



FACULTAD DE INGENIERÍA

Escuela Académico Profesional de Ingeniería de Sistemas e Informática

Proyecto:

Documentos de Lanzamiento del Proyecto – *Revisor Automático de Escritura Académica*

Profesor:

Daniel Gamarra Moreno

PRESENTADO POR:

- Medina Vertiz Yerson Yassir
- Rivera Velazco Mauricio Gabriel

Huancayo – Perú

2025

2.1. Project Charter (Acta de Proyecto)

2.1.1. Propósito y Justificación del Proyecto

El Revisor Automático de Escritura Académica tiene como propósito principal mejorar la calidad de los trabajos académicos en instituciones educativas mediante el uso de Inteligencia Artificial y automatización.

La justificación se basa en:

- La creciente necesidad de agilizar la revisión académica, evitando los tiempos prolongados de corrección manual.
- La importancia de garantizar originalidad en los escritos mediante la detección de plagio.
- La necesidad de fortalecer el aprendizaje autónomo, ofreciendo retroalimentación inmediata y detallada a los estudiantes.
- El beneficio de proveer a docentes y administradores informes estandarizados y confiables.

El proyecto busca reemplazar revisiones manuales y correctores básicos por un sistema inteligente, accesible desde navegador y escalable.

2.1.2. Objetivos Generales

- Desarrollar un MVP en 12 semanas que brinde servicios de corrección automática y validación académica.
- Integrar NLP avanzado para revisión gramatical, coherencia y estilo en español e inglés.
- Implementar detección de plagio por similitud semántica y validación de citas en APA/IEEE.
- Automatizar el flujo de trabajo (carga → revisión → informe → notificación) con n8n.
- Generar reportes descargables en PDF, accesibles tanto para estudiantes como docentes.
- Lograr satisfacción $\geq 80\%$ en prueba piloto con estudiantes y docentes.

2.1.3. Alcance Inicial

Incluye en el MVP (fase 1):

- Módulo de registro/autenticación de usuarios.
- Carga de documentos en formatos .docx y .pdf.
- Revisión automática con IA (gramática, coherencia, estilo).
- Módulo de detección de plagio y validación de citas.
- Generación de reportes en PDF.
- Flujo automatizado con n8n (notificación, informes, estadísticas).

Excluye (para fases posteriores):

- Sugerencias avanzadas de estilo (más allá de corrección básica).
- Soporte para otros idiomas distintos al español e inglés.
- Integración con LMS (Moodle, Blackboard, etc.).

2.1.4. Supuestos y Restricciones

Supuestos:

- Los usuarios tendrán acceso estable a internet.
- Se contará con APIs de IA accesibles a bajo costo (OpenAI/HuggingFace).
- Los usuarios aceptarán compartir sus documentos bajo normativas de protección de datos.

Restricciones:

- Tiempo máximo: 12 semanas (del 09/09/2025 al 01/12/2025).
- Presupuesto máximo: USD 1,730.
- Cumplimiento obligatorio de GDPR y LOPD.
- Alcance limitado al MVP (gramática, plagio y citas).

2.1.5. Identificación de los Interesados (Stakeholders)

Interesado	Rol / Interés	Necesidad
Estudiantes	Usuarios principales	Retroalimentación inmediata y clara para mejorar sus trabajos.
Docentes	Validadores académicos	Informes confiables para corrección y detección de plagio.
Administradores académicos	Usuarios secundarios	Acceso a reportes y estadísticas de uso.
Equipo de desarrollo (Rivera y Medina)	Implementadores	Cumplir cronograma, calidad técnica y entregables.

2.1.6. Roles y Responsabilidades del Equipo

- Medina Vertiz – Gerente de Proyecto / Desarrollador Backend
 - Planificación y gestión del cronograma.
 - Integración de APIs de IA.
 - Automatización de procesos con n8n.
 - Despliegue técnico en la nube.
- Rivera Velazco – Patrocinador / Desarrollador Frontend
 - Validación de entregables y comunicación con stakeholders.
 - Diseño y desarrollo de interfaz de usuario.
 - Ejecución de pruebas piloto con estudiantes y docentes.
 - Documentación y soporte de usabilidad.

2.2.1. Identificación del Epic Central y Descomposición en Historias

Epic Central:

Como estudiante quiero cargar un documento académico en la plataforma, para recibir un análisis automático con correcciones gramaticales, detección de plagio y validación de citas, de modo que pueda mejorar la calidad de mi trabajo antes de entregarlo.

Descomposición en historias (ya alineadas con INVEST):

- Registro y autenticación de usuarios.
- Carga de documentos académicos.

- Corrección gramatical automática.
- Análisis de coherencia y estilo.
- Detección de plagio.
- Validación de citas académicas.
- Generación de reportes PDF.
- Notificación automática por correo.

2.2.2. Lista de Historias de Usuario Priorizadas con INVEST

Prioridad	Historia de Usuario	Criterios de Aceptación
1	H1 – Registro y autenticación: Como estudiante quiero registrarme e iniciar sesión en la plataforma para acceder a mis documentos y revisiones.	<ul style="list-style-type: none"> - Se puede registrar con nombre, correo y contraseña. - El login valida credenciales. - Se puede recuperar contraseña.
2	H2 – Carga de documentos: Como estudiante quiero cargar un documento en formato PDF/DOCX para que sea analizado.	<ul style="list-style-type: none"> - El sistema acepta .pdf y .docx. - El documento no supera 20 MB. - Se confirma la carga exitosa.
3	H3 – Corrección gramatical automática: Como estudiante quiero recibir sugerencias de corrección gramatical para mejorar la escritura.	<ul style="list-style-type: none"> - El sistema detecta errores gramaticales. - Propone correcciones con ejemplos.
4	H4 – Análisis de coherencia y estilo: Como estudiante quiero recibir un análisis de estilo y coherencia para mejorar la calidad de mi redacción.	<ul style="list-style-type: none"> - Se detectan redundancias y frases incoherentes. - Se generan métricas de legibilidad.
5	H5 – Detección de plagio: Como estudiante quiero conocer el porcentaje de plagio en mi documento para asegurar originalidad.	<ul style="list-style-type: none"> - Se muestra porcentaje total de plagio. - Se resaltan frases sospechosas.
6	H6 – Validación de citas: Como estudiante quiero que mis citas sean verificadas en formato APA/IEEE para cumplir con normas académicas.	<ul style="list-style-type: none"> - El sistema reconoce citas. - Marca citas mal formateadas. - Recomendación de correcciones.

7	H7 – Generación de reportes PDF: Como estudiante quiero descargar un reporte en PDF con los resultados de la revisión.	<ul style="list-style-type: none">- El sistema genera un PDF con correcciones.- El reporte incluye gramática, plagio y citas.
8	H8 – Notificación automática: Como estudiante quiero recibir una notificación cuando la revisión haya finalizado para acceder al informe.	<ul style="list-style-type: none">- Se envía correo de confirmación.- Se muestra notificación en el sistema.

2.3. Infraestructura y entorno técnico
C4 (nivel contenedores)

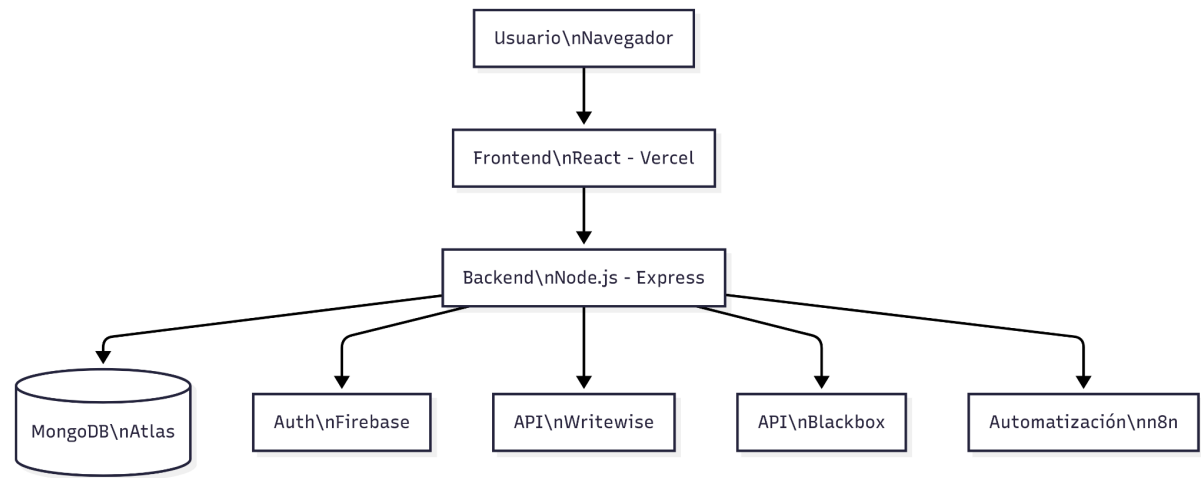
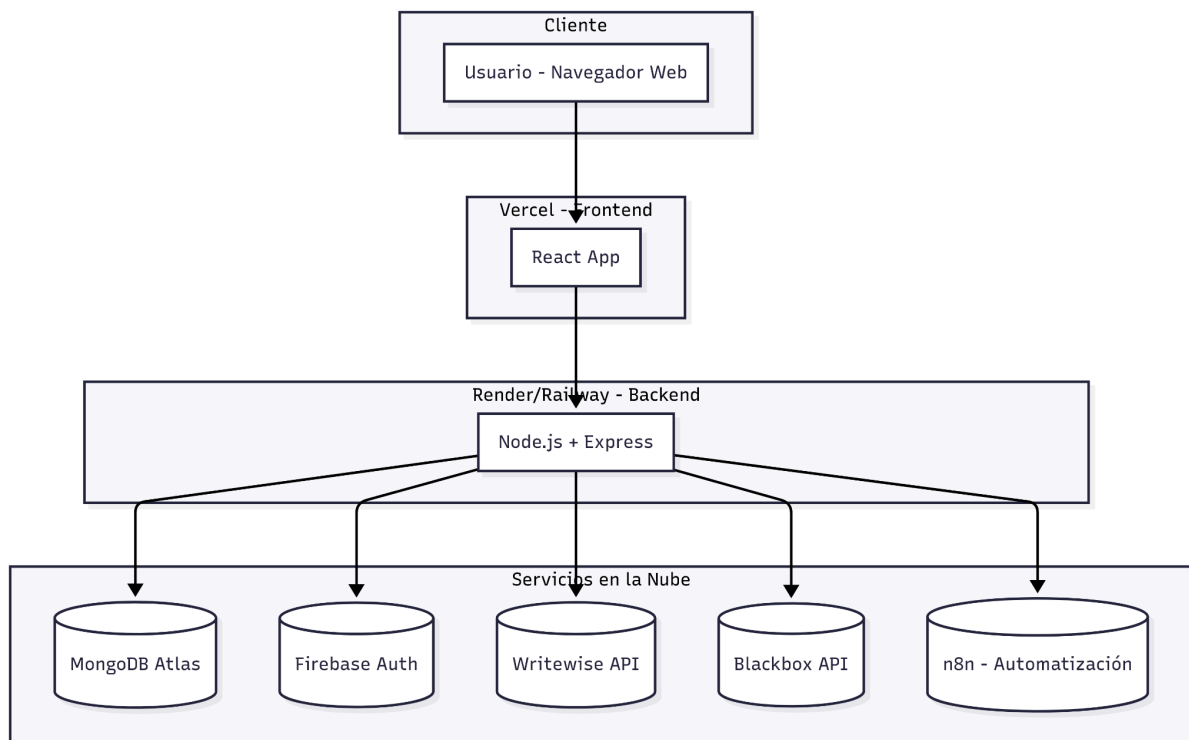


Diagrama de Despliegue



2.3.1. Creación de repositorio en GitHub .

G3007 – Proyecto MERN: Revisor Académico

Descripción del Proyecto

Revisor Académico es un proyecto web desarrollado con el stack MERN (MongoDB, Express, React, Node.js), orientado a estudiantes y docentes universitarios para mejorar la calidad de los escritos académicos.

Este repositorio contiene la **Iteración 0**, incluyendo documentación del proyecto, backlog inicial, acuerdos de equipo y el Walking Skeleton funcional (frontend + backend + conexión a base de datos).

Objetivos de la Iteración 0

- Diseñar y documentar la Iteración 0 de un proyecto web con MERN bajo un enfoque ágil.
- Elaborar el Project Charter como documento base del proyecto.
- Configurar la estructura inicial del proyecto, incluyendo repositorio, ramas y dependencias mínimas.
- Implementar un **Walking Skeleton** funcional: backend con Express + MongoDB y frontend con React.

g3007.vercel.app

Readme
Activity
0 stars
0 watching
0 forks
Report repository

Releases

No releases published

Packages

No packages published

Link: <https://github.com/EIMau201003/G3007.git>

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



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



2.4. Procesos de Equipo

2.4.1. Definición de Canales de Comunicación

Para garantizar una comunicación fluida y efectiva, se establecen los siguientes canales:

- Slack → canal oficial de comunicación del equipo. Usado para coordinación diaria, anuncios rápidos, compartir avances y resolver dudas.
- Google Meet → plataforma principal para las reuniones sincrónicas de Sprint Planning, Daily, Review y Retrospectiva.
- Correo institucional → medio formal para comunicación con stakeholders y envío de reportes oficiales.
- Jira → herramienta de gestión de tareas y seguimiento del backlog del producto y sprints.

2.4.2. Acuerdos de Trabajo

Horarios de reunión

- Daily Scrum: todos los días hábiles, 10 minutos a las 9:00 a.m. vía Slack/Meet.
- Sprint Planning: primer día de cada sprint, 1 hora.
- Sprint Review: último día de cada sprint, 1 hora con stakeholders invitados.
- Sprint Retrospective: inmediatamente después de la Review, 45 minutos.

Definition of Done (DoD)

Una historia de usuario se considera terminada cuando cumple con:

1. Código desarrollado, probado y versionado en repositorio Git.
2. Cumplimiento de criterios de aceptación definidos en el backlog.
3. Pruebas unitarias ejecutadas sin errores.
4. Documentación mínima actualizada (README, manual breve de uso o comentario técnico en Jira).
5. Validación en entorno de prueba por al menos un miembro del equipo distinto al desarrollador principal.

Roles iniciales

- Product Owner (PO): Rivera Velazco
 - Responsable de priorizar el backlog y representar los intereses de los estudiantes y docentes.

- Scrum Master: Medina Vertiz
 - Asegura el cumplimiento de la metodología Scrum, facilita reuniones y elimina impedimentos.
- Development Team: Rivera Velazco y Medina Vertiz (roles mixtos Backend/Frontend según la necesidad).

2.5. Arquitectura mínima (Walking Skeleton)

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

- Modelo físico:

1. usuarios

```
{
  "_id": ObjectId,
  "nombre": "Juan",
  "apellido": "Pérez",
  "correo": "juan@example.com",
  "rol": "estudiante", // estudiante | docente | admin
  "fecha_registro": ISODate("2025-09-09T20:00:00Z"),
  "auth_provider": "firebase",
  "firebase_uid": "uid_12345"
}
```

2. documentos

```
{
  "_id": ObjectId,
  "usuario_id": ObjectId, // referencia a usuarios
  "titulo": "Ensayo de Literatura",
  "archivo_url": "https://storage.vercel.app/docs/ensayo1.pdf",
  "estado": "pendiente", // pendiente | en_revision | finalizado
  "fecha_subida": ISODate("2025-09-09T20:10:00Z")
}
```

3. revisiones

```
{
  "_id": ObjectId,
  "documento_id": ObjectId, // referencia a documentos
  "fecha_inicio": ISODate("2025-09-09T20:15:00Z"),
  "fecha_fin": ISODate("2025-09-09T20:20:00Z"),
  "precision_gramatica": 92.5,
}
```

```

"similitud_plagio": 8.3,
"estado": "completada", // en_proceso | completada | fallida
"errores_gramaticales": [
  {
    "tipo": "Concordancia",
    "descripcion": "El verbo no concuerda con el sujeto",
    "sugerencia": "Cambiar a 'fueron' en lugar de 'fue'"
  }
],
"plagio": [
  {
    "fuente": "https://revistas.universidad.edu/articulo123",
    "porcentaje": 12.5,
    "fragmento": "El aprendizaje profundo ha demostrado..."
  }
],
"citass": [
  {
    "cita_texto": "(Smith, 2020)",
    "formato_detectado": "APA",
    "es_valido": true,
    "sugerencia": ""
  }
]
}

```

4. reportes

```

{
  "_id": ObjectId,
  "revision_id": ObjectId,
  "ruta_pdf": "https://storage.vercel.app/reports/rev123.pdf",
  "fecha_generacion": ISODate("2025-09-09T20:25:00Z")
}

```

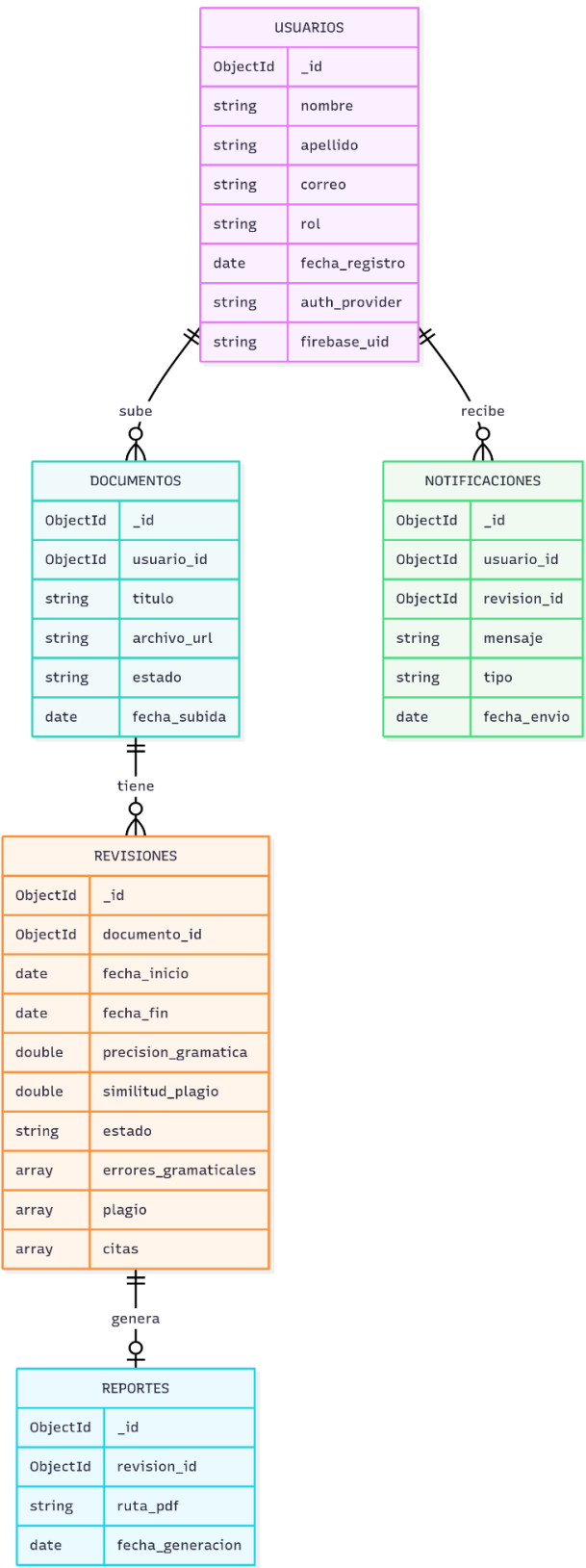
5. notificaciones

```

{
  "_id": ObjectId,
  "usuario_id": ObjectId,
  "revision_id": ObjectId,
  "mensaje": "Tu revisión ha finalizado",
  "tipo": "correo", // correo | sistema
  "fecha_envio": ISODate("2025-09-09T20:30:00Z")}

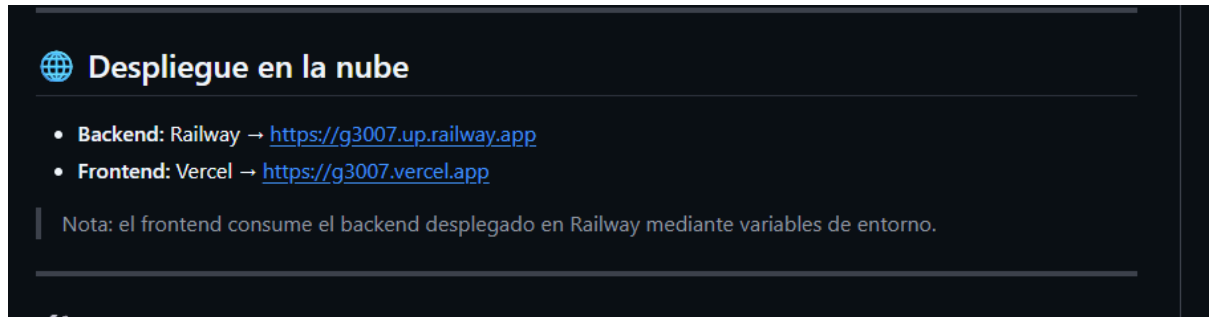
```

- Modelo lógico:

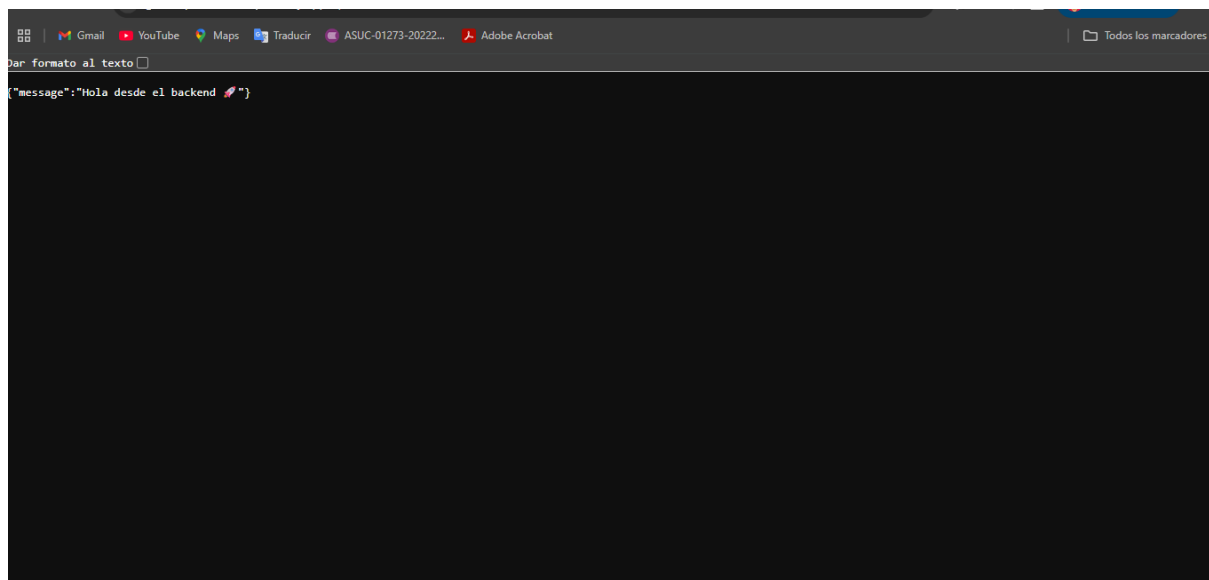


2.5.2. Evidencia de despliegue inicial en un servicio gratuito (ej. Render, Vercel, Railway).

Despliegue:



Render:



Vercel:



Revisor Académico

Conexión con backend:

Hola desde el backend 🚀