

“Año de la recuperación y consolidación de la economía peruana”



**Universidad
Continental**

Tutor virtual de lectura crítica

Asignatura:

Taller de Proyectos 2

Integrantes del equipo de trabajo:

1. Huaccho Mancilla Steven José
2. Lazo Maraví Nilton Joel
3. Poma Goche Abigail Karim
4. Ramirez Basualdo Lenin Sebasthian
5. Robles Sanchez Britney Sheyla
6. Rojas Mellado Andrea Mirella

Huancayo - Perú

2025

índice

1. Enfoque de desarrollo	4
1.1. Enfoque	4
1.2. Variables clave	4
1.2.1. Tipo de Proyecto	4
1.2.2. Requisitos	5
1.2.3. Cliente	6
1.2.4. Cronograma	6
2. Lanzamiento del proyecto	7
2.1. Declaración de la visión del proyecto	7
2.2. Project Charter (Acta de Proyecto)	8
2.2.1. Datos generales del proyecto	8
2.2.2. Propósito del proyecto	8
2.2.3. Descripción general del proyecto	8
2.2.4. Límites del proyecto	8
2.2.5. Entregables principales	9
2.2.6. Requerimientos de alto nivel	9
2.2.7. Riesgos generales del proyecto	9
2.2.8. Hitos del proyecto	10
2.2.9. Criterios de éxito	10
2.2.10. Alcance	10
2.2.11. Tiempo	11
2.2.12. Costos	11
2.2.13. Otros	12
2.2.14. Objetivos del proyecto	12
2.2.15. Recursos financieros preaprobados	13
2.2.16. Interesados	13
2.2.17. Criterios de cierre del proyecto	13
2.2.18. Nivel de autoridad del gerente del proyecto	14
2.3. Suposiciones y restricciones	15
2.4. Declaración del equipo del proyecto	16
3. Infraestructura y entorno técnico	20
3.1. Creación de repositorio en GitHub	20
3.2. Configuración inicial de ramas	21
3.3. Estructura base del proyecto MERN	21
3.3.1. Carpetas	21
3.3.2. Dependencias mínimas	22
4. Procesos de equipo	23
4.1. Definición de canales de comunicación	23
4.2. Acuerdos de trabajo: horarios de reunión, definition of done, roles iniciales	25
5. Arquitectura mínima	26
5.1. Configuración básica: conexión Express-MongoDB, Hello World en React	26

5.2. Evidencia del despliegue inicial en un servicio gratuito	26
5.3. Diagrama de arquitectura con IA y automatización	27
5.3.1. Diagrama de componentes principales	27
5.3.2. Diagrama de arquitectura (ASCII)	27
5.3.3. Flujos principales	27
6. Anexos	28
6.1. Planificación del Proyecto - Jira	28

1. Enfoque de desarrollo

1.1. Enfoque

El proyecto adopta un **enfoque ágil híbrido**, combinando elementos de **Scrum** y **Kanban** con el propósito de optimizar tanto la planificación como la ejecución del trabajo. Por un lado, se utiliza Scrum como marco de referencia para organizar el desarrollo en sprints iterativos de corta duración (1 a 2 semanas), con roles definidos (Product Owner, Scrum Master y equipo de desarrollo), eventos formales (planificación, reuniones diarias, revisiones y retrospectivas) y la entrega de incrementos funcionales del sistema. Por otro lado, se incorpora **Kanban** como método visual de gestión del flujo, mediante un tablero con columnas (To Do, In Progress, Done) y límites de trabajo en curso (WIP), lo cual permite dar seguimiento a las tareas de forma clara y transparente. Esta combinación, conocida como **Scrumban**, brinda la disciplina estructurada de Scrum junto con la flexibilidad de Kanban, lo que resulta especialmente útil en proyectos académicos con tareas impredecibles y con la necesidad de un flujo continuo de valor.

1.2. Variables clave

1.2.1. Tipo de Proyecto

El presente trabajo corresponde a un proyecto de desarrollo de software de carácter aplicativo, orientado a la creación de una solución tecnológica innovadora en el ámbito educativo. Su naturaleza es tecnológica y académica, ya que combina la implementación de un sistema web full-stack basado en el stack MERN con la integración de técnicas de inteligencia artificial (generación de preguntas y detección de sesgos) y herramientas de automatización (n8n). Asimismo, puede considerarse un proyecto con componente investigativo-aplicado, dado que no se limita a construir un producto funcional, sino que también busca validar la utilidad de la tecnología para fortalecer habilidades de lectura crítica y pensamiento analítico en estudiantes universitarios.

1.2.2. Requisitos

Se necesitará una aplicación que desarrolle habilidades de pensamiento crítico. Los usuarios leen textos y el sistema genera preguntas, detecta sesgos y evalúa comprensión, n8n programa sesiones, envía recordatorios y registra progreso.

1) Requisitos generales (Historias épicas)

- Como estudiante, quiero que un tutor virtual me haga preguntas sobre el texto que leí para desarrollar mi análisis crítico.
- Como docente, quiero asignar textos con actividades automáticas para evaluar el progreso.
- Como usuario, quiero que el sistema detecte sesgos o falacias lógicas en un artículo
- Como sistema, quiero que n8n automatice el envío de textos diarios y el registro de respuestas para crear hábitos.
- Como administrador, quiero ver un panel con el nivel de comprensión promedio de los usuarios.

2) Requisitos de inteligencia artificial

- Generación de preguntas con NLP (Hugging Face).
- Clasificación de sesgos en textos (modelo entrenado o API).

3) Requisitos de automatización

- Flujo: asignación → lectura → actividad → evaluación → notificación.

4) Historias de usuario orientada a atributos de calidad

ID	Historia de usuario (NFR)	Criterios de aceptación
NFR-Perf-01	Como estudiante, quiero que la generación de preguntas tarde como máximo 3 segundos para mantener el flujo de lectura.	Dado un texto de ≤ 1500 palabras, cuando solicite preguntas, entonces el p95 de latencia es ≤ 3 s medido en 20 intentos.
NFR-Disp-01	Como usuario, quiero que la plataforma esté disponible al menos 99% durante la semana de la demo para usarla sin interrupciones.	Monitoreo con healthcheck cada 1 min; uptime semanal $\geq 99\%$.
NFR-Seg-01	Como administrador, quiero que el acceso use JWT y contraseñas seguras para proteger la información.	Contraseñas ≥ 8 caracteres, 1 mayúscula, 1 minúscula, 1 dígito; Tokens expiran en 60 min.

NFR-Usab-01	Como usuario con lector de pantalla, quiero etiquetas ARIA y contraste AA.	Chequeo WCAG 2.2 AA: contraste \geq 4.5:1; navegación completa por teclado.
NFR-Mant-01	Como desarrollador, quiero linting y tests automáticos.	ESLint/Prettier en pre-commit; CI bloquea si fallan; cobertura \geq 70%.
NFR-Obs-01	Como operador, quiero logs estructurados y métricas básicas.	Logs JSON con request-id; dashboard simple de latencia y errores 5xx.

5) Historias de usuario para la gestión de usuarios

ID	Historia de usuario	Criterios de aceptación
HU-User-01	Como usuario, quiero registrarme con correo y contraseña.	Registro con verificación de correo; mensaje de confirmación al activar cuenta.
HU-User-02	Como usuario, quiero iniciar sesión de forma segura.	Autenticación JWT; bloqueo tras 5 intentos fallidos; “remember me” opcional.
HU-User-03	Como administrador, quiero gestionar roles (estudiante, docente, admin).	Rutas protegidas; acceso no autorizado devuelve 403.
HU-User-04	Como usuario, quiero restablecer mi contraseña.	Token de un solo uso (30 min) enviado al correo.
HU-User-05	Como usuario, quiero editar mi perfil (nombre, avatar, preferencias).	Actualización validada; avatar opcional con límite de tamaño.

1.2.3. Cliente

El cliente objetivo del proyecto está representado por los estudiantes universitarios, quienes constituyen los principales beneficiarios de la solución. Son ellos quienes interactuarán con el tutor virtual de lectura crítica para desarrollar habilidades de comprensión lectora y pensamiento crítico a partir de textos digitales. En este sentido, el sistema se orienta a responder a sus necesidades de contar con una herramienta accesible, interactiva y automatizada que les permita practicar la lectura comprensiva, recibir retroalimentación inmediata y reforzar hábitos de estudio de manera continua.

1.2.4. Cronograma

Fase	Ventana	Objetivo principal	Entregables clave
Inicio	01–05 Sep	Alinear visión y marco del proyecto	Visión aprobada, Acta de proyecto, Stakeholders & roles

Planificación	06–12 Sep	Preparar ejecución y base técnica	Backlog inicial (épicas/HU), Arquitectura MERN mínima, Repo GitHub y ramas
Sprint 1	10 Sep–18 Oct	MVP de preguntas y reporting inicial	HU01 (generación de preguntas: API+UI+pruebas), HU05 (reportes básicos), Accesibilidad básica, QA S1
Sprint 2	09 Oct–17 Nov	IA de retroalimentación y modelo de sesgos (baseline)	HU02 (feedback y scoring), HU03 (progreso acumulado), HU07 (dataset+modelo baseline+endpoint+UI), QA S2, pruebas de carga ligera
Sprint 3	05–22 Nov	Gestión docente y recordatorios (n8n) + panel comprensión	HU04 (asignación docente), HU08 (flujos n8n de recordatorios), HU09 (dashboard comprensión + usabilidad)
Integración y despliegue	23 Nov–04 Dic	Asegurar calidad y publicar versión estable	CI/CD (GitHub Actions), Staging (Vercel/Heroku) + smoke tests, Integración global, Monitoreo básico y backups
Cierre y presentación	05–14 Dic	Documentar, presentar y cerrar	Documentación técnica + manual de usuario, Slides + demo, Presentación (12 Dic), Lecciones aprendidas y cierre

2. Lanzamiento del proyecto

2.1. Declaración de la visión del proyecto

2.1.1. Versión 1

Para estudiantes **quienes** desean mejorar su comprensión lectora y desarrollar el pensamiento crítico, **el Tutor Virtual de Lectura Crítica es una** aplicación de última generación **que** fomenta la reflexión, el análisis y la detección de sesgos en los textos.

A diferencia de otras plataformas que solo ofrecen lecturas o preguntas estáticas, **nuestro producto** integrará inteligencia artificial para generar preguntas personalizadas, identificar falacias, registrar progreso automáticamente y ofrecer retroalimentación en tiempo real.

2.1.2. Versión 2

Para estudiantes **que** buscan fortalecer sus habilidades de análisis y comprensión, **el Tutor Virtual de Lectura Crítica es una** aplicación innovadora **que** transforma la manera de leer, cuestionar y entender los textos.

A diferencia de herramientas tradicionales que solo miden la memoria, **nuestro Producto** tutor utiliza inteligencia artificial para detectar sesgos y falacias, programar sesiones automatizadas y acompañar al usuario en la creación de un hábito de lectura reflexiva.

2.2. Project Charter (Acta de Proyecto)

2.2.1. Datos generales del proyecto

- 1) Título del proyecto:** Tutor virtual de lectura crítica
- 2) Patrocinador del proyecto:** Ramirez Basualdo, Lenin Sebasthian
- 3) Gerente del proyecto:** Lazo Maraví, Nilton Joel
- 4) Fecha de elaboración:** 25 de agosto de 2025 (1° semana de desarrollo)
- 5) Cliente de proyecto:** Estudiantes

2.2.2. Propósito del proyecto

Fomentar el desarrollo de habilidades de pensamiento crítico en estudiantes universitarios mediante una aplicación web interactiva que apoye la lectura y comprensión de textos, utilizando inteligencia artificial y procesos automatizados.

2.2.3. Descripción general del proyecto

Se desarrollará una aplicación web full-stack con tecnología MERN (MongoDB, Express.js, React.js y Node.js) que funcione como un tutor virtual de lectura crítica. La plataforma permitirá a los usuarios leer textos asignados y recibir preguntas generadas automáticamente, identificar sesgos o falacias en los contenidos y evaluar su nivel de comprensión. Además, integrará un sistema de automatización (n8n) para programar sesiones, enviar recordatorios y registrar el progreso académico.

2.2.4. Límites del proyecto

- **Incluye:** Desarrollo del frontend y backend, base de datos, integración de IA para generación de preguntas y detección de sesgos, flujos automatizados con n8n, panel de seguimiento para

administradores, pruebas automatizadas y documentación técnica.

- **No incluye:** Generación de textos propios, tutoría presencial, soporte fuera de la plataforma, ni módulos de pago o monetización.

2.2.5. Entregables principales

- 1) Aplicación web MERN funcional.
- 2) Dos funcionalidades de IA implementadas (generación de preguntas y clasificación de sesgos).
- 3) Automatización con n8n para notificaciones y registro de actividades.
- 4) Documentación técnica (README, diagramas, informe técnico).
- 5) Repositorio público en GitHub.
- 6) Informe de impacto ambiental del proyecto.
- 7) Demostración oral con video y presentación.

2.2.6. Requerimientos de alto nivel

- 1) Frontend en React.js con diseño responsive.
- 2) Backend con Node.js y Express.js.
- 3) Base de datos en MongoDB Atlas.
- 4) Uso de NLP (Hugging Face u OpenAI API) para generación de preguntas.
- 5) Clasificación de sesgos mediante modelo entrenado o API.
- 6) Automatización de flujos con n8n.
- 7) Pruebas unitarias (Jest) y E2E (Cypress).
- 8) Contenerización con Docker y docker-compose.

2.2.7. Riesgos generales del proyecto

- 1) **Riesgo técnico:** errores en la integración de IA o baja precisión de los modelos.
- 2) **Riesgo de tiempo:** retrasos por curva de aprendizaje en IA y Docker.
- 3) **Riesgo operativo:** falta de adopción por parte de los estudiantes.

- 4) **Riesgo de calidad:** cobertura insuficiente en pruebas automatizadas.

2.2.8. Hitos del proyecto

- 1) **Semana 1-2:** Definición de requisitos y diseño de arquitectura.
- 2) **Semana 3-4:** Desarrollo del backend y base de datos.
- 3) **Semana 5-6:** Desarrollo del frontend y diseño responsive.
- 4) **Semana 7-8:** Implementación de la primera funcionalidad de IA: generación de preguntas.
- 5) **Semana 9-10:** Implementación de la segunda funcionalidad de IA: clasificación de sesgos.
- 6) **Semana 11-12:** Automatización con 8n8 e integración de flujos.
- 7) **Semana 13-14:** Contenerización, documentación y entrega final.

2.2.9. Criterios de éxito

- 1) **Aplicación MERN funcional:** El sistema debe estar desplegado y operativo en un entorno accesible para los usuarios.
- 2) **Inteligencia Artificial implementada:** Al menos dos funcionalidades de IA completadas:
 - a) Generación de preguntas con NLP.
 - b) Clasificación de sesgos en textos.
- 3) **Automatización operativa:** Flujos clave implementados en n8n: asignación → lectura → actividad → evaluación → notificación.
- 4) **Pruebas y calidad asegurada:** Cobertura mínima del 70% en pruebas unitarias e integrales.
- 5) **Documentación completa:** Incluir README, documentos de planificación, implementación, presupuesto, y demás.
- 6) **Entrega final:** Presentación del proyecto con demo funcional, video explicativo y validación del patrocinador.

2.2.10. Alcance

- 1) Dentro del alcance
 - Desarrollo de una aplicación web con el stack MERN.

- Implementación de funcionalidades de IA: generación de preguntas y clasificación de sesgos.
- Automatización de flujos clave con n8n.
- Pruebas de calidad (unitarias, integrales y E2E).
- Documentación técnica y presentación final del proyecto.

2) Fuera del alcance

- Desarrollo de una aplicación móvil nativa.
- Integración con plataformas LMS externas.
- Soporte técnico presencial u offline.
- Módulos de monetización o pasarelas de pago.

2.2.11. Tiempo

1) Duración: 14 semanas

2) Línea de tiempo:

- **Sprint 1 (Semana 1-2):** Definición de requisitos y diseño de arquitectura.
- **Sprint 2 (Semana 3-4):** Desarrollo del backend y base de datos.
- **Sprint 3 (Semana 5-6):** Desarrollo del frontend y diseño responsive.
- **Sprint 4 (Semana 7-8):** Implementación de la primera funcionalidad de IA (generación de preguntas).
- **Sprint 5 (Semana 9-10):** Implementación de la segunda funcionalidad de IA (clasificación de sesgos).
- **Sprint 6 (Semana 11-12):** Automatización con n8n e integración de flujos.
- **Sprint 7 (Semana 13-14):** Contenerización, documentación y entrega final.

2.2.12. Costos

- 1) Costo financieros y directos: Aplican costos que se detallarán en los siguientes ítems del documento.
- 2) Inversión principal: Esfuerzo humano (horas de desarrollo, pruebas, documentación y presentación).

3) Recursos:

- *Equipo de desarrollo:* estudiantes asignados al proyecto.
- *Tiempo de dedicación:* 14 semanas distribuidas en 7 sprints.
- Infraestructura en la nube gratuita o con planes básicos (ej. MongoDB Atlas Free Tier, Hugging Face API con versión gratuita, n8n self-hosted).

2.2.13. Otros

- **Sostenibilidad:** El proyecto contribuye a la reducción del uso de papel mediante la digitalización de materiales de lectura y actividades.
- **Impacto educativo:** Fomenta el desarrollo de habilidades de pensamiento crítico en estudiantes universitarios a través de la práctica continua.
- **Innovación:** Combina inteligencia artificial y automatización de procesos (IA + n8n) dentro de un entorno web interactivo.
- **Escalabilidad:** La solución podrá evolucionar en el futuro para integrar nuevas funcionalidades de IA, un panel más avanzado o una aplicación móvil.

2.2.14. Objetivos del proyecto

En esta sección podrá visualizar los objetivos del proyecto. Están relacionados con las fechas de cumplimiento en cada sprint y se despliegan utilizando la técnica SMART (Specific, Measurable, Achievable, Relevant, Time-bound).

1. Implementar una aplicación web MERN que funcione como tutor virtual.
2. Integrar al menos dos funcionalidades de IA (generación de preguntas y clasificación de sesgos).
3. Automatizar procesos clave con n8n.
4. Garantizar accesibilidad y usabilidad en la interfaz.
5. Cumplir con pruebas automatizadas y documentación técnica.

Código	Objetivo (SMART)	Métrica / Indicador	Línea Base	Meta	Sprint	Método de verificación
O1	Desplegar una aplicación MERN funcional accesible	URL activa + uptime ≥ 99% semana de demo	0%	URL operativa + uptime ≥ 99% en	7	Ping uptime/healthcheck + demo en clase

	públicamente antes del 14/12			semana demo		
O2	Implementar generación de preguntas con IA con latencia ≤ 3 s	p95 latencia generación (s)	N/A	$p95 \leq 3$ s	5	Logs de API + cronómetro en demo
O3	Implementar clasificación de sesgos con $F1 \geq 0,70$ (con dataset de prueba)	F1-score en conjunto de validación	N/A	$F1 \geq 0,70$	5	Informe de evaluación + notebook de pruebas
O4	Automatizar recordatorios y registro con n8n para 3 flujos críticos	3 workflows activos: asignación, notificación, registro	0	3 workflows operativos	6	Evidencia n8n (capturas + ejecución program.)
O5	Asegurar calidad con cobertura de pruebas $\geq 70\%$	Cobertura total Jest + Cypress (%)	N/A	$\geq 70\%$	7	Reporte de cobertura en CI (GitHub Actions)

2.2.15. Recursos financieros preaprobados

- 1) **Fondos asignados:** No aplican, ya que el proyecto utilizará herramientas y tecnologías gratuitas (open-source).
- 2) **Infraestructura:** Servicios gratuitos como MongoDB Atlas, Hugging Face API (plan gratuito) y n8n self-hosted.
- 3) **Inversión en recursos humanos:** El esfuerzo provendrá del equipo de desarrollo, distribuido en 14 semanas (7 sprints).

2.2.16. Interesados

Stakeholder(s)	Role
Daniel Gamarra Moreno	Patrocinador del proyecto
Nilton Joel Lazo Maravi	Gerente del proyecto
Estudiantes	Clientes del proyecto
Usuarios de la plataforma	Usuarios finales
Equipo de desarrollo	Desarrolladores responsables de backend, frontend, IA y automatización

2.2.17. Criterios de cierre del proyecto

- La aplicación web MERN está desplegada en un entorno accesible públicamente (publicly accessible deployment).

- Las dos funcionalidades de IA (generación de preguntas y clasificación de sesgos) están implementadas y operativas (AI features implemented and operational).
- Los flujos de automatización con n8n funcionan correctamente: asignación → lectura → actividad → evaluación → notificación (automation workflows functional).
- Se cuenta con documentación completa: técnica, de usuario y de impacto ambiental (complete documentation).
- Se realizó una presentación final con demostración en vivo y material audiovisual (final presentation and live demo).
- Validación de los entregables por parte del patrocinador del proyecto (sponsor validation).

2.2.18. Nivel de autoridad del gerente del proyecto

1) Decisiones de personal

El Gerente del Proyecto podrá asignar tareas y responsabilidades dentro del equipo de desarrollo, definir prioridades en los sprints y aprobar cambios en la distribución del trabajo.

2) Gestión del presupuesto y variaciones

No aplica, ya que el proyecto no cuenta con presupuesto financiero directo. En su lugar, el Gerente gestionará el uso de recursos gratuitos (open-source) y la optimización del tiempo del equipo.

3) Decisiones técnicas

El Gerente del Proyecto tiene autoridad para definir la arquitectura de software, seleccionar librerías, integrar herramientas de IA y establecer estándares de desarrollo y pruebas.

4) Resolución de conflictos

Los conflictos internos del equipo se resolverán primero mediante consenso. Si no se logra acuerdo, el Gerente del Proyecto tendrá la última decisión. En casos mayores que afecten el rumbo del proyecto, se escalará al patrocinador.

5) Autoridad del patrocinador

El Patrocinador es la máxima autoridad del proyecto. Tendrá la capacidad de aprobar los entregables finales, validar cambios significativos en el alcance y dar por concluido el proyecto.

2.3. Suposiciones y restricciones

ID	Categoría	Supuesto/Restricción	Estado	Comentarios
C1	Tiempo	El proyecto debe completarse en 14 semanas (7 sprints) .	Activo	La duración está definida en el Project Charter.
A1	Recursos	Se supone que todos los miembros del equipo cumplirán con sus roles asignados.	Confirmado	Declarado en la Declaración del Equipo.
A2	Técnico	Se supone que no habrá retrasos significativos en la curva de aprendizaje de IA/Docker.	Pendiente	Relacionado con riesgos de tiempo.
A3	Equipo	Se supone que habrá disponibilidad de los estudiantes para dedicar horas suficientes.	Pendiente	Clave para cumplir los sprints definidos.
C2	Financiero	El proyecto debe ejecutarse usando tecnologías y recursos gratuitos (open-source) .	Activo	No hay presupuesto asignado.
A4	Operaciones	Se supone que MongoDB Atlas, Hugging Face API y n8n tendrán disponibilidad gratuita.	Pendiente	Dependencia de planes "free tier".
A5	Usuarios	Se supone que los estudiantes tendrán acceso a internet estable para usar la plataforma.	Pendiente	Impacta adopción y pruebas.
A6	Equipo	Se supone que el patrocinador y docentes apoyarán con validación académica sin costo.	Confirmado	Ayuda a cumplir con calidad sin afectar presupuesto.
C3	Alcance	Solo se desarrollará la aplicación web MERN , no app móvil ni integración LMS.	Activo	Definido en los límites del Charter.
A7	Técnico	Se supone que la arquitectura MERN cubrirá las necesidades del tutor virtual.	Confirmado	Alcance realista con las tecnologías elegidas.
A8	Operaciones	Se supone que los flujos automatizados con n8n son suficientes para los procesos clave.	Pendiente	No se contemplan integraciones externas.
A9	Equipo	Se supone que no se requerirá soporte técnico presencial ni offline.	Confirmado	Alineado con el "fuera de alcance" del Charter.
C4	Calidad	El sistema debe lograr al menos 70% de cobertura en pruebas automatizadas .	Activo	Estándar de calidad definido.
A10	Técnico	Se supone que Jest y Cypress funcionarán adecuadamente con el stack MERN.	Confirmado	Son frameworks ya probados en proyectos similares.
A11	Recursos	Se supone que el equipo podrá diseñar y ejecutar los casos de prueba dentro del plazo.	Pendiente	Riesgo por carga académica del equipo.

A12	Equipo	Se supone que la comunicación constante (Trello, WhatsApp, Gmail) permitirá detectar errores rápido.	Confirmado	Mejora la probabilidad de cumplir con el estándar de calidad.
-----	--------	--	------------	---

2.4. Declaración del equipo del proyecto

1) Visión del proyecto

El proyecto Tutor virtual de lectura crítica es una herramienta educativa innovadora que fomenta el desarrollo de habilidades de pensamiento crítico en usuarios de diversos perfiles (estudiantes, docentes y administradores), mediante el uso de inteligencia artificial y automatización. El objetivo a largo plazo es democratizar el acceso a una educación personalizada y efectiva, donde las personas puedan analizar textos de manera autónoma, identificar sesgos y falacias lógicas, y mejorar su comprensión lectora de forma habitual y medible. Esto contribuiría a una sociedad más informada y capaz de discernir información en un mundo saturado de contenidos, integrando tecnologías como NLP y flujos automatizados para hacer el aprendizaje interactivo, accesible y sostenible.

2) Objetivo del equipo

El objetivo principal del equipo es diseñar, desarrollar e implementar una aplicación integral que combine funcionalidades de IA con automatizaciones mediante n8n (para programar sesiones, enviar recordatorios y registrar progreso). Esto se enfoca en satisfacer las necesidades de los usuarios a través de historias épicas específicas: proporcionar preguntas analíticas para estudiantes, actividades automáticas para docentes, detección de sesgos para usuarios generales, hábitos diarios automatizados y paneles de monitoreo para administradores.

3) Miembros del equipo y roles

Nombre	Rol	Responsabilidades clave
Lazo Maravi Nilton Joel	Gerente de Proyecto	Lidera el proyecto, gestiona el cronograma, presupuesto y riesgos; coordina a los líderes de equipo.

Huaccho Mancilla Steven Jose	Líder de Construcción	Supervisa la renovación y construcción del proyecto
Robles Sanchez Britney Sheyla	Líder de Sistemas	Responsable del desarrollo e implementación del proyecto.
Poma Goche Abigail Karim	Líder de Operaciones	Supervisa la integración y coordinación de los recursos humanos y técnicos necesarios para el desarrollo y despliegue del tutor virtual.
Rojas Mellado Andrea Mirella	Líder de Comunicaciones	Facilita la comunicación efectiva entre los equipos de desarrollo, los usuarios (estudiantes, docentes, administradores) y otras partes interesadas
Ramirez Basualdo Lenin Sebasthian	Patrocinador	Proporciona el respaldo estratégico y financiero para el desarrollo del tutor virtual, aprobando decisiones clave relacionadas con el alcance, presupuesto y cronograma del proyecto.

4) Normas del equipo (Team Norms)

- a) **Respeto mutuo:** Escuchamos activamente, valoramos todas las opiniones y evitamos interrupciones.
- b) **Puntualidad:** Las reuniones comienzan y terminan a tiempo, y se respetan los plazos de entrega.
- c) **Transparencia:** Compartimos avances, riesgos y problemas de forma oportuna y honesta.
- d) **Colaboración:** Trabajamos de manera unida, apoyándonos mutuamente y distribuyendo cargas de manera equitativa, incluso en modalidades híbridas.
- e) **Comunicación asertiva:** Las diferencias de opinión se expresan de forma constructiva, con enfoque en soluciones.
- f) **Responsabilidad:** Cumplimos con las tareas asignadas y asumimos la ownership de nuestros resultados.
- g) **Flexibilidad:** Estamos abiertos a adaptarnos a cambios y nuevas ideas según las necesidades del proyecto.
- h) **Enfoque en resultados:** Priorizamos la productividad y el cumplimiento de los objetivos en cada sesión de trabajo.

5) Frecuencia y tipos de reuniones

Tipo de reunión	Frecuencia	Duración	Participantes
Revisión semanal de avances	Semanal (miércoles)	1.5 horas	Equipo de proyecto

Reunión mensual con patrocinador	Mensual	1.5 horas	Equipo de proyecto, sponsor
Revisión de riesgos y presupuesto	Mensual	2.5 horas	Equipo de proyecto, sponsor, stakeholders

6) Herramientas de trabajo

- Gestión de proyectos: Jira
- Comunicación: Discord, Whatsapp
- Documentos: Documentos de Google, Google Drive

7) Toma de decisiones

El equipo se compromete a utilizar un modelo de toma de decisiones que priorice el consenso y la participación de todos los integrantes. El proceso se regirá por los siguientes lineamientos:

- **Identificación y Análisis:** Cualquier miembro del equipo puede plantear una decisión necesaria. Se expondrá el tema de manera clara, se discutirán las alternativas disponibles y se analizarán sus pros y contras de manera objetiva.
- **Búsqueda de Consenso:** Se buscará llegar a un acuerdo unánime que satisfaga los objetivos del proyecto y sea aceptable para todos los miembros. Se fomentará la discusión abierta y respetuosa hasta agotar las posibilidades de consenso.
- **Votación Democrática:** Si no se logra un consenso después de una discusión razonable, se procederá a una votación democrática donde cada miembro tendrá un voto. La decisión se tomará por mayoría simple (mitad más uno de los votos).
- **Decisión del Líder o Project Manager:** En caso de un empate en la votación, o en situaciones de urgencia crítica que requieran una acción inmediata, el Líder o Project Manager del proyecto tendrá la facultad de tomar la decisión final, la cual será acatada por todo el equipo.
- **Compromiso con la Decisión:** Una vez tomada una decisión (por consenso, votación o por el líder), todos los miembros del equipo se

comprometen a apoyar y ejecutar de manera cohesionada, dejando de lado las discrepancias previas.

8) Valores del equipo

Nuestro equipo se fundamenta en un conjunto de valores compartidos que guiarán nuestra conducta, nuestras interacciones y nuestro trabajo a lo largo del desarrollo del proyecto:

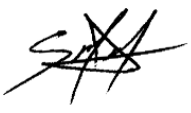





- **Respeto:** Valoramos las opiniones, el tiempo y el esfuerzo de cada integrante. Fomentamos un ambiente donde todos se sientan escuchados y libres de expresar sus ideas sin temor a la crítica destructiva.
- **Responsabilidad:** Nos hacemos cargo de nuestras tareas asignadas y somos accountable de nuestros resultados. Cumplimos con los plazos establecidos y comunicamos proactivamente cualquier impedimento.
- **Comunicación Asertiva y Transparente:** Priorizamos una comunicación clara, honesta y oportuna. Creemos en dar y recibir feedback constructivo de manera respetuosa para la mejora continua del proyecto y del equipo.
- **Compromiso:** Estamos dedicados al éxito del proyecto y al cumplimiento de los objetivos comunes. Ponemos nuestro mejor esfuerzo y dedicación en cada tarea que realizamos.
- **Colaboración y Apoyo Mutuo:** Trabajamos de manera sinérgica, entendiendo que el éxito es colectivo. Estamos dispuestos a ayudarnos entre nosotros para superar obstáculos y equilibrar cargas de trabajo cuando sea necesario.
- **Profesionalismo y Ética:** Actuamos con integridad, honestidad académica y seriedad en todas las fases del proyecto, respetando las normas del curso y los derechos de autor.

9) Compromiso del equipo

Nosotros, los integrantes del equipo, declaramos de manera voluntaria y unánime nuestro compromiso inquebrantable con el éxito de este proyecto. Nos obligamos a participar con puntualidad y asistencia constante en todas las reuniones de trabajo, notificando con antelación

cualquier impedimento. Nos comprometemos a cumplir de manera diligente con las tareas y entregables asignados, dentro de los plazos establecidos y manteniendo una comunicación proactiva sobre nuestro avance o cualquier obstáculo encontrado. Asumimos el compromiso de resolver cualquier diferencia de manera madura y constructiva, priorizando siempre los objetivos comunes. Finalmente, nos dedicamos a aportar nuestro máximo esfuerzo y capacidad de colaboración para alcanzar las metas propuestas, actuando en todo momento con ética, lealtad al equipo y dedicación al proyecto.

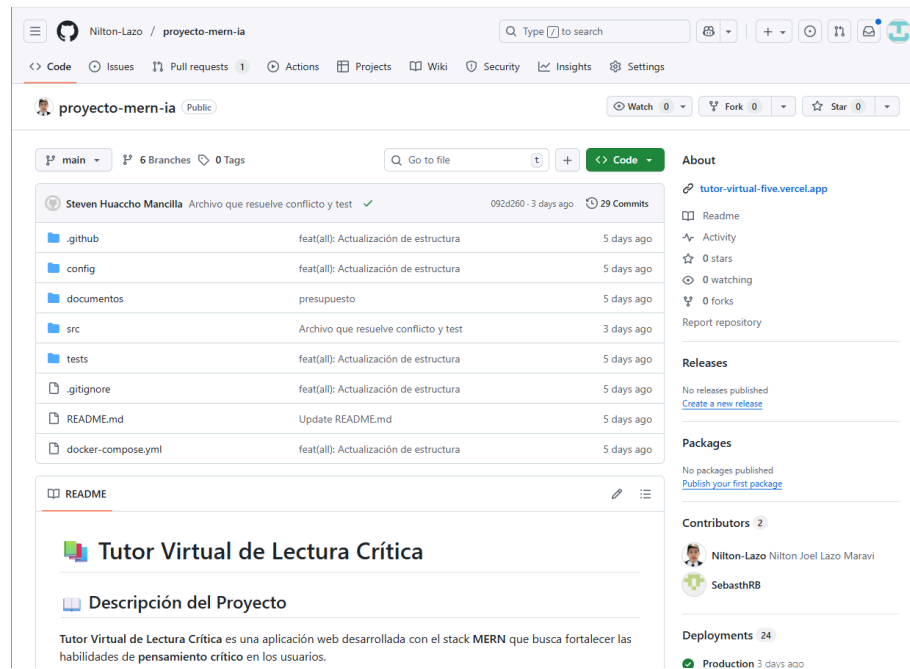
a) Firmas del equipo de trabajo

Integrantes	Firma
Huaccho Mancilla Steven José	
Lazo Maraví Nilton Joel	
Poma Goche Abigail Karim	
Ramirez Basualdo Lenin Sebastian	
Robles Sanchez Britney Sheyla	
Rojas Mellado Andrea Mirella	

3. Infraestructura y entorno técnico

3.1. Creación de repositorio en GitHub

Se realizó la creación del repositorio en el GitHub subiendo la base de creación del sistema para el proyecto “Tutor virtual de lectura crítica”.



3.2. Configuración inicial de ramas

Para el desarrollo de este punto cada uno de los integrantes mediante un proceso de mejora de la organización, mediante codificación y otros medios, creamos ramas para poner trabajar en conjunto y paralelamente según las ventajas que ofrece la herramienta GitHub.

Active branches					
Branch	Updated	Check status	Behind	Ahead	Pull request
rama-andrea	3 minutes ago	✓ 1 / 1	0	0	...
rama-britney	8 minutes ago	✓ 1 / 1	1	0	...
rama-lenin	14 minutes ago	✓ 1 / 1	0	0	...
rama-abigail	20 minutes ago	✓ 1 / 1	0	0	...
rama-steven	28 minutes ago	✓ 2 / 2	0	1	...

3.3. Estructura base del proyecto MERN

3.3.1. Carpetas



3.3.2. Dependencias mínimas

- Fronted

```
1 {
2   "name": "frontend",
3   "private": true,
4   "version": "0.0.0",
5   "type": "module",
6   "scripts": {
7     "dev": "vite",
8     "build": "tsc -b && vite build",
9     "lint": "eslint .",
10    "preview": "vite preview"
11  },
12  "dependencies": {
13    "react": "^19.1.1",
14    "react-dom": "^19.1.1"
15  },
16  "devDependencies": {
17    "@eslint/js": "^9.33.0",
18    "@types/react": "^19.1.10",
19    "@types/react-dom": "^19.1.7",
20    "@vitejs/plugin-react": "^5.0.0",
21    "eslint": "^9.33.0",
22    "eslint-plugin-react-hooks": "^5.2.0",
23    "eslint-plugin-react-refresh": "^0.4.20",
24    "globals": "^16.3.0",
25    "typescript": "~5.8.3",
26    "typescript-eslint": "^8.39.1",
27    "vite": "^7.1.2"
28  }
29 }
```

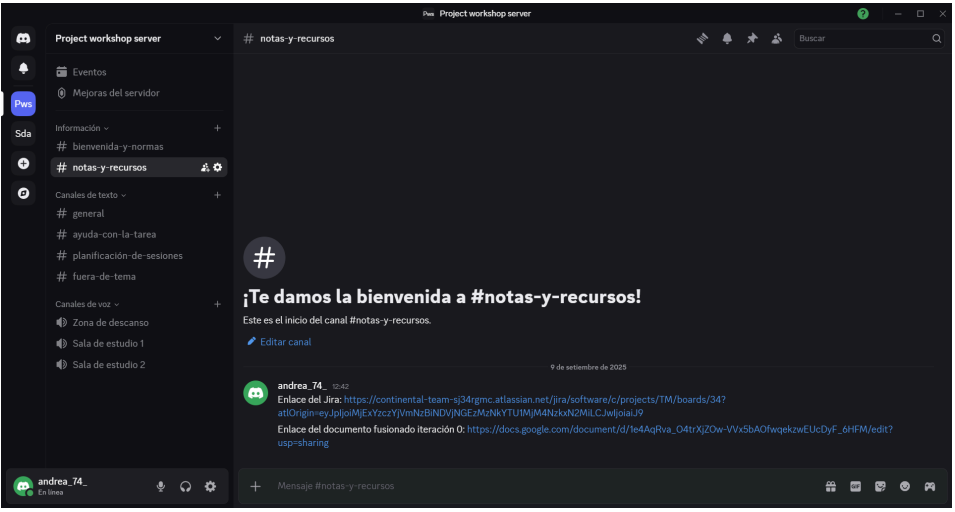
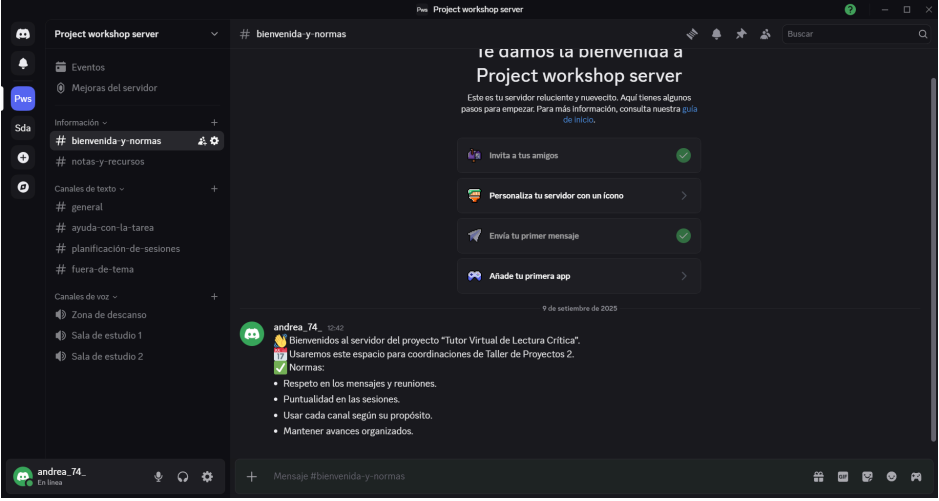
- Backend

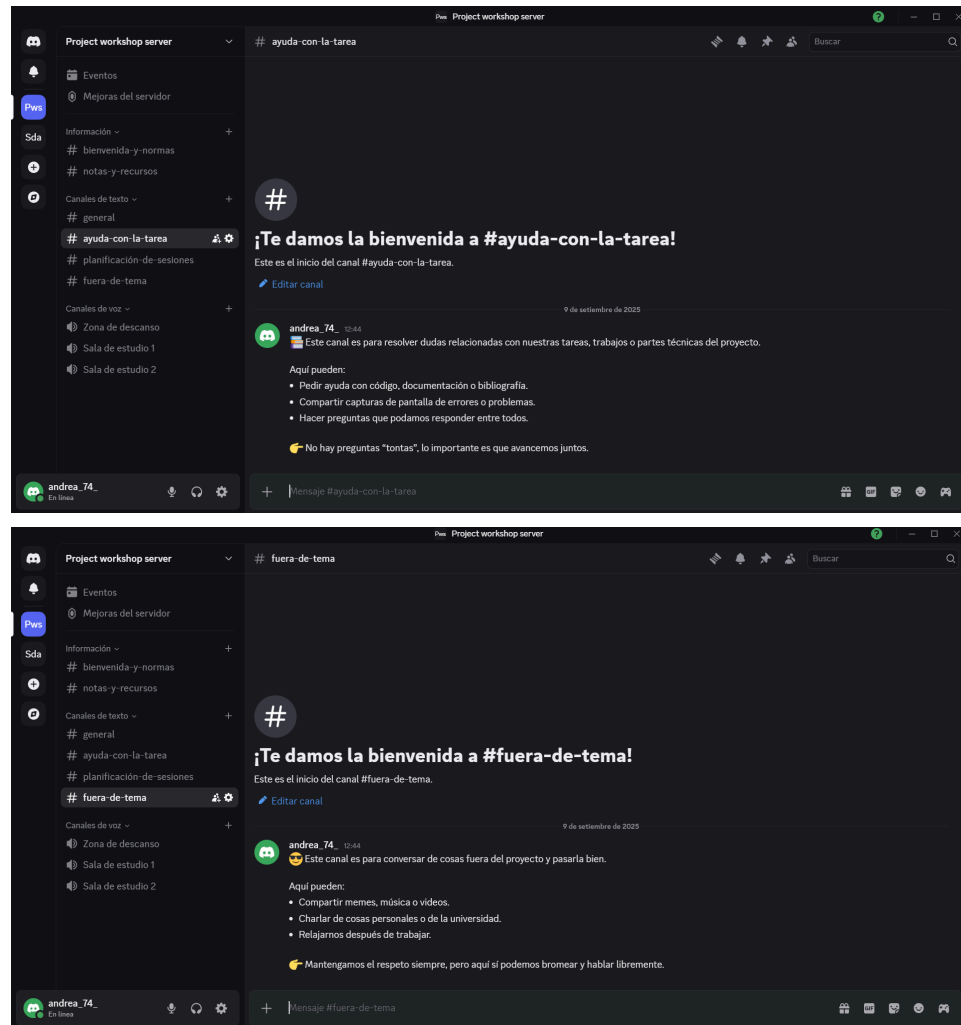
```
1 {
2   "name": "backend",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "start": "node index.js",
8     "dev": "nodemon index.js",
9     "test": "echo \\\"Error: no test specified\\\" && exit 1"
10  },
11  "keywords": [],
12  "author": "",
13  "license": "ISC",
14  "packageManager": "pnpm@10.15.1",
15  "dependencies": {
16    "express": "^5.1.0",
17    "mongoose": "^8.18.0"
18  },
19  "devDependencies": {
20    "nodemon": "^3.1.10"
21  }
22 }
```

4. Procesos de equipo

4.1. Definición de canales de comunicación

Se eligió el canal de comunicación Discord, a motivo de la facilidad del grupo de utilizar debido al conocimiento de los miembros del equipo para utilizar esta herramienta.





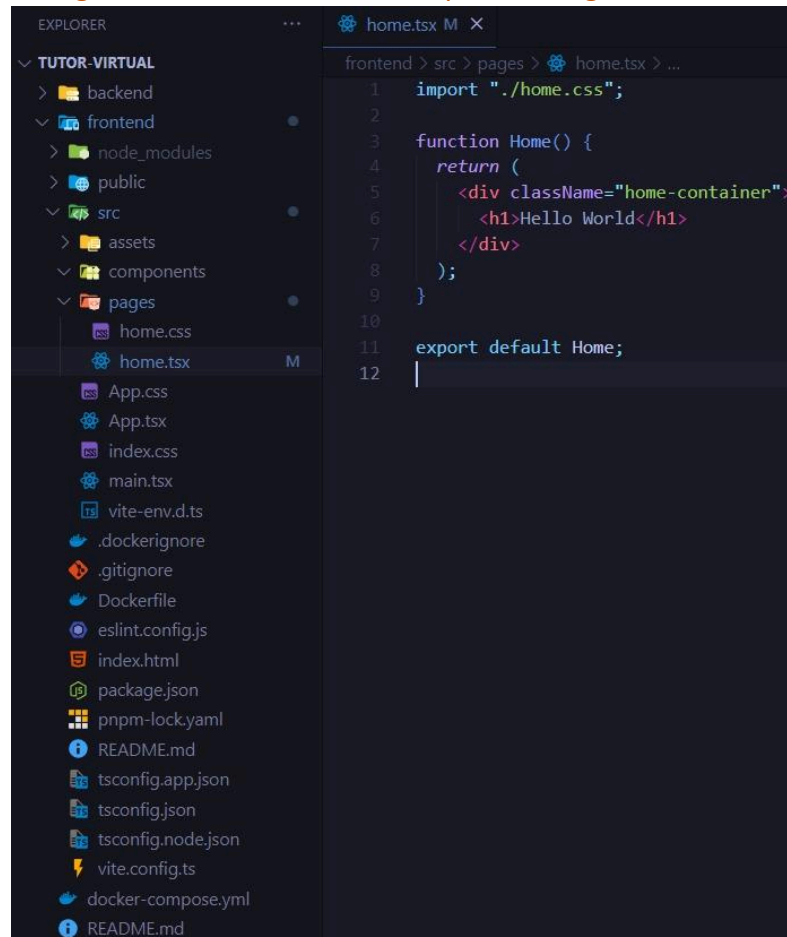
4.2. Acuerdos de trabajo: horarios de reunión, definition of done, roles iniciales

1. **Reuniones:** Se realizarán dos veces por semana (martes y viernes a las 8:00 p. m.) con todo el equipo, para revisar avances, resolver impedimentos y coordinar entregables.
2. **Definition of Done para historias de usuario:** Una historia de usuario se considerará terminada únicamente cuando cumpla con todos sus criterios de aceptación, el código haya sido revisado por pares, se hayan ejecutado pruebas unitarias e integrales con una cobertura mínima del 70 %, está documentada en el repositorio (README/Swagger) y la funcionalidad se encuentre desplegada en el entorno de staging, lista para demostración.
3. **Definition of Done para sprints:** Un sprint se dará por concluido solo si el 100 % de las historias comprometidas cumplen su definición de hecho, se presenta un incremento funcional potencialmente desplegable en demo al patrocinador/docente, se han corregido errores críticos, se realizó la retrospectiva documentando acciones de mejora, y el backlog queda refinado para el siguiente sprint.

4. **Roles iniciales:** Los roles fueron asignados según el punto 1.6 de este documento, manteniendo responsabilidades claras para gerente de proyecto, líderes y miembros del equipo.

5. Arquitectura mínima

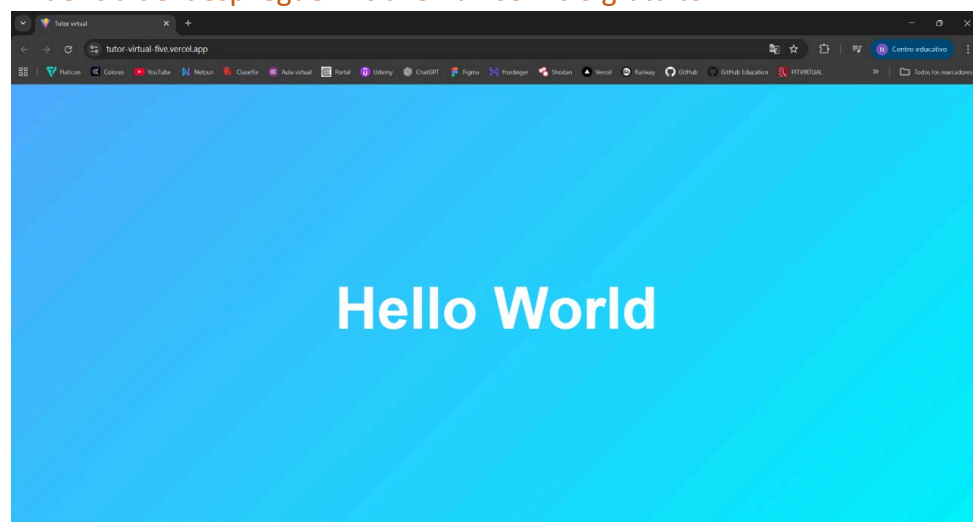
5.1. Configuración básica: conexión Express-MongoDB, Hello World en React



The screenshot shows the VS Code interface. On the left, the Explorer sidebar displays the project structure for 'TUTOR-VIRTUAL'. The 'pages' folder is expanded, showing 'home.css' and 'home.tsx'. The 'home.tsx' file is selected and its content is displayed in the main editor. The code defines a 'Home' function that returns a JSX element with a 'div' containing an 'h1' with the text 'Hello World'. The function is then exported as the default export.

```
1 import "../home.css";
2
3 function Home() {
4   return (
5     <div className="home-container">
6       <h1>Hello World</h1>
7     </div>
8   );
9 }
10
11 export default Home;
12
```

5.2. Evidencia del despliegue inicial en un servicio gratuito



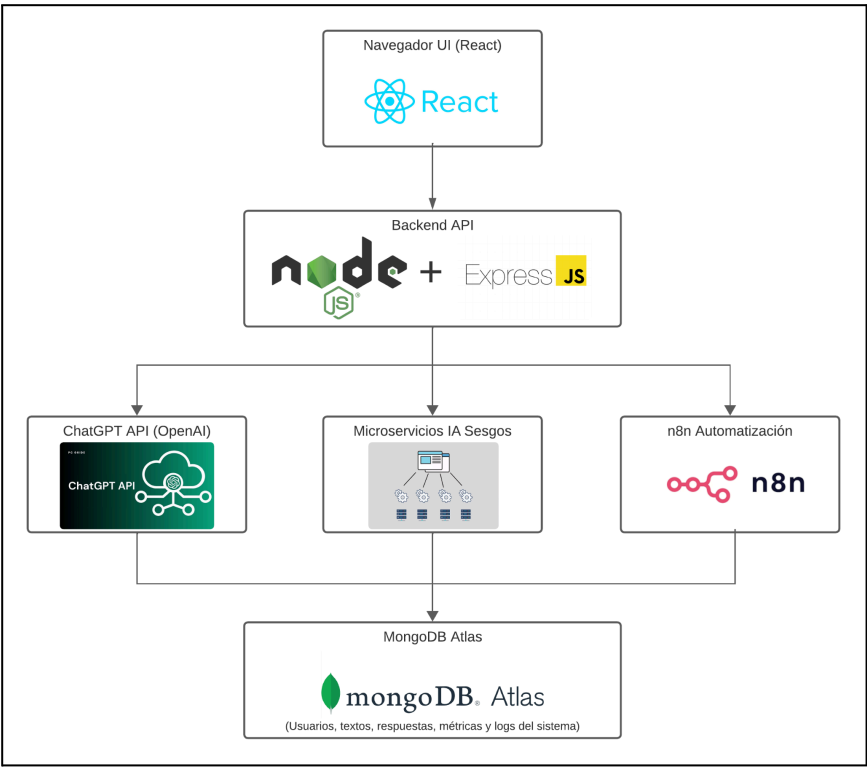
Enlace: <https://tutor-virtual-five.vercel.app/>

5.3. Diagrama de arquitectura con IA y automatización

5.3.1. Diagrama de componentes principales

Componente	Tecnología	Función
Frontend (SPA)	React (Vercel)	Interfaz de usuario, lectura de textos, presentación de preguntas, reportes.
Backend API	Node.js + Express	Gestión de usuarios, autenticación JWT, lógica de negocio, conexión con IA y automatización.
Base de datos	MongoDB Atlas	Almacenamiento de usuarios, roles, textos, respuestas, resultados de IA y logs.
IA – Generación de preguntas	ChatGPT API (NLP)	Genera automáticamente preguntas a partir de textos.
IA – Clasificación de sesgos	Microservicios IA (Python)	Detecta sesgos y falacias en los textos enviados.
Automatización	n8n (self-hosted)	Orquesta recordatorios, asignación de textos y registro automático de actividades.
CI/CD	GitHub Actions + Docker	Integración continua, pruebas automatizadas y despliegue en entornos de staging/producción.

5.3.2. Diagrama de arquitectura (ASCII)



5.3.3. Flujos principales

1. Flujo de generación de preguntas (IA):

- El usuario accede desde el **Frontend (React)**.
- El **Backend** envía el texto a la **API de ChatGPT (OpenAI)** para generar preguntas.

- c. La IA devuelve preguntas personalizadas → se almacenan en **MongoDB** y se muestran en la interfaz de usuario.

2. Flujo de detección de sesgos (IA):

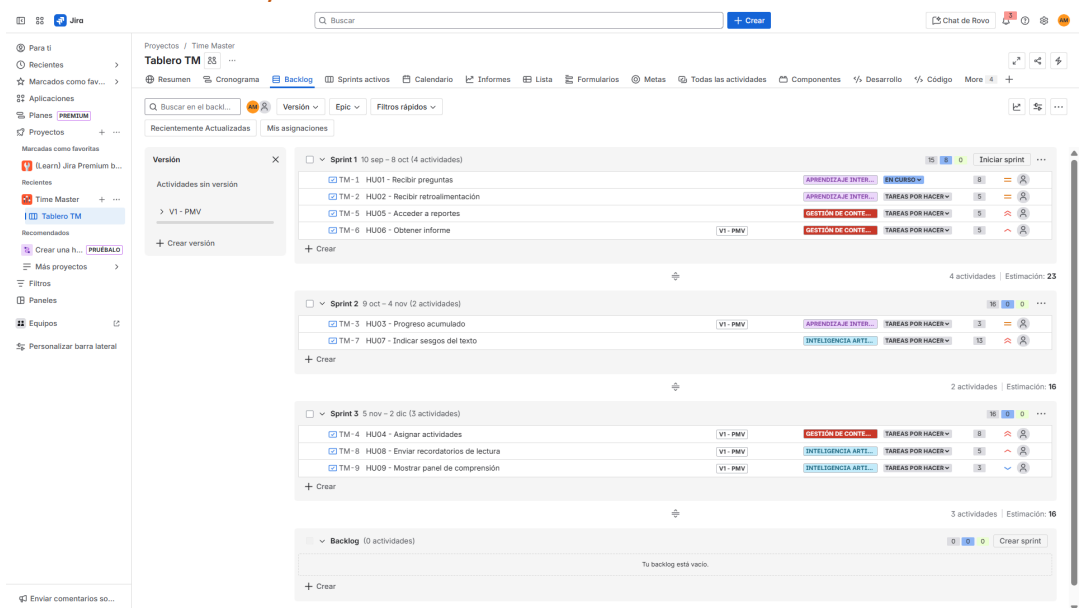
- a. El usuario analiza un texto desde el Frontend.
- b. El **Backend** lo envía al **Microservicio de Sesgos**.
- c. El modelo devuelve etiquetas o probabilidades de sesgo → se registran en **MongoDB** para su consulta posterior.

3. Flujo de automatización (n8n):

- a. **n8n** programa y envía notificaciones o recordatorios (por ejemplo, lecturas y actividades asignadas).
- b. También registra automáticamente en la **base de datos** los textos enviados y las respuestas recibidas.
- c. El **Backend** consulta estos registros y los presenta en los reportes de progreso para docentes y administradores.

6. Anexos

6.1. Planificación del Proyecto - Jira



<https://continental-team-sj34rgmc.atlassian.net/jira/software/c/projects/TM/boards/34/backlog?atlOrigin=eyJpIjoiYjYxNjcwZDVjNTBiNGFmOWEzMjUzODA4NjgzNmRiMTEiLCJwIjoiaj9>