**PROJECT CHARTER — TUTOR VIRTUAL DE LECTURA CRÍTICA**

1. **PROPÓSITO DEL PROYECTO**

Proveer a estudiantes, docentes y usuarios interesados una aplicación web full-stack (MERN) que funcione como un tutor virtual de lectura crítica, cuyo objetivo es fortalecer la comprensión lectora y el pensamiento crítico. Utilizando modelos de NLP/IA para generar preguntas automáticamente, detectar sesgos y falacias en los textos, y ofrecer retroalimentación inmediata y personalizada integrando automatizaciones con n8n para gestionar sesiones, enviar recordatorios y registrar el progreso del usuario, facilitando el seguimiento y la evaluación por parte del docente.

Fortalecer la comprensión lectora y el pensamiento crítico de estudiantes mediante una aplicación web MERN con IA (NLP) y automatización (n8n) que genere preguntas, detecte sesgos y brinde retroalimentación inmediata, reduciendo la carga docente y mejorando el seguimiento del progreso.

1. **DESCRIPCIÓN GENERAL DEL PROYECTO**

El proyecto consiste en el diseño, desarrollo y despliegue de una aplicación web full-stack utilizando el stack MERN (MongoDB, Express.js, React.js, Node.js), cuyo propósito es brindar un tutor virtual interactivo para fortalecer la lectura crítica en estudiantes y usuarios interesados.

Este proyecto incluye todo el trabajo necesario para:

* **Tipo de solución:** construcción de una aplicación web full-stack con arquitectura cliente-servidor moderna.
* **Funcionalidad central (MVP):**
* Lectura de textos dentro de la plataforma.
* Generación automática de preguntas mediante procesamiento de lenguaje natural (NLP).
* Detección de sesgos y falacias en los textos.
* Retroalimentación inmediata y registro de avances individuales.
* **Módulos complementarios:**
* Panel docente con métricas y reportes de progreso.
* Flujos de automatización con **n8n** para envío de lecturas, recordatorios y consolidación de resultados.
* Sistema de autenticación y gestión de usuarios con distintos roles (estudiantes, docentes, administradores).
* **Metodología:**
* Uso de Scrum con entregas incrementales.
* Gestión de versiones mediante Git Flow.
* Pruebas automatizadas (unitarias y end-to-end) para asegurar calidad.
* Contenerización con Docker para garantizar portabilidad y despliegue eficiente.

1. **REQUISITOS DE ALTO NIVEL**

**Funcionales**

* Generación de al menos 10 preguntas por lectura, combinando formatos de selección múltiple y preguntas abiertas cortas.
* Retroalimentación inmediata sobre las respuestas del estudiante.
* Clasificación de al menos 5 tipos de sesgos o falacias (ej. ad hominem, generalización apresurada, apelación a la autoridad, falacia de causa falsa, apelación a la emoción).
* Capacidad de asignar textos con actividades automáticas para evaluar el progreso de los alumnos.
* Visualización de un panel de métricas con el nivel de comprensión promedio de los usuarios.
* **Flujo n8n operativo**

1. asignación,
2. lectura,
3. actividad,
4. evaluación,
5. notificación.

* Como sistema: automatización del envío de textos diarios y registro de respuestas para crear hábitos de lectura crítica.
* Autenticación segura y diferenciación de roles (estudiante, docente, administrador).
* Almacenamiento y visualización de histórico de resultados y evolución individual.

**No funcionales**

* **Usabilidad y accesibilidad:**
* Interfaz responsive.
* Puntaje mínimo de 90 en accesibilidad (Lighthouse).
* **Calidad:**
* Pruebas unitarias y end-to-end (E2E).
* Cobertura mínima de **70%** en pruebas automatizadas.
* **Despliegue y portabilidad:**
* Contenerización con docker-compose para frontend, backend, base de datos y n8n.
* **Seguridad:**
* Manejo seguro de credenciales y variables de entorno.
* Validación de entradas en la API.
* Cifrado de contraseñas en la base de datos.
* **Escalabilidad:**
* Arquitectura diseñada para soportar un crecimiento progresivo de usuarios, con posibilidad de despliegue en la nube.
* **Mantenibilidad:**
* Código estructurado bajo buenas prácticas (Clean Code, Git Flow, documentación interna).
* **Cumplimiento ético y legal:**
* Protección de datos personales.
* Uso responsable de modelos de IA y transparencia en los resultados generados.

1. **OBJETIVOS DEL PROYECTO Y CRITERIOS DE ÉXITO**

**Objetivo General**

Implementar un tutor virtual interactivo basado en inteligencia artificial y automatización que fomente la lectura crítica en estudiantes, docentes y usuarios interesados.

**ObjetivoS Específicos**

* Diseñar una interfaz intuitiva y accesible con React.js, asegurando compatibilidad responsive y buenas prácticas de UX/UI.
* Implementar un backend con Node.js y Express que gestione usuarios, actividades, resultados y seguridad de la información.
* Incorporar dos funcionalidades de IA:
* Generación automática de preguntas a partir de textos (NLP con Hugging Face).
* Detección de sesgos y falacias lógicas en artículos.
* Integrar flujos de automatización con n8n para optimizar la gestión del aprendizaje (envío de lecturas, recordatorios y consolidación de resultados).
* Garantizar la calidad del sistema mediante pruebas automatizadas unitarias y E2E, alcanzando una cobertura mínima del 70%.
* Documentar el proyecto siguiendo buenas prácticas de ingeniería de software y sostenibilidad.

**Criterios de Éxito**

* **Entrega del MVP (Semana 8):**
  + Lectura de textos disponible.
  + Generación de preguntas con IA operativa.
  + Registro y visualización básica de resultados.
  + Éxito: demo funcional con un conjunto de lecturas y un tablero básico de progreso.
* **Integración de IA y n8n (Semanas 9–11):**
  + Funcionalidad de generación de preguntas y detección de sesgos funcionando.
  + Flujo de automatización en n8n probado en entorno de desarrollo.
  + Éxito: ambos módulos de IA operativos + al menos un flujo automatizado implementado.
* **Calidad y accesibilidad (Semanas 12–13):**
  + Cobertura de pruebas automatizadas ≥70%.
  + Lighthouse Accesibilidad ≥90.
  + Éxito: validación de calidad y usabilidad cumplida.
* **Entrega final y exposición (Semanas 15–16):**
  + Producto completo y estable.
  + Documentación técnica final (incluyendo impacto ambiental).
  + Presentación oral y video demostrativo.
  + Éxito: backlog crítico cumplido, sistema estable y exposición satisfactoria.

1. **RIESGOS DE ALTO NIVEL**

* **R1 (Técnico):** Baja precisión de los modelos NLP en generación de preguntas o detección de sesgos.
  + **Mitigación:** uso de datasets de prueba, ajuste de umbrales de confianza y validación manual inicial.
* **R2 (Integración):** Fricciones entre frontend, backend e integración con IA.
  + **Mitigación:** definición clara de contratos de API, uso de mocks para pruebas tempranas y pull requests pequeños y frecuentes.
* **R3 (Automatización):** Errores de orquestación en los flujos de n8n (ej. envíos duplicados o pérdida de datos).
  + **Mitigación:** comenzar con flujos simples, habilitar logs detallados, aplicar reintentos automáticos y validaciones en puntos críticos.
* **R4 (Equipo):** Participación desigual de los integrantes o retrasos en la entrega de tareas.
  + **Mitigación:** normas claras de colaboración, entregas parciales por sprint y rotación de tareas para balancear cargas.
* **R5 (Tiempo):** Acumulación de deuda técnica que retrase avances.
  + **Mitigación:** definición estricta de criterios de “hecho” (Definition of Done) incluyendo pruebas y linting desde el inicio.
* **R6 (Calidad de datos):** Textos mal estructurados o con ruido pueden afectar la precisión del análisis crítico.
  + **Mitigación:** preprocesamiento de textos, normalización y pruebas con diferentes fuentes.
* **R7 (Seguridad):** Manejo inadecuado de credenciales, datos de usuarios o fallas de autenticación.
  + **Mitigación:** uso de variables de entorno seguras, cifrado de contraseñas, validación de entradas y auditorías periódicas.
* **R8 (Escalabilidad):** Posible limitación de rendimiento al aumentar el número de usuarios o textos procesados.
  + **Mitigación:** contenerización con Docker, monitoreo de performance y escalado horizontal en caso de ser necesario.

1. **CRONOGRAMA RESUMIDO DE HITOS**

| **Semana** | **Actividades principales** |
| --- | --- |
| **1–2** | Elaboración del acta del proyecto, identificación de stakeholders, definición de arquitectura inicial (MERN + n8n) y construcción del backlog inicial. |
| **3–4** | Desarrollo del **backend base** (Node.js + Express: autenticación y CRUD de textos) y del **frontend base** (React: layout, rutas, estado global). |
| **5–7** | Implementación de la **IA #1** (generación automática de preguntas con NLP), desarrollo del tablero de progreso básico y pruebas iniciales. |
| **8** | **Entrega del MVP:** flujo completo lectura → preguntas IA → registro de resultados. |
| **9–11** | Desarrollo de la **IA #2** (detección de sesgos/falacias), mejoras de UX/UI y validación con pruebas de usuario. |
| **12–13** | Integración de flujos de **automatización en n8n** (recordatorios, registro de respuestas), ejecución de pruebas unitarias y E2E con cobertura ≥70%. |
| **14** | Configuración de **Docker** (docker-compose: frontend, backend, base de datos y n8n), pruebas de despliegue y hardening de seguridad. |
| **15** | Desarrollo de la **versión final**: estabilización del sistema, cierre de backlog crítico y documentación técnica completa. |
| **16** | **Exposición y entrega final:** presentación oral, demo estable y documentación entregada. |

1. **PRESUPUESTO RESUMIDO**

| **Recurso / Herramienta** | **Detalle / Uso** | **Costo aproximado (USD)** | **Notas** |
| --- | --- | --- | --- |
| **Licencia Windows 11 Pro** | SO para laptops/PC del equipo | $199 c/u (suponiendo 8 integrantes: $1,592) | Estudiantes pueden usar **Windows Education gratis** (DreamSpark/MS Imagine). |
| **MongoDB Atlas** | Base de datos en la nube | $0 (free tier 500 MB) → $57/mes (M10 cluster básico) | Free tier suficiente para MVP, pago solo si se escala. |
| **Hugging Face API** | NLP: generación de preguntas y detección de sesgos | $0 (free, límite 30k tokens/mes) → $9–99/mes según plan | Se puede usar **modelos locales gratis** con GPU (si hay recurso). |
| **n8n Cloud** | Orquestación y automatización | $20/mes plan básico | Alternativa: **self-host en Docker gratis**. |
| **Docker Desktop** | Contenerización y despliegue local | $0 (gratis para uso educativo/personal) | Empresas > 250 empleados pagan licencia. |
| **GitHub** | Control de versiones, CI/CD | $0 (GitHub Student Pack o repos privados free) | Incluye Actions para CI/CD. |
| **Jest + Cypress** | Pruebas unitarias y E2E | $0 (open source) | Gratis. |
| **VS Code** | IDE de desarrollo | $0 | Gratis. |
| **Figma** | Diseño UI/UX | $0 (educational/free) | Plan Pro $12/mes si necesitan features extra. |
| **Hosting / Dominio** | Despliegue final (ej. Vercel, Netlify, Render, Railway) + dominio “.com” | $12/año (dominio) + $0–$20/mes hosting | Hosting puede ser gratis con límites. |
| **Infra extra (opcional)** | AWS/GCP/Azure para pruebas con más GPU/CPU | $50–$100 créditos estudiantiles | GitHub Student Pack da $100 en Azure y Heroku. |
| **Materiales físicos** | Laptops personales, internet, bibliografía | $0 (ya disponibles) | Si no, laptop dev: $700–1000 c/u. |

1. **PRINCIPALES iNTERESADOS**

### **Internos**

* **Sponsor (Docente):** Responsable de validar la alineación del proyecto con los objetivos académicos.
* **Project Manager:** Encargado de la planificación, coordinación y seguimiento del cronograma.
* **Product Owner:** Define los requerimientos, prioriza el backlog y asegura el valor del producto.
* **Scrum Master:** Facilita la metodología ágil, remueve impedimentos y guía al equipo en buenas prácticas.
* **Equipo de Desarrollo:**
  + **Frontend:** Implementación de la interfaz en React.js (UI/UX).
  + **Backend:** Desarrollo de servicios y APIs con Node.js/Express.
  + **Base de Datos (BD):** Gestión y modelado en MongoDB.
  + **QA/Documentación:** Pruebas unitarias, E2E y elaboración de documentación técnica.
  + **DevOps:** Configuración de CI/CD, Docker y despliegues.

### **Externos**

* **Estudiantes (usuarios finales):** Beneficiarios directos; usan la app para practicar lectura crítica y análisis de textos.
* **Docentes (evaluadores):** Utilizan el sistema para asignar lecturas, monitorear el progreso y evaluar desempeño.
* **Instituciones educativas (clientes):** Escuelas o universidades interesadas en aplicar el sistema en sus cursos.
* **Proveedores tecnológicos:**
  1. Hugging Face (modelos NLP).
  2. MongoDB Atlas (base de datos en la nube).
  3. n8n (automatización de flujos).
  4. Docker (contenerización).
* **Comunidad académica:** Fuente de retroalimentación, validación de resultados y potenciales replicadores del proyecto.

1. **REQUISITOS DE APROBACIÓN DEL PROYECTO**

El proyecto será considerado **aprobado** cuando se cumplan los siguientes criterios:

* **Entregables funcionales:**
  + MVP entregado en la **Semana 8** con lectura, generación automática de preguntas (IA #1) y registro de resultados.
  + Versión final en la **Semana 15** con ambas funcionalidades de IA (preguntas y detección de sesgos) y al menos un flujo automatizado en **n8n** operativo.
* **Calidad técnica:**
  + Cobertura de pruebas (unitarias + E2E) igual o superior al **70%**.
  + Evaluación de accesibilidad con **Lighthouse ≥90**.
  + Correcta contenerización con **Docker** (docker-compose: FE, BE, BD y n8n).
* **Documentación:**
  + **README** con instrucciones claras de instalación y uso.
  + Diagramas de arquitectura, flujo de datos y automatización.
  + Informe técnico con buenas prácticas de ingeniería y **análisis del impacto ambiental**.
  + Guía de despliegue en Docker con pasos reproducibles.
* **Exposición y defensa:**
  + Presentación en la **Semana 16** con demo estable y funcional.
  + Capacidad del equipo para responder preguntas del jurado, mostrando dominio técnico y de gestión.

**GERENTE DE PROYECTO ASIGNADO, RESPONSABILIDAD Y NIVEL DE AUTORIDAD**

**Nombre:** Llacza Isidro Miguel:

* **Autoridad:** priorizar backlog junto con PO; gestionar cronograma y riesgos; validar pull requests críticos; coordinar releases.
* **No puede aprobar** cambios de alcance mayores sin visto bueno del **Sponsor**/**PO**.

**NOMBRE Y AUTORIDAD DEL PATROCINADOR**

**Sponsor:** Daniel Gamarra Moreno

**Autoridad:** aprobar hitos, cambios de alcance mayores y criterios de aceptación; arbitrar conflictos escalados por el PM.

| Gamarra Moreno Daniel  Patrocinador | Quispe Ubaldo Alfredo  Frontend Developer / DevOps |
| --- | --- |
| Llacza Isidro José Miguel  Project Manager | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Villena Ochoa Yorshyo  Backend Developer |
| Roman Briceño Jhonn Arturo  Product Owner | Contreras Bullón Arnaldo  Database Admin |
| Palacios Quincho Melany  Scrum Master | Castro Velarde Anyulieth / Iturrizaga Campeán Geraldine  QA/Tester & Documentación |

## 

## 