entregue valor.

Scrum Master / Programador:

Crespo Soto Fred Jhossep

- ✓ Facilita la metodología ágil (Scrum), eliminando impedimentos del equipo.
- ✓ Organiza las reuniones (daily, sprint planning, review, retrospectiva).
- ✓ Asegura que se respeten las prácticas ágiles y fomenta la mejora continua.
- ✓ Como programador, apoya en el desarrollo backend y automatización con n8n.

Programador / Especialista en IA

Belito Ramirez Mori Octavio

- ✓ Encargado de diseñar e implementar los módulos de inteligencia artificial:
 - ✓ Chatbot académico (OpenAI/Hugging Face).
 - ✓ Resúmenes automáticos con NLP/Transformers.
- ✓ Integra modelos de machine learning al backend.
- ✓ Colabora en la arquitectura de datos y optimización de algoritmos.

Programador

Tacunan Palomino Jean Pool

- ✓ Desarrollador frontend y backend con el stack MERN.
- ✓ Diseña pantallas en React.js con responsive design.
- ✓ Implementa la conexión con la API (Node.js/Express.js) y la base de datos MongoDB.
- ✓ Desarrollador frontend y backend con el stack MERN.

Tester



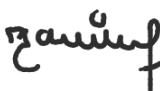



Guevara Quintana Jenkins Jhoseph

- ✓ Diseña y ejecuta planes de prueba (unitarias, integración, E2E).
- ✓ Usa Cypress/Playwright para validar la funcionalidad end-to-end.
- ✓ Verifica criterios de aceptación de cada historia de usuario.
- ✓ Documenta defectos y colabora en la mejora de la calidad del software.

Documentación

Ore Jimenez Jeison Juan

- ✓ Encargado de la documentación técnica y de gestión del proyecto.
- ✓ Realiza auditorías internas de cumplimiento de estándares (PMI, seguridad, buenas prácticas de código).
- ✓ Da seguimiento a métricas de avance y reporta desviaciones.
- ✓ Prepara informes de auditoría y control de calidad para el patrocinador.

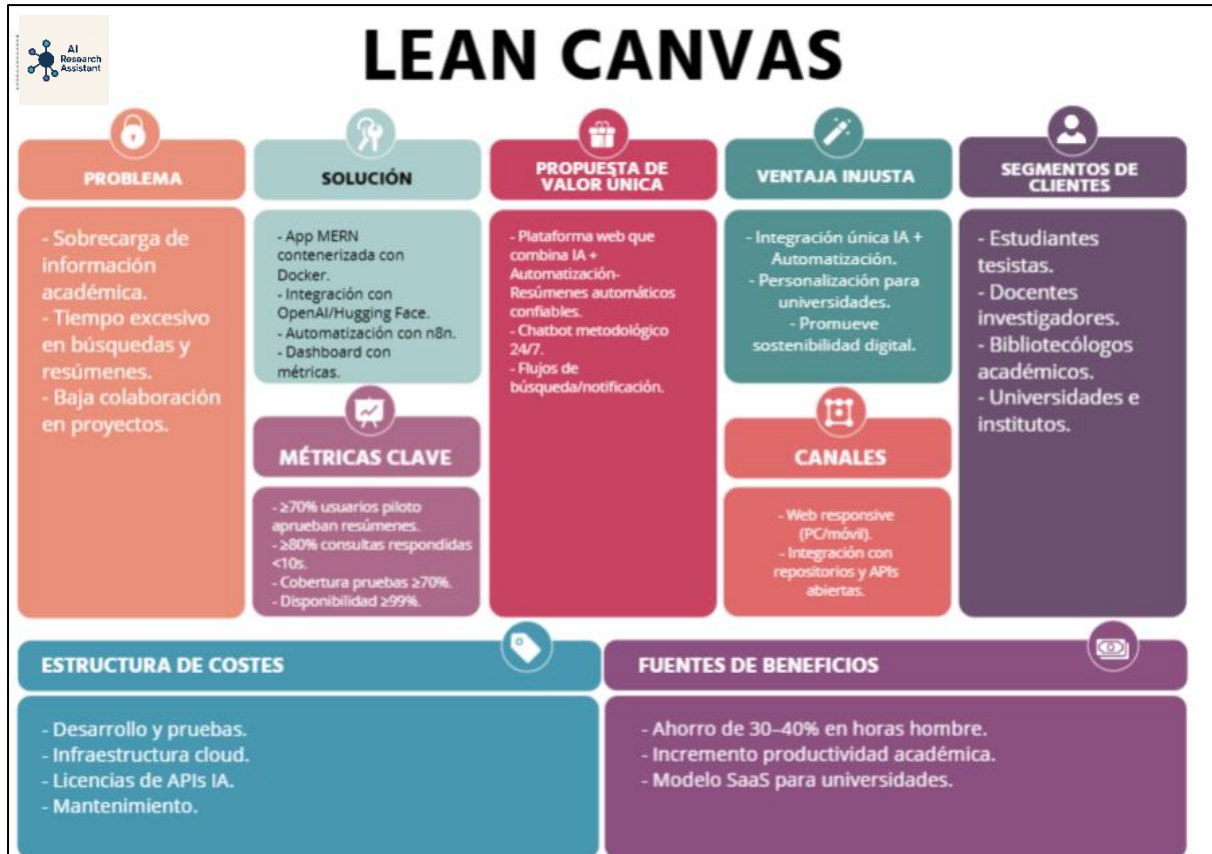
Auditora		Torres Aguilar Cielo Alessandra
<ul style="list-style-type: none"> ✓ Redacta los informes finales y manuales de usuario. ✓ Apoya en la preparación de presentaciones para stakeholders. ✓ Documenta el backlog, métricas de productividad y resultados del proyecto. ✓ Garantiza que el conocimiento quede registrado para futuras mejoras. 		
Aprobaciones:		
		
Firma Project Manager	Firma Scrum Master / Programador	
		
Firma Programador / Especialista en IA	Firma Programador	
		
Firma Tester	Firma Documentación	
		
Firma Auditora		

1.1.6. Identificación de los interesados (stakeholders).

Enlace del documento detallado:

<https://docs.google.com/spreadsheets/d/1rwHWCG0OMSk8oADlwd8yd5CMnldkin-q/edit?usp=sharing&oid=110211996957209144380&rtpof=true&sd=true>

1.1.7. Lean Project Canvas – Asistente de Investigación Académica con IA



1.2. Backlog inicial

1.2.1. Identificación del Epic central y descomposición en historias.

Épico Central:

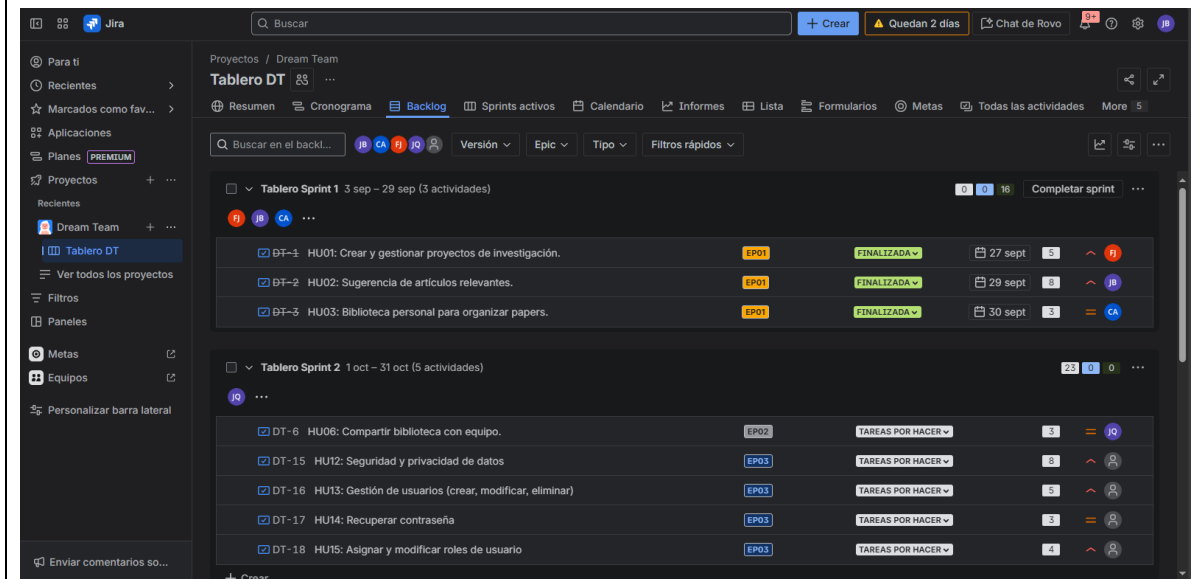
"Como investigador quiero contar con una plataforma digital que integre inteligencia artificial y automatización, para facilitar la búsqueda, organización y análisis de información académica, con el fin de optimizar mi tiempo y mejorar la calidad de mis investigaciones."

Este épico se descompone en tres grandes áreas:

- **Épica 01** – Gestión de Información Académica
- **Épica 02** – Funcionalidades con IA
- **Épica 03** – Automatización y Control

Enlace de Backlog:

<https://continental-team-th2kkles.atlassian.net/jira/software/c/projects/DT/boards/35/backlog?atlOrigin=eyJpIjoiNDY1NmE2Y2NiYmYwNDg1MDg4NTUxMzdmNzdjYjRkZGMiLCJwIjoiaj9>



1.2.2. Lista de 6 a 8 historias de usuario priorizadas con INVEST.

Se aplica lo siguiente para las historias:

- **Independiente:** cada funcionalidad (chatbot, resúmenes, biblioteca, etc.) puede desarrollarse y probarse sin depender de otra.
- **Negociable:** los detalles de cada historia pueden ajustarse durante los sprints según retroalimentación.
- **Valiosa:** todas las historias aportan un beneficio claro al usuario (ahorro de tiempo, organización, seguridad, colaboración).
- **Estimable:** cada historia tiene puntos de usuario y duración definida (1–3 semanas).
- **Small:** se descompuso en historias manejables (2 a 8 puntos), factibles dentro de un sprint.
- **Testeable:** cada historia cuenta con criterios de aceptación claros que permiten verificar su cumplimiento.

Código	Historia de Usuario	Puntos	Duración	Criterios de Aceptación	Prioridad
Épica 01 – Gestión de Información Académica					
HU01	Como usuario, quiero crear y gestionar proyectos de investigación en la plataforma.	5	2 semanas	- Crear proyecto con título, descripción y área temática.	Alta – base estructural.
				- Guardar, editar y eliminar proyectos.	

				- Acceso multiplataforma.	
HU02	Como investigador, quiero que el sistema me sugiera artículos relevantes según mi tema.	8	3 semanas	- Consulta en Google Scholar/PubMed.	Alta – valor académico clave.
				- Muestra al menos 5 artículos relevantes.	
				- Guardar en biblioteca.	
HU03	Como usuario, quiero organizar mis papers en una biblioteca personal.	3	1 semana	- Cargar y etiquetar PDFs.	Media – gestión personal.
				- Filtrar y buscar por palabras clave.	
Épica 02 – Funcionalidades con IA					
HU04	Como estudiante, quiero un chatbot con IA que me oriente en metodología de investigación.	8	3 semanas	- Responde preguntas frecuentes.	Alta – diferenciador pedagógico.
				- Integración con OpenAI/Hugging Face.	
				- Respuesta en <10 segundos.	
HU05	Como usuario, quiero que el sistema resuma automáticamente papers cargados.	5	2 semanas	- Resumen automático ≤ 300 palabras.	Alta – ahorro de tiempo.
				- Guardar junto al documento original.	
HU06	Como colaborador, quiero compartir mi biblioteca con el equipo de investigación.	3	1 semana	- Invitar a otros usuarios.	Media – colaboración.
				- Gestionar acceso (lectura/edición).	
Épica 03 – Automatización y Control					
HU07	Como usuario, quiero recibir notificaciones automáticas de nuevos artículos.	5	2 semanas	- Alertas al encontrar nuevos papers.	Media-Alta – actualización constante.
				- Notificaciones vía email o app.	

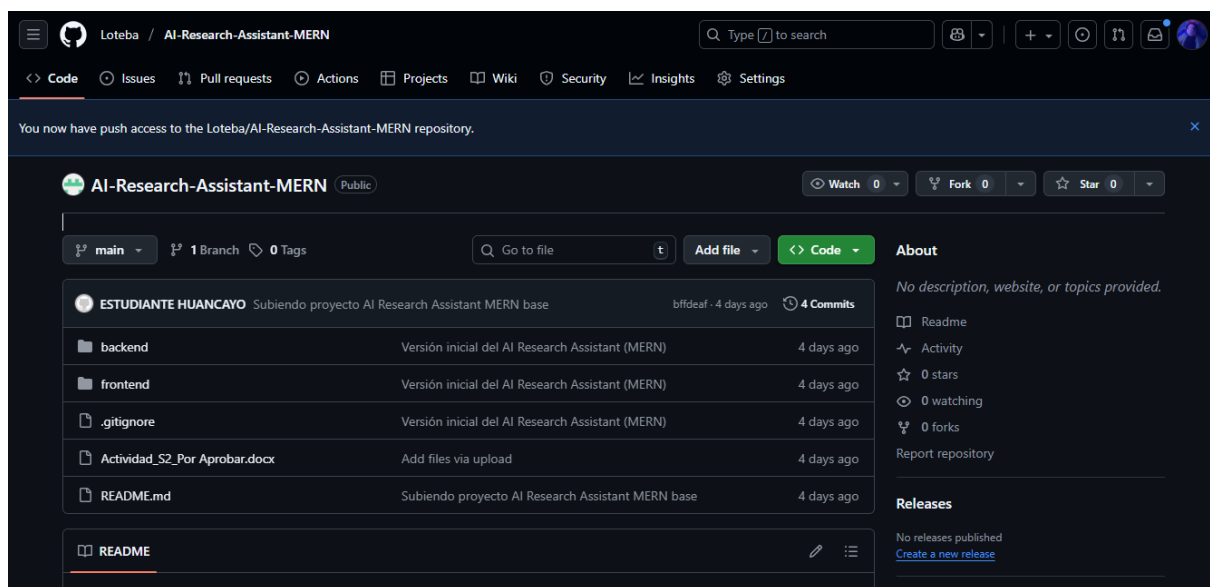
HU08	Como administrador, quiero un dashboard con métricas de productividad.	3	1 semana	- Mostrar proyectos creados, papers leídos, resúmenes generados.	Media evaluación institucional.	-
				- Exportar métricas.		
HU09	Como usuario, quiero iniciar sesión seguro con roles definidos.	2	1 semana	- Registro/login con credenciales cifradas.	Alta seguridad básica.	-
				- Roles: usuario estándar y administrador.		
UH – Calidad y Gestión de Usuarios						
HU10	Como usuario, quiero que la aplicación tenga alta disponibilidad para acceder al sistema en cualquier momento.	5	2 semanas	- El sistema estará disponible $\geq 99\%$ del tiempo.	Alta – atributo de calidad crítico.	
				- Se despliega en servicios cloud escalables.		
HU11	Como usuario, quiero que los tiempos de respuesta sean rápidos para no interrumpir mi flujo de trabajo.	5	2 semanas	- Consultas con IA responden en ≤ 3 segundos en el 80% de los casos.	Alta experiencia de usuario.	-
				- Las búsquedas en la base de datos responden en ≤ 2 segundos.		
HU12	Como usuario, quiero que mis datos estén protegidos para garantizar seguridad y privacidad.	8	3 semanas	- Cifrado de contraseñas en base de datos (bcrypt/argon2).	Muy alta – requisito de seguridad.	
				- Uso de HTTPS en todas las conexiones.		
				- Cumplimiento de la Ley de Protección de Datos Personales (LPDP).		
HU13	Como administrador, quiero gestionar usuarios (crear, modificar, eliminar) para controlar el acceso al sistema.	5	2 semanas	- CRUD completo de usuarios desde un panel de administración.	Alta – control de sistema.	
				- Validaciones de roles antes de realizar acciones.		

HU14	Como usuario, quiero recuperar mi contraseña en caso de olvido para poder acceder de nuevo.	3	1 semana	- Funcionalidad de recuperación vía correo electrónico.	Media usabilidad.	-
				- Token temporal de seguridad válido por 15 minutos.		
HU15	Como administrador, quiero asignar y modificar roles de usuario para gestionar permisos de acceso.	4	1 semana	- Roles definidos: Administrador, Investigador, Estudiante.	Alta seguridad y control.	-
				- Cambios de rol registrados en logs de auditoría.		

1.3. Infraestructura y entorno técnico

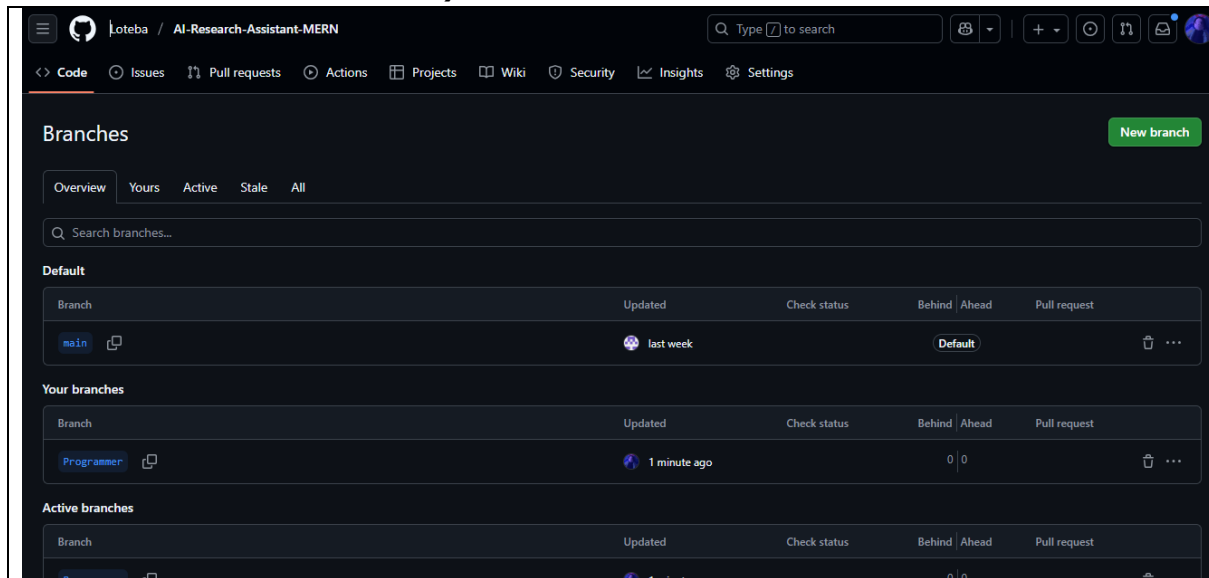
1.3.1. Creación de repositorio en GitHub (un integrante crea, el resto clona).

Enlace de Repositorio:



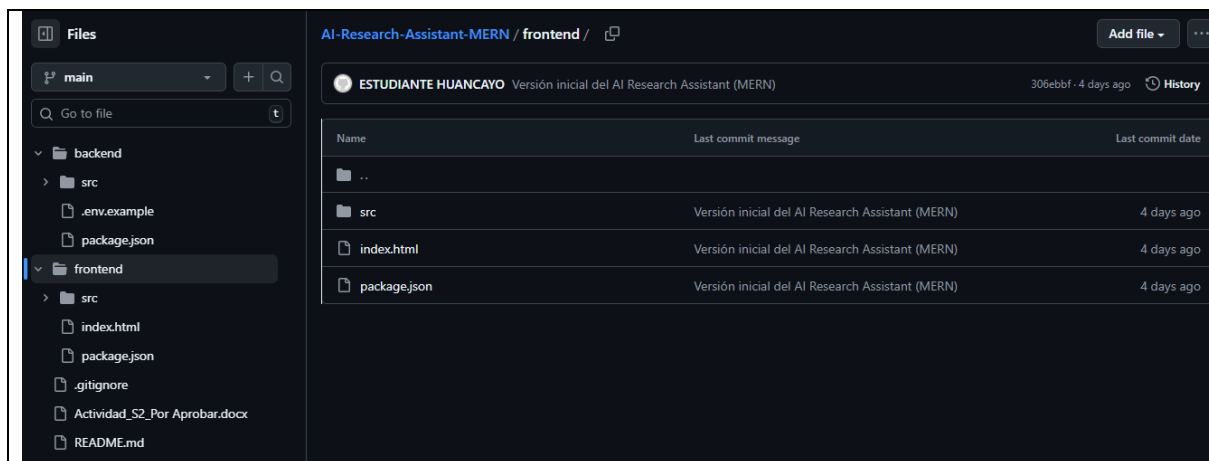
<https://github.com/Loteba/AI-Research-Assistant-MERN.git>

1.3.2. Configuración inicial de ramas (main, develop, ramas por funcionalidad).



- Rama Principal – Programador de IA
- Rama de Programador
- Rama de Tester

1.3.3. Estructura base del proyecto MERN (carpetas, dependencias mínimas).



1.4. Procesos de equipo

1.4.1. Definición de canales de comunicación (ej. Slack, Discord).

Definición de canales de comunicación:	
WhatsApp	Comunicación rápida e inmediata para coordinar tareas diarias.

Google Meet	Reuniones virtuales cuando no sea posible reunirse presencialmente.
Correo institucional	Envío de documentos oficiales y reportes.
GitHub	Comunicación técnica mediante issues, pull requests y comentarios en el código.

1.4.2. Acuerdos de trabajo: horarios de reunión, Definition of Done, roles iniciales.)

Acuerdos de trabajo	
Horarios de reunión:	Reuniones de seguimiento: Martes y miércoles, 2 veces por semana.
	Daily breve: Al inicio de cada clase (5–10 minutos).
	Reuniones extraordinarias: Convocadas únicamente en caso de incidencias críticas.
Definition of Done (DoD):	Repositorio: Código subido en la rama correspondiente de GitHub.
	Revisión: Validado mediante pull request por al menos un integrante.
	Pruebas: Ejecución exitosa de pruebas unitarias y E2E.
	Documentación: Actualización en Jira o comentarios en el código.
	Validación final: Aprobación del Project Manager o Scrum Master.
Roles iniciales (ya definidos en la Declaración del Equipo):	<ul style="list-style-type: none"> • Project Manager: Coronel Burgos Javier Daniel. • Scrum Master / Programador: Crespo Soto Fred Jhosseph. • Programador / IA: Belito Ramírez Mori Octavio. • Programador MERN: Tacunan Palomino Jean Pool. • Tester: Guevara Quintana Jenkins Jhoseph. • Documentación: Ore Jiménez Jeison Juan. • Auditora: Torres Aguilar Cielo Alessandra.

Métricas de Calidad	
Métrica	Descripción
Cobertura de pruebas	El equipo garantiza que como mínimo el 70% del código está probado y validado. Esto da confianza de que el sistema es estable y reduce riesgos de errores en producción.

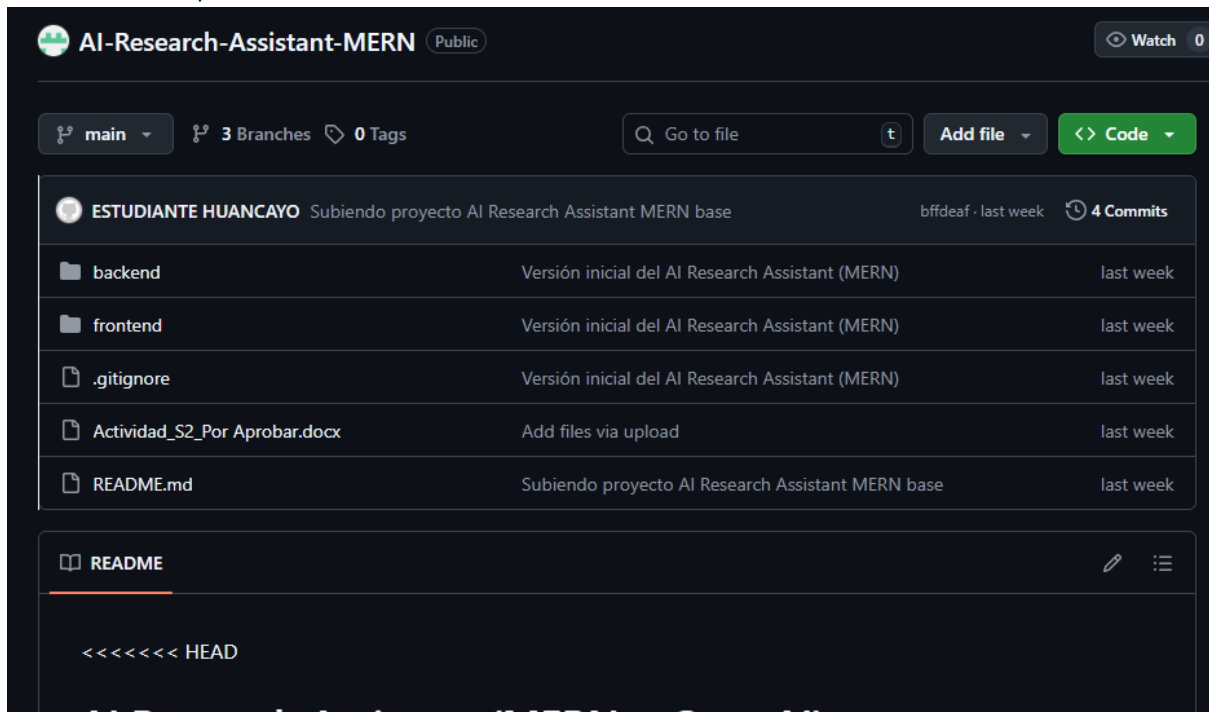
Cumplimiento de plazos	Entrega de $\geq 90\%$ de los entregables en las fechas establecidas.
Documentación actualizada	Avances acompañados de documentación técnica y de usuario vigente.

Herramientas de seguimiento	
Herramienta	Uso principal
Jira	Planificación de sprints, asignación de tareas y control del progreso.
Google Drive	Almacenamiento y edición colaborativa de documentos oficiales.
GitHub Projects	Control de incidencias, gestión de versiones y revisiones de código.

1.5. Arquitectura mínima (Walking Skeleton)

1.5.1. Configuración básica: conexión Express–MongoDB, Hello World en React.

Enlace de Repositorio:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

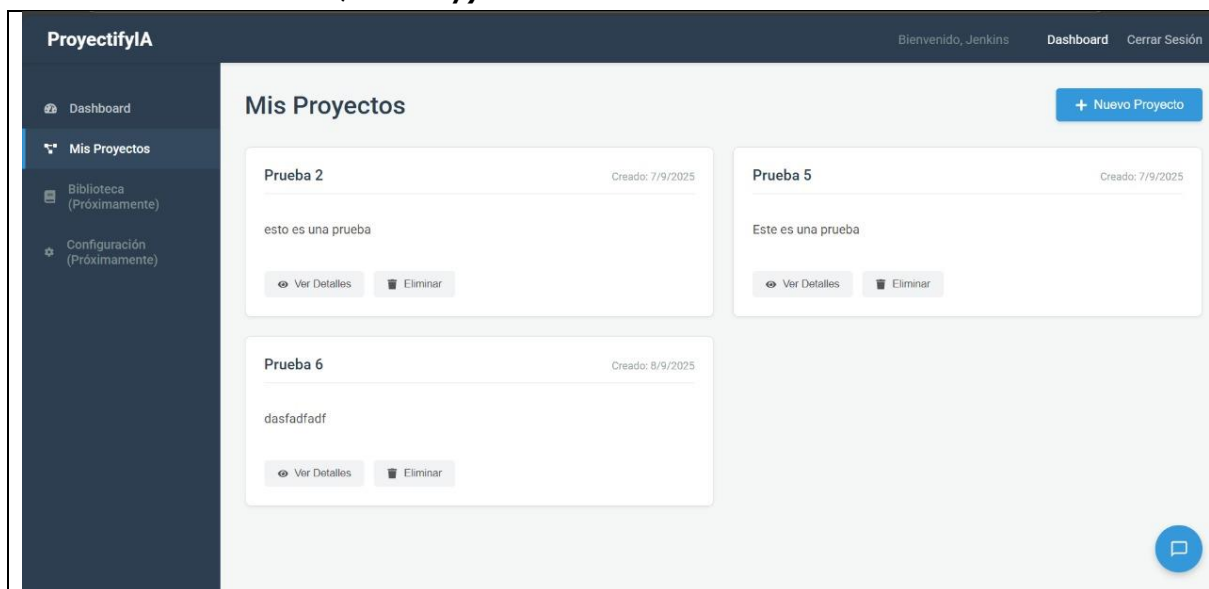
ASUS@Jenkins MINGW64 /d/Proyectos2/backend
$ npm start

> backend@1.0.0 start
> nodemon server.js

[nodemon] 3.1.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node server.js`
[dotenv@17.2.2] injecting env (3) from .env -- tip: 🐼 observe env with Radar: https://dotenvx.com/radar
Servidor corriendo en el puerto 5000
MongoDB Conectado: ac-aud39wo-shard-00-00.8wsn9fe.mongodb.net
█

https://github.com/Loteba/AI-Research-Assistant-MERN.git
```

1.5.2. Evidencia de despliegue inicial en un servicio gratuito (ej. Render, Vercel, Railway).



1.6. Entregables

1.6.1. Documento (Markdown, PDF o Google Docs) con:

- Project Charter completo.
- Backlog inicial y acuerdos de equipo.

Enlace:

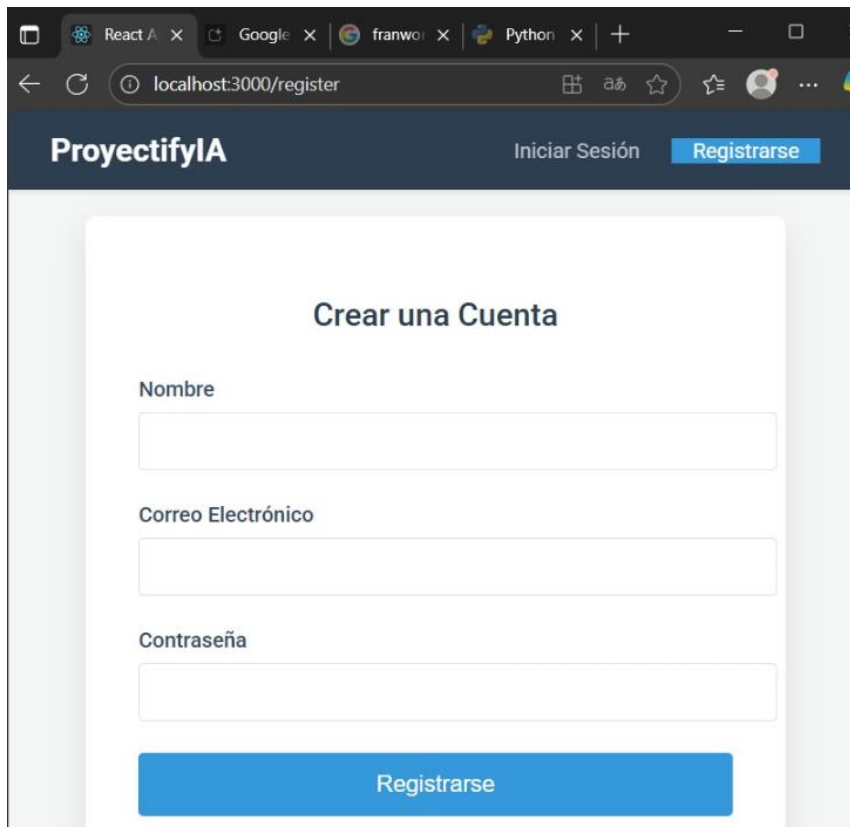
https://docs.google.com/document/d/1-Yp7Eqg5lhXjvY9CqFaecdc47wocKYP1/edit?usp=drive_link&ouid=110211996957209144380&rtpof=true&sd=true

1.6.2. URL del repositorio GitHub con:

- Código base MERN configurado.
- Readme con instrucciones de instalación.

<https://github.com/Loteba/AI-Research-Assistant-MERN>

1.6.3. Evidencia del despliegue inicial (screenshot o URL):



The screenshot shows a web browser window with the URL `localhost:3000/register`. The page has a dark blue header with the logo 'ProyectifyLA' on the left, and links for 'Iniciar Sesión' and 'Registrarse' on the right. The main content area is a white card titled 'Crear una Cuenta'. It contains three input fields labeled 'Nombre', 'Correo Electrónico', and 'Contraseña'. Below these fields is a blue button labeled 'Registrarse'.