**PROJECT CHARTER**

**Título del proyecto:** x

| **Patrocinador del proyecto:** Daniel Gamarra Moreno | **Fecha de elaboración:** 25 de agosto de 2025 |
| --- | --- |
| **Gerente del proyecto:** Nilton Joel Lazo Maravi | **Cliente del proyecto:** Estudiantes |
| **Propósito del proyecto:** |  |

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

**Descripción general del proyecto:**

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

**Límites del proyecto:**

| * **Incluye:** Desarrollo del frontend y backend, base de datos, integración de IA para generación de preguntas y detección de sesgos, flujos automatizados con n8n, panel de seguimiento para administradores, pruebas automatizadas y documentación técnica. * **No incluye:** Generación de textos propios, tutoría presencial, soporte fuera de la plataforma, ni módulos de pago o monetización. |
| --- |

**Entregables principales:**

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

**Requerimientos de alto nivel:**

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

**Riesgos generales del proyecto:**

| 1. **Riesgo técnico:** errores en la integración de IA o baja precisión de los modelos. 2. **Riesgo de tiempo:** retrasos por curva de aprendizaje en IA y Docker. 3. **Riesgo operativo:** falta de adopción por parte de los estudiantes. 4. **Riesgo de calidad:** cobertura insuficiente en pruebas automatizadas. |
| --- |

| **Hitos del proyecto**   1. Semana 1-2: Definición de requisitos y diseño de arquitectura. 2. Semana 3-4: Desarrollo del backend y base de datos. 3. Semana 5-6: Desarrollo del frontend y diseño responsive. 4. Semana 7-8: Implementación de la primera funcionalidad de IA: generación de preguntas. 5. Semana 9-10: Implementación de la segunda funcionalidad de IA: clasificación de sesgos. 6. Semana 11-12: Automatización con 8n8 e integración de flujos. 7. Semana 13-14: Contenerización, documentación y entrega final | **Criterios de éxito**   1. Aplicación MERN funcional: El sistema debe estar desplegado y operativo en un entorno accesible para los usuarios. 2. Inteligencia Artificial implementada: Al menos dos funcionalidades de IA completadas:   **2.1.** Generación de preguntas con NLP.  **2.2.** Clasificación de sesgos en textos.   1. Automatización operativa: Flujos clave implementados en n8n: asignación → lectura → actividad → evaluación → notificación. 2. Pruebas y calidad asegurada: Cobertura mínima del 70% en pruebas unitarias e integrales. 3. Documentación completa: Incluir README, diagramas, manual técnico y reporte de impacto ambiental. 4. Entrega final: Presentación del proyecto con demo funcional, video explicativo y validación del patrocinador. |
| --- | --- |

**Alcance:**

| **Dentro del alcance:**   * Desarrollo de una aplicación web con el stack MERN. * Implementación de funcionalidades de IA: generación de preguntas y clasificación de sesgos. * Automatización de flujos clave con n8n. * Pruebas de calidad (unitarias, integrales y E2E). * Documentación técnica y presentación final del proyecto. | **Fuera del alcance:**   * Desarrollo de una aplicación móvil nativa. * Integración con plataformas LMS externas. * Soporte técnico presencial u offline. * Módulos de monetización o pasarelas de pago. |
| --- | --- |

**Tiempo:**

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

**Costos:**

| **Costos financieros directos:**  No aplican, ya que se utilizarán herramientas y tecnologías open-source.  **Inversión principal:**  Esfuerzo humano (horas de desarrollo, pruebas, documentación y presentación). | **Recursos:**   * Equipo de desarrollo: estudiantes asignados al proyecto. * Tiempo de dedicación: 14 semanas distribuidas en 7 sprints. * Infraestructura en la nube gratuita o con planes básicos (ej. MongoDB Atlas Free Tier, Hugging Face API con versión gratuita, n8n self-hosted). |
| --- | --- |

**Otros:**

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

| **Objetivos del proyecto** | **Fecha de cumplimiento** |
| --- | --- |
| Implementar una aplicación web MERN que funcione como tutor virtual. | Sprint 7 |
| Integrar al menos dos funcionalidades de IA (generación de preguntas y clasificación de sesgos). | Sprint 5 |
| Automatizar procesos clave con n8n. | Sprint 6 |
| Garantizar accesibilidad y usabilidad en la interfaz. | Sprint 3 |
| Cumplir con pruebas automatizadas y documentación técnica. | Sprint 7 |

**Recursos financieros preaprobados:**

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

| **Stakeholder(s)** | **Role** |
| --- | --- |
| Daniel Gamarra Moreno | Patrocinador del proyecto |
| Nilton Joel Lazo Maravi | Gerente del proyecto |
| Estudiantes | Clientes del proyecto |
| Usuarios de la plataforma | Usuarios finales |
| Equipo de desarrollo | Desarrolladores responsables de backend, frontend, IA y automatización |
| Docentes y expertos en lectura crítica | Colaboradores académicos para validar contenidos |

**Criterios de cierre del proyecto:**

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

**Nivel de autoridad de gerente del proyecto:**

Decisiones de personal:

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

Gestión del presupuesto y variaciones:

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

Decisiones técnicas:

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

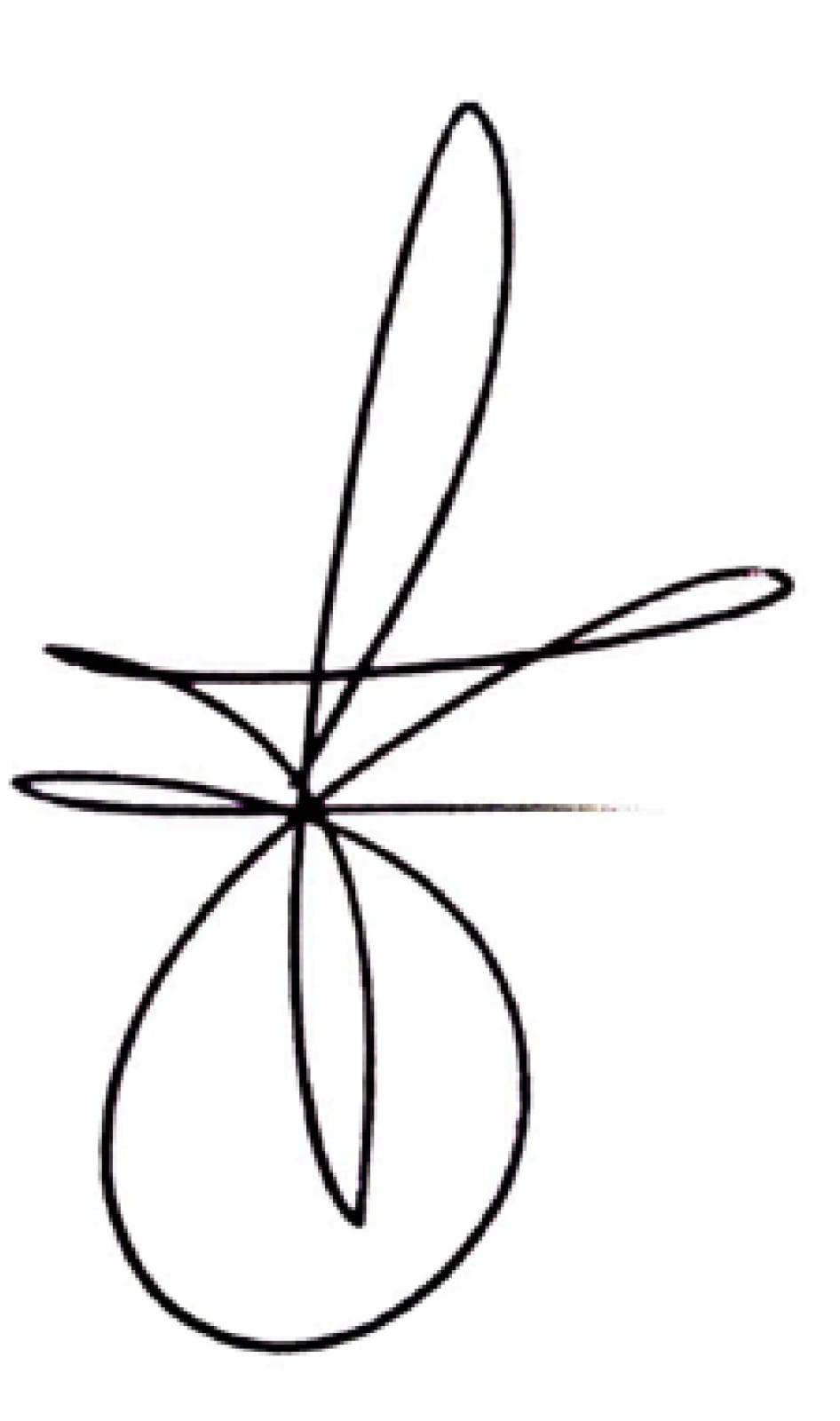
Resolución de conflictos:

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

Autoridad del patrocinador:

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

**Aprobaciones:**



|  |  | Daniel Gamarra Moreno |
| --- | --- | --- |

Firma del Gerente del Proyecto        Firma del Patrocinador o Solicitante

| Nilton Joel Lazo Maravi |  | Daniel Gamarra Moreno |
| --- | --- | --- |

Nombre del Gerente del Proyecto Nombre del Patrocinador o Solicitante

|  |  |  |
| --- | --- | --- |

| Fecha: 27 de Agosto de 2025 |  | Fecha: 27 de Agosto de 2025. |
| --- | --- | --- |