

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



Facultad de Ingeniería de Sistemas E Informática

**"Tutor Virtual De Lectura  
Crítica"**

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

**DOCENTE:** GAMARRA MORENO, JOB DANIEL

**NRC:** 62100

**ESTUDIANTE:**

APELLIDOS Y NOMBRES	CÓDIGO
Dueñas Guerra Jesús Korlant	72122566
Gutierrez Medina Jesús Manuel	75826567
Chavez Estrella José Jhovanni	75557750
Lopez Idone Jose Gianmarco	71698090
Reymundo Rodriguez Cristhian Jhon	75049277
Asistente Virtual	

**HUANCAYO - 2025**

**CONTENIDO**

<b>1. Consigna</b>	<b>3</b>
<b>1.1. Project Charter (Acta de Proyecto)</b>	<b>3</b>
1.1.1. Propósito y justificación del proyecto	3
1.1.2. Objetivos generales	3
1.1.3. Alcance Inicial	4
1.1.4. Supuestos y restricciones	4
<b>Límites del Proyecto</b>	<b>5</b>
1.1.5. Identificación de los interesados (stakeholders)	6
1.1.6. Requisitos de Alto Nivel	7
1.1.7. Roles y responsabilidades del equipos	10
<b>1.2. Backlog inicial</b>	<b>10</b>
1.2.1. Identificación del Epic central y descomposición en historias.	10
1.2.2. Lista de 6 a 8 historias de usuario priorizadas con INVEST.	10
<b>1.3. Infraestructura y entorno técnico</b>	<b>13</b>
1.3.1. Creación de repositorio en GitHub (un integrante crea, el resto clona)	13
1.3.2. Configuración inicial de ramas (main, develop, ramas por funcionalidad)	13
1.3.3. Estructura base del proyecto MERN (carpetas, dependencias mínimas)	13
<b>1.4. Procesos de equipo</b>	<b>14</b>
1.4.1. Definición de canales de comunicación (ej. Slack, Discord)	14
1.4.2. Acuerdos de trabajo: horarios de reunión, Definition of Done, roles iniciales.)	14
<b>1.5. Arquitectura mínima (Walking Skeleton)</b>	<b>16</b>
1.5.1. Configuración básica: conexión Express–MongoDB, Hello World en React.	16
1.5.2. Evidencia de despliegue inicial en un servicio gratuito (ej. Render, Vercel,Railway).	17
<b>1.6. Entregables</b>	<b>17</b>
1.6.1. Documento (Markdown, PDF o Google Docs) con:	17
1.6.2. URL del repositorio GitHub con	18
1.6.3. Evidencia del despliegue inicial (screenshot o URL)	18

## 1. Consigna

### 1.1. Project Charter (Acta de Proyecto)

**Nombre del Proyecto:** Tutor Virtual de Lectura Crítica

**Fecha de Inicio:** 28/08/2025

**Fecha de Finalización:** Prevista: 04/12/2025

**Sponsor del Proyecto:** Universidad Continental

**Gerente del Proyecto:** Dueñas Guerra Jesus Korlant

**Equipo del Proyecto:** Equipo de desarrollo full-stack MERN

#### 1.1.1. Propósito y justificación del proyecto

Desarrollar una aplicación web interactiva que sirva como tutor virtual de lectura crítica, con apoyo de inteligencia artificial (IA de NLP) y automatización (n8n), orientada a estudiantes universitarios. El sistema busca mejorar la comprensión lectora y el pensamiento crítico, ofreciendo rutas personalizadas de estudio y retroalimentación inmediata.

#### 1.1.2. Objetivos generales

- Desarrollar una aplicación web con stack MERN (MongoDB, Express.js, React.js, Node.js).
- Entregar PMV funcional en 4 sprints
- Integrar al menos dos funcionalidades de IA: generación de preguntas y detección de sesgos.
- Implementar automatización con n8n para gestionar sesiones, notificaciones y seguimiento.
- Asegurar usabilidad, accesibilidad y diseño responsive.
- Cumplir con las prácticas de Git Flow, pruebas automatizadas y contenerización con Docker.

Objetivo	Fecha de Vencimiento
Desarrollar una aplicación web full-stack con el stack MERN (MongoDB, Express, React, Node.js) que permita evaluar y mejorar la comprensión lectora de estudiantes universitarios.	Semana 1 – 15
Implementar un módulo de generación automática de preguntas críticas y de opción múltiple mediante IA (OpenAI/Hugging Face).	Semana 6
Integrar un sistema de detección de sesgos y falacias en los textos cargados, con retroalimentación automática.	Semana 8
Configurar flujos de automatización con n8n para enviar recordatorios y registrar el progreso de los estudiantes.	Semana 10

Contenerizar la solución con Docker y establecer un pipeline CI/CD para garantizar portabilidad y despliegues reproducibles.	Semana 12
Desarrollar y ejecutar pruebas unitarias y E2E (Jest, Cypress) con al menos 70% de cobertura.	Semana 14
Documentar el sistema con manuales de usuario, guía técnica y métricas de usabilidad y accesibilidad (WCAG 2.1 AA).	Semana 15

### 1.1.3. Alcance Inicial

- Frontend en React.js con estado gestionado (Context API o Redux).
- Backend con Express.js y API REST.
- Base de datos MongoDB.
- Integración con modelos de NLP (Hugging Face/OpenAI) para IA.
- Automatización de flujos con n8n: asignación → lectura → actividad → evaluación → notificación.
- Pruebas unitarias y E2E con cobertura ≥70%.
- Documentación técnica y diagramas de arquitectura.

### 1.1.4. Supuestos y restricciones

#### Supuestos

Categoría	Descripción	Estado/Comentario
Tecnológica	Los modelos de IA de Hugging Face/OpenAI estarán disponibles vía API.	Validado, APIs accesibles en capa gratuita.
Recursos	El equipo cuenta con conocimientos técnicos en MERN e integración de IA.	En proceso de fortalecimiento con autoestudio y tutoriales.
Conectividad	Los usuarios tendrán acceso a internet estable para usar la plataforma.	Probablemente, se requiere piloto inicial en campus con WiFi.
Automatización	n8n permitirá la orquestación de flujos sin requerir licencia premium.	Confirmado, plan free disponible.
Académico	Se contará con retroalimentación del docente en cada sprint.	Pendiente de coordinación formal.

## Restricciones

Categoría	Descripción	Estado/Comentario
Tiempo	El desarrollo se limita al semestre (4 meses, 5 sprints).	Fijo, no ampliable.
Presupuesto	Presupuesto \$0: solo recursos gratuitos (MongoDB Atlas, Hugging Face, etc.).	Permanente.
Tecnológica	Uso exclusivo de stack MERN y software open-source.	Aceptada por el equipo.
Calidad	Debe cumplir con $\geq 70\%$ cobertura en pruebas unitarias/E2E.	Restricción técnica.
Accesibilidad	La aplicación debe ser responsive y cumplir con WCAG 2.1 nivel AA.	Normativa interna.

## Límites del Proyecto

Incluye:

- Desarrollo de la aplicación web con stack MERN (MongoDB, Express, React, Node.js).
- Implementación de módulos de IA para generación de preguntas y detección de sesgos/falacias.
- Automatización con n8n para recordatorios y notificaciones de avance.
- Panel de progreso para estudiantes y docentes.
- Pruebas unitarias y end-to-end con cobertura mínima del 70%.
- Documentación técnica y manual de usuario.

No incluye:

- Desarrollo de aplicación móvil nativa.
- Integración con sistemas de gestión académica institucional.
- Acceso a todas las bases de datos de pago (se usará solo contenido abierto).
- Soporte offline o aplicaciones de escritorio.

Resultado esperado	Indicador de éxito
Aplicación web funcional desplegada en la nube (Render/Vercel).	Acceso a través de una URL pública y operativa.
Generación automática de preguntas de comprensión lectora.	Mínimo 5 preguntas por texto con nivel literal, inferencial y crítico.
Módulo de detección de falacias y sesgos implementado.	Identificación correcta en al menos el 80% de un set de prueba docente.
Automatización de recordatorios y seguimiento en n8n.	Notificaciones enviadas automáticamente a estudiantes según cronograma.
Pruebas unitarias y E2E ejecutadas con éxito.	Cobertura mínima del 70% en CI/CD.
Documentación y manual de usuario finalizados.	Documento entregado con guías de instalación y uso.

#### 1.1.5. Identificación de los interesados (stakeholders)

Stakeholder	Rol en el proyecto	Interés principal	Influencia	Estrategia de gestión/comunicación
<b>Universidad Continental (Facultad de Ingeniería de Sistemas e Informática)</b>	Sponsor académico	Que el proyecto cumpla con los objetivos de curso y estándares académicos	Alta	Reuniones de revisión en cada entrega, entrega de informes de avance
<b>Docente del curso (Daniel Gamarra)</b>	Stakeholder clave y evaluador	Recibir entregables completos y en línea con la rúbrica	Alta	Informes semanales, revisiones de sprint, retroalimentación continua
<b>Estudiantes universitarios (usuarios finales)</b>	Usuarios principales	Mejorar su comprensión lectora y pensamiento crítico con la aplicación	Media-Alta	Pilotos controlados, encuestas de satisfacción, capacitaciones breves

<b>Docentes universitarios</b>	Usuarios secundarios	Evaluar progreso de sus estudiantes y tener un panel de métricas	Media	Pruebas piloto con retroalimentación, guías de usuario
<b>Equipo de proyecto (integrantes)</b>	Desarrolladores	Cumplir con plazos, roles y responsabilidades	Alta	Reuniones semanales (daily, sprint review), uso de Trello/Jira
<b>Asistente Virtual (herramienta IA)</b>	Facilitador técnico	Integración efectiva con NLP y automatización	Media	Monitoreo técnico, documentación de APIs y pruebas unitarias
<b>Servicios en la nube (MongoDB Atlas, Vercel, Render, n8n)</b>	Proveedores de infraestructura	Ofrecer disponibilidad, accesibilidad y compatibilidad gratuita	Media	Seguimiento técnico y validación periódica
<b>Docentes externos (evaluadores de pruebas piloto)</b>	Stakeholders de validación	Confirmar aplicabilidad y beneficios del proyecto en un entorno académico	Baja-Media	Sesiones demostrativas, cuestionarios de retroalimentación

### 1.1.6. Requisitos de Alto Nivel

- Requisitos Funcionales

- El sistema debe permitir el registro y autenticación de usuarios (estudiantes y docentes).
- El sistema debe generar preguntas contextuales a partir de los textos cargados.
- El sistema debe detectar sesgos y falacias en los textos procesados.
- El sistema debe ofrecer rutas de estudio personalizadas.
- El sistema debe enviar notificaciones y recordatorios automáticos mediante flujos en n8n.
- El sistema debe registrar y mostrar el progreso de los estudiantes en la plataforma.
- El sistema debe permitir la evaluación de comprensión lectora en base a actividades programadas

- Requisitos No Funcionales
  - La aplicación debe desarrollarse con el stack MERN (MongoDB, Express.js, React.js, Node.js).
  - El frontend debe estar desarrollado en React.js con manejo de estado mediante Context API o Redux.
  - El backend debe implementarse con Express.js exponiendo servicios a través de una API REST.
  - La base de datos debe ser MongoDB en la nube (ejemplo: MongoDB Atlas).
  - El sistema debe integrar modelos de NLP (Hugging Face / OpenAI) a través de APIs externas.
  - La aplicación debe ser responsive y cumplir con las normas de accesibilidad WCAG 2.1 AA.
  - El sistema debe tener al menos 70% de cobertura en pruebas unitarias y E2E.
  - Se debe entregar documentación técnica completa (README, diagramas de arquitectura, manual de usuario).
  - El sistema debe desplegarse en contenedores Docker.
- Entregables Principales:
  - Código fuente en GitHub.
  - Documentación (README, diagramas, informe técnico).
  - Demo funcional con video explicativo (opcional para sobresaliente).
  - Informe de impacto ambiental.
- Hitos y cronograma tentativo
  - Sprint 1 (28/08 – 11/09): Configuración inicial, base de datos, frontend básico.
  - Sprint 2 (12/09 – 25/09): API REST, primeras integraciones de IA (preguntas contextuales).
  - Sprint 3 (26/09 – 09/10): Detección de sesgos, pruebas iniciales, automatización con n8n.
  - Sprint 4 (10/10 – 23/10): Mejoras de accesibilidad, responsive, pruebas E2E.
  - Sprint 5 (24/10 – 07/11): Documentación, CI/CD, despliegue en Docker.
  - Entrega final (04/12): Presentación y socialización de resultados.
- Riesgos identificados
  - Técnicos: problemas de integración con APIs externas de IA.
  - Recursos: limitación de tiempo del equipo dentro del semestre académico.
  - Usuarios: resistencia inicial al uso de herramientas digitales.
  - Mitigación: pruebas tempranas, documentación clara, sesiones de retroalimentación con usuarios piloto.

Presupuesto

Categoría	Detalle	Costo Estimado (S.)
Infraestructura	Servicios en la nube gratuitos (Render/Vercel, Railway, MongoDB Atlas). En caso de ampliación a planes básicos de pago.	800.00
Licencias de IA / APIs	OpenAI API, Hugging Face, o equivalentes (créditos iniciales para pruebas y despliegues).	1,200.00
Herramientas de Automatización	n8n (open-source, hosting en servidor propio gratuito).	0.00
Contenerización y CI/CD	Docker y GitHub Actions (gratuito en plan educativo).	0.00
Gestión de Proyecto	Herramientas de trabajo colaborativo (Trello, Jira, Google Drive).	0.00
Recursos Humanos	Dedicación estimada de 7 integrantes (trabajo académico no remunerado).	0.00
Reserva de Contingencia (15%)	Para imprevistos técnicos o ampliación de uso de API.	375.00
Reserva de Gestión (10%)	Fondo adicional para cambios de alcance menores.	250.00

- El presupuesto es académico y simulado, ya que la mayoría de servicios son gratuitos en sus planes iniciales.
- Se incluye un monto base para APIs .
- La reserva de contingencia cubre cualquier imprevisto técnico.
- La reserva de gestión permite flexibilidad ante cambios menores en el alcance.

Criterios de Aceptación:

- Aplicación funcional y desplegable con Docker.
- Cumplimiento de los requisitos técnicos y funcionales.
- Cobertura de pruebas  $\geq 70\%$ .
- Documentación completa y clara.

### 1.1.7. Roles y responsabilidades del equipos

Rol	Responsabilidades
Líder de Proyecto Dueñas Guerra Jesús Kirlant	Coordinación general, seguimiento de sprints, comunicación con stakeholders.
Frontend Developer Chavez Estrella José Jhovanni	Desarrollo de la interfaz en React.js, estado con Context API/Redux, pruebas.
Backend Developer Lopez Idone Jose Gianmarco	API REST con Express.js, integración con MongoDB, lógica de negocio.
Especialista en IA Gutierrez Medina Jesús Manuel	Integración de modelos de NLP, generación de preguntas, detección de sesgos.
DevOps/Automation Gutierrez Medina Jesús Manuel	Configuración de Docker, n8n, CI/CD, despliegue.
Tester Cristhian Jhon Reymundo Rodriguez	Pruebas unitarias, E2E, documentación de casos de prueba.

## 1.2. Backlog inicial

### 1.2.1. Identificación del Epic central y descomposición en historias.

- Épica 01 – Evaluación de Textos
- Épica 02 – Funcionalidades con IA
- Épica 03 – Seguimiento y Automatización

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

"Como estudiante universitario, quiero contar con una plataforma digital con inteligencia artificial y automatización, para mejorar mi comprensión lectora y pensamiento crítico, de manera que pueda reforzar mis habilidades académicas de forma autónoma."

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.
- **Testable:** cada historia cuenta con criterios de aceptación claros que permiten verificar su cumplimiento. Código Historia de Usuario Puntos Duración Criterio

#### Épica 01 – Evaluación de textos

Código	Historia de Usuario	Duración	Criterios de Aceptación	Prioridad
HU01	Como estudiante, quiero subir un texto (PDF/DOCX/TXT) para que el sistema lo procese.	1 semana	- Subida ≤10 MB.- Texto guardado en BD.- Mensaje de confirmación.	Alta – base estructural

#### Épica 02 – Funcionalidades con IA

Código	Historia de Usuario	Duración	Criterios de Aceptación	Prioridad
HU02	Como estudiante, quiero que el sistema genere preguntas de comprensión lectora de diferentes niveles.	3 semanas	- ≥5 preguntas por texto.- Preguntas clasificadas (literal, inferencial, crítico).- Exportación en CSV.	Alta – valor académico
HU03	Como estudiante, quiero que el sistema detecte falacias y sesgos en el texto cargado.	3 semanas	- Reporte con tipo de falacia.- Destacar fragmento afectado..- Precisión ≥80% en pruebas docentes.	Alta – diferenciador

## Épica 03 – Seguimiento y Automatización

Código	Historia de Usuario	Duración	Criterios de Aceptación	Prioridad
HU04	Como estudiante, quiero ver un dashboard con mi progreso en las prácticas.	2 semanas	- Métricas de intentos completados.- Historial de notas por intento.- Visualización gráfica.	Media-Alta
HU05	Como estudiante, quiero recibir notificaciones automáticas para recordar mis actividades.	1 semana	- Integración con n8n.- Recordatorios vía correo o app.- Configuración personalizada.	Media
HU06	Como docente, quiero acceder a un panel para visualizar el rendimiento de mis estudiantes.	2 semanas	- Ver progreso agregado por curso.- Exportar métricas en PDF/Excel.- Roles diferenciados (docente/estudiante).	Media
HU07	Como usuario, quiero registrarme e iniciar sesión de forma segura.	1 semana	- Registro/login con credenciales cifradas.- Roles: estudiante/docente.- Sesión segura con JWT.	Alta – seguridad básica

### 1.3. Infraestructura y entorno técnico

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

Nombre APP : AppComprende

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

The screenshot shows the GitHub repository page for 'Comprende-IA'. At the top, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. Below the navigation is a search bar and a 'Type to search' input field. The main content area shows the 'main' branch with 2 branches and 0 tags. A commit from 'JosCh1999' titled 'Update README.md' is listed, along with other commits from 'backend' and 'frontend'. To the right, there's an 'About' section with a brief description of the project: 'Aplicación web de tutoría de lectura crítica que utiliza inteligencia artificial que busca mejorar la comprensión lectora de estudiantes y docentes'. Below the 'About' section are links for Readme, Activity, and Stars.

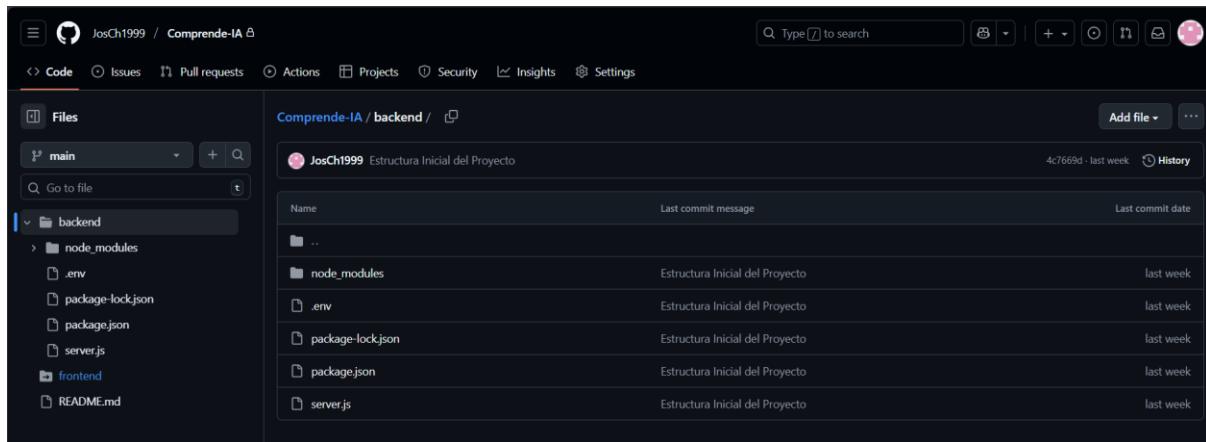
### 1.3.2. Configuración inicial de ramas (main, develop, ramas por funcionalidad)

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

The screenshot shows the GitHub 'Branches' page. At the top, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. Below the navigation is a search bar and a 'Type to search' input field. The main content area shows two sections: 'Default' and 'Your branches'. Under 'Default', there's a table for the 'main' branch showing its status: Updated 24 minutes ago, Check status 0 / 1, Behind/Ahead Default, and a Pull request button. Under 'Your branches', there's a table for the 'develop' branch showing its status: Updated last week, Check status 0 / 1, Behind/Ahead 2 / 0, and a Pull request button. There's also a 'New branch' button at the top right of the 'Default' section.

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

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



## 1.4. Procesos de equipo

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

Comunicación sincrónica: GoogleMeet (reuniones de voz/video).

Comunicación asincrónica: WhatsApp (mensajes rápidos).

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

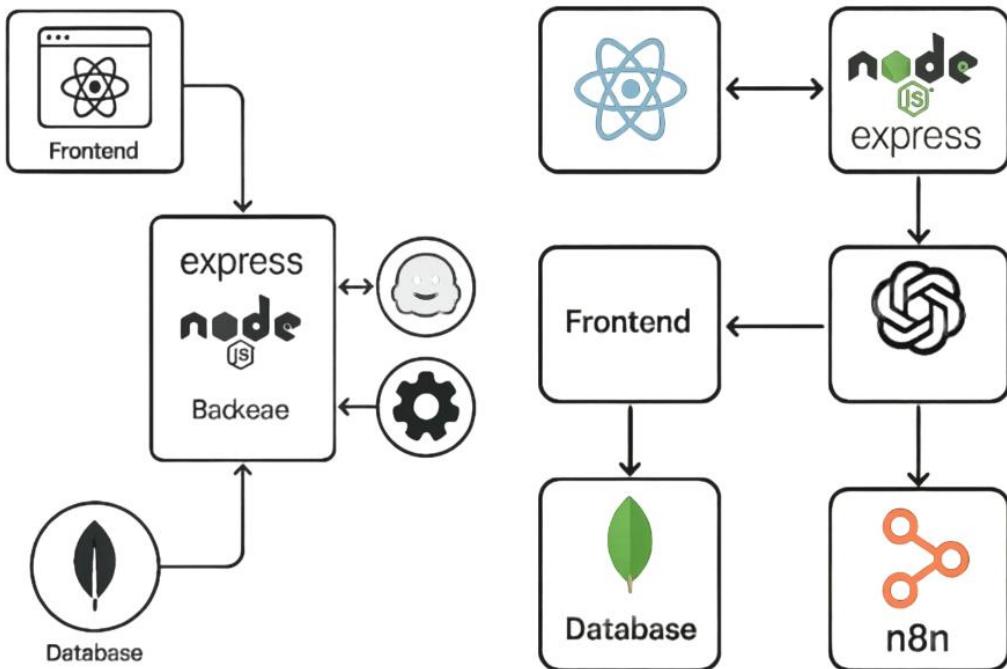
Acuerdos de trabajo	
	<b>Reuniones de seguimiento:</b> Martes y miércoles, 2 veces por semana.
Horarios de reunión:	<b>Daily breve:</b> Al inicio de cada clase (5–10 minutos).
	<b>Reuniones extraordinarias:</b> Convocadas únicamente en caso de incidencias críticas.
	<b>Repositorio:</b> Código subido en la rama correspondiente de GitHub.
	<b>Revisión:</b> Validado mediante pull request por al menos un integrante.
Definition of Done (DoD):	<b>Pruebas:</b> Ejecución exitosa de pruebas unitarias y E2E.
	<b>Documentación:</b> Actualización en Jira o comentarios en el código.
	<b>Validación final:</b> Aprobación del Project Manager o Scrum Master.

<b>Roles iniciales (ya definidos en la Declaración del Equipo):</b>	<ul style="list-style-type: none"> <li>• <b>Project Manager:</b> Dueñas Guerra Jesús Koralnt</li> <li>• <b>Scrum Master / Programador:</b> Chavez Estrella José Jhovanni</li> <li>• <b>Programador / IA:</b> Gutierrez Medina Jesús Manuel</li> <li>• <b>Programador MERN:</b> Chavez Estrella José Jhovanni</li> <li>• <b>Tester:</b> Cristhian Jhon Reymundo Rodriguez</li> <li>• <b>Documentación:</b> Gutierrez Medina Jesús Manuel</li> <li>• <b>Auditor:</b> Dueñas Guerra Jesús Koralnt</li> </ul>
---	---

<b>Métricas de Calidad</b>	
<b>Métrica</b>	<b>Descripción</b>
<b>Cobertura de pruebas</b>	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.
<b>Cumplimiento de plazos</b>	Entrega de $\geq 90\%$ de los entregables en las fechas establecidas.
<b>Documentación actualizada</b>	Avances acompañados de documentación técnica y de usuario vigente.

<b>Herramientas de seguimiento</b>	
<b>Herramienta</b>	<b>Uso principal</b>
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.

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

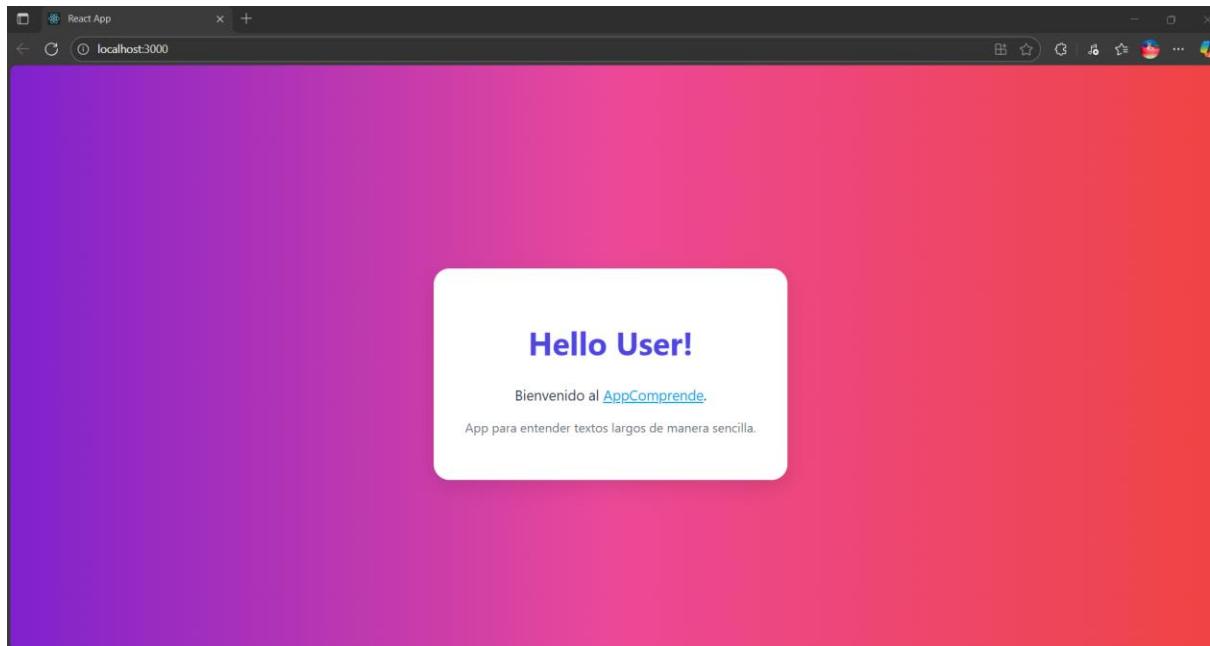
```
● PS D:\Comprende-AI> cd backend
○ PS D:\Comprende-AI\backend> npm start
> start
> node server.js

[dotenv@17.2.2] injecting env (1) from .env -- tip: * enable debug logging with { debug: true }
(node:11684) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:11684) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Conexión exitosa a MongoDB
Servidor escuchando en el puerto 5000
```

The screenshot shows the MongoDB Atlas Data Services interface. The top navigation bar includes 'Atlas' (with a green icon), date '2025-09-04 ...', 'Access Manager', 'Billing', 'All Clusters', 'Get Help', and a user profile 'José Jhovanni'. The main menu on the left has sections for 'DATABASE' (Clusters, Services), 'SERVICES' (Atlas Search, Stream Processing, Triggers, Migration, Data Federation), and 'SECURITY' (Quickstart, Backup, Database Access). The 'Clusters' section is selected. The central area shows 'Clusters' for 'Project 0' with a search bar 'Find a database deployment...'. A prominent chart titled 'Cluster0' shows metrics: R: 0, W: 0, Connections: 3.0, In: 5.20 B/s, Out: 1.00 KB/s, and Data Size: 116.51 MB / 512.00 MB (23%). Buttons for 'Edit Config', '+ Create', 'Explore Charts', and 'Dismiss' are visible.

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

Evidencia del despliegue inicial (screenshot o URL)



### 1.6. Entregables

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

Enlace Primer entregable:

[https://docs.google.com/document/d/1YKUKI70HdEFNpOx0U80AB2ZlpWEKRsktzeblZGGI5b8/edit?usp=drive\\_link](https://docs.google.com/document/d/1YKUKI70HdEFNpOx0U80AB2ZlpWEKRsktzeblZGGI5b8/edit?usp=drive_link)

Enlace de Minuta de Reunion:

[https://drive.google.com/file/d/1nDPITxC9zA93xiZb3qWW0LcP6n8zkSKT/view?usp=drive\\_link](https://drive.google.com/file/d/1nDPITxC9zA93xiZb3qWW0LcP6n8zkSKT/view?usp=drive_link)

### 1.6.2. URL del repositorio GitHub con

Github: <https://github.com/JosCh1999/Comprende-IA>

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

