

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



**Universidad  
Continental**

**PROYECTO**

**“Tutor Virtual de Lectura Comprensiva Escolar - I.E.P. San Carlos”**

**PRESENTADO POR:**

<b>APELLIDOS Y NOMBRES</b>	<b>CÓDIGO</b>
<b>Huaccho Mancilla Steven José</b>	<b>72620878</b>
<b>Lazo Maraví Nilton Joel</b>	<b>74091904</b>
<b>Poma Goche Abigail Karim</b>	<b>72541050</b>
<b>Ramirez Basualdo Lenin Sebastian</b>	<b>72695645</b>
<b>Robles Sanchez Britney Sheyla</b>	<b>72855637</b>
<b>Rojas Mellado Andrea Mirella</b>	<b>75411510</b>

**ASESOR:**

**Daniel Gamarra Moreno**

**HUANCAYO – PERÚ**

**2023**

## LISTA DE CONTENIDO

<b>PORADA.....</b>	<b>1</b>
<b>LISTA DE CONTENIDO.....</b>	<b>2</b>
<b>CAPÍTULO 1 PLANTEAMIENTO DEL ESTUDIO.....</b>	<b>5</b>
1.1. Aspectos Generales de la Empresa.....	5
1.1.1. Organigrama.....	5
1.1.2. Misión y Visión.....	5
1.2. Diagnóstico del Problema.....	6
1.3. Procesos de la Empresa.....	6
1.4. Oportunidad Encontrada.....	8
1.5. Detalles del Proyecto.....	9
<b>CAPÍTULO 2 ESTUDIO DE FACTIBILIDAD.....</b>	<b>10</b>
2.1. Alternativas de Solución.....	10
2.2. Factibilidad Técnica.....	10
2.2.1. Recursos tecnológicos.....	11
2.2.2. Requerimientos del entorno.....	11
2.3. Factibilidad Económica.....	11
2.3.1. Gastos Generales.....	12
2.4. Factibilidad Operacional.....	12
2.4.1. Sistema de Tutor de lectura Crítica.....	12
<b>CAPÍTULO 3 ANÁLISIS DE REQUERIMIENTOS.....</b>	<b>13</b>
3.1. Metas del Sistema de Información.....	13
3.2. Requisitos del Sistema.....	13
3.2.1. Requerimientos Funcionales.....	13
3.2.2. Requerimientos no Funcionales.....	14
3.3. Identificación de Actores del Sistema.....	15
3.3.1. Relación con los procesos Educativos.....	15
<b>CAPÍTULO 4 PLANIFICACIÓN DEL PROYECTO.....</b>	<b>15</b>
4.1. Definición de Roles de Trabajo.....	15
4.1.1. Product Owner.....	15
4.1.2. Scrum master.....	15
4.1.3. Team member.....	16
4.1.4. Tester.....	16
4.2. Product Backlog.....	16
4.3. Sprint Backlog.....	17
4.3.1. Sprint 1.....	17
4.3.2. Sprint 2.....	17
4.3.3. Sprint 3.....	17
4.4. Planificación de Sprints.....	17
4.4.1. Historias de usuario.....	17
4.4.2. Priorización de historias de usuario.....	18

4.5. Cronograma de Actividades.....	18
4.6. Gestión de riesgos.....	19
4.7. Aseguramiento de la calidad.....	20
4.7.1. Heurísticas de Nielsen (10 ítems, Likert 1–5).....	20
4.7.2. SUS – System Usability Scale (10 ítems, Likert 1–5).....	20
<b>CAPÍTULO 5 DISEÑO DEL SISTEMA DE INFORMACIÓN.....</b>	<b>21</b>
5.1. Diseño de Diagramas UML.....	21
5.1.1. Diagramas de casos de uso.....	21
5.1.2. Diagrama de secuencia.....	22
5.1.3. Diagramas de colaboración.....	22
5.1.4. Diagramas de clases.....	23
5.2. Diseño de base de datos.....	24
5.2.1. Diseño conceptual (E/R).....	24
5.2.2. Diseño lógico.....	24
5.2.3. Diseño físico.....	25
5.2.4. Modelado de base de datos.....	26
5.3. Diseño de Interfaces Básicas.....	30
5.3.1. Pantalla página de inicio.....	30
5.3.2. Pantalla de generación de preguntas y retroalimentación.....	31
5.3.3. Pantalla de registro - nuevo usuario.....	31
5.3.4. Pantalla de logueo / Inicio de sesión.....	31
5.3.5. Pantalla de reporte de respuestas.....	32
5.3.6. Pantalla de bloqueo al no iniciar sesión.....	32
5.3.7. Pantalla de actividades del estudiante.....	32
5.3.8. Pantalla de actividad asignada al estudiante.....	33
5.3.9. Pantalla Panel Docente.....	33
5.3.10. Pantalla de asignación de actividad del estudiante por parte del docente.....	33
<b>CAPÍTULO 6 CODIFICACIÓN DEL SOFTWARE.....</b>	<b>34</b>
6.1. Desarrollo del Sprint 1.....	34
6.1.1. Sprint planning.....	34
6.1.2. Sprint backlog.....	34
6.1.3. Historias de usuario.....	35
6.1.4. Taskboard.....	36
6.1.5. Daily scrum.....	36
6.1.6. Sprint review.....	37
6.1.7. Criterios de aceptación.....	37
6.1.8. Resultados del sprint.....	38
6.1.8.1. Evidencias.....	38
6.1.8.2. Prueba de desarrollo.....	38
6.1.9. Sprint retrospective.....	38
6.2. Desarrollo del Sprint 2.....	39
6.2.1. Sprint planning.....	39
6.2.2. Sprint backlog.....	39
6.2.3. Historias de usuario.....	40

6.2.4. Taskboard.....	40
6.2.5. Daily scrum.....	41
6.2.6. Sprint review.....	41
6.2.7. Criterios de aceptación.....	41
6.2.8. Resultados del sprint.....	42
6.2.8.1. Evidencias.....	42
6.2.8.2. Prueba de desarrollo.....	42
6.2.9. Sprint retrospective.....	42
6.3. Desarrollo del Sprint 3.....	43
6.3.1. Sprint planning.....	43
6.3.2. Sprint backlog.....	43
6.3.3. Historias de usuario.....	44
6.3.4. Taskboard.....	45
6.3.5. Daily scrum.....	45
6.3.6. Sprint review.....	46
6.3.7. Criterios de aceptación.....	46
6.3.8. Resultados del sprint.....	46
6.3.8.1. Evidencias.....	46
6.3.8.2. Prueba de desarrollo.....	46
6.3.9. Sprint retrospective.....	47
<b>CAPÍTULO 7 PRUEBAS DE SOFTWARE.....</b>	<b>47</b>
7.1. Plan de Pruebas.....	47
7.1.1. Objetivo del plan de pruebas.....	48
7.1.2. Alcance.....	48
7.1.3. Estrategia general de pruebas.....	48
7.1.4. Tipos de pruebas y herramientas.....	50
a) Pruebas unitarias (frontend y backend).....	50
b) Pruebas de integración de API.....	50
c) Pruebas de API con Postman / cliente REST.....	50
d) Pruebas end-to-end (E2E).....	51
e) Pruebas de usabilidad.....	52
7.1.5. Ambiente de pruebas y datos.....	52
7.1.6. Criterios de aceptación y salida.....	53
7.1.7. Trazabilidad con los requisitos.....	53
<b>CONCLUSIONES.....</b>	<b>54</b>
<b>RECOMENDACIONES.....</b>	<b>54</b>
<b>ANEXOS.....</b>	<b>54</b>
Anexo 01. Manual Técnico.....	54
Anexo 02. Manual de Usuario.....	54
<b>LISTA DE TABLAS.....</b>	<b>54</b>
<b>LISTA DE FIGURAS.....</b>	<b>54</b>

# CAPÍTULO 1 PLANTEAMIENTO DEL ESTUDIO

## 1.1. Aspectos Generales de la Empresa

### 1.1.1. Organigrama

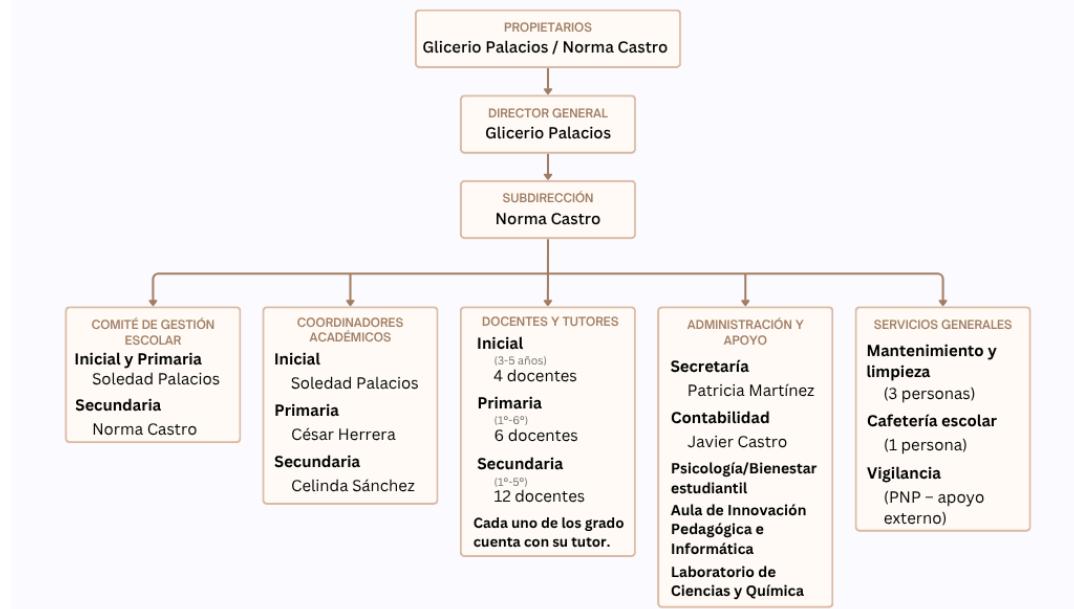


FIG.1 Organigrama

### 1.1.2. Misión y Visión

#### - Misión

“Somos una Institución Educativa que formamos líderes con una sólida formación en valores que les permita enfrentar con éxito los retos de este nuevo milenio.”

#### - Visión

“Ser una institución Educativa, acreditada y líder, con lo más exigentes estándares de calidad que contribuya al logro de un sociedad desarrollada; con justicia, libertad, soberanía y verdadera democracia”

## 1.2. Diagnóstico del Problema

En la Institución Educativa “San Carlos” se ha identificado una problemática relacionada con el desarrollo limitado de la comprensión lectora y el pensamiento crítico en los estudiantes de los niveles inicial, primaria y secundaria.

Aunque el colegio promueve la lectura como eje de su propuesta pedagógica, existen dificultades en el seguimiento individual del progreso lector, la creación de actividades personalizadas y la retroalimentación inmediata a los alumnos.

Actualmente, los docentes elaboran manualmente las preguntas y ejercicios para cada texto, lo que demanda tiempo y dificulta una atención diferenciada según el ritmo de aprendizaje. Además, las lecturas se realizan de forma tradicional, sin registros sistemáticos de avance ni herramientas que permitan analizar resultados.

Como consecuencia, muchos estudiantes no consolidan habilidades de análisis e inferencia, y los docentes carecen de medios automatizados para monitorear y mejorar la comprensión lectora.

Frente a ello, se plantea el desarrollo del Tutor Virtual de Lectura Comprensiva Escolar, una aplicación web que integra inteligencia artificial y automatización para generar preguntas automáticas, brindar retroalimentación y ofrecer a los docentes un panel de seguimiento del rendimiento lector. El proyecto busca optimizar la gestión del proceso lector, fortalecer las competencias comunicativas y fomentar el hábito de lectura desde edades tempranas mediante el uso innovador de la tecnología educativa.

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

En la Institución Educativa “San Carlos”, el proceso de evaluación constituye uno de los pilares fundamentales para medir el avance académico y la formación integral de los estudiantes en los niveles de inicial, primaria y secundaria.

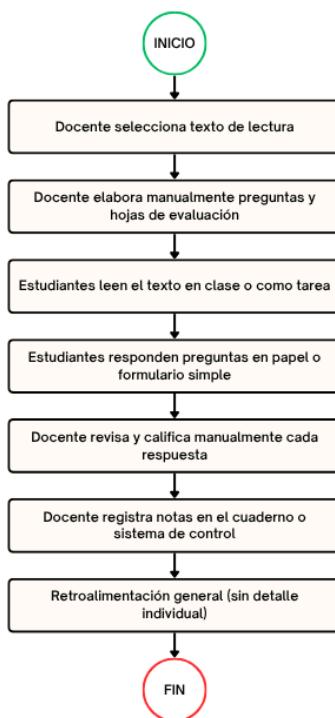
Actualmente, el sistema de evaluación se desarrolla mediante evaluaciones mensuales y bimestrales, complementadas en el nivel secundario con exámenes tipo admisión cada quince días, los cuales permiten medir la preparación de los alumnos frente a evaluaciones externas y reforzar el aprendizaje continuo.

Si bien este modelo permite mantener un control periódico del rendimiento académico, presenta limitaciones en la evaluación de competencias lectoras y de pensamiento crítico, ya que la valoración se centra principalmente en resultados cuantitativos y no en el proceso de comprensión o análisis del texto. Además, la corrección manual de los ejercicios y exámenes demanda un esfuerzo considerable del docente, dificultando la entrega de retroalimentación inmediata al estudiante.

Frente a ello, se identifica la necesidad de **optimizar el proceso de evaluación de la lectura comprensiva**, integrando herramientas tecnológicas que permitan **automatizar la generación de preguntas, calificación y seguimiento del desempeño lector**.

El **Tutor Virtual de Lectura Comprensiva Escolar** surge como una solución que complementa los procesos evaluativos tradicionales, permitiendo a los docentes realizar evaluaciones formativas continuas con apoyo de inteligencia artificial, mientras que los estudiantes pueden practicar la lectura y recibir retroalimentación automática antes de las evaluaciones mensuales o tipo admisión.

DIAGRAMA DEL PROCESO ACTUAL  
"Evaluación de Comprensión Lectora Tradicional"



#### DIAGRAMA DEL PROCESO MEJORADO

"Evaluación Automatizada con el Tutor Virtual de Lectura Comprensiva Escolar"

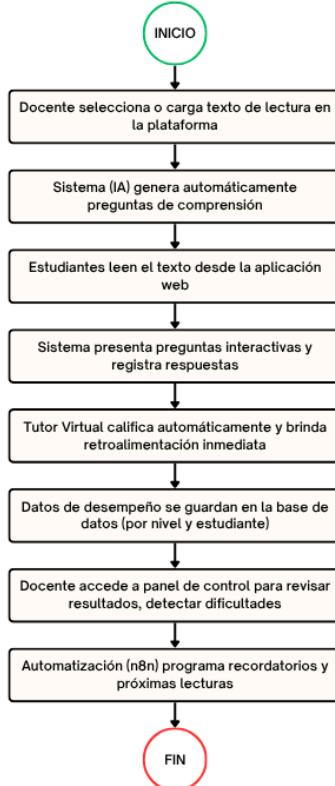


FIG.2 Procesos

#### 1.4. Oportunidad Encontrada

La Institución Educativa “San Carlos” cuenta con un entorno académico comprometido con la mejora continua del aprendizaje, lo que representa una oportunidad favorable para incorporar **herramientas tecnológicas que fortalezcan la comprensión lectora y el pensamiento crítico**.

El uso de metodologías tradicionales en las evaluaciones lectoras ha evidenciado la necesidad de **modernizar los procesos de enseñanza y evaluación**, permitiendo que los estudiantes practiquen de forma autónoma y los docentes dispongan de información precisa sobre el rendimiento de sus alumnos.

En este contexto, la implementación del **Tutor Virtual de Lectura Comprensiva Escolar** surge como una oportunidad para **integrar la inteligencia artificial y la automatización** dentro del proceso educativo, facilitando la generación de preguntas, la retroalimentación inmediata y el seguimiento del progreso lector por niveles.

De esta forma, el colegio puede **optimizar su gestión académica, reducir la carga manual de los docentes y mejorar la calidad del aprendizaje**, consolidándose

como una institución innovadora que incorpora tecnología educativa al servicio de la formación integral de sus estudiantes.

## **1.5. Detalles del Proyecto**

El proyecto Tutor Virtual de Lectura Comprensiva Escolar consiste en el desarrollo de una aplicación web educativa basada en el stack MERN (MongoDB, Express, React y Node.js), que busca fortalecer las habilidades de lectura y comprensión de los estudiantes de los niveles inicial, primaria y secundaria.

El sistema integra inteligencia artificial (IA) para generar preguntas automáticas a partir de textos y proporcionar retroalimentación inmediata, así como automatización mediante n8n para el envío de recordatorios, registro de actividades y programación de sesiones lectoras.

Además, incluye un panel de control para docentes, donde se visualiza el progreso de los estudiantes, su nivel de comprensión y los resultados de las evaluaciones por periodo (mensual, bimestral o tipo admisión).

La solución se desplegará en la nube, permitiendo acceso desde cualquier dispositivo con conexión a internet. Su diseño busca ser intuitivo, accesible y adaptable a cada nivel educativo, con el fin de promover la lectura comprensiva y reducir la carga administrativa de los docentes.

En conjunto, el proyecto contribuirá a digitalizar el proceso de evaluación lectora, mejorar la gestión académica del colegio y fomentar el uso responsable e innovador de la tecnología en el aula.

## CAPÍTULO 2 ESTUDIO DE FACTIBILIDAD

### 2.1. Alternativas de Solución

Tras analizar la problemática identificada en la Institución Educativa “San Carlos”, se evaluaron distintas alternativas orientadas a mejorar el proceso de evaluación lectora y el desarrollo de habilidades de comprensión en los estudiantes:

- **Alternativa 1: Mantener el proceso actual de evaluación manual.**

Consiste en continuar utilizando exámenes impresos o formularios digitales básicos elaborados por los docentes. Aunque no requiere inversión tecnológica, mantiene las mismas limitaciones: alta carga de trabajo docente, ausencia de retroalimentación inmediata y escaso seguimiento individual del progreso lector.

- **Alternativa 2: Implementar plataformas externas de lectura.**

Implica adoptar herramientas o sitios web ya existentes para practicar comprensión lectora. Sin embargo, la mayoría de estas plataformas están orientadas a contextos generales o internacionales, carecen de adaptación curricular al entorno nacional y no ofrecen control directo a los docentes del colegio.

- **Alternativa 3: Desarrollar el Tutor Virtual de Lectura Comprensiva Escolar.**

Propone crear una plataforma propia e integrada al modelo pedagógico institucional. El sistema automatiza la generación de preguntas mediante inteligencia artificial, evalúa las respuestas en tiempo real y proporciona a los docentes un panel de seguimiento del progreso lector. Además, incorpora automatización (n8n) para programar recordatorios, registrar resultados y facilitar la gestión académica.

Entre las alternativas analizadas, esta última resulta la más viable e innovadora, ya que digitaliza el proceso evaluativo, optimiza el tiempo docente y fortalece las competencias lectoras de los estudiantes en todos los niveles educativos.

### 2.2. Factibilidad Técnica

El proyecto **Tutor Virtual de Lectura Comprensiva Escolar** es técnicamente viable, dado que utiliza tecnologías probadas y de libre acceso, lo que reduce significativamente los costos de desarrollo y mantenimiento.

El sistema se desarrollará bajo la arquitectura **MERN (MongoDB, Express, React y Node.js)**, con integración de **inteligencia artificial** mediante API NLP (Hugging Face u Ollama) y **automatización** con **n8n**, todas herramientas disponibles en versiones gratuitas o de código abierto.

Asimismo, el proyecto se apoyará en plataformas de despliegue **Vercel** y **MongoDB Atlas Free Tier**, lo que garantiza accesibilidad sin requerir inversión en infraestructura. Las pruebas y control de calidad se gestionan con **Jest** y **Cypress**, y la integración continua se automatizará mediante **GitHub Actions** y **Docker**.

El equipo cuenta con los conocimientos necesarios en desarrollo web full-stack, control de versiones y metodologías ágiles, por lo que la factibilidad técnica del proyecto es **alta**.

### 2.2.1. Recursos tecnológicos

- **Frontend:** React.js con diseño responsive y componentes reutilizables.
- **Backend:** Node.js con Express, autenticación JWT y conexión a MongoDB Atlas.
- **Base de datos:** MongoDB (Free Tier en la nube).
- **IA:** Modelos NLP (Hugging Face / Ollama) para generación de preguntas.
- **Automatización:** n8n self-hosted para notificaciones y registro automático.
- **Integración y despliegue:** GitHub, Docker y Vercel.

### 2.2.2. Requerimientos del entorno

- **Hardware mínimo de desarrollo:** Procesador Intel i5 o superior, 8 GB RAM, 512 GB SSD.
- **Software base:** Node.js 20+, MongoDB Compass, VS Code, Docker Desktop.
- **Recursos en la nube:** Vercel y MongoDB Atlas (planes gratuitos).
- **Compatibilidad:** Navegadores modernos (Chrome, Edge, Firefox).

El entorno propuesto asegura una ejecución estable, escalable y sin costo operativo significativo.

## 2.3. Factibilidad Económica

El proyecto es económicamente viable, ya que aprovecha herramientas gratuitas y la participación de un equipo académico multidisciplinario.

Según el presupuesto consolidado (informe técnico y hoja de cálculo), el **costo total del proyecto** asciende a **4 586 USD**, distribuidos en recursos humanos, desarrollo, pruebas y documentación.

### **2.3.1. Gastos Generales**

<b>Fase</b>	<b>Costo estimado (USD)</b>
Inicio	122
Planificación	308
Planificación 2	265
Desarrollo	2 776
Pruebas y entrega	1 115
<b>Total</b>	<b>4 586 USD</b>

Tabla 1 Gastos generales

Los principales gastos se asocian al tiempo de dedicación del equipo (horas de programación, diseño y pruebas), sin requerir inversiones en infraestructura o licencias.

El uso de tecnologías open source y servicios en la nube gratuitos permite mantener la sostenibilidad del sistema en el mediano plazo.

Por tanto, la factibilidad económica se califica como alta, con retorno académico y funcional inmediato.

## **2.4. Factibilidad Operacional**

La solución propuesta es operativamente viable, ya que se integra de forma natural a la dinámica pedagógica del colegio. Los docentes podrán utilizar la plataforma para asignar lecturas y realizar evaluaciones automáticas, mientras que los estudiantes accederán a las actividades desde cualquier dispositivo con conexión a internet.

El sistema está diseñado con una **interfaz intuitiva y adaptable por nivel educativo**, lo que garantiza su aceptación tanto por estudiantes de primaria y secundaria como por el personal docente. Además, los flujos automatizados reducen la carga administrativa y facilitan la evaluación continua.

### **2.4.1. Sistema de Tutor de lectura Crítica**

El sistema permitirá:

- Generar y presentar **preguntas automáticas** sobre los textos leídos.

- Brindar **retroalimentación inmediata** según las respuestas del estudiante.
- Registrar resultados y progreso en la **base de datos central**.
- Ofrecer un **panel docente** para monitorear comprensión y rendimiento.
- Automatizar notificaciones y recordatorios mediante **n8n**.

Gracias a su diseño modular y escalable, el sistema puede implementarse progresivamente en todos los niveles del colegio, consolidando una **gestión educativa moderna, eficiente y digitalizada**.

## CAPÍTULO 3 ANÁLISIS DE REQUERIMIENTOS

### 3.1. Metas del Sistema de Información

El proyecto Tutor Virtual de Lectura Comprensiva Escolar tiene como meta principal fortalecer las habilidades de comprensión lectora y pensamiento crítico de los estudiantes de la Institución Educativa “San Carlos”, mediante una plataforma web con inteligencia artificial que automatice el proceso de evaluación y retroalimentación.

### 3.2. Requisitos del Sistema

El sistema se desarrolla con la arquitectura MERN (MongoDB, Express, React y Node.js) y servicios complementarios como n8n para automatización y modelos de IA para análisis de texto.

Estas herramientas permiten un sistema escalable, seguro y compatible con los recursos del colegio sin requerir grandes inversiones.

#### 3.2.1. Requerimientos Funcionales

Código	Requerimiento funcional	Descripción
RF01	Gestión de usuarios	Permitir el registro e inicio de sesión de alumnos y docentes
RF02	Asignación de lecturas	El docente podrá subir o seleccionar textos según grado y nivel
RF03	Generación automática de preguntas	El sistema, mediante IA, generará preguntas de

		comprensión lectora adaptadas al texto.
RF04	Resolución de actividades	El alumno podrá responder las preguntas y enviar sus resultados.
RF05	Retroalimentación inmediata	El sistema mostrará comentarios automáticos sobre aciertos y errores.
RF06	Registro de resultados	Guardar el puntaje y desempeño de cada estudiante.
RF07	Reportes de desempeño	Generar gráficos e informes por alumno, aula o nivel educativo.
RF08	Gestión de contenidos	El coordinador podrá modificar o eliminar textos y usuarios.
RF09	Comunicación docente–alumno	Habilitar mensajes automáticos de recordatorio o aviso de evaluación.

Tabla 2 Requerimientos funcionales

### 3.2.2. Requerimientos no Funcionales

Código	Requerimiento no funcional	Descripción
RNF01	Accesibilidad	Interfaz intuitiva, adaptable a dispositivos móviles.
RNF02	Rendimiento	El sistema responderá a cada solicitud en menos de 3 segundos.
RNF03	Seguridad	Autenticación JWT y contraseñas cifradas con bcrypt.
RNF04	Escalabilidad	Permitir la incorporación de nuevos grados y usuarios sin reconfiguración.
RNF05	Disponibilidad	Acceso en línea 24/7 mediante Vercel y MongoDB Atlas Free Tier.
RNF06	Usabilidad	Diseñado con componentes claros y navegación guiada.

RNF07	Mantenibilidad	Código modular y documentado bajo estándares MERN.
-------	----------------	--

Tabla 3 Requerimientos no funcionales

### 3.3. Identificación de Actores del Sistema

El usuario cuenta con tres actores principales:

Actor	Descripción	Funciones Principales
Coordinador Académico	Encargado de la gestión global del sistema.	Administra usuarios (docentes/alumnos), supervisa reportes y aprueba contenidos
Docente	Responsable de asignar lecturas y evaluar resultados.	Asigna textos, revisa reportes, orienta a los estudiantes y analiza el avance.
Alumno	Usuario principal del sistema. Estudiantes de primaria o secundaria.	Inicia sesión, realiza lecturas, responde preguntas, revisa retroalimentación y consulta su progreso.

Tabla 4 Actores del sistema

#### 3.3.1. Relación con los procesos Educativos

El sistema se integra con los procesos de evaluación mensual y bimestral que realiza la institución, automatizando la parte de comprensión lectora y generando resultados comparativos por grupo y periodo.

## CAPÍTULO 4 PLANIFICACIÓN DEL PROYECTO

### 4.1. Definición de Roles de Trabajo

El proyecto “Tutor Virtual de Lectura Comprensiva Escolar - I.E.P. San Carlos” se gestiona en Jira con enfoque Scrum. Los roles que se consignan a continuación provienen de las actas y documentos del equipo.

#### 4.1.1. Product Owner

**Nombre:** Nilton Joel Lazo Maraví

**Responsabilidades:** Proteger la visión del producto; priorizar y mantener el Product Backlog en Jira, validar criterios de aceptación, aceptar o rechazar incrementos, coordinar con las partes interesadas académicas y velar por el valor entregado en cada sprint.

#### 4.1.2. Scrum master

**Nombre:** Rol compartido por el equipo

**Responsabilidades:** Facilitar ceremonias, proteger el time-box, remover impedimentos, promover la mejora continua del proceso, limitar reuniones a ≤60 minutos y dar a conocer los criterios de aceptación en el refinamiento.

#### 4.1.3. Team member

**Nombres:** Huaccho Mancilla Steven José, Robles Sánchez Britney Sheyla, Poma Goche Abigail Karim, Rojas Mellado Andrea Mirella

**Responsabilidades:** análisis y diseño, desarrollo MERN (React + Vite + Tailwind, Express, MongoDB) e integración de Ollama y n8n, pruebas unitarias, de integración y E2E, documentación técnica y demos de sprint.

#### 4.1.4. Tester

**Nombre:** Rol compartido por el equipo

**Responsabilidades:** Definir y mantener DoR/DoD, diseñar y ejecutar pruebas, resguardar evidencia en Jira, medir cobertura, apoyar la automatización (CI/CD con Jest) y el aseguramiento de calidad.

**Patrocinio y partes interesadas:** Ramírez Basualdo Lenin Sebastian (patrocinador); docente evaluador del curso (interesado académico).

## 4.2. Product Backlog

**Criterios de priorización:** Valor educativo (impacto en estudiantes y docentes), factibilidad técnica (modelo local con Ollama y automatizaciones en n8n) y necesidad de analítica para seguimiento institucional.

Historias de usuario y estado actual:

- **HU01 – Recibir preguntas (8 pts, Finalizada):** El sistema genera  $\geq 5$  preguntas por texto; si falla el procesamiento, muestra error sin resultados vacíos.
- **HU02 – Recibir retroalimentación (5 pts, Finalizada):** Tras responder, muestra correcto/incorrecto con breve explicación; si no hay respuestas, advierte sin bloquear.
- **HU05 – Acceder a reportes (5 pts, Finalizada):** Reporte por estudiante con % de comprensión y actividad; si no hay datos, lo informa.
- **HU06 – Obtener informe (5 pts, Finalizada):** Dashboard con promedios de curso; si no hay estudiantes, muestra “No hay información disponible”.
- **HU03 – Progreso acumulado (3 pts, Finalizada):** Visualización del progreso del estudiante; si no hay registros, “Sin datos disponibles”.
- **HU07 – Indicar sesgos del texto (13 pts, Finalizada):** Detección y etiquetas de sesgos/falacias; si no se puede analizar, notifica sin resultados vacíos.
- **HU04 – Asignar actividades (8 pts, Finalizada):** El docente sube texto y lo asigna a estudiantes; si el formato no es soportado, se notifica y no se asigna.
- **HU08 – Enviar recordatorios de lectura (5 pts, En curso):** n8n agenda y envía recordatorios; ante fallo de envío, registra y reintenta.
- **HU09 – Mostrar panel de comprensión (3 pts, En curso):** KPIs agregados (promedio general); si no hay datos suficientes, se indica.

## 4.3. Sprint Backlog

### 4.3.1. Sprint 1

**Objetivo:** MVP navegable con generación de preguntas, retroalimentación y visualización mínima de desempeño.

**Resultado:** HU01 (8), HU02 (5), HU05 (5), HU06 (5): 23 puntos entregados.

### 4.3.2. Sprint 2

**Objetivo de producto:** Analítica personal (HU03) y análisis de sesgos con IA (HU07).

**Resultado:** HU03 (3) y HU07 (13): 16 puntos entregados.

#### 4.3.3. Sprint 3

**Objetivo de producto:** Operación docente (HU04), hábitos de lectura con n8n (HU08) y panel global (HU09).

**Carga planificada:** 16 puntos.

**Hito de calendario:** Martes 2 de diciembre del 2025.

### 4.4. Planificación de Sprints

#### 4.4.1. Historias de usuario

Todas las HU están documentadas en Jira con descripción y criterios de aceptación verificables. La DoR exige historia comprensible, dependencias explícitas y criterios de prueba; la DoD exige pruebas ejecutadas, evidencia en Jira y demo en la revisión de sprint.

#### 4.4.2. Priorización de historias de usuario

Orden establecido: (1) aprendizaje esencial del estudiante y retroalimentación (HU01, HU02), (2) visibilidad docente mínima (HU05, HU06), (3) analítica personal y sesgos (HU03, HU07), (4) flujo operativo docente y hábitos (HU04, HU08), (5) analítica institucional (HU09). Este orden maximiza valor temprano y reduce riesgo técnico.

### 4.5. Cronograma de Actividades

Fase / Sprint	Fechas	Entregables principales
Sprint 1	10/sept - 08/oct	HU01, HU02, HU05, HU06 (MVP validado)
Sprint 2	09/oct - 04/nov	HU03, HU07 (analítica personal + IA de sesgos)
Sprint 3	05/nov - 02/dic	HU04, HU08, HU09 (operación docente, hábitos con n8n y panel global)
Cierre	03/dic	Demo final, documentación, empaquetado Docker, checklist de calidad

Tabla 5 Cronograma de actividades

#### 4.6. Gestión de riesgos

ID	Riesgo	Probabilidad	Impacto	Respuesta / Mitigación	Responsable
R1	Latencia/precisión del modelo (Ollama) afecta experiencia (HU01/HU02/HU07).	Media	Alta	Spike técnico de latencia; profiling; caching; control de tokens; métricas p95; mensajes de estado claros.	Equipo (dev)
R2	Cobertura de pruebas insuficiente retrasa cierres DoD.	Media	Alta	CI/CD con Jest en cada PR; umbral objetivo $\geq 80\%$ ; casos de prueba alineados a criterios de aceptación.	Equipo (QA compartido)
R3	Datos insuficientes para reportes/panel (HU05–HU09).	Media	Media	Estados “sin datos”; datos semilla para pruebas; validación previa a demo.	Equipo (dev)
R4	Reuniones extensas reducen capacidad efectiva del sprint.	Media	Media	Time-box $\leq 60$ min; agenda y objetivos por ceremonia; seguimiento del cumplimiento.	Rol SM (compartido)
R5	Integración y conectividad de n8n (HU08) genera fallas de envío.	Media	Media	Reintentos con backoff; logging y trazabilidad; cola de reenvíos; notificación de error al usuario.	Equipo (dev)
R6	Limitaciones de free-tiers (Atlas, n8n, etc.) producen cuellos de botella.	Baja	Media	Plan B local; monitoreo de cuotas; tareas batch fuera de horas pico.	PO + equipo

Tabla 6 Gestión de riesgos

## 4.7. Aseguramiento de la calidad

### 4.7.1. Heurísticas de Nielsen (10 ítems, Likert 1–5)

1. El sistema informa claramente al usuario sobre las acciones en curso o resultados.
2. Los términos usados son comprensibles para el usuario promedio.
3. El sistema permite deshacer o cancelar acciones fácilmente.
4. Colores, tipografías y estilos son coherentes.
5. El sistema evita errores mediante validaciones previas.
6. Se muestran alertas antes de ejecutar acciones críticas.
7. El sistema ofrece menús que evitan depender de la memoria del usuario.
8. Existen atajos o funciones rápidas para usuarios expertos.
9. La interfaz evita elementos innecesarios que distraen.
10. Los mensajes de error indican la causa y la posible solución.

**Aplicación y reporte:** Al cierre del Sprint 3 (post-demo) con estudiantes y docentes, calcular media por ítem, promedio global y desviación estándar, presentar 3 hallazgos clave con acciones de mejora asociadas.

### 4.7.2. SUS – System Usability Scale (10 ítems, Likert 1–5)

1. Creo que me gustaría usar este sistema con frecuencia.
2. Encontré el sistema innecesariamente complejo.
3. Considero que el sistema fue fácil de usar.
4. Pienso que necesitaría apoyo técnico para usar el sistema.
5. Las funciones del sistema estaban bien integradas.
6. Me parece que hubo demasiada inconsistencia en el sistema.
7. Imagino que la mayoría de las personas aprenderían a usarlo rápidamente.
8. Encontré el sistema muy engorroso de usar.
9. Me sentí confiado al usar el sistema.
10. Necesité aprender muchas cosas antes de poder usar el sistema.

**Cálculo del puntaje SUS:** recodificar ítems positivos (1,3,5,7,9) como (respuesta – 1) e ítems negativos (2,4,6,8,10) como (5 – respuesta); sumar y multiplicar × 2.5 → SUS (0–100). Interpretación: ≥68 aceptable, ≥80 excelente.

# CAPÍTULO 5 DISEÑO DEL SISTEMA DE INFORMACIÓN

## 5.1. Diseño de Diagramas UML

El diseño UML describe gráficamente la estructura y el comportamiento del **Tutor Virtual de Lectura Comprensiva Escolar**, permitiendo visualizar las interacciones entre alumnos, docentes y el sistema.

El modelo se elaboró considerando los requerimientos funcionales definidos y las principales operaciones: **asignar lecturas, generar preguntas automáticas, responder actividades, registrar resultados y brindar retroalimentación**.

### 5.1.1. Diagramas de casos de uso

El Tutor Virtual de Lectura Comprensiva Escolar permite a los alumnos realizar lecturas interactivas y responder preguntas automáticas, a los docentes asignar lecturas y monitorear resultados, y al administrador gestionar usuarios y contenidos.

#### Actores principales:

- **Docente:** Asigna lecturas, revisa resultados y genera reportes.
- **Alumno:** Lee textos, responde preguntas y recibe retroalimentación.
- **Coordinador:** Supervisa cuentas, reportes y controla el correcto funcionamiento del sistema.
- **Sistema (IA / n8n):** Genera preguntas automáticas, evalúa respuestas y envía recordatorios.

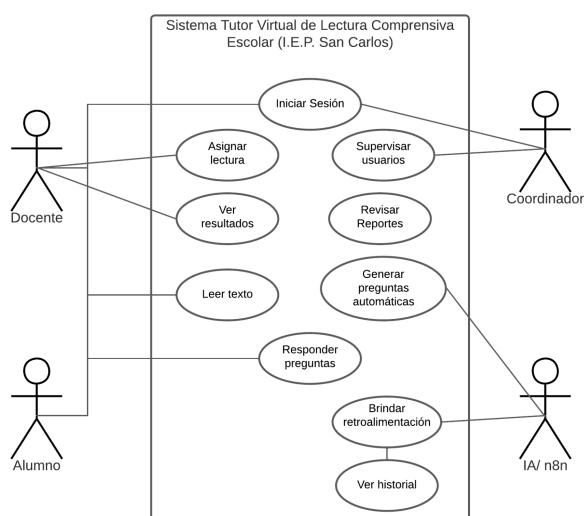


FIG.3 Casos de uso

### 5.1.2. Diagrama de secuencia

El diagrama de secuencia muestra cómo interactúan los actores (Alumno, Docente, Coordinador y el Sistema con IA/n8n) en el flujo principal de uso: evaluar la comprensión lectora.

Este proceso comienza cuando el docente asigna una lectura y culmina con la generación automática de resultados y retroalimentación para el alumno.

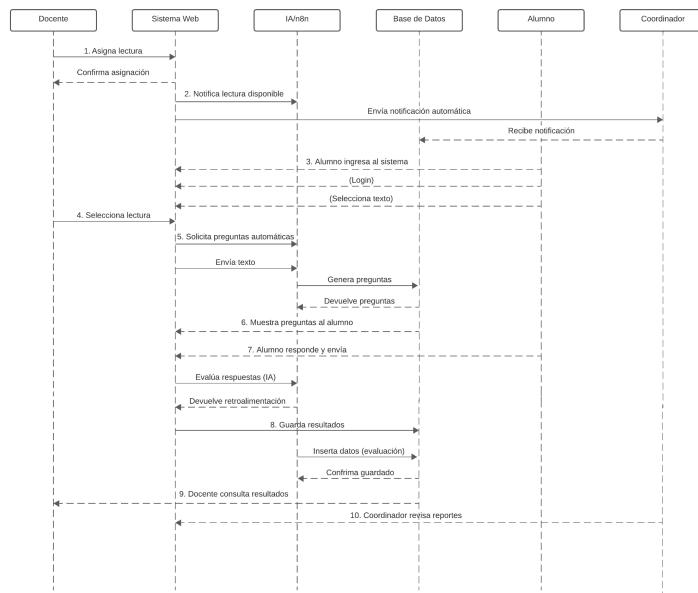


FIG.4 Diagrama de secuencia

### 5.1.3. Diagramas de colaboración

El diagrama de colaboración muestra cómo interactúan los actores y objetos del sistema Tutor Virtual de Lectura Comprensiva Escolar. Representa el intercambio de mensajes entre el alumno, el módulo de IA, la base de datos y el panel docente durante el proceso de evaluación. Resume la cooperación de los componentes para generar preguntas, registrar respuestas y actualizar resultados en tiempo real.

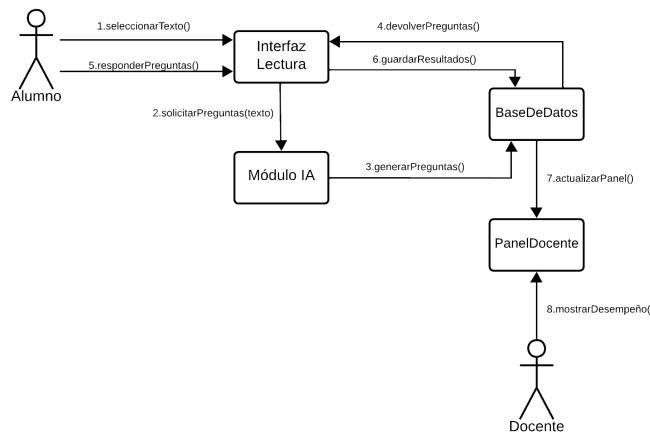


FIG.5 Diagrama de colaboración

#### 5.1.4. Diagramas de clases

El siguiente diagrama de clases presenta la estructura del sistema Tutor Virtual de Lectura Comprensiva Escolar a partir de las colecciones principales de MongoDB. Incluye las clases con sus atributos, métodos y relaciones (herencia, agregación y composición) que conectan los módulos de usuarios, actividades, preguntas, respuestas y entregas. Este modelo orientado a objetos facilita la organización, coherencia y trazabilidad de la información del sistema.

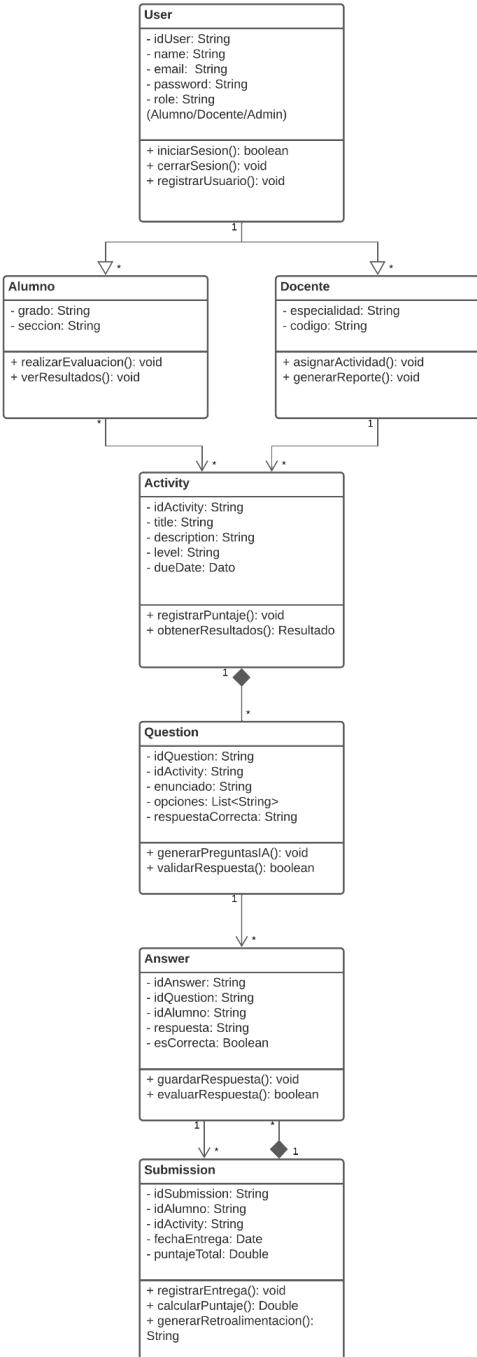


FIG.6 Diagrama de clases

## 5.2. Diseño de base de datos

### 5.2.1. Diseño conceptual (E/R)

El siguiente diagrama Entidad–Relación describe el modelo conceptual del sistema Tutor Virtual de Lectura Comprensiva Escolar. Representa las entidades

principales y sus relaciones, mostrando cómo alumnos y docentes interactúan con actividades, preguntas, respuestas y entregas. Este diseño asegura organización estructural, coherencia funcional e integridad referencial en los datos gestionados dentro de la base de datos MongoDB.

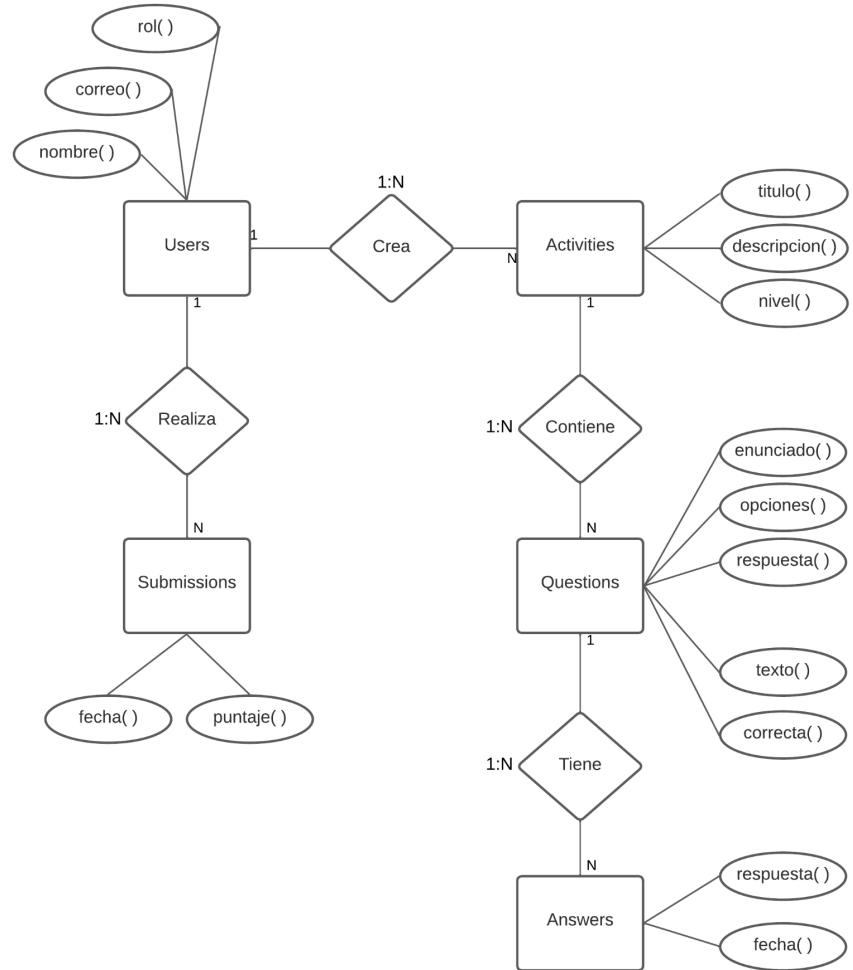


FIG.7 Diseño conceptual

### 5.2.2. Diseño lógico

El diseño lógico se deriva del modelo conceptual, transformando las entidades y relaciones del diagrama E/R en estructuras tabulares normalizadas.

En este nivel se definen las tablas, campos, tipos de datos, claves primarias (PK) y claves foráneas (FK) que se implementarán en la base de datos MongoDB, conservando las reglas de integridad referencial y optimizando la

organización de los datos del sistema Tutor Virtual de Lectura Comprensiva Escolar.

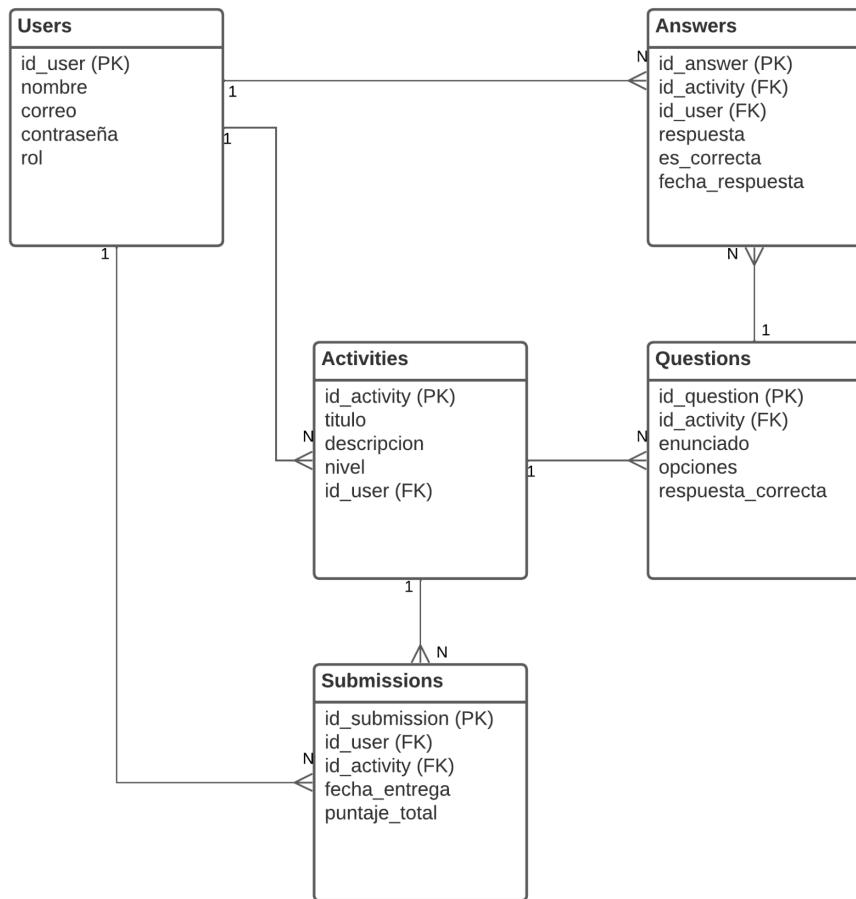


FIG.8 Diseño lógico

### 5.2.3. Diseño físico

El diseño físico muestra la estructura final de la base de datos del sistema Tutor Virtual de Lectura Comprensiva Escolar, donde se definen las tablas, campos, tipos de datos y relaciones entre entidades. Este modelo garantiza la integridad y eficiencia en el almacenamiento de información sobre usuarios, actividades, preguntas, respuestas y evaluaciones, optimizando el rendimiento y la coherencia del sistema.

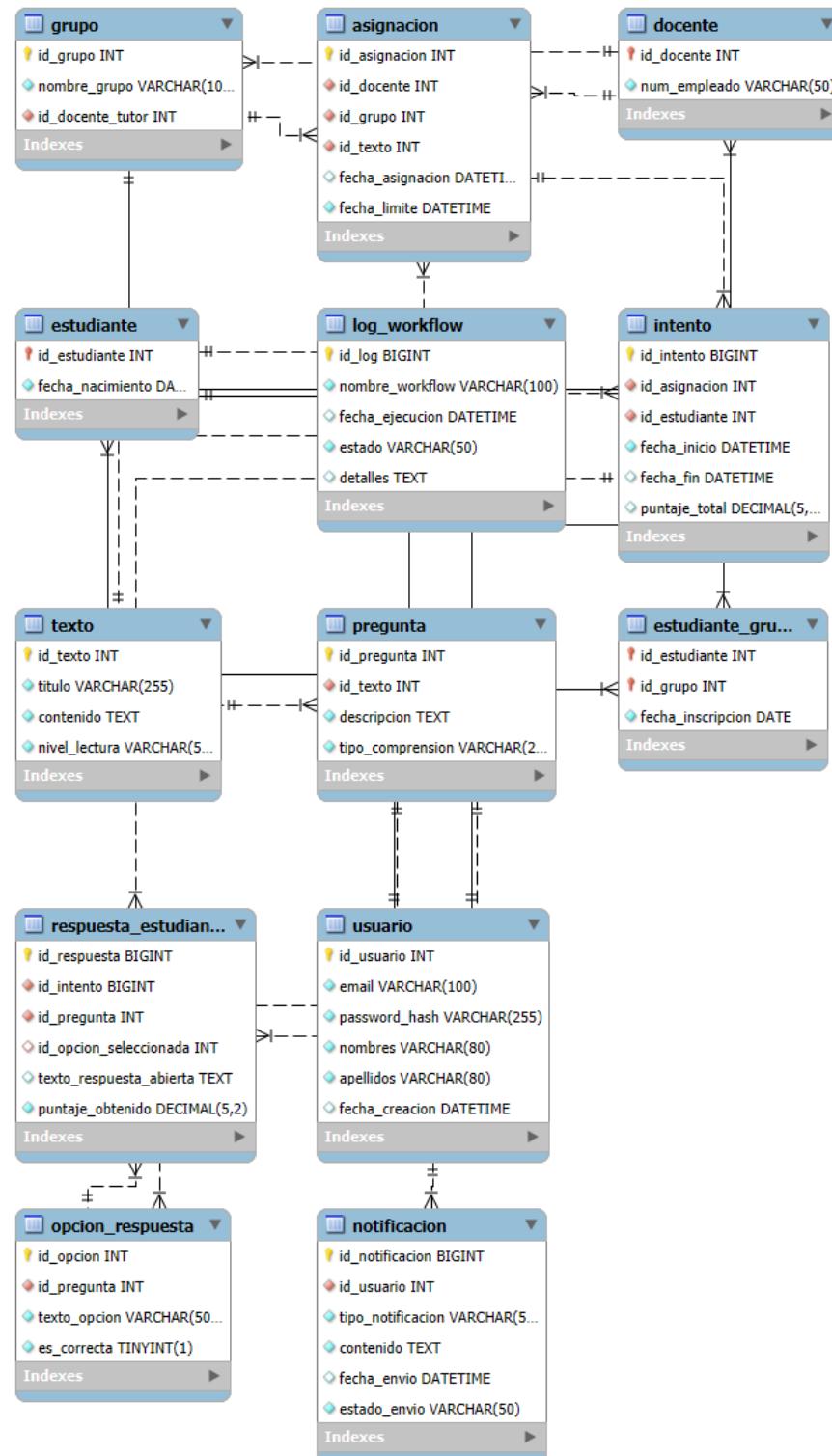


FIG.9 Diseño físico

#### 5.2.4. Modelado de base de datos

El modelado de la base de datos representa la estructura final implementada en MySQL (Diseño Físico), asegurando la integridad, consistencia y trazabilidad de

la información del sistema "Tutor Virtual de Lectura Comprensiva Escolar - I.E.P. San Carlos".

## 1. Diagrama Entidad-Relación Lógico-Físico (DER)

Este diagrama es el resultado de la función Reverse Engineer realizada sobre el esquema tutor\_virtual\_iepsancarlos en MySQL. Muestra las entidades (tablas), sus atributos clave (PK/FK) y la cardinalidad de las relaciones implementadas.

## 2. Diccionario de Datos

El diccionario de datos es la documentación técnica que describe cada tabla y cada campo de la base de datos, detallando el tipo de dato físico utilizado y sus restricciones.

### a. Mòdulo de usuarios y grupos

Tabla: USUARIO	Propósito: Almacena datos generales de acceso (padre de Docente y Estudiante)
Campo	Tipo de Dato (MySQL)
id_usuario	INT
email	VARCHAR(100)
password_hash	VARCHAR(255)
nombres	VARCHAR(80)
apellidos	VARCHAR(80)
fecha_creacion	DATETIME

Tabla: DOCENTE	Propósito: Datos específicos del profesor
Campo	Tipo de Dato (MySQL)
id_docente	INT
num_empleado	VARCHAR(50)

Tabla: ESTUDIANTE	Propósito: Datos específicos del alumno
Campo	Tipo de Dato (MySQL)
id_estudiante	INT
fecha_nacimiento	DATE

<b>Tabla: GRUPO</b>	<b>Propósito:</b> Representa los salones o aulas
<b>Campo</b>	<b>Tipo de Dato (MySQL)</b>
id_grupo	INT
nombre_grupo	VARCHAR(100)
id_docente_tutor	INT

<b>Tabla: ESTUDIANTE_GRUPO</b>	<b>Propósito:</b> Relaciona estudiantes con grupos (N:M)
<b>Campo</b>	<b>Tipo de Dato (MySQL)</b>
id_estudiante	INT
id_grupo	INT
fecha_inscripcion	DATE

b. Módulo de Contenido y Evaluación

<b>Tabla: TEXTO</b>	<b>Propósito:</b> Almacena los materiales de lectura
<b>Campo</b>	<b>Tipo de Dato (MySQL)</b>
id_texto	INT
titulo	VARCHAR(255)
contenido	TEXT
nivel_lectura	VARCHAR(50)

<b>Tabla: PREGUNTA</b>	<b>Propósito:</b> Define las preguntas de evaluación
<b>Campo</b>	<b>Tipo de Dato (MySQL)</b>
id_pregunta	INT
id_texto	INT
descripcion	TEXT
tipo_comprehension	VARCHAR(20)

<b>Tabla: OPCION_RESPUESTA</b>	<b>Propósito:</b> Opciones para preguntas de selección múltiple
<b>Campo</b>	<b>Tipo de Dato (MySQL)</b>
id_opcion	INT

id_pregunta	INT
texto_opcion	VARCHAR(500)
es_correcta	BOOLEAN

c. Módulo de Asignaciones y Resultados

Tabla: ASIGNACION	Propósito: Define una tarea específica de lectura (Docente + Grupo + Texto)
Campo	Tipo de Dato (MySQL)
id_asignacion	INT
id_docente	INT
id_grupo	INT
id_texto	INT
fecha_asignacion	DATETIME
fecha_limite	DATETIME

Tabla: INTENTO	Propósito: Sesión de evaluación realizada por un estudiante
Campo	Tipo de Dato (MySQL)
id_intento	BIGINT
id_asignacion	INT
id_estudiante	INT
fecha_inicio	DATETIME
fecha_fin	DATETIME
puntaje_total	DECIMAL(5, 2)

Tabla: RESPUESTA_ESTUDIANTE	Propósito: Detalle de la respuesta a cada pregunta en un intento
Campo	Tipo de Dato (MySQL)
id_respuesta	BIGINT
id_intento	BIGINT
id_pregunta	INT
id_opcion_seleccionada	INT
texto_respuesta_abierta	TEXT
puntaje_obtenido	DECIMAL(5, 2)

d. Módulo de Trazabilidad y Logs

<b>Tabla: LOG_WORKFLOW</b>	<b>Propósito: Registro de la ejecución de flujos automatizados (n8n)</b>
<b>Campo</b>	<b>Tipo de Dato (MySQL)</b>
id_log	BIGINT
nombre_workflow	VARCHAR(100)
fecha_ejecucion	DATETIME
estado	VARCHAR(50)
detalles	TEXT

<b>Tabla: NOTIFICACION</b>	<b>Propósito: Trazabilidad de correos electrónicos y avisos enviados</b>
<b>Campo</b>	<b>Tipo de Dato (MySQL)</b>
id_notificacion	BIGINT
id_usuario	INT
tipo_notificacion	VARCHAR(50)
contenido	TEXT
fecha_envio	DATETIME

Tabla 7 Diccionario de datos

## 5.3. Diseño de Interfaces Básicas

### 5.3.1. Pantalla página de inicio



FIG.10 Página de inicio

### 5.3.2. Pantalla de generación de preguntas y retroalimentación

The screenshot shows a web application interface for generating questions and receiving feedback. At the top, there's a header with the logo of 'Universidad Continental' and the text 'Tutor Virtual de Lectura Crítica'. Below the header, the navigation menu includes 'Inicio', 'Preguntas', 'Reportes', 'Acceder', and 'Registrarse'. The main content area has a title 'Generador de preguntas y retroalimentación' with a subtitle 'Pega un fragmento del texto, obtén preguntas automáticas, identifica sesgos argumentativos y recibe retroalimentación inmediata sobre tus respuestas.' A blue button labeled 'Consejo rápido' with the text 'Usa entre 2 y 5 párrafos y evita listas de viñetas para obtener mejores resultados de IA.' is visible. The interface is divided into three main sections: 1) 'Ingresar texto a analizar' (Copy a fragment of the text. You can edit it before generating questions.), 2) 'Responde las preguntas generadas' (Answer the generated questions and receive immediate feedback for each one.), and 3) 'Sesgos detectados en el texto' (Generate questions to analyze the text and detect possible argumentative biases). Each section contains detailed instructions and examples.

FIG.11 Generación de preguntas

### 5.3.3. Pantalla de registro - nuevo usuario

The screenshot shows the 'Crear cuenta' (Create account) form. The title 'Crear cuenta' is at the top. Below it are five input fields: 'Nombres', 'Apellidos', 'Centro de estudios (opcional)', 'Correo', and 'Contraseña'. There is also a dropdown menu 'Soy estudiante'. A large blue 'Registrarme' (Register) button is centered below the inputs. At the bottom left, there's a link '¿Ya tienes cuenta? Inicia sesión' (Already have an account? Log in). The footer contains links to 'Tutor Virtual', 'Reportes', 'Generar preguntas', and social media icons for YouTube, LinkedIn, and X (Twitter). The footer also includes copyright information: '© 2025 Tutor Virtual de Lectura Crítica' and 'Hecho con React + TypeScript + Tailwind + Node'.

FIG.12 Nuevo usuario

### 5.3.4. Pantalla de logueo / Inicio de sesión

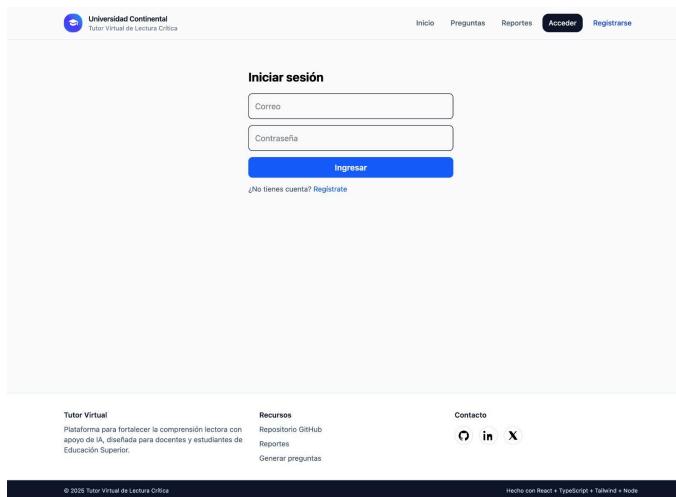


FIG.13 Inicio de sesión

### 5.3.5. Pantalla de reporte de respuestas

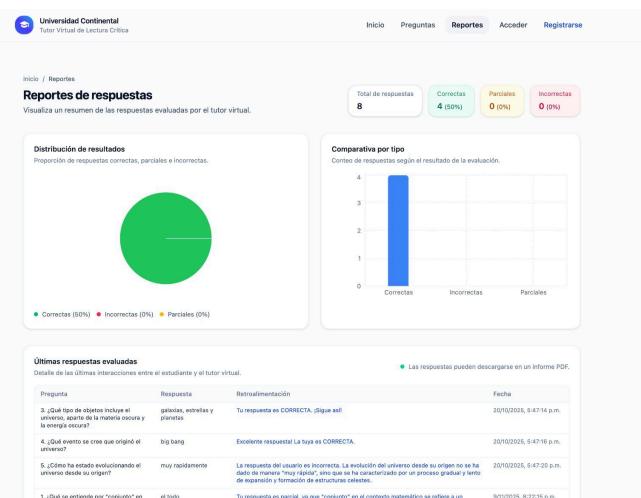


FIG.14 Reporte de respuestas

### 5.3.6. Pantalla de bloqueo al no iniciar sesión

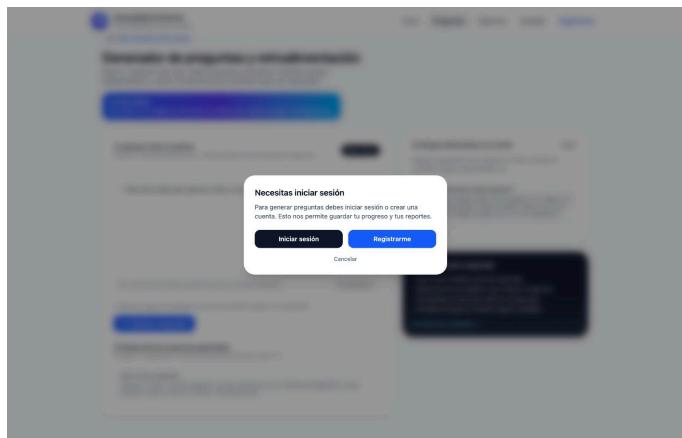


FIG.15 Pantalla de bloqueo

### 5.3.7. Pantalla de actividades del estudiante

A screenshot of a student's activity dashboard. At the top, there's a header with the university logo, the course name "Tarea Virtual de Lectura Crítica", and navigation links like "Inicio", "Preguntas", "Reportes", "Mis actividades", and "Salir". A user profile icon indicates "Habla, lee, evalúa" and shows "Estudiante". Below the header, a section titled "Mis actividades" says "Revise las lecturas asignadas, su fecha límite y tu avance." It shows a progress bar for "Progreso promedio" at 10%, with 2 activities completed and 19 in progress. A table lists 21 assigned activities with columns for "Actividad", "Vence", "Progreso", and "Estado". Most activities have a due date of "18/11/2025" and are marked as "En progreso". One activity is listed as "Entregada".

FIG.16 Actividades del estudiante

### 5.3.8. Pantalla de actividad asignada al estudiante

The screenshot shows a web-based learning platform interface. At the top, there is a navigation bar with links for 'Inicio', 'Preguntas', 'Reportes', 'Mis actividades', and a user profile 'Hola, Joel Estudiante'. Below the navigation bar, the page title is 'Inicio / Mis actividades / Detalle'. The main content area is titled 'Energía' and has a status indicator 'En progreso' (In progress) with a progress bar at 0%. The text of the reading is as follows:

**Texto de la lectura**

Srinivasa Ramanujan fue un matemático indio autodidacta que hizo contribuciones revolucionarias al análisis matemático, la teoría de los números y las series infinitas, a pesar de su corta vida y la falta de formación universitaria formal. Su genio fue reconocido por matemáticos como G.H. Hardy de Cambridge, quien lo invitó a Inglaterra y colaboró con él. Ramanujan creía que sus descubrimientos matemáticos eran inspirados por la diosa hindú Namagiri.

Below the text, there is a section for 'Tu respuesta / notas' with a note: 'Puedes escribir ideas, respuestas o un borrador antes de enviar definitivamente la actividad.' A text input field contains the placeholder 'Escribe aquí tu respuesta o notas...'. At the bottom of the activity card, there are two buttons: 'Guardar borrador' and 'Enviar actividad'.

FIG.17 Actividad asignada al estudiante

### 5.3.9. Pantalla Panel Docente

The screenshot shows the teacher dashboard with a header 'Hola, Luis — Panel docente' and a note 'Gestiona actividades, revisa avances y asigna nuevas lecturas.' There is a green circular icon indicating 'Datos actualizados en tiempo real' (Real-time updated data).

The dashboard features three main statistics boxes:

- ACTIVIDADES**: 23 (Promedio de 0.1 estudiantes por lectura)
- ESTUDIANTES**: 2 (2 entregas recibidas)
- PROGRESO PROMEDIO**: 14% (Bajo)

Below these stats, there are two buttons: '+ Asignar nueva actividad' and 'Ver reportes'.

The main content area is titled 'Actividades asignadas' and includes a search bar and a table of assigned activities:

ACTIVIDAD	ASIGNADOS	ENTREGAS	PROGRESO	ESTADO	CREADA
Actividad E2E Cypress 1763946304672 ID: 6923ae...	1	1	<div style="width: 100%;">100%</div>	Alto	23/11/2025
Actividad E2E Cypress 1763946240175 ID: 6923ae...	1	0	<div style="width: 0%;">0%</div>	Bajo	23/11/2025
Actividad E2E Cypress 1763946195992 ID: 6923ae...	1	0	<div style="width: 0%;">0%</div>	Bajo	23/11/2025
Actividad E2E Cypress 1763946174564 ID: 6923ae...	1	0	<div style="width: 0%;">0%</div>	Bajo	23/11/2025
Actividad E2E Cypress 1763946098154 ID: 6923ae...	1	0	<div style="width: 0%;">0%</div>	Bajo	23/11/2025

FIG.18 Pantalla panel docente

### 5.3.10. Pantalla de asignación de actividad del estudiante por parte del docente

The screenshot shows the 'Asignar actividad' (Assign activity) page. At the top, there's a header with the university logo, navigation links (Inicio, Preguntas, Reportes, Panel docente, Asignar actividades), and a user session (Hola, Luis Docente). Below the header, the page title 'Asignar actividad' is displayed with a subtitle 'Crea una lectura, define instrucciones y asignala a tus estudiantes.' A 'Sesión' dropdown shows 'Luis Docente'. The main form is divided into two sections:

- 1. DETALLES DE LA ACTIVIDAD**: Fields include 'Título' (Title) with placeholder 'Ej.: "Energía y cambio climático"', 'Instrucciones (opcional)' (Optional instructions) with placeholder 'Describe qué espera que haga el estudiante (por ejemplo, responder todas las preguntas en una sola sesión)...', and 'Texto de la lectura' (Text of the reading) with a text area and placeholder 'Pega aquí el texto que será utilizado para generar preguntas...'. A note at the bottom says 'Si necesitas PDF o Word, podemos integrarlo en una siguiente iteración. Por ahora, usa texto plano o archivos Markdown.'
- 2. CONFIGURACIÓN**: Fields include 'Fecha límite (opcional)' (Optional deadline) with placeholder 'dd/mm/aaaa', 'Supervisión rápida' (Quick supervision) with a note about using it as a diagnostic initial or reinforcement, and 'Selección estudiantes' (Select students) with a list of two students: 'Joel Lazo Maraví' (joel@gmail.com) and 'José Mancilla' (jose@gmail.com). A note says 'Los estudiantes verán esta fecha como referencia para completar la actividad.' and 'Puedes usar esta actividad como diagnóstico inicial o como refuerzo después de una clase de lectura crítica.'

A large blue 'Asignar actividad' (Assign activity) button is at the bottom right. A note below it says 'La actividad aparecerá en el panel de "Mis actividades" de cada estudiante seleccionado.'

FIG.19 Pantalla de asignación de actividad

## CAPÍTULO 6 CODIFICACIÓN DEL SOFTWARE

### 6.1. Desarrollo del Sprint 1

#### 6.1.1. Sprint planning

##### Objetivo del Sprint 1:

Desarrollar el MVP (Producto Mínimo Viable) del Tutor Virtual de Lectura Comprensiva Escolar, que permita al estudiante recibir preguntas generadas automáticamente, obtener retroalimentación inmediata y al docente visualizar reportes e informes básicos de desempeño.

**Duración:** 10 de septiembre – 8 de octubre de 2025

##### Historias de usuario planificadas:

- **HU01 – Recibir preguntas:** Generación automática de preguntas mediante IA a partir del texto proporcionado.
- **HU02 – Recibir retroalimentación:** Mostrar al estudiante comentarios automáticos sobre sus respuestas.
- **HU05 – Acceder a reportes:** Permitir la visualización de resultados individuales por parte del docente.
- **HU06 – Obtener informe:** Mostrar un dashboard con indicadores generales de comprensión.

##### Objetivos técnicos:

- Implementar el flujo básico de interacción entre frontend y backend.
- Conectar la API de generación de preguntas (modelo local con Ollama).
- Implementar el módulo de retroalimentación en tiempo real.
- Crear endpoints para reportes e informes de comprensión lectora.
- Diseñar el dashboard docente inicial con estadísticas básicas.

#### 6.1.2. Sprint backlog

Nº	Historia	Tarea técnica	Responsable
1	HU01	Crear endpoint /api/generate-questions con conexión a modelo IA local (Ollama).	Joel
2	HU01	Implementar vista de lectura y renderizado dinámico de preguntas en React.	Steven
3	HU02	Diseñar función de evaluación automática de respuestas y mostrar retroalimentación.	Andrea
4	HU02	Conectar respuestas del usuario con la API y guardar resultados en MongoDB.	Britney
5	HU05	Crear endpoint /api/reports/student/:id para devolver datos de desempeño individual.	Karim
6	HU06	Implementar componente DashboardDocente.tsx con gráficos básicos (Chart.js).	Steven
7	—	Validar comunicación front-back y control de errores (status 400–500).	Equipo
8	—	Pruebas unitarias con Jest y documentación técnica en Jira.	Equipo

Tabla 8 Sprint 1 backlog

### 6.1.3. Historias de usuario

#### **HU01 – Recibir preguntas**

*Como estudiante, quiero recibir preguntas generadas automáticamente según el texto que lea, para evaluar mi comprensión.*

##### **Criterio de aceptación:**

El sistema genera al menos 5 preguntas coherentes por texto y muestra error controlado si falla el procesamiento.

#### **HU02 – Recibir retroalimentación**

*Como estudiante, quiero recibir comentarios automáticos después de responder, para conocer mis aciertos y errores.*

##### **Criterio de aceptación:**

Tras enviar las respuestas, el sistema muestra resultado correcto/incorrecto con explicación breve.

#### **HU05 – Acceder a reportes**

*Como docente, quiero acceder a reportes individuales de los estudiantes, para analizar su nivel de comprensión.*

**Criterio de aceptación:**

El sistema genera reportes por estudiante con porcentaje de comprensión y actividades realizadas.

#### **HU06 – Obtener informe**

*Como docente, quiero visualizar un panel con indicadores generales del aula, para tener una vista global del desempeño.*

**Criterio de aceptación:**

El dashboard muestra promedio de comprensión por curso y el estado de participación de los estudiantes.

#### **6.1.4. Taskboard**

**Tareas “Done”:**

- Endpoint /api/generate-questions funcional con conexión al modelo Ollama local.
- Interfaz de lectura y preguntas en React implementada.
- Lógica de retroalimentación automática completada.
- Reporte individual y dashboard docente funcionales.
- Validaciones y manejo de errores implementados.
- Evidencias de ejecución y pruebas registradas en Jira.

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

Reuniones diarias realizadas durante el sprint:

Día	Temas clave
1	Definición de arquitectura MERN y flujo general de endpoints.
3	Integración inicial con modelo IA (Ollama) y control de respuestas vacías.

5	Diseño del componente de retroalimentación y pruebas unitarias iniciales.
7	Pruebas de conexión entre backend y frontend, corrección de latencias.
9	Implementación del dashboard y revisión de métricas en MongoDB.
10	Validación final del MVP y preparación para la demo del sprint.

Tabla 9 Daily Scrum 1

#### 6.1.6. Sprint review

##### Logros principales:

- Generación automática de preguntas funcional y estable.
- Retroalimentación inmediata operativa con respuestas correctas/incorrectas.
- Reportes individuales y dashboard docente accesibles.
- Flujo de datos coherente entre el backend (Node.js/Express) y el frontend (React).

##### Feedback recibido:

- Mejorar la variedad de preguntas generadas por la IA.
- Incorporar colores diferenciados en las respuestas correctas e incorrectas.
- Añadir opción de exportar los reportes en formato PDF en sprints posteriores.

#### 6.1.7. Criterios de aceptación

- El sistema genera un mínimo de 5 preguntas por texto.
- Las respuestas muestran retroalimentación inmediata.
- Los reportes individuales y globales son accesibles sin errores.
- Los datos se almacenan correctamente en MongoDB Atlas.
- Todos los endpoints responden en menos de 3 segundos.
- La demo del MVP es navegable y funcional.

#### 6.1.8. Resultados del sprint

##### 6.1.8.1. Evidencias

- Capturas del módulo de lectura mostrando preguntas generadas.
- Evidencia del flujo de retroalimentación con respuestas correctas/incorrectas.

- Capturas del dashboard docente mostrando métricas básicas de comprensión.
- Logs de API /api/generate-questions, /api/reports/student y /api/reports/general.
- Video corto del MVP funcional (Demo Sprint 1).

#### 6.1.8.2. Prueba de desarrollo

Caso	Acción	Resultado esperado
1	Solicitar generación de preguntas para un texto cargado	Devuelve $\geq 5$ preguntas, status 200 OK
2	Enviar respuestas vacías	Muestra mensaje “Debes completar todas las preguntas”
3	Evaluuar respuestas correctas/incorrectas	Muestra retroalimentación visual inmediata
4	Consultar reporte individual de estudiante	Retorna % de comprensión y lista de actividades
5	Acceder al dashboard docente	Muestra gráficos con promedios de comprensión
6	Forzar error en conexión IA	Muestra mensaje de error controlado sin vacíos

Tabla 10 Prueba de desarrollo 1

#### 6.1.9. Sprint retrospective

##### Fortalezas:

- Excelente coordinación del equipo durante el desarrollo.
- Integración efectiva del modelo IA con el backend.
- Cumplimiento del 100% de las HU planificadas
- Base sólida para los siguientes incrementos (Sprint 2 y 3).

##### Debilidades:

- Falta de validación avanzada de calidad de preguntas.
- UI inicial simple, con margen para mejoras de usabilidad.

##### Acciones para el siguiente sprint:

- Implementar barra de progreso por actividad lectora (HU03).
- Mejorar la precisión del análisis de sesgos (HU07).
- Añadir pruebas automatizadas para generación de preguntas.

## 6.2. Desarrollo del Sprint 2

### 6.2.1. Sprint planning

#### Objetivo del Sprint 2:

Implementar la gestión y visualización del progreso del estudiante y la detección de sesgos en los textos de lectura, conectando la base de datos con el panel de análisis interactivo.

**Duración:** 9 de octubre – 4 de noviembre de 2025

#### Historias de usuario planificadas:

- **HU03 – Progreso acumulado:** permitir que el sistema registre y muestre el avance porcentual del estudiante según las actividades desarrolladas.
- **HU07 – Indicar sesgos del texto:** habilitar el análisis asistido por IA para detectar posibles sesgos o subjetividades dentro del texto leído.

#### Objetivos técnicos:

- Integrar la API backend con autenticación JWT.
- Diseñar endpoints REST para obtener y actualizar el progreso del estudiante (/student/activities, /student/progress).
- Implementar el middleware de control de roles (docente / estudiante).
- Desarrollar componentes React para mostrar el avance y análisis textual.

### 6.2.2. Sprint backlog

Nº	Historia	Tarea técnica	Responsable
1	HU03	Crear modelo <b>Submission</b> para registrar progreso	Steven
2	HU03	Endpoint <b>GET / student/activities</b> y <b>POST / student/progress</b>	Joel
3	HU07	Endpoint <b>POST / ai/analyze-text</b> para detectar sesgos con IA (local stub)	Karim
4	HU07	Integrar resultados de IA en interfaz de lectura	Andrea

5	HU03	Mostrar porcentaje de progreso en la interfaz <b>StudentActivities.tsx</b>	Sebas
6	—	Pruebas de API y conexión front-back	Sheyla

Tabla 11 Sprint 2 backlog

### 6.2.3. Historias de usuario

#### HU03 – Progreso acumulado

Como estudiante, quiero ver mi progreso general en cada actividad de lectura para conocer cuánto he avanzado.

**Criterio de aceptación:** El sistema guarda el porcentaje de avance y lo actualiza en tiempo real según las respuestas enviadas  
**HU07 – Indicar sesgos del texto**

Como docente, quiero que el sistema resalte posibles sesgos o subjetividades en el texto analizado para guiar la discusión en clase.

**Criterio de aceptación:** El asistente virtual marca en el texto las frases subjetivas o de juicio de valor.

### 6.2.4. Taskboard

#### Tareas “Done”:

- Endpoints /student/activities y /student/progress.
- Middleware auth.js con JWT y roles.
- Interfaz de progreso en panel de estudiante.
- Prototipo de IA para detección de sesgos.

### 6.2.5. Daily scrum

Reuniones diarias para revisar avances:

Día	Temas clave
1	Revisión de estructura de modelos y autenticación.
3	Integración de middleware y autorización por rol.
5	Pruebas de carga de progreso y visualización en frontend.

Tabla 12 Daily Scrum 2

### 6.2.6. Sprint review

#### Logros:

- Los estudiantes pueden visualizar sus actividades y el porcentaje de progreso.
- Docentes observan los sesgos textuales analizados por el módulo de IA.
- Integración funcional entre backend y frontend.

#### Feedback:

- Reforzar la precisión del análisis de sesgos para el sprint siguiente.
- Agregar métricas visuales de progreso por color o barras.

### 6.2.7. Criterios de aceptación

- El estudiante puede ver todas sus actividades asignadas con su porcentaje de avance.
- Solo los usuarios autenticados acceden a sus datos.
- El docente recibe el resultado del análisis de sesgos en texto.
- Los datos se guardan en MongoDB y se actualizan sin errores de autorización.

### 6.2.8. Resultados del sprint

#### 6.2.8.1. Evidencias

- Capturas del panel de progreso del estudiante.
- Logs de peticiones POST / student/progress y GET / student/activities.
- Demostración de análisis de sesgos IA en texto de lectura.
- Validación exitosa de tokens JWT y roles.

### **6.2.8.2. Prueba de desarrollo**

Caso	Acción	Resultado
1	Enviar progreso desde interfaz	201 Created + % actualizado
2	Acceder sin token	401 No autorizado
3	Usuario sin rol válido	403 No autorizado
4	Analizar texto con IA local	Retorna palabras marcadas por sesgo

Tabla 13 Prueba de desarrollo 2

### **6.2.9. Sprint retrospective**

#### **Fortalezas:**

- Integración completa entre API, roles y base de datos.
- Buen manejo de autenticación y validación de roles.

#### **Debilidades:**

- Tiempo limitado para ajustar el análisis IA.
- Falta de visualización gráfica más intuitiva del progreso.

#### **Acciones para el Sprint 3:**

- Consolidar el panel docente para asignar actividades (HU04).
- Añadir módulo de recordatorios automáticos (HU08).
- Generar reportes con métricas de comprensión (HU09).

## **6.3. Desarrollo del Sprint 3**

### **6.3.1. Sprint planning**

#### **Objetivo del Sprint 3:**

Implementar la funcionalidad completa de gestión docente, incluyendo la asignación de actividades, el sistema de recordatorios automatizados mediante n8n y el panel de comprensión global para administradores, consolidando así todas las capacidades del Tutor Virtual de Lectura Comprensiva.

**Duración:** 5 de noviembre - 2 de diciembre de 2025

**Historias de usuario planificadas:**

- **HU04 – Asignar actividades:** Permitir al docente subir textos y asignarlos a estudiantes específicos.
- **HU08 – Enviar recordatorios de lectura:** Implementar flujos automatizados con n8n para programar y enviar notificaciones.
- **HU09 – Mostrar panel de comprensión:** Desarrollar dashboard con métricas agregadas de comprensión lectora.

**Objetivos técnicos:**

- Implementar módulo completo de gestión docente con interfaz React.
- Configurar flujos de n8n para recordatorios automatizados.
- Desarrollar panel administrativo con gráficos avanzados.
- Integrar sistema de notificaciones por correo electrónico.
- Realizar pruebas de integración end-to-end.

### 6.3.2. Sprint backlog

Nº	Historia	Tarea técnica	Responsable
1	HU04	Crear endpoint POST /api/assignments para asignar textos	Joel
2	HU04	Implementar componente AssignmentForm en React con validaciones	Steven
3	HU08	Configurar workflow n8n para recordatorios automáticos	Karim
4	HU08	Integrar webhooks n8n con backend para registro de actividades	Andrea
5	HU09	Desarrollar componente AdminDashboard con Chart.js	Britney
6	HU09	Crear endpoint GET /api/admin/metrics para datos agregados	Sebas
7	-	Implementar sistema de notificaciones por email	Equipo

8	-	Pruebas E2E con Cypress para flujos completos	Sheyla
---	---	---	--------

Tabla 14 Sprint 3 backlog

### 6.3.3. Historias de usuario

#### **HU04 – Asignar actividades**

Como docente, quiero asignar textos con actividades a mis estudiantes para gestionar su aprendizaje.

##### **Criterio de aceptación:**

- El docente puede subir texto en formatos soportados (txt, pdf, docx).
- Puede seleccionar estudiantes o grupos específicos para asignar.
- Si el formato no es soportado, muestra error claro y no asigna.

#### **HU08 – Enviar recordatorios de lectura**

Como sistema, quiero programar y enviar recordatorios con n8n para mantener el engagement de los estudiantes.

##### **Criterio de aceptación:**

- n8n envía notificaciones en fecha/hora configurada.
- Ante fallo de envío, registra error y realiza reintentos automáticos.
- Integración completa con base de datos para registro de actividades.

#### **HU09 – Mostrar panel de comprensión**

Como administrador, quiero ver un panel con métricas de comprensión promedio para tener una visión global del rendimiento.

##### **Criterio de aceptación:**

- El panel muestra KPIs agregados (promedio general, tendencias).

- Si no hay datos suficientes, indica "No hay datos suficientes para mostrar".
- Gráficos interactivos con filtros por periodo y nivel educativo.

#### **6.3.4. Taskboard**

**Tareas “Done”:**

- Endpoint /api/assignments funcional con validación de archivos.
- Interfaz de asignación docente implementada en React.
- Workflows n8n configurados y operativos para recordatorios.
- Panel administrativo con métricas de comprensión.
- Sistema de notificaciones por email integrado.
- Pruebas E2E ejecutadas y aprobadas.

#### **6.3.5. Daily scrum**

Reuniones diarias realizadas durante el sprint:

Día	Temas clave
1	Definición de estructura de datos para asignaciones
3	Integración inicial de n8n con backend MongoDB
5	Desarrollo de componentes de gráficos para dashboard
7	Pruebas de flujo completo: asignación → notificación → registro
9	Optimización de rendimiento del panel administrativo
10	Validación final y preparación para demo del sprint

Tabla 15 Daily Scrum 3

### **6.3.6. Sprint review**

#### **Logros:**

- Módulo completo de gestión docente operativo.
- Sistema de recordatorios automatizados funcionando con n8n.
- Panel administrativo con métricas de comprensión implementado.
- Integración exitosa de todos los flujos del sistema.

#### **Feedback:**

- Mejorar la interfaz de asignación para selección masiva de estudiantes.
- Agregar más tipos de gráficos en el panel administrativo.
- Optimizar frecuencia de recordatorios para no saturar a usuarios.

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

- Los docentes pueden asignar textos a estudiantes/grupos específicos.
- El sistema envía recordatorios automáticos según programación.
- El panel administrativo muestra métricas agregadas de comprensión.
- Todos los flujos funcionan de extremo a extremo sin errores.
- La integración n8n-backend es estable y confiable.

### **6.3.8. Resultados del sprint**

#### **6.3.8.1. Evidencias**

- Capturas del módulo de asignación docente.
- Logs de ejecución de workflows n8n.
- Demostración del panel administrativo con datos reales.
- Evidencias de notificaciones enviadas por correo.

#### **6.3.8.2. Prueba de desarrollo**

Caso	Acción	Resultado
1	Docente asigna texto a grupo	201 Created + notificación programada
2	n8n ejecuta recordatorio	Email enviado + registro en BD
3	Administrador consulta panel	Métricas mostradas correctamente
4	Intento de asignar archivo inválido	Error 400 con mensaje claro

Tabla 16 Pruebas de desarrollo 3

### 6.3.9. Sprint retrospective

#### Fortalezas:

- Excelente coordinación en la integración de componentes complejos.
- n8n demostró ser una herramienta robusta para automatización.
- Panel administrativo superó expectativas de funcionalidad.
- Cumplimiento del 100% de las HU planificadas.

#### Debilidades:

- Curva de aprendizaje inicial con n8n más pronunciada de lo esperado.
- Tiempo limitado para optimizaciones de rendimiento en gráficos.

#### Acciones para mejora continua:

- Documentar mejores prácticas de uso de n8n.
- Planificar sprint de optimización post-entrega.
- Establecer monitoreo continuo de flujos automatizados.

#### Lecciones aprendidas:

- La planificación detallada de flujos de automatización es crucial.
- Las pruebas E2E son esenciales para sistemas con múltiples integraciones.
- La retroalimentación temprana de usuarios mejora significativamente la usabilidad.

## CAPÍTULO 7 PRUEBAS DE SOFTWARE

### 7.1. Plan de Pruebas

El plan de pruebas del sistema “Tutor Virtual de Lectura Comprensiva Escolar – I.E.P. San Carlos” define la estrategia, tipos de pruebas, herramientas, ambiente y criterios de aceptación utilizados para asegurar la calidad del producto desarrollado sobre la arquitectura MERN con integración de IA y n8n.

#### 7.1.1. Objetivo del plan de pruebas

Verificar que las funcionalidades implementadas cumplan con los requerimientos funcionales y no funcionales definidos en los capítulos anteriores (RF01–RF09, RNF01–RNF07), asegurando que:

- Los flujos críticos de estudiante, docente y coordinador se ejecuten sin errores.
- La información registrada en la base de datos sea consistente y trazable.
- El desempeño y la usabilidad del sistema sean aceptables para su contexto educativo.
- Las historias de usuario planificadas en los tres sprints cumplan sus criterios de aceptación y la DoD.

#### 7.1.2. Alcance

El plan de pruebas abarca:

- **Frontend (React + Vite + Tailwind):** componentes de interfaz para login, panel docente, generador de preguntas, panel de reportes y actividades del estudiante.
- **Backend (Node.js + Express + MongoDB Atlas):** controladores, servicios, modelos y endpoints REST.
- **Servicios de IA y automatización:** integración con la API de generación de preguntas y módulo de sesgos, así como flujos n8n para recordatorios.
- **Flujos de extremo a extremo (E2E):** desde el login de usuario hasta la entrega de actividades y visualización de reportes.
- **Pruebas de usabilidad:** aplicación de heurísticas de Nielsen y escala SUS a la versión final del sistema.

Fuera de alcance quedan pruebas de rendimiento a gran escala (stress testing con miles de usuarios concurrentes) y pruebas de compatibilidad en navegadores legacy.

### 7.1.3. Estrategia general de pruebas

Se ha seguido una estrategia **incremental y por capas**, alineada a los sprints del proyecto:

#### 1. Sprint 1 – Pruebas sobre MVP de lectura y reportes básicos

- a. Pruebas unitarias de backend (Jest) para generación de preguntas, retroalimentación y cálculo de métricas.
- b. Pruebas de API con **Supertest** para endpoints `/api/auth/login`, `/api/ai/questions`, `/api/reports/student/:id` y `/api/reports/summary`.
- c. Pruebas manuales y con Postman/cliente REST desde VS Code para verificar códigos de estado, formatos JSON y manejo de errores.
- d. Primeros casos E2E con **Cypress**: flujo de estudiante que genera preguntas y docente que accede a reportes.

#### 2. Sprint 2 – Pruebas sobre progreso del estudiante y análisis de sesgos

- a. Nuevas pruebas unitarias Jest para el modelo *Submission*, servicios de cálculo de progreso y módulo de sesgos.
- b. Pruebas de integración con Supertest para los endpoints `/api/student/activities`, `/api/student/progress` y `/api/ai/biases`.
- c. Pruebas E2E Cypress para:
  - Visualización de actividades del estudiante y actualización de barras de progreso.
  - Visualización de sesgos detectados en la interfaz de lectura.

#### 3. Sprint 3 – Pruebas sobre gestión docente, recordatorios y panel global

- a. Pruebas unitarias sobre lógica del módulo de asignación de actividades, panel administrativo y adaptadores hacia n8n.
- b. Pruebas de integración para `/api/teacher/activities`, `/api/teacher/students`, `/api/admin/metrics` y webhooks conectados a n8n.
- c. Pruebas E2E Cypress para:

- Flujo de docente que asigna una lectura a estudiantes.
  - Flujo de estudiante que ve la actividad, guarda borrador y entrega.
  - Acceso a reportes y panel de comprensión global.
- d. Aplicación de pruebas de usabilidad (Heurísticas de Nielsen y SUS) con usuarios representativos (docentes y estudiantes).

#### 7.1.4. Tipos de pruebas y herramientas

##### a) Pruebas unitarias (frontend y backend)

- **Herramienta:** Jest.
- **Objetivo:** Validar funciones, componentes y servicios aislados.
- **Alcance principal:**
  - Backend: servicios de generación de preguntas, cálculo de porcentaje de progreso, resumen de reportes, validación de roles y autenticación.
  - Frontend: componentes de React para formularios de login, generador de preguntas, tarjetas de actividades y gráficos de reportes (render básico y manejo de estados).

##### b) Pruebas de integración de API

- **Herramienta:** Jest + Supertest.
- **Objetivo:** Verificar que los controladores Express, modelos MongoDB y middleware de autenticación trabajen correctamente como una unidad.
- **Endpoints principales probados:**
  - Autenticación: `/api/auth/login`, `/api/auth/register`.
  - Estudiante: `/api/student/activities`, `/api/student/progress`.
  - Docente: `/api/teacher/students`, `/api/teacher/activities`.
  - Analítica: `/api/ai/questions`, `/api/ai/biases`,  
`/api/ai/reports`, `/api/admin/metrics`.

### c) Pruebas de API con Postman / cliente REST

- **Herramientas:** Colecciones Postman o cliente REST en VS Code.
- **Objetivo:** Ejecutar *smoke tests* y pruebas exploratorias sobre el backend desplegado, validando:
  - Códigos de respuesta (2xx, 4xx, 5xx).
  - Estructura del JSON (campos obligatorios, mensajes de error).
  - Reglas de autenticación y autorización (requiere token JWT, validación de rol docente/estudiante).
- Se generaron colecciones agrupadas por módulo (Auth, Student, Teacher, Reports) y se usaron en cada entrega de sprint.

### d) Pruebas end-to-end (E2E)

- **Herramienta:** Cypress.
- **Objetivo:** Validar el funcionamiento del sistema desde la perspectiva del usuario final, interactuando con el frontend real y el backend en ejecución.
- **Principales flujos cubiertos:**

#### 1. Autenticación y redirecciones por rol

- Camino feliz docente: login real y acceso a */teacher/dashboard*.
- Camino feliz estudiante: login real y acceso a */questions*.
- Caminos infelices: acceso sin sesión o con rol incorrecto redirige al login o ruta autorizada.

#### 2. Panel docente y asignación de actividades

- El docente visualiza métricas de actividades, lista de lecturas asignadas y acciones rápidas.
- Camino feliz: docente crea una actividad desde */teacher/assign* seleccionando al menos un estudiante, con validación de campos obligatorios.

- Camino infeliz: intento de asignar sin completar campos muestra mensajes de error sin romper la página.

### 3. Generador de preguntas para estudiantes

- Camino feliz: estudiante ingresa un texto, la interfaz llama a `/api/ai/questions` y se generan preguntas, manteniendo estable la sección de sesgos.
- Camino infeliz: estudiante intenta generar sin texto; el sistema mantiene la página estable y evita peticiones inválidas.

### 4. Página de reportes

- Camino feliz: el docente ingresa a `/reports`, se obtiene la información desde `/api/ai/reports` (códigos 200/304) y se muestran totales y gráficos.
- Camino infeliz: se simula fallo del API (500) y se comprueba el mensaje “No se pudieron cargar los reportes. Intenta nuevamente más tarde”.

### 5. Flujo completo del estudiante

- Se crea una actividad real por API para un estudiante de prueba.
- El estudiante inicia sesión, ingresa a “Mis actividades”, abre la actividad, responde, guarda borrador y envía la actividad, verificando que el estado cambie a “Entregada” y el progreso llegue a 100%.

## e) Pruebas de usabilidad

- **Herramientas:** cuestionarios físicos/digitales, hoja de cálculo.
- **Enfoque:**
  - **Heurísticas de Nielsen (10 ítems, Likert 1–5):** se evaluaron coherencia visual, prevención de errores, claridad de mensajes y visibilidad del estado del sistema. Se calcularon promedios por ítem y hallazgos clave para propuestas de mejora.
  - **System Usability Scale – SUS:** se aplicó a docentes y estudiantes que utilizaron el sistema al cierre del Sprint 3, obteniendo un puntaje global SUS (0–100) para clasificar la usabilidad como aceptable o excelente.

### 7.1.5. Ambiente de pruebas y datos

#### Entorno de ejecución:

- Backend Node.js 20+, Express, MongoDB Atlas (cluster de pruebas).
- Frontend React + Vite en modo desarrollo y build de producción.
- n8n desplegado localmente o en servicio remoto para flujos de recordatorios.

#### Datos de prueba:

- Usuarios docentes y estudiantes configurados en la base de datos (por ejemplo, docente de prueba y estudiante “joel@gmail.com”).
- Textos de lectura de diferentes niveles (inicial, primaria y secundaria).
- Actividades y suscripciones de prueba para verificar distintos estados (en progreso, entregada, sin fecha, etc.).

### 7.1.6. Criterios de aceptación y salida

- Todos los **casos de prueba críticos** (autenticación, generación de preguntas, envío de respuestas, visualización de reportes, flujo de entrega de actividades) deben ejecutarse sin errores.
- Cobertura objetivo de pruebas unitarias:  $\geq 80\%$  en módulos centrales de backend (servicios de IA, progresos, reportes).
- Ningún **defecto crítico (severidad alta)** puede permanecer abierto al momento de la entrega final.
- La **prueba SUS** debe arrojar un puntaje global  $\geq 68$  (usabilidad aceptable) y no se deben detectar problemas graves de navegación según las heurísticas de Nielsen.

### 7.1.7. Trazabilidad con los requisitos

Cada historia de usuario (HU01–HU09) está asociada a uno o más casos de prueba unitarios, de integración y/o E2E. La matriz de trazabilidad se gestiona en Jira, vinculando:

- Requerimientos funcionales (RF01–RF09)  $\leftrightarrow$  Historias de usuario  $\leftrightarrow$  Casos de prueba.
- Resultados de ejecución  $\leftrightarrow$  Evidencias (capturas de pantalla, logs de Cypress, colecciones Postman).

Esta trazabilidad permite demostrar que el Tutor Virtual de Lectura Comprensiva Escolar cumple con las funcionalidades previstas y que las pruebas se han realizado de manera sistemática y verificable.

## **CONCLUSIONES**

El Tutor Virtual de Lectura Comprensiva Escolar representa una solución efectiva y viable para mejorar la evaluación y el fortalecimiento de la comprensión lectora en los estudiantes de la I.E.P. San Carlos. La integración de inteligencia artificial para la generación automática de preguntas y retroalimentación inmediata permitió optimizar el proceso de enseñanza y aprendizaje.

El sistema reduce significativamente la carga operativa del docente, al automatizar la elaboración de preguntas, la calificación y el seguimiento del avance lector. Esto permitió que el profesor enfoque su tiempo en actividades pedagógicas de mayor valor, como el acompañamiento personalizado.

Las pruebas funcionales, de integración, E2E y de usabilidad demostraron que la plataforma es estable, intuitiva y técnicamente sólida, logrando cumplir con los requerimientos planteados y manteniendo una alta compatibilidad con entornos educativos reales.

La arquitectura MERN y las herramientas utilizadas (MongoDB, React, Node.js, n8n, Ollama/Hugging Face) evidenciaron alta factibilidad técnica, económica y operativa, pues se basan en tecnologías gratuitas, mantenibles y escalables, lo que facilita su sostenibilidad a futuro.

A lo largo del desarrollo del proyecto, el enfoque Scrum permitió una gestión ordenada, iterativa y transparente, facilitando la entrega de incrementos funcionales, la identificación temprana de riesgos y la mejora continua del producto.

El sistema contribuye directamente a los objetivos institucionales, pues fortalece la comprensión lectora, promueve la autonomía del estudiante y moderniza el proceso evaluativo, alineándose con la visión de innovación educativa de la institución.

## **RECOMENDACIONES**

Implementar el sistema a nivel institucional de manera progresiva, iniciando por los grados superiores de primaria y secundaria, para posteriormente extenderlo a todos los niveles una vez validado su impacto pedagógico.

Capacitar a docentes y estudiantes en el uso de la plataforma, especialmente en la interpretación de la retroalimentación generada por la IA y en el aprovechamiento de los reportes generados por el sistema.

Expandir el banco de textos y actividades, incorporando lecturas por competencias, niveles de dificultad y temas transversales, a fin de complementar y enriquecer las prácticas lectoras.

Monitorear periódicamente el desempeño del sistema, realizando pruebas de mantenimiento, revisión de logs y actualización de modelos de IA para evitar errores y mejorar la precisión de la retroalimentación.

Optimizar la infraestructura de despliegue en caso de mayor demanda, migrando a planes superiores de los servicios en la nube si es necesario, garantizando disponibilidad, seguridad y estabilidad ante mayor número de usuarios.

Integrar nuevas funcionalidades como gamificación, badges, lectura guiada o analítica predictiva que permita identificar estudiantes en riesgo, potenciando el impacto educativo del sistema.

Acompañar la implementación con una política institucional de uso de tecnologías educativas, para asegurar coherencia, continuidad y sostenibilidad del proyecto en el tiempo.

## **ANEXOS**

### **Anexo 01. Manual Técnico**

#### **1.1. Datos generales del sistema**

- **Nombre del sistema:** Tutor virtual de lectura crítica
- **Tipo de sistema:** Aplicación web educativa con soporte de inteligencia artificial
- **Usuarios principales:** Estudiantes y docentes de lectura crítica / comprensión lectora
- **Desarrollador:** Equipo del Proyecto 3 – Taller de Proyectos II

El sistema permite a los estudiantes practicar la lectura crítica mediante la presentación de textos, la generación automática de preguntas y la entrega de retroalimentación generada por un modelo de inteligencia artificial. A su vez, los docentes pueden asignar lecturas, revisar el avance de los estudiantes y generar reportes de desempeño.

#### **1.2. Objetivo del manual de usuario**

El presente manual tiene como objetivo **orientar al usuario final** (estudiante, docente o administrador) en el uso correcto del sistema “Tutor virtual de lectura crítica”.

En particular, el manual describe:

- Cómo acceder al sistema.
- Cómo registrarse e iniciar sesión.
- Cómo utilizar las funcionalidades principales según el tipo de usuario.
- Recomendaciones básicas de uso y solución de problemas frecuentes.

#### **1.3. Requisitos para el uso del sistema**

##### **1.3.1. Requisitos de hardware y software**

- Computadora o laptop con al menos:
  - Procesador de 2 núcleos o superior.

- 4 GB de memoria RAM o más.
- Conexión estable a Internet.
- Navegador web actualizado (Google Chrome, Microsoft Edge o Mozilla Firefox).
- Resolución de pantalla recomendada: 1366 x 768 o superior.

### 1.3.2. Requisitos de cuenta de usuario

- Contar con un **correo electrónico válido** (institucional o personal).
- Disponer de un **usuario y contraseña** registrados en el sistema o proporcionados por el docente/administrador.

## 1.4. Acceso al sistema

1. Abrir el navegador web.
2. Escribir en la barra de direcciones la **URL del sistema** (proporcionada por el docente o administrador).
3. Esperar a que cargue la pantalla de inicio, donde se mostrará el formulario de **inicio de sesión** y, en algunos casos, la opción de **registro**.

Nota: Si el sistema se ejecuta de forma local (por ejemplo, en un laboratorio), la dirección puede ser similar a <http://localhost:5173>, configurada por el equipo técnico.

## 1.5. Gestión de usuarios

### 1.5.1. Registro de nuevo usuario

Si el sistema permite el registro directo de usuarios, se deben seguir los siguientes pasos:

1. En la pantalla de inicio, hacer clic en el botón “**Registrarse**” o “**Crear cuenta**”.
2. Completar el formulario con los datos solicitados:
  - Nombre y apellidos.
  - Correo electrónico.
  - Rol (por ejemplo, estudiante o docente), si el formulario lo requiere.

- Contraseña.
3. Confirmar el registro con el botón “**Registrar**”.
  4. Si el registro es exitoso, el sistema mostrará un mensaje de confirmación y redirigirá a la pantalla de inicio de sesión.

En algunos entornos, el registro de cuentas puede realizarse únicamente por parte del **administrador**. En ese caso, el usuario recibirá sus credenciales ya generadas.

#### **1.5.2. Inicio de sesión**

1. En la pantalla principal, ingresar el **correo electrónico** y la **contraseña** registrados.
2. Hacer clic en el botón “**Iniciar sesión**”.
3. Si los datos son correctos, el sistema mostrará el **panel principal** correspondiente al rol del usuario (estudiante, docente o administrador).

#### **1.5.3. Cierre de sesión**

1. Ubicar el menú de usuario, generalmente en la esquina superior derecha de la pantalla.
2. Seleccionar la opción “**Cerrar sesión**”.
3. El sistema redirigirá a la pantalla de inicio.

### **1.6. Uso del sistema como ESTUDIANTE**

#### **1.6.1. Panel principal del estudiante**

Al iniciar sesión como estudiante, se muestra un panel donde el usuario puede visualizar:

- Sus **últimas actividades de lectura**.
- Un resumen de su **desempeño** (por ejemplo, número de actividades realizadas o porcentaje de respuestas correctas).
- Accesos a opciones como:
  - **Nueva actividad de lectura**.

- **Historial de actividades.**
- **Reportes o progreso personal.**

#### **1.6.2. Crear una nueva actividad de lectura**

1. En el panel del estudiante, hacer clic en “**Nueva lectura**” o “**Nueva actividad**”.
2. Ingresar el texto sobre el cual se desea trabajar. El sistema puede permitir:
  - Pegar directamente el texto en un cuadro de entrada.
  - Seleccionar un texto previamente asignado por el docente.
3. Verificar que el texto esté completo y en el idioma correcto.
4. Hacer clic en el botón “**Generar preguntas**” (o similar).

El sistema procesará el texto y generará un conjunto de preguntas orientadas a evaluar la comprensión y el pensamiento crítico.

#### **1.6.3. Responder las preguntas generadas**

1. Una vez que el sistema muestre las preguntas, leer detenidamente cada una.
2. Escribir la respuesta en el campo correspondiente.
3. Repetir el proceso para todas las preguntas asociadas a la lectura.
4. Finalmente, hacer clic en “**Enviar**”, “**Guardar respuestas**” o “**Finalizar actividad**”.

#### **1.6.4. Visualizar la retroalimentación de la IA**

1. Tras enviar las respuestas, el sistema mostrará para cada pregunta:
  - La respuesta del estudiante.
  - Un comentario de la **IA**, indicando si la respuesta es adecuada, parcial o incorrecta, y sugiriendo mejoras.
2. El estudiante debe leer esta retroalimentación para identificar:
  - Ideas mal interpretadas.

- Argumentos incompletos.
- Oportunidades de mejorar su análisis crítico.

#### **1.6.5. Consultar historial y reportes de progreso**

1. Desde el menú principal, seleccionar la opción “**Historial**” o “**Mis actividades**”.
2. El sistema listará las lecturas realizadas, con información como:
  - Fecha de la actividad.
  - Cantidad de preguntas respondidas.
  - Resultado o evaluación global.
3. En la sección de “**Reportes**” o “**Progreso**”, el estudiante podrá ver un resumen de su desempeño a lo largo del tiempo.

### **1.7. Uso del sistema como DOCENTE**

#### **1.7.1. Panel principal del docente**

Al iniciar sesión como docente, el usuario puede acceder a funcionalidades adicionales, entre ellas:

- **Asignar textos o actividades** a sus estudiantes.
- **Revisar el progreso** de cada estudiante o grupo.
- **Generar reportes**, que pueden exportarse o imprimirse.

#### **1.7.2. Asignación de textos**

1. En el panel del docente, seleccionar la opción “**Asignar texto**” o “**Nueva actividad para estudiantes**”.
2. Completar la información solicitada:
  - Título de la lectura.
  - Texto completo o enlace al texto.
  - Indicaciones específicas para los estudiantes (por ejemplo, número mínimo de preguntas a responder o fecha límite).

3. Seleccionar el grupo o lista de estudiantes a los que se asignará la actividad, si el sistema lo permite.
4. Guardar la asignación.

#### **1.7.3. Revisión del desempeño de los estudiantes**

1. Acceder al módulo “**Reportes**”, “**Progreso de estudiantes**” o similar.
2. Seleccionar la actividad o el grupo de interés.
3. Visualizar la información disponible, por ejemplo:
  - Número de actividades completadas por cada estudiante.
  - Porcentaje de respuestas correctas, parciales e incorrectas.
  - Retroalimentación generada por la IA.
4. Utilizar esta información para:
  - Identificar estudiantes que requieren refuerzo.
  - Evaluar la efectividad de las lecturas asignadas.
  - Apoyar la toma de decisiones pedagógicas.

### **1.8. Uso del sistema como ADMINISTRADOR**

Esta sección aplica cuando existe un rol de administrador responsable de la configuración y mantenimiento del sistema.

#### **1.8.1. Funciones principales**

El administrador puede:

- Crear, modificar y desactivar cuentas de usuario.
- Asignar roles (estudiante, docente, administrador).
- Supervisar el uso general del sistema.

#### **1.8.2. Gestión de cuentas**

1. Iniciar sesión con un usuario con privilegios de administrador.

2. Ingresar al módulo “**Gestión de usuarios**”.
3. Realizar las operaciones necesarias:
  - **Crear usuario:** completar los datos básicos y asignar rol.
  - **Editar usuario:** actualizar información o cambiar rol.
  - **Desactivar usuario:** impedir el acceso de manera temporal o definitiva.

## **1.9. Recomendaciones de uso**

- Mantener la **contraseña en reserva** y no compartirla con otras personas.
- Cerrar sesión al terminar de usar el sistema, especialmente en computadoras compartidas.
- Leer atentamente las preguntas y la retroalimentación de la IA para aprovechar al máximo la herramienta.
- En caso de encontrar errores o comportamientos extraños, informar al docente o al administrador.

## **1.10. Preguntas frecuentes (FAQ)**

### **1. ¿Qué hago si olvidé mi contraseña?**

Solicitar al docente o administrador el restablecimiento de la contraseña, según el procedimiento definido para el sistema.

### **2. ¿Por qué no veo las actividades asignadas por el docente?**

Verificar que se haya iniciado sesión con el usuario correcto. Si el problema persiste, comunicarse con el docente para confirmar la asignación.

### **3. ¿Qué hago si el sistema no carga o se queda en blanco?**

Comprobar la conexión a Internet y actualizar la página. Si el problema continúa, probar con otro navegador o contactar al administrador.

### **4. ¿La IA puede equivocarse en la retroalimentación?**

Sí, la IA es una herramienta de apoyo y puede cometer errores. Se recomienda que el estudiante utilice la retroalimentación como guía y, ante dudas, consulte con su docente.

## **1.11. Contacto y soporte**

Para cualquier inconveniente técnico o duda relacionada con el funcionamiento del sistema, el usuario debe comunicarse con:

- El docente responsable del curso, o
- El administrador del sistema, según se haya definido en la institución.

## Anexo 02. Manual de Usuario

**Manual de usuario**

**Sistema: Tutor Virtual de Lectura Comprensiva Escolar**

### Introducción

El Tutor Virtual de Lectura Comprensiva Escolar es una plataforma web educativa que utiliza inteligencia artificial para generar preguntas automáticas, ofrecer retroalimentación inmediata y permitir a los docentes monitorear el avance lector de los estudiantes.

Este manual tiene como finalidad orientar a estudiantes, docentes y administradores en el uso adecuado del sistema.

### Objetivo del Manual

- Guiar al usuario en el acceso y navegación del sistema.
- Explicar las funciones principales según el rol asignado (Estudiante, Docente, Administrador).
- Presentar soluciones a problemas frecuentes.
- Facilitar el aprendizaje del uso del software sin necesidad de asistencia técnica.

### Requisitos del Sistema

<b>Hardware</b> <ul style="list-style-type: none"> <li>• PC o laptop con procesador de dos núcleos o superior.</li> <li>• 4 GB de memoria RAM (mínimo).</li> <li>• Conexión estable a Internet.</li> <li>• Resolución recomendada: 1366x768 o superior.</li> </ul>	<b>Software</b> <ul style="list-style-type: none"> <li>• Navegadores compatibles:           <ul style="list-style-type: none"> <li>Google Chrome (recomendado)</li> <li>Mozilla Firefox</li> <li>Microsoft Edge</li> </ul> </li> <li>• Cuenta de usuario activa (correo + contraseña).</li> </ul>
--	---

### Gestión de Usuarios

**Inicio de Sesión**

1. Ingrese su correo electrónico.
2. Ingrese su contraseña.
3. Haga clic en Iniciar Sesión.
4. Automáticamente será redirigido a su panel principal.



### Uso del Sistema-Estudiante

**Realizar una Actividad de Lectura**

Paso 1: Seleccionar una actividad

1. Ingrese al módulo Mis Actividades.
2. Seleccione la lectura asignada.

Paso 2: Generar preguntas

1. Haga clic en Generar Preguntas.
2. El sistema procesará el texto y mostrará preguntas automáticas.

Paso 3: Responder preguntas

1. Lea cada pregunta generada.
2. Escriba su respuesta en el campo correspondiente.
3. Presione Enviar actividad para finalizar.

**Panel Principal del Estudiante**

En esta vista encontrará:

- Actividades asignadas por el docente.
- Porcentaje de avance por actividad.
- Panel de progreso lector general.

**Ver Retroalimentación Automática**

Después del envío, el sistema mostrará:

- Respuesta del estudiante
- Evaluación de la IA (correcta / parcial / incorrecta)
- Sugerencias de mejora
- Indicadores de comprensión

**Revisar Progreso Personal**

En el menú Mi Progreso, podrá consultar:

- Actividades realizadas
- Resultados obtenidos
- Porcentaje acumulado
- Tendencias de mejora

### Uso del Sistema-Docente

**Panel Principal del Docente**

Incluye accesos a:

- Crear y asignar actividades
- Revisar entregas
- Ver reportes por estudiante y por grupo
- Panel de comprensión global

**Asignar un Texto a los Estudiantes**

Paso 1: Ir a "Nueva Asignación".

1. Haga clic en Asignar lectura.

Paso 2: Completar información

- Título del texto
- Contenido (ingresado o cargado)
- Fecha límite (opcional)
- Grupo o estudiantes seleccionados

Paso 3: Guardar

- Presione Crear Asignación.
- El sistema notificará automáticamente a los estudiantes

**Revisar Actividades Entregadas**

- Acceso directo a Proyecto de Estudiantes.
- Seleccionar el estudiante o grupo.
- Vista de:

  - Actividades individuales
  - Retroalimentación automática
  - Puntaje total
  - Reporte de avance
  - 4 GB de memoria RAM (mínimo).
  - Resolución recomendada: 1366x768 o superior.

**Generar Reportes**

- El docente puede consultar:
- Reporte por estudiante
- Promedio del grupo
- Evolución de comprensión
- Actividades pendientes

### Uso del Sistema-ADMINISTRADOR

**Funciones Principales**

- Crear, editar y desactivar cuentas
- Asignar roles
- Monitorear el uso general del sistema

**Gestión de Usuarios**

1. Ir a Administración → Usuarios
2. Seleccionar la acción correspondiente:
  - Crear usuario
  - Editar información
  - Desactivar / reactivar

### Preguntas Frecuentes

**¿Qué hago si olvidé mi contraseña?**

Debe comunicarlo al administrador o docente para restablecerla.

**¿Por qué no aparecen mis actividades?**

Verifique que ingresó con el usuario correcto. Si persiste, el docente debe confirmar la asignación.

**El sistema no carga o se queda en blanco.**

Verifique su conexión a Internet. Actualice la página. Pruebe con otro navegador.

**¿La inteligencia artificial puede equivocarse?**

Sí. La retroalimentación es una guía. Cualquier duda debe consultarse con el docente.

**Recomendaciones de Uso**

- No comparta su contraseña.
- Cierre sesión al finalizar.
- Lea cuidadosamente cada pregunta antes de responder.
- Mantenga su navegador actualizado.
- Reporte cualquier error al docente o administrador.

**Contacto y Soporte**

Ante problemas o consultas, comunicarse con:

- Docente responsable del aula, o
- Administrador del sistema designado por la institución.

## **LISTA DE TABLAS**

Tabla 1 Gastos generales	12
Tabla 2 Requerimientos funcionales	14
Tabla 3 Requerimientos no funcionales	15
Tabla 4 Actores del sistema	15
Tabla 5 Cronograma de actividades	18
Tabla 6 Gestión de riesgos	19
Tabla 7 Diccionario de datos	31
Tabla 8 Sprint 1 backlog	38
Tabla 9 Daily Scrum 1	40
Tabla 10 Prueba de desarrollo 1	41
Tabla 11 Sprint 2 backlog	43
Tabla 12 Daily Scrum 2	44
Tabla 13 Prueba de desarrollo 2	45
Tabla 14 Sprint 3 backlog	47
Tabla 15 Daily Scrum 3	48
Tabla 16 Pruebas de desarrollo 3	50

## **LISTA DE FIGURAS**

FIG.1 Organigrama	5
FIG.2 Procesos	8
FIG.3 Casos de uso	22
FIG.4 Diagrama de secuencia	22
FIG.5 Diagrama de colaboración	23
FIG.6 Diagrama de clases	24
FIG.7 Diseño conceptual	25
FIG.8 Diseño lógico	26
FIG.9 Diseño físico	27
FIG.10 Página de inicio	32
FIG.11 Generación de preguntas	32
FIG.12 Nuevo usuario	33
FIG.13 Inicio de sesión	33
FIG.14 Reporte de respuestas	34
FIG.15 Pantalla de bloqueo	34
FIG.16 Actividades del estudiante	35
FIG.17 Actividad asignada al estudiante	35
FIG.18 Pantalla panel docente	36
FIG.19 Pantalla de asignación de actividad	36