

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



**PROYECTO**

**“Implementación de un Sistema de Información para la Revisión Automática de Documentos Académicos mediante Inteligencia Artificial”**

**PRESENTADO POR:**

APELLIDOS Y NOMBRES	CÓDIGO
Camargo Ambicho Joussepe Josué	76943911
Lorenzo Masgo Josue Elí	74974962

**ASESOR:**

**Gamarra Moreno Daniel**

**HUANCAYO – PERÚ**

**2025**

## LISTA DE CONTENIDO

PORADA	1
LISTA DE CONTENIDO	2
LISTA DE TABLAS	7
LISTA DE FIGURAS	8
CAPÍTULO 1	9
PLANTEAMIENTO DEL ESTUDIO	9
1.1. Aspectos Generales de la Empresa	9
1.1.1. Organigrama	9
1.1.2. Misión y visión	10
1.2. Diagnóstico del Problema	10
1.3. Procesos de la Empresa	10
1.4. Oportunidad Encontrada	11
1.5. Detalles del Proyecto	11
CAPÍTULO 2	13
ESTUDIO DE FACTIBILIDAD	13
2.1. Alternativas de Solución	13
2.2. Factibilidad Técnica	13
2.2.1. Infraestructura tecnológica disponible	13
2.2.2. Conocimientos del equipo	13
2.2.3. Herramientas y software	13
2.2.4. Riesgos técnicos	13
2.3. Factibilidad Económica	13
2.3.1. Escenario sin gastos (Teórico)	14
2.3.2. Gastos generales	14
2.4. Factibilidad Operacional	15
2.4.1. Sistemas de soporte existentes	15
2.4.2. Recursos humanos	15
2.4.3. Procesos	15
2.4.4. Ventajas operacionales	15
CAPÍTULO 3	16

<b>ANÁLISIS DE REQUERIMIENTOS</b>	16
3.1. Metas del Sistema de Información	16
3.2. Requisitos del Sistema	16
3.2.1. Requerimientos funcionales	16
3.2.2. Requerimientos no funcionales	17
3.3. Identificación de Actores del Sistema	17
3.3.1. Estudiantes	17
3.3.2. Docentes	18
3.3.3. Administrador académico	18
3.3.4. Equipo de desarrollo	18
3.3.5. Proveedores de IA	18
3.3.6. Plataforma de automatización (n8n)	18
3.3.7. Área de TI / Soporte técnico	18
3.3.8. Dirección académica	18
<b>CAPÍTULO 4</b>	19
<b>PLANIFICACIÓN DEL PROYECTO</b>	19
4.1. Definición de Roles de Trabajo	19
4.1.1. Product owner	19
4.1.2. Scrum master	19
4.1.3. Team member	20
4.1.4. Tester	20
4.2. Product Backlog	21
4.3. Sprint Backlog	21
4.3.1. Sprint 1	22
4.3.2. Sprint 2	22
4.3.3. Sprint 3	23
4.3.4. Sprint 4	23
4.3.5. Sprint 5	24
4.3.6. Sprint 6	24
4.4. Planificación de Sprints	25
4.4.1. Historias de usuario	25
4.4.2. Priorización de historias de usuario	26

4.5. Cronograma de Actividades	28
4.6. Gestión de Riesgos	28
<b>CAPÍTULO 5</b>	<b>30</b>
<b>DISEÑO DEL SISTEMA DE INFORMACIÓN</b>	<b>30</b>
5.1. Diseño de Diagramas UML	30
5.1.1. Diagramas de casos de uso	30
5.1.2. Diagramas de secuencia	31
5.1.3. Diagramas de colaboración	32
5.1.4. Diagramas de clases	33
5.2. Diseño de Base de Datos	34
5.2.1. Diseño conceptual (E/R)	34
5.2.2. Diseño lógico	35
5.2.3. Diseño físico	35
5.2.4. Modelado de base de datos	36
5.3. Diseño de Interfaces Básicas	37
5.3.1. Acceso login	38
5.3.2. Interfaz de visualización de documentos	38
<b>CAPÍTULO 6</b>	<b>40</b>
<b>CODIFICACIÓN DEL SOFTWARE</b>	<b>40</b>
6.1. Desarrollo del Sprint 1	40
6.1.1. Sprint planning	40
Historias seleccionadas:	40
6.1.2. Sprint backlog	40
6.1.3. Historias de usuarios	40
HU01 – Detección de errores gramaticales (Sprint 1)	40
HU02 – Coherencia y estilo (Sprint 1)	41
HU04 – Registro y acceso (Sprint 1)	41
6.1.4. Taskboard	41
6.1.5. Daily scrum	42
6.1.6. Sprint review	42
6.1.7. Criterios de aceptación	42
6.1.8. Resultados del sprint	42

6.1.8.1. Evidencias.	42
6.1.8.2. Prueba de desarrollo.	43
6.1.9. Sprint retrospective	43
6.2. Desarrollo del Sprint 2	44
6.2.1. Sprint planning	44
6.2.2. Sprint backlog	44
6.2.3. Historias de usuarios	44
HU03 – Informe descargable (Sprint 2)	44
HU06 – Notificación a docentes (Sprint 2)	45
HU11 – Flujo automático (Sprint 2)	45
6.2.4. Taskboard	45
6.2.5. Daily scrum	45
6.2.6. Sprint review	46
6.2.7. Criterios de aceptación	46
6.2.8. Resultados del sprint	46
6.2.8.1. Evidencias.	46
6.2.8.2. Prueba de desarrollo.	46
6.2.9. Sprint retrospective	47
6.3. Desarrollo del Sprint 3	47
6.3.1. Sprint planning	47
Historias seleccionadas	47
6.3.2. Sprint backlog	47
6.3.3. Historias de usuarios	48
HU-05 – Panel del Docente	48
HU-12 – Generación automática del informe	48
HU-13 – Archivado automático de informes	48
6.3.4. Taskboard	49
6.3.5. Daily scrum	49
6.3.6. Sprint review	49
6.3.7. Criterios de aceptación	49
6.3.8. Resultados del sprint	50
6.3.8.1. Evidencias.	50

6.3.8.2. Prueba de desarrollo.	50
6.3.9. Sprint retrospective	50
Aspectos positivos	50
Aspectos a mejorar	50
6.4. Desarrollo del Sprint 4	50
6.4.1. Sprint planning	50
6.4.2. Sprint backlog	51
6.4.3. Historias de usuarios	51
HU-07 – Detección de similitudes con otros documentos	51
HU-08 – Validación de referencias APA	51
6.4.4. Taskboard	52
6.4.5. Daily scrum	52
6.4.6. Sprint review	52
6.4.7. Criterios de aceptación	53
6.4.8. Resultados del sprint	53
6.4.8.1. Evidencias.	53
6.4.8.2. Prueba de desarrollo.	54
6.4.9. Sprint retrospective	54
6.5. Desarrollo del Sprint 5	54
6.5.1. Sprint planning	54
6.5.2. Sprint backlog	54
6.5.3. Historias de usuarios	55
HU-09 – Validación IEEE	55
HU-10 – Citas faltantes	55
6.5.4. Taskboard	55
6.5.5. Daily scrum	56
6.5.6. Sprint review	56
6.5.7. Criterios de aceptación	56
6.5.8. Resultados del sprint	56
6.5.8.1. Evidencias.	56
6.5.8.2. Prueba de desarrollo.	57
6.5.9. Sprint retrospective	57

6.6. Desarrollo del Sprint 6	57
6.6.1. Sprint planning	57
6.6.2. Sprint backlog	58
6.6.3. Historias de usuarios	58
HU-14 – Estadísticas de errores	58
HU-15 – Filtros avanzados	58
HU-16 – Exportación PDF/Excel	58
6.6.4. Taskboard	59
6.6.5. Daily scrum	59
6.6.6. Sprint review	59
6.6.7. Criterios de aceptación	59
6.6.8. Resultados del sprint	59
6.6.8.1. Evidencias.	59
6.6.8.2. Prueba de desarrollo.	59
6.6.9. Sprint retrospective	60
CAPÍTULO 7	61
PRUEBAS DE SOFTWARE	61
7.1. Plan de Pruebas	61
7.1.1. Objetivos de las pruebas	61
7.1.2. Tipos de Pruebas Aplicadas	62
7.2. Pruebas Unitarias (Jest)	62
7.3. Pruebas End-to-End (Cypress)	63
7.4. Pruebas de API (Postman)	67
7.5. Pruebas de Rendimiento (Lighthouse)	69
7.6. Pruebas de Aceptación	70
Criterios validados	70
CONCLUSIONES	71
RECOMENDACIONES	72
ANEXOS	74
Anexo 01. Manual Técnico	74
Anexo 02. Manual de Usuario	75

## LISTA DE TABLAS

**Tabla 1.** Costos aproximados de herramientas e infraestructura

**Tabla 2.** Funcionalidades probadas en pruebas unitarias (Jest)

**Tabla 3.** Casos de prueba end-to-end (Cypress)

**Tabla 4.** Endpoints probados mediante Postman

**Tabla 5.** Resultados Lighthouse de rendimiento

## LISTA DE FIGURAS

**Figura 1.** Organigrama de la Universidad Continental

**Figura 2.** Estado de historias de usuario – Sprint 1 (JIRA)

**Figura 3.** Interfaz de inicio de sesión (Login del sistema)

**Figura 4.** Evidencia de prueba: Register

**Figura 5.** Evidencia de prueba: Login

**Figura 6.** Flujo automatizado en N8N – Sprint 2

**Figura 7.** Estado de HU en Sprint 2 (JIRA)

**Figura 8.** Captura de Request en Postman

**Figura 9.** Evidencias Sprint 3 – Estado de HU en JIRA

**Figura 10.** Panel del docente – Vista inicial

**Figura 11.** Nodo N8N para envío automático de correo

**Figura 12.** Nodo N8N para backup automático

**Figura 13.** Vista individual de documento – Prueba Cypress

**Figura 14.** Análisis con IA – Prueba Cypress

**Figura 15.** Revisión de citas IEEE – Sprint 5

**Figura 16.** Evidencias Sprint 6 – Estado en JIRA

# CAPÍTULO 1

## PLANTEAMIENTO DEL ESTUDIO

### 1.1. Aspectos Generales de la Empresa

**Empresa:** Universidad Continental

**Rubro:** Educación Superior / Academia

#### 1.1.1. Organigrama

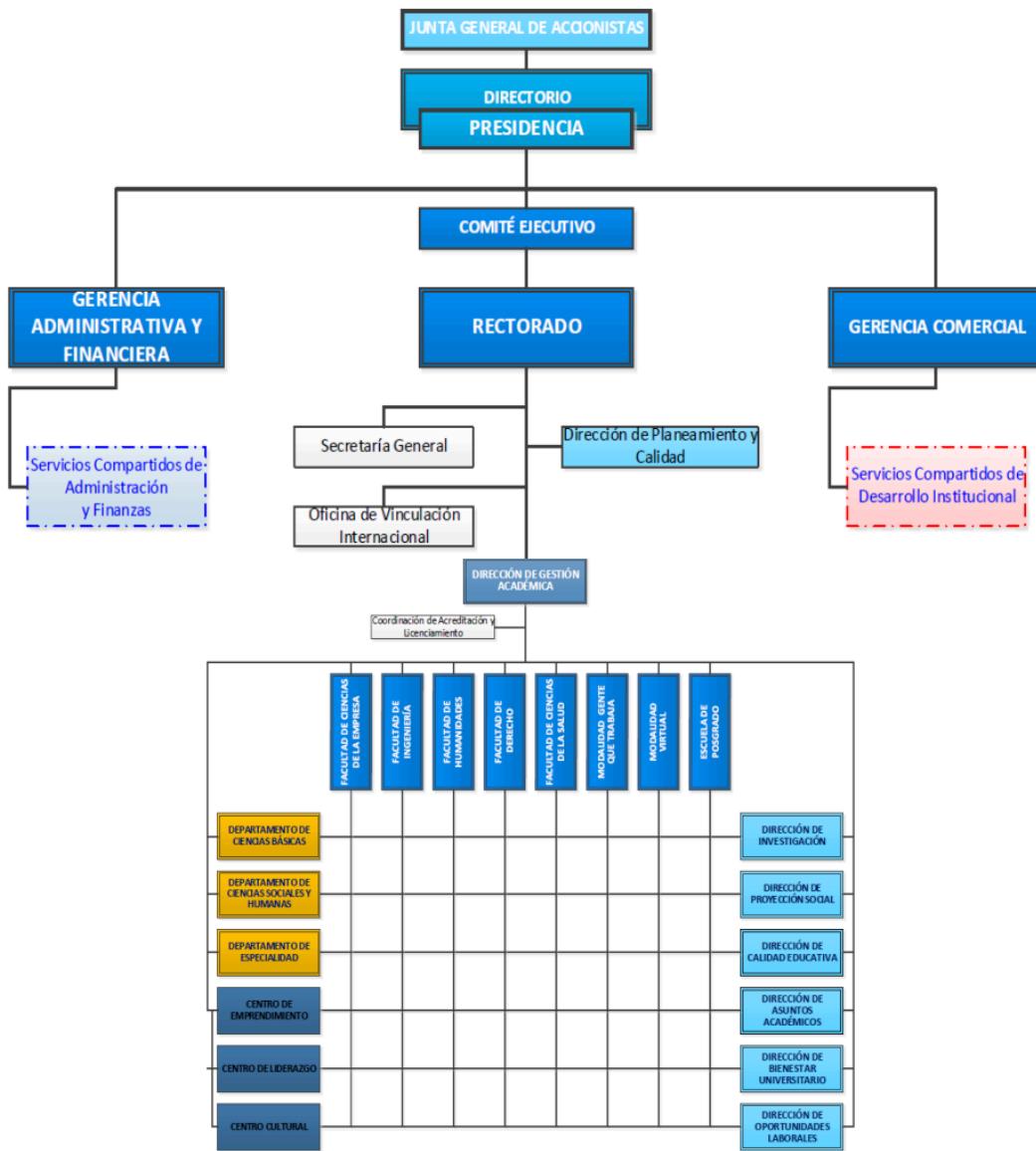


Figura 1. Organigrama de la Universidad Continental

### **1.1.2. Misión y visión**

Misión: Ser una organización de educación superior que impulsa y acompaña a las personas en la realización de su propósito de vida a través de una oferta educativa de calidad global, accesible y pertinente, que genera impacto positivo a lo largo de sus vidas.

Visión: Ser la mejor organización de educación superior posible que brinda oportunidades para el aprendizaje, la investigación y que une personas e ideas que buscan hacer realidad sueños y aspiraciones de prosperidad en un entorno incierto.

### **1.2. Diagnóstico del Problema**

Actualmente, los procesos de revisión académica en la Universidad Continental presentan los siguientes desafíos:

- Revisión manual que consume mucho tiempo de docentes y estudiantes.
- Errores frecuentes en gramática, coherencia, estilo y citación.
- Falta de uniformidad y consistencia en los informes de revisión.
- Limitaciones en la trazabilidad de cambios y retroalimentación de los documentos.

Estos problemas generan retrasos, aumento de carga laboral para docentes y estudiantes, y afectan la calidad académica.

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

La Universidad Continental cuenta con procesos organizativos y académicos que garantizan el funcionamiento eficiente de la institución. Entre los principales procesos se incluyen:

#### **1. Procesos académicos:**

- Planificación y programación de cursos y asignaturas.

- Gestión de matrículas y registro académico de estudiantes.
- Evaluación y seguimiento del desempeño académico.
- Gestión de docentes y asignación de responsabilidades académicas.

## **2. Procesos administrativos:**

- Gestión financiera, presupuestos y pagos.
- Administración de recursos humanos y contratación de personal.
- Control de infraestructura y mantenimiento de instalaciones.
- Gestión de compras y adquisiciones de materiales y servicios.

## **3. Procesos de soporte tecnológico:**

- Mantenimiento de sistemas de información institucionales.
- Soporte a usuarios (docentes, estudiantes y administrativos).
- Gestión de la plataforma de e-learning y herramientas digitales.

## **4. Procesos de aseguramiento de la calidad:**

- Monitoreo de cumplimiento de normas académicas y legales.
- Evaluación y acreditación de programas educativos.
- Gestión de auditorías internas y externas para garantizar estándares institucionales.

### **1.4. Oportunidad Encontrada**

Existe la oportunidad de automatizar y optimizar el proceso de revisión académica mediante una plataforma inteligente que reduzca errores, agilice tiempos de revisión y proporcione retroalimentación estandarizada a estudiantes y docentes.

### **1.5. Detalles del Proyecto**

**Nombre del Proyecto:** ReviDoc

**Breve Descripción:**

ReviDoc es una plataforma web full-stack con inteligencia artificial que permite a estudiantes y docentes cargar documentos académicos para recibir correcciones

automáticas de gramática, coherencia, estilo, citaciones y plagio, generando informes automáticos.

**Alcance Preliminar:**

- Incluye: Desarrollo de plataforma web, integración de IA, flujos automatizados, metodologías ágiles.
- No incluye: Sistemas de calificación docente ni integraciones externas fuera del entorno universitario.

**Resultados Esperados / Beneficios:**

- Plataforma funcional y confiable.
- Reducción de errores en forma y citación.
- Optimización del tiempo de revisión.
- Mejora de la calidad académica y transparencia.
- Fortalecimiento de buenas prácticas de investigación.

## CAPÍTULO 2

### ESTUDIO DE FACTIBILIDAD

#### **2.1. Alternativas de Solución**

Se identifican las siguientes alternativas para implementar el proyecto ReviDoc:

- Desarrollo interno completo: El equipo de la universidad desarrolla toda la plataforma desde cero utilizando el stack MERN y herramientas de creación de IA.
- Uso de soluciones comerciales existentes: Contratar software de revisión académica disponible en el mercado y adaptarlo a la universidad.
- Desarrollo híbrido: Combinar componentes internos con APIs externas (IA, automatización) para reducir tiempo de desarrollo y costos.

#### **2.2. Factibilidad Técnica**

Se evaluó la capacidad técnica de la universidad para implementar ReviDoc:

##### **2.2.1. Infraestructura tecnológica disponible**

Servidores de prueba, acceso a nube (MongoDB Atlas), conectividad de alta velocidad.

##### **2.2.2. Conocimientos del equipo**

Especialistas en MERN, IA y flujos automatizados con n8n.

##### **2.2.3. Herramientas y software**

Node.js, React, Express, MongoDB, Docker, n8n, APIs de IA.

##### **2.2.4. Riesgos técnicos**

Complejidad de integración de IA y automatización, disponibilidad de APIs externas.

#### **2.3. Factibilidad Económica**

Se analizan los gastos generales del proyecto en dos escenarios

### 2.3.1. Escenario sin gastos (Teórico)

Categoría	Categoría	Monto total (S/)	Mensual (S/)
Servicios en la nube	Dominio, MongoDB Atlas, servidor	0	0
APIs de IA	Consultas masivas (~1000/día)	0	0
Herramientas de automatización	n8n	0	0
Infraestructura y contenerización	Docker, entornos de prueba	0	0
Software y utilitarios	Gemini Free, Trello, librerías premium	0	0
Documentación y presentación	Informe técnico, video demo	0	0
Total		0	0

### 2.3.2. Gastos generales

Categoría	Categoría	Monto total (S/)	Mensual (S/)
Servicios en la nube	Dominio, MongoDB Atlas, servidor	1200	300
APIs de IA	Consultas masivas (~1000/día con GPT-4o mini)	600 - 1800	150 - 450
Herramientas de automatización	n8n (suscripción/servidor)	500	125
Infraestructura y contenerización	Docker, entornos de prueba y despliegue	400	100
Software y utilitarios	Gemini Pro, Trello, librerías premium	400	100
Documentación y presentación	Informe técnico, video demo	0	125

Total	3100 - 4500	900 - 1250
-------	-------------	------------

## **2.4. Factibilidad Operacional**

Se evaluó la capacidad de la universidad para integrar ReviDoc en su operación diaria:

### **2.4.1. Sistemas de soporte existentes**

Plataformas de e-learning, repositorios académicos y correo institucional.

### **2.4.2. Recursos humanos**

Docentes, estudiantes y personal de TI capacitados para interactuar con la plataforma.

### **2.4.3. Procesos**

La plataforma se integrará al flujo académico actual sin afectar los sistemas de ventas ni procesos administrativos existentes.

### **2.4.4. Ventajas operacionales**

Reducción de carga manual, retroalimentación más rápida y estandarización de revisiones académicas.

## CAPÍTULO 3

### ANÁLISIS DE REQUERIMIENTOS

#### **3.1. Metas del Sistema de Información**

El sistema ReviDoc busca cumplir con las siguientes metas:

- Automatizar la revisión académica de documentos, incluyendo gramática, coherencia, estilo, citaciones y detección de plagio.
- Reducir el tiempo que estudiantes y docentes invierten en la revisión de trabajos académicos.
- Garantizar la consistencia y calidad de las revisiones académicas.
- Generar informes automáticos de retroalimentación para estudiantes y docentes.
- Integrar flujos de automatización que optimicen la gestión de documentos y notificaciones.
- Facilitar la escalabilidad y sostenibilidad del sistema mediante contenerización y metodologías ágiles.

#### **3.2. Requisitos del Sistema**

##### **3.2.1. Requerimientos funcionales**

ID	Requerimiento Funcional	Descripción
RF1	Registro y autenticación de usuarios	Permitir que estudiantes y docentes creen cuentas y accedan al sistema mediante login seguro.
RF2	Carga de documentos	Permitir a los usuarios subir documentos en formatos Word y PDF.
RF3	Revisión automática	Analizar documentos con IA para corrección de gramática, coherencia y estilo.
RF4	Verificación de citaciones	Comprobar referencias según normas APA, IEEE u otras.
RF5	Detección de plagio	Identificar contenido duplicado o no original dentro del documento.
RF6	Generación de informes	Crear reportes automáticos con

		retroalimentación detallada.
RF7	Notificaciones	Informar a los usuarios sobre el estado de la revisión de sus documentos.
RF8	Gestión de usuarios y roles	Administrar permisos y accesos para estudiantes, docentes y administradores.

### 3.2.2. Requerimientos no funcionales

ID	Requerimiento No Funcional	Descripción
RNF1	Rendimiento	El sistema debe procesar revisiones en menos de 2 minutos por documento promedio.
RNF2	Escalabilidad	La plataforma debe soportar crecimiento en número de usuarios y documentos sin degradar el rendimiento..
RNF3	Disponibilidad	Debe garantizarse un 99% de uptime durante horarios académicos.
RNF4	Seguridad	Protección de datos personales y académicos según GDPR/LPDP.
RNF5	Usabilidad	Interfaz intuitiva y accesible desde PC, tablets y móviles.
RNF6	Mantenibilidad	Código modular y documentado para facilitar futuras mejoras y actualizaciones.
RNF7	Automatización	Integración de flujos automatizados con n8n para gestión de documentos y notificaciones.

## 3.3. Identificación de Actores del Sistema

### 3.3.1. Estudiantes

Son los usuarios finales que cargan sus documentos académicos al sistema y reciben retroalimentación automática sobre gramática, estilo, coherencia y citaciones.

### **3.3.2. Docentes**

Actúan como usuarios clave, supervisando las revisiones realizadas por la plataforma, validando los informes generados y proporcionando retroalimentación adicional cuando sea necesario.

### **3.3.3. Administrador académico**

Es el supervisor del sistema, responsable de gestionar los permisos de usuarios, configurar el sistema y asegurar el correcto funcionamiento de la plataforma dentro del entorno universitario.

### **3.3.4. Equipo de desarrollo**

Se encarga de implementar, mantener y mejorar la plataforma, asegurando que cumpla con los requerimientos funcionales y no funcionales.

### **3.3.5. Proveedores de IA**

Servicios externos que proveen las funcionalidades de inteligencia artificial para corrección gramatical, verificación de coherencia y detección de plagio.

### **3.3.6. Plataforma de automatización (n8n)**

Actor externo encargado de gestionar los flujos automáticos del sistema, incluyendo la carga de documentos, la ejecución de revisiones y la generación de notificaciones e informes.

### **3.3.7. Área de TI / Soporte técnico**

Responsable de mantener la infraestructura tecnológica, resolver incidencias y asegurar la disponibilidad del sistema.

### **3.3.8. Dirección académica**

Alta dirección que supervisa el impacto académico del proyecto y valida los resultados obtenidos por la plataforma.

## CAPÍTULO 4

### PLANIFICACIÓN DEL PROYECTO

#### **4.1. Definición de Roles de Trabajo**

En el proyecto ReviDoc se asignaron roles claros para garantizar la correcta ejecución bajo la metodología ágil (Scrum). Cada rol tiene responsabilidades específicas:

##### **4.1.1. Product owner**

El Product Owner (PO) es responsable de definir y priorizar los requerimientos del producto, asegurando que el equipo de desarrollo entregue valor de manera continua. Entre sus funciones destacan:

- Definir la visión del producto y comunicarla al equipo.
- Priorizar el Product Backlog y establecer las historias de usuario más importantes.
- Asegurar que el producto final cumpla con las expectativas de los usuarios y stakeholders.
- Tomar decisiones sobre cambios en el alcance durante el desarrollo.

En este proyecto, los Product Owners son: Camargo Ambicho Joussepe Josué y Lorenzo Masgo Josue Eli.

##### **4.1.2. Scrum master**

El Scrum Master (SM) se encarga de facilitar el proceso ágil y eliminar obstáculos que puedan afectar el rendimiento del equipo. Sus principales responsabilidades incluyen:

- Asegurar que se sigan las prácticas de Scrum correctamente.
- Facilitar las ceremonias ágiles (Daily Scrum, Sprint Planning, Sprint Review, Sprint Retrospective).
- Apoyar al equipo en la resolución de problemas y conflictos.
- Servir como enlace entre el equipo de desarrollo y los stakeholders.

En este proyecto, los Scrum Masters son: Camargo Ambicho Joussepe Josué y Lorenzo Masgo Josue Eli.

#### **4.1.3. Team member**

Los Team Members son los integrantes del equipo de desarrollo que implementan las funcionalidades del sistema. Sus responsabilidades incluyen:

- Desarrollar el backend y frontend del sistema según las historias de usuario.
- Realizar pruebas unitarias y de integración.
- Participar activamente en reuniones ágiles y retroalimentaciones.
- Documentar el código y colaborar en la mejora continua del sistema.

En este proyecto, los Team Members son: Camargo Ambicho Joussepe Josué y Lorenzo Masgo Josue Eli..

#### **4.1.4. Tester**

El Tester es responsable de garantizar la calidad del sistema, verificando que todas las funcionalidades cumplan con los requisitos. Sus funciones incluyen:

- Diseñar y ejecutar pruebas funcionales, unitarias y de extremo a extremo (E2E).
- Detectar errores y reportarlos al equipo de desarrollo.
- Validar que el sistema cumpla con los requerimientos no funcionales (rendimiento, usabilidad, seguridad).
- Participar en la planificación de Sprints para asegurar la calidad desde el inicio.

En este proyecto ReviDoc, el rol de Tester es desempeñado por miembros el cual son: Camargo Ambicho Joussepe Josué y Lorenzo Masgo Josue Eli.

## 4.2. Product Backlog

ID	DESCRIPCIÓN	ÉPICA	PRIORIDAD	COMPLEJIDAD
HU-01	Gramática	EP - 01	1	5
HU-02	Coherencia	EP - 01	1	8
HU-03	Informe	EP - 01	1	5
HU-04	Registro usuarios	EP - 02	2	3
HU-05	Notificaciones básica	EP - 02	2	5
HU-06	Flujo automático básico	EP - 02	2	5
HU-07	Detección de plagio	EP - 03	3	8
HU-08	Citas APA	EP - 03	3	5
HU-09	Citas IEEE	EP - 03	3	5
HU-10	Errores de citación	EP - 03	3	8
HU-11	Envío de informes	EP - 04	4	5
HU-12	Panel docente	EP - 04	4	5
HU-13	Archivo de documentos revisados	EP - 04	4	5
HU-14	Estadísticas de errores	EP - 05	5	5
HU-15	Filtro de estadísticas	EP - 05	5	5
HU-16	Exportación de estadísticas	EP - 05	5	5

## 4.3. Sprint Backlog

Una vez definido el Product Backlog, la metodología SCRUM establece la necesidad de organizar el Sprint Backlog, el cual consiste en una planificación específica de los requerimientos que serán desarrollados en cada iteración (Sprint). Esta selección se realiza en la Sprint Planning Meeting, con la participación del Product Owner, Scrum Master y el Scrum Team. En este proyecto se han planificado cuatro sprints principales:

#### **4.3.1. Sprint 1**

La primera iteración tiene una duración de 20 días. El equipo de desarrollo realizará reuniones diarias de seguimiento de 20 minutos, para comunicar los avances en las tareas asignadas.

El objetivo de este Sprint es implementar las funcionalidades básicas para la gramática, coherencia y registro de usuarios.

Nro.	Requisitos	Descripción
HU-01	Gramática	El sistema detecta errores gramaticales.
HU-02	Coherencia	El sistema detecta problemas de coherencia y estilo
HU-04	Registro de usuarios	El usuario se registra y acceder con credenciales

#### **4.3.2. Sprint 2**

La primera iteración tiene una duración de 20 días. El equipo de desarrollo realizará reuniones diarias de seguimiento de 20 minutos, para comunicar los avances en las tareas asignadas.

El objetivo de este Sprint es implementar las funcionalidades básicas para el Informe, flujo automático básico y envío de informes.

Nro.	Requisitos	Descripción
HU-03	Informe	El sistema genera un informe descargable con las sugerencias
HU-06	Flujo automático básico	El sistema al subir un documento, activa automáticamente el flujo de revisión
HU-11	Envío de informes	El sistema genera automáticamente el informe y lo

		envíe al estudiante
--	--	---------------------

#### 4.3.3. Sprint 3

La primera iteración tiene una duración de 20 días. El equipo de desarrollo realizará reuniones diarias de seguimiento de 20 minutos, para comunicar los avances en las tareas asignadas.

El objetivo de este Sprint es implementar las funcionalidades básicas para las notificaciones básicas, panel docente y archivo de documentos revisados.

Nro.	Requisitos	Descripción
HU-05	Notificaciones básica	El docente acceder a un panel para ver los trabajos de los estudiantes
HU-12	Panel docente	El docente accede al apartado del panel a ver los resultados de trabajos de los estudiantes
HU-13	Archivo de documentos revisados	El sistema archiva los informes y documentos revisados

#### 4.3.4. Sprint 4

La primera iteración tiene una duración de 20 días. El equipo de desarrollo realizará reuniones diarias de seguimiento de 20 minutos, para comunicar los avances en las tareas asignadas.

El objetivo de este Sprint es implementar las funcionalidades básicas para la detección de plagio y citas APA.

Nro.	Requisitos	Descripción
HU-07	Detección de plagio	El sistema detecte similitudes con otros documentos
HU-08	Citas APA	El sistema valide las referencias

		en formato APA
--	--	----------------

#### 4.3.5. Sprint 5

La primera iteración tiene una duración de 20 días. El equipo de desarrollo realizará reuniones diarias de seguimiento de 20 minutos, para comunicar los avances en las tareas asignadas.

El objetivo de este Sprint es implementar las funcionalidades básicas para la gramática, coherencia y registro de usuarios.

Nro.	Requisitos	Descripción
HU-09	Citas IEEE	El sistema valide las referencias en formato IEEE
HU-10	Errores de citación	El sistema marque las citas faltantes o mal referenciadas

#### 4.3.6. Sprint 6

La primera iteración tiene una duración de 20 días. El equipo de desarrollo realizará reuniones diarias de seguimiento de 20 minutos, para comunicar los avances en las tareas asignadas.

El objetivo de este Sprint es implementar las funcionalidades básicas para la gramática, coherencia y registro de usuarios.

Nro.	Requisitos	Descripción
HU-14	Estadísticas de errores	El sistema mostrará estadísticas de los errores más comunes.
HU-15	Filtro de estadísticas	El sistema filtra estadísticas por curso, carrera o fecha.
HU-16	Exportación de estadísticas	El sistema exporta estadísticas en PDF/Excel

## 4.4. Planificación de Sprints

Según la Metodología SCRUM, el equipo de trabajo debe definir y plasmar las historias de usuario, con sus respectivas tareas técnicas y criterios de aceptación. Esta planificación permite gestionar el desarrollo del sistema de forma iterativa e incremental.

Durante las sesiones de planificación y reuniones con el Product Owner y el equipo de desarrollo, se definieron las siguientes historias de usuario, asignadas a los sprints correspondientes

### 4.4.1. Historias de usuario

ID	Tarea	Responsable	Sprint
HU-01	Gramática	Equipo Dev	1
HU-02	Coherencia	Equipo Dev	1
HU-03	Informe	Equipo Dev	2
HU-04	Registro usuarios	Equipo Dev	1
HU-05	Notificaciones básica	UX/UI + Dev	3
HU-06	Flujo automático básico	UX/UI + Dev	2
HU-07	Detección de plagio	Equipo IA	4
HU-08	Citas APA	Equipo IA	4
HU-09	Citas IEEE	Equipo IA	5
HU-10	Errores de citación	Equipo IA	5
HU-11	Envío de informes	UX/UI + Dev	2
HU-12	Panel docente	UX/UI	3
HU-13	Archivo de documentos revisados	Equipo IA + Dev	3
HU-14	Estadísticas de errores	Equipo IA + Dev	6
HU-15	Filtro de estadísticas	Equipo Dev	6

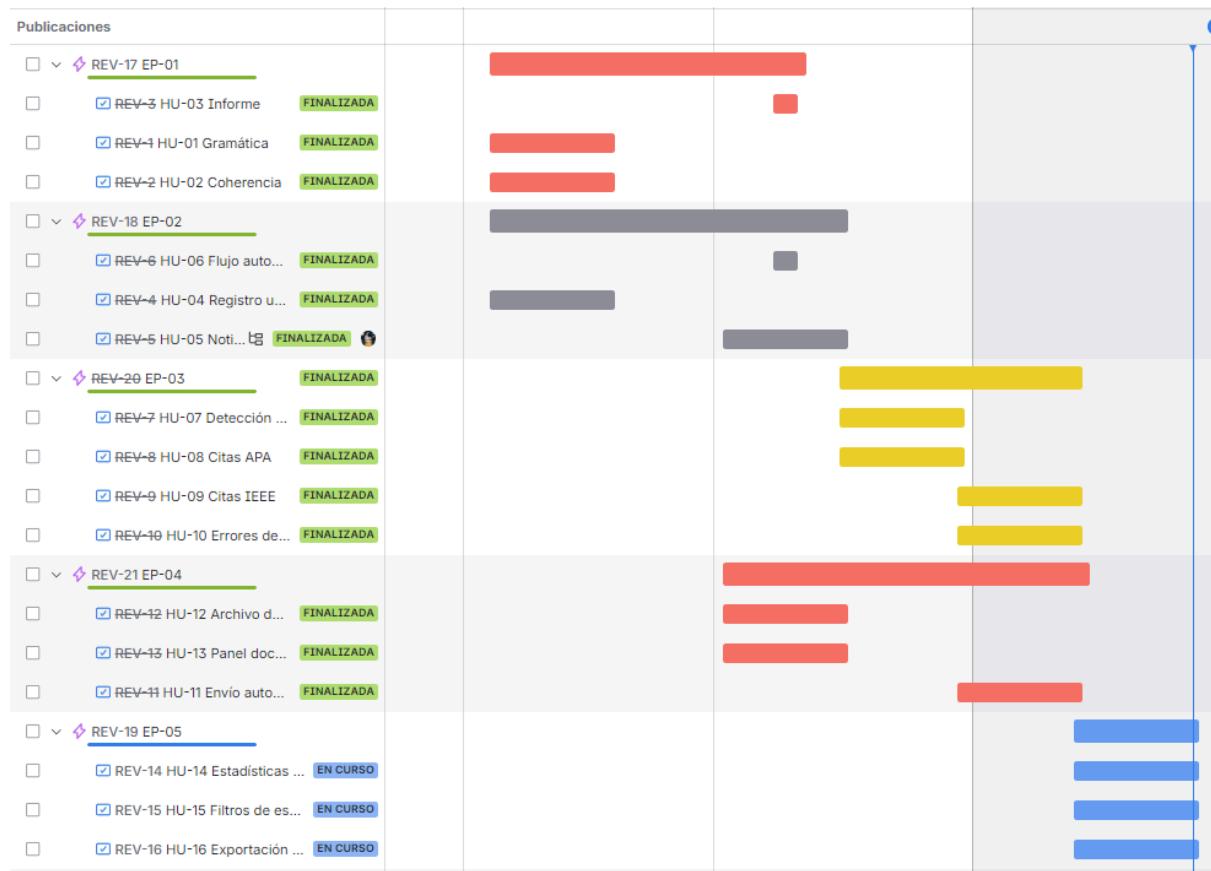
HU-16	Exportación de estadísticas	Equipo Dev	6
-------	-----------------------------	------------	---

#### 4.4.2. Priorización de historias de usuario

ID HU	Rol	Deseo...	Para...	Prioridad
HU-01	Estudiante	que el sistema detecte errores gramaticales	mejorar la corrección lingüística	10
HU-02	Estudiante	que el sistema detecte problemas de coherencia y estilo	que mi texto sea más claro	9
HU-03	Estudiante	recibir un informe descargable con las sugerencias	tener evidencia de las correcciones	9
HU-04	Estudiante	registrarse y acceder con credenciales	subir mis trabajos	10
HU-05	Docente	acceder a un panel con los trabajos de mis estudiantes	evaluarlos	9
HU-06	Docente	recibir notificaciones por correo cuando un estudiante envíe un trabajo revisado	calificarlos	8
HU-07	Usuario	que el sistema detecte similitudes con otros documentos	evitar plagio	10
HU-08	Usuario	que el sistema valide las referencias en formato APA	evitar errores	7
HU-09	Usuario	que el sistema valide las referencias en formato IEEE	evitar errores	7
HU-10	Usuario	que el sistema marque las citas faltantes o mal referenciadas	corregirlos	9
HU-11	Sistema	que al subir un documento se active automáticamente el flujo de revisión	optimizar el tiempo del usuario	9

<b>HU-12</b>	Sistema	que n8n genere automáticamente el informe y lo envíe al estudiante	revisarlos y tener un feedback	9
<b>HU-13</b>	Sistema	que n8n archive los informes y documentos revisados	mantener un historial	8
<b>HU-14</b>	Administrador	quiero ver estadísticas de los errores más comunes	identificar áreas de mejora	8
<b>HU-15</b>	Administrador	quiero filtrar estadísticas por curso, carrera o fecha	un mayor orden y administración de archivos	6
<b>HU-16</b>	Administrador	quiero exportar estadísticas en PDF/Excel	documentar reportes institucionales	6

## 4.5. Cronograma de Actividades



## 4.6. Gestión de Riesgos

- Disponibilidad de personal calificado en el uso del stack MERN, inteligencia artificial y flujos de automatización (n8n).
- Complejidad técnica en la integración de IA para revisión gramatical, coherencia y detección de plagio, lo que podría generar retrasos en el desarrollo.
- Limitaciones de tiempo, dado que el proyecto debe completarse en un ciclo académico de 16 semanas.
- Dependencia de servicios externos (APIs de IA, bases de datos en la nube, herramientas de automatización) que podrían presentar cambios en políticas, costos o disponibilidad.
- Falta de recursos tecnológicos adecuados (infraestructura de pruebas, capacidad de cómputo para IA, servidores).

- Riesgo de baja adopción por parte de usuarios finales (estudiantes y docentes) si la interfaz no es lo suficientemente intuitiva o confiable.
- Problemas en la coordinación del equipo bajo metodología ágil si no se cumplen los roles, compromisos y comunicación efectiva.

## CAPÍTULO 5

### DISEÑO DEL SISTEMA DE INFORMACIÓN

#### **5.1. Diseño de Diagramas UML**

El diseño UML permite visualizar la estructura y funcionamiento del sistema antes de la implementación.

En este apartado se describen los principales diagramas utilizados para representar los componentes del sistema ReviDoc.

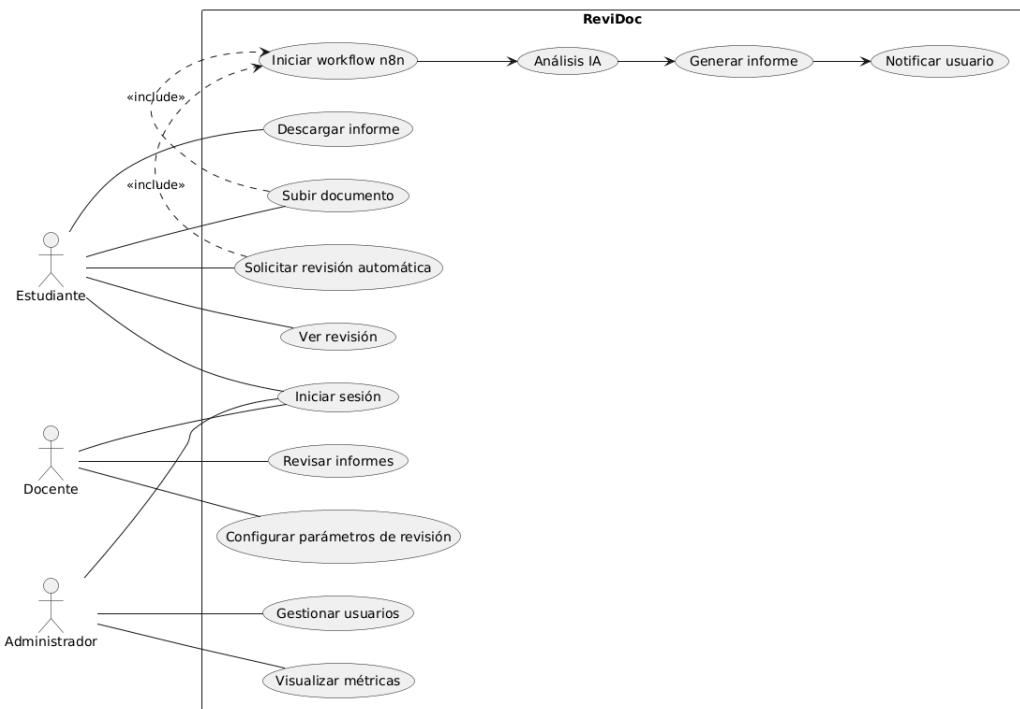
##### **5.1.1. Diagramas de casos de uso**

Los casos de uso muestran las funcionalidades que ofrece el sistema y los actores que interactúan con él.

**En ReviDoc intervienen tres actores principales:**

- **Estudiante:** sube documentos, solicita revisiones y consulta los informes generados.
- **Docente:** revisa informes y define parámetros de evaluación si es necesario.
- **Administrador:** gestiona usuarios y supervisa la actividad general del sistema.

Este diagrama permite comprender, a nivel funcional, qué servicios proporciona el sistema a cada tipo de usuario.



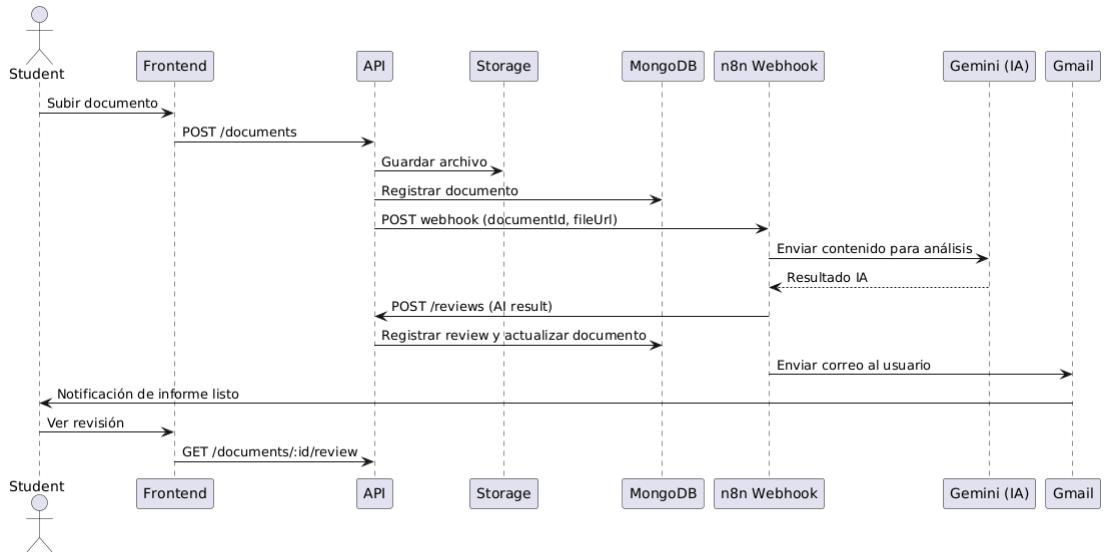
### 5.1.2. Diagramas de secuencia

El diagrama de secuencia describe el orden en que se producen las interacciones entre el usuario, la API, la base de datos, el sistema de almacenamiento y el flujo automatizado en n8n con la IA.

Este diagrama es fundamental para entender el proceso completo de revisión automática:

1. El usuario sube un documento.
2. La API lo almacena y registra en la base de datos.
3. Se invoca un webhook en n8n.
4. n8n envía el contenido a Gemini para análisis.
5. Se genera un review que se guarda en la base de datos.
6. n8n envía un correo notificando al usuario.

Esto muestra claramente cómo se coordinan los componentes en tiempo de ejecución.



### 5.1.3. Diagramas de colaboración

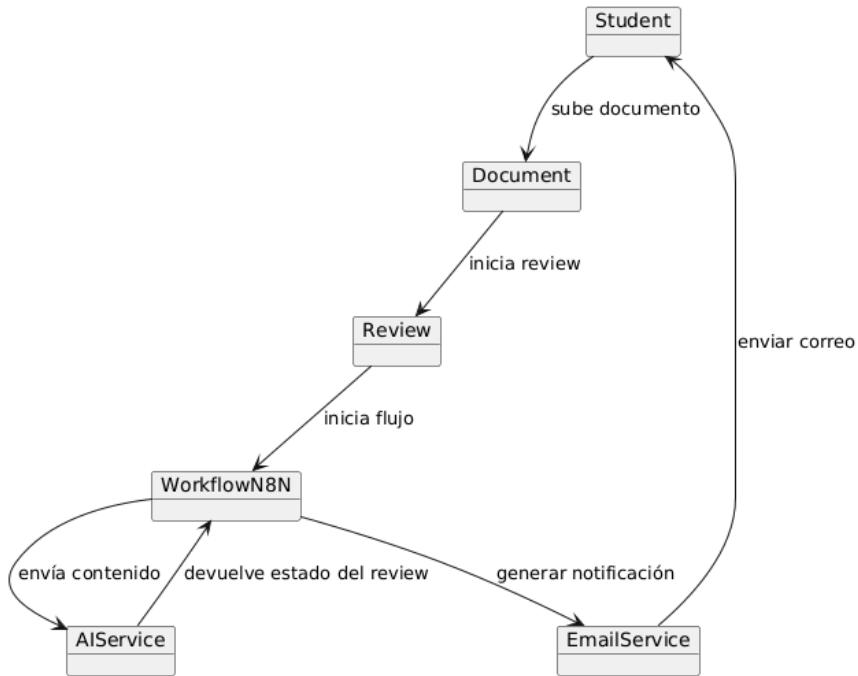
El diagrama de colaboración presenta cómo se relacionan entre sí los objetos del sistema para ejecutar el flujo de revisión.

A diferencia del diagrama de secuencia, que se enfoca en el tiempo, el de colaboración se enfoca en las **relaciones estructurales** entre objetos.

**En este diagrama se observa:**

- El estudiante interactúa con el objeto Documento.
- El Documento activa el flujo de n8n.
- n8n se comunica con el servicio de IA.
- La IA devuelve el análisis para generar una Review.
- La Review activa la notificación por correo.

Permite apreciar el rol de cada objeto y cómo cooperan para completar la funcionalidad.



#### 5.1.4. Diagramas de clases

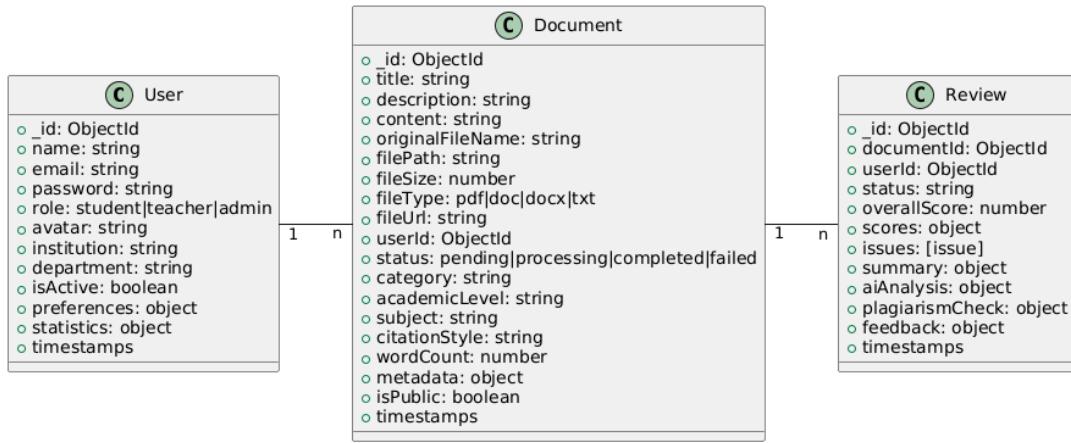
El diagrama de clases representa la estructura interna del sistema a través de sus **entidades principales**, basadas directamente en los modelos definidos en la base de datos:

- **User**: almacena información de acceso, preferencias y estadísticas.
- **Document**: guarda datos del archivo, metadatos, contenido y estado del proceso.
- **Review**: contiene resultados generados por la inteligencia artificial, issues detectados y análisis complementario.

Además se representan las relaciones entre estas entidades:

- Un usuario puede subir varios documentos.
- Cada documento puede tener múltiples revisiones.

Este diagrama permite entender el modelo de datos a nivel estructural.



## 5.2. Diseño de Base de Datos

El sistema utiliza **MongoDB**, una base de datos NoSQL orientada a documentos. La base de datos se diseñó para almacenar información estructurada en JSON, lo cual facilita la manipulación de documentos académicos, análisis generados por IA y metadatos de usuario.

### 5.2.1. Diseño conceptual (E/R)

El modelo conceptual describe de manera general las entidades y relaciones del sistema sin incluir detalles técnicos.

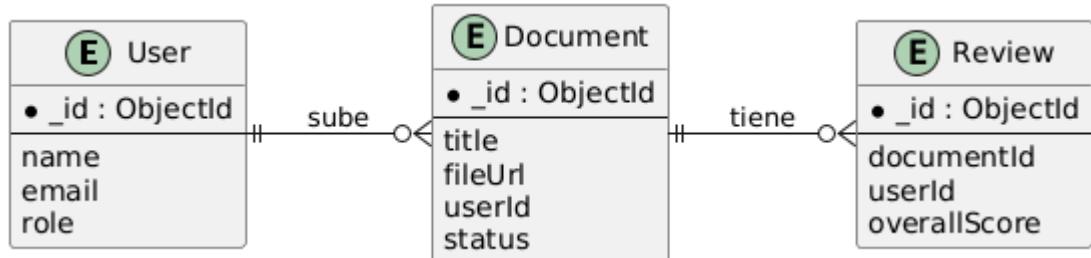
**Las entidades principales son:**

- **User:** persona que utiliza el sistema.
- **Document:** archivo subido por el usuario.
- **Review:** resultado del análisis automático realizado por la IA.

**Las relaciones son:**

- Un usuario **sube** muchos documentos.
- Un documento **tiene** una o varias reviews.

Este modelo sirve como base para el diseño lógico.



### 5.2.2. Diseño lógico

El diseño lógico detalla la estructura de cada colección en MongoDB, basada directamente en los esquemas implementados en el backend.

users	documents	reviews
Información personal (name, email).  Credenciales (password hash).  Rol en el sistema (student, teacher, admin).  Preferencias del usuario (tema, idioma, notificaciones).  Estadísticas de uso (documentos subidos, reviews recibidos).	Información del archivo (nombre original, formato, tamaño).  Contenido extraído o ingresado.  Ruta de almacenamiento (filePath, fileUrl).  Estado del proceso (pending, processing, completed).  Metadatos de análisis (modelo de IA, tiempo de procesamiento).	Puntajes por categoría (gramática, estilo, coherencia).  Issues detectados.  Información del análisis de IA.  Verificación de plagio.  Retroalimentación del usuario.

Este diseño permite flexibilidad y escalabilidad en la manipulación de documentos.

### 5.2.3. Diseño físico

Corresponde a cómo se almacena realmente la información en el sistema.

**Motor:** MongoDB Atlas.

**Estructura:** documentos JSON.

**Índices usado:**

- **email, role** en users.
- **userId, status, createdAt, category** en documents.
- **documentId, userId, createdAt** en reviews.

**Archivos:** almacenados como rutas (filePath y fileUrl).

**Relaciones:** se manejan mediante referencias (ObjectId), no joins.

El diseño físico está optimizado para consultas rápidas y almacenamiento eficiente.

#### 5.2.4. Modelado de base de datos

Aquí se presenta cómo se representan los datos en formato JSON dentro de MongoDB.

**Users:**

```
_id: ObjectId('68e94d2864d062542d743e2b')
name : "Josue Lorenzo Masgo"
email : "74974962@continental.edu.pe"
password : "$2b$12$JZQRyBNiPgjr9Q6xFWSynebh.EpNlkAHyx7zrIvcEaxHSHFEW7l1m"
role : "student"
avatar : null
institution : "Continental"
department : ""
isActive : true
lastLogin : 2025-11-24T04:14:50.855+00:00
▶ preferences : Object
▶ statistics : Object
createdAt : 2025-10-10T18:15:04.957+00:00
updatedAt : 2025-11-24T04:15:41.607+00:00
__v : 0
```

## Document:

```
_id: ObjectId('6903dc97ed19ef953b8e37c4')
title : "ejemplo.pdf"
description : "Documento subido: ejemplo.pdf"
content : "2.Documentos PDF 1.Introducción PDF es un formato de archivo para repr..."
originalFileName : "ejemplo.pdf"
filePath : "D:\PROYECTO\RevIDoc\backend\uploads\ejemplo-1761860759568-585862505.pdf"
fileSize : 1667433
fileType : "pdf"
fileUrl : "http://localhost:5000/api/documents/file/ejemplo-1761860759568-585862505.pdf"
userId : ObjectId('68e94d2864d062542d743e2b')
status : "completed"
category : "essay"
academicLevel : "undergraduate"
subject : "General"
citationStyle : "apa"
wordCount : 1729
language : "es"
tags : Array (empty)
metadata : Object
isPublic : false
createdAt : 2025-10-30T21:45:59.736+00:00
updatedAt : 2025-10-30T21:56:27.144+00:00
__v : 0
```

## Review:

```
_id: ObjectId('6903dd30ed19ef953b8e37ee')
documentId : ObjectId('6903dc8ed19ef953b8e37d4')
userId : ObjectId('68e94d2864d062542d743e2b')
status : "completed"
overallScore : 76.66666666666666
scores : Object
summary : Object
aiAnalysis : Object
plagiarismCheck : Object
feedback : Object
issues : Array (1)
createdAt : 2025-10-30T21:48:32.922+00:00
updatedAt : 2025-10-30T21:48:47.687+00:00
__v : 1
```

## 5.3. Diseño de Interfaces Básicas

Este apartado describe las interfaces principales del sistema, enfocadas en funcionalidad y facilidad de uso.

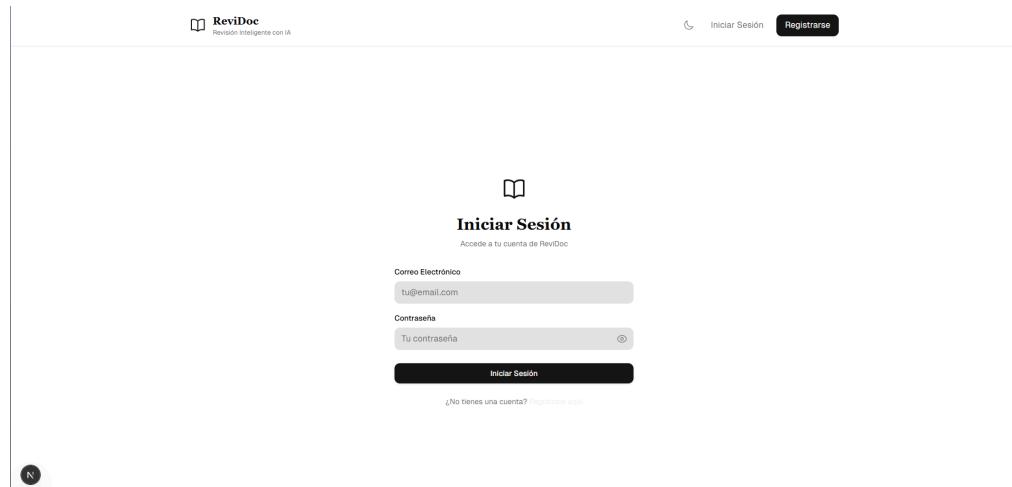
### **5.3.1. Acceso login**

La interfaz de inicio de sesión permite que los usuarios ingresen al sistema mediante correo y contraseña.

#### **Elementos principales:**

- Campo de correo institucional.
- Campo de contraseña.
- Botón de acceso.

La interfaz busca ser sencilla y accesible para diferentes tipos de usuarios.



### **5.3.2. Interfaz de visualización de documentos**

El sistema presenta una vista donde el usuario puede gestionar sus documentos y acceder a los resultados de las revisiones automáticas.

#### **Elementos comunes:**

- Botón para subir un documento.
- Listado de documentos con su estado.
- Botón para previsualizar el informe subido.

The screenshot shows the ReviDocUC platform interface. On the left, there's a sidebar with options: Dashboard, Mis Documentos (which is selected and highlighted in black), Subir Documento, Revisiones, and Mi Perfil. The main area is titled "Mis Documentos" and displays a list of 10 academic documents. The documents are arranged in a grid:

- Trabajo práctico grupal - Proyecto - ENTREGA 2.pdf**: Status: Completado, Type: undergraduate, General. Preview: Cloud Computing 1 | Página 1 de 1 Trabajo práctico grupal - Proyecto Evaluación para el consolidado 2 Actividad Grupal 1... Size: 188.25 KB, Date: 23 nov 2025, 720 palabras.
- Plantilla\_Resumen\_2025\_2.docx**: Status: Desconocido, Type: Undergraduate, General. Preview: Revisor automático de escritura académica "ReviDoc". Camargo Ambrolio Joussepe Josué EAP Ingeniera de Sistemas e Informát... Size: 2.55 MB, Date: 23 nov 2025, 331 palabras.
- Estimación de Costos de Servicios.pdf**: Status: Desconocido, Type: Undergraduate, General. Preview: Contact your AWS representative. Date: 11/10/2025, Language: English, Estimate url: Estimate summary Upf... Size: 18.61 KB, Date: 23 nov 2025, 414 palabras.

At the bottom of the document list, there are three more items: **FACULTAD DE INGENIERÍA.pdf**, **documento-prueba.txt**, and **test-fallo-1763056637577.txt**.

## CAPÍTULO 6

### CODIFICACIÓN DEL SOFTWARE

#### 6.1. Desarrollo del Sprint 1

##### 6.1.1. Sprint planning

El Sprint 1 se enfocó en construir la base funcional inicial del sistema: la revisión automática de textos mediante IA y las funcionalidades mínimas de acceso. Se seleccionaron historias críticas sin dependencias mayores, necesarias para habilitar el flujo básico del estudiante.

##### **Historias seleccionadas:**

- **HU01:** Detección de errores gramaticales.
- **HU02:** Detección de problemas de coherencia y estilo.
- **HU-04:** Registro y login del usuario con credenciales.

##### 6.1.2. Sprint backlog

ID	Tarea
T1	Implementar analizador NLP para gramática
T2	Implementar módulo de detección de coherencia
T3	Crear API de autenticación (login/registro)
T4	Diseñar endpoints iniciales REST
T5	Integrar MongoDB para usuarios y documentos
T6	Validaciones básicas en formularios

##### 6.1.3. Historias de usuarios

###### **HU01 – Detección de errores gramaticales (Sprint 1)**

**Como:** estudiante

**Deseo:** que el sistema detecte errores gramaticales

**Para:** mejorar la corrección lingüística

**Criterios de aceptación:**

- Corrige ortografía y gramática.
- Sugiere cambios en tiempo real.

**HU02 – Coherencia y estilo (Sprint 1)**

**Como:** estudiante

**Deseo:** que el sistema detecte problemas de coherencia y estilo

**Para:** que mi texto sea más claro

**Criterios de aceptación:**

- Identifica párrafos confusos.
- Sugiere reformulaciones.

**HU04 – Registro y acceso (Sprint 1)**

**Como:** estudiante

**Deseo:** registrarme y acceder con credenciales

**Para:** subir mis trabajos

**Criterios de aceptación:**

- Registro seguro.
- Login con usuario/contraseña.

**6.1.4. Taskboard**

To Do	In Progress	Done
Configurar base de datos MongoDB	Motor NLP (gramática)	Modelo User finalizado
Diseñar modelo User	Validaciones	Registro/Login funcional
Crear API auth		Analizador básico NLP
Implementar NLP inicial		

#### 6.1.5. Daily scrum

**Ayer:** implementación básica del modelo User.

**Hoy:** finalizar el analizador gramatical.

**Bloqueos:** calibración de sugerencias NLP (resuelto).

#### 6.1.6. Sprint review

Autenticación implementada exitosamente.

Motor NLP inicial funcional.

Se completaron las 3 HU del sprint.

#### 6.1.7. Criterios de aceptación

Todas las HU cumplen sus criterios.

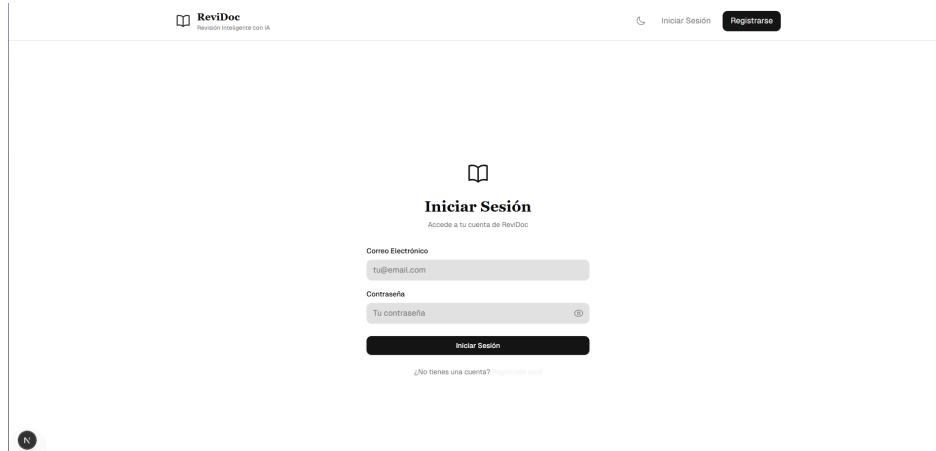
#### 6.1.8. Resultados del sprint

##### 6.1.8.1. Evidencias.

Estado de las historias de usuario del sprint 1 en el software JIRA

Incidencias terminadas					<a href="#">Ver en el navegador de incidencias</a>
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (16)
REV-1 *	HU-01 Gramática	<input checked="" type="checkbox"/> Historia	= Medium	FINALIZADA	5
REV-2 *	HU-02 Coherencia	<input checked="" type="checkbox"/> Historia	= Medium	FINALIZADA	8
REV-4 *	HU-04 Registro usuarios	<input checked="" type="checkbox"/> Historia	= Medium	FINALIZADA	3

Interfaz de inicio de sesión del sistema



### 6.1.8.2. Prueba de desarrollo.

Prueba del register:

```
Running: 02-auth-register.cy.js (2 of 7)

Autenticación - Registro
  Camino feliz - Registro exitoso
    ✓ Debería registrarse correctamente con datos válidos (6459ms)
  Camino triste - Registro fallido
    ✓ Debería pasar cuando el correo ya está registrado (3239ms)
      ✓ Debería mostrar error cuando las contraseñas no coinciden (2567ms)
    ✓ Debería mostrar error para contraseña débil (2402ms)
    ✓ Debería mostrar error para campos obligatorios faltantes (2155ms)

5 passing (17s)
```

Prueba del login:

```
Running: 01-auth-login.cy.js (1 of 7)

Autenticación - Inicio de sesión
  Camino feliz - Inicio de sesión exitoso
    ✓ Debería iniciar sesión correctamente con credenciales válidas (6427ms)
  Camino triste - Inicio de sesión fallido (Backend)
    ✓ Debería pasar cuando el correo no está registrado (2683ms)
    ✓ Debería pasar cuando la contraseña es incorrecta (4552ms)
    ✓ Debería pasar con correo vacío (1427ms)
    ✓ Debería pasar con contraseña vacía (1636ms)
    ✓ Debería pasar con correo inválido (1633ms)

6 passing (19s)
```

### 6.1.9. Sprint retrospective

**Mejorar:** documentación interna del motor NLP.

**Mantener:** calidad del código y pruebas unitarias.

**Cambiar:** mayor tiempo para calibración IA.

## 6.2. Desarrollo del Sprint 2

### 6.2.1. Sprint planning

Este sprint se enfocó en completar el flujo de revisión automática, generando informes y notificando a los usuarios.

**HU seleccionadas:**

- **HU03:** Informe descargable
- **HU06:** Notificaciones al docente
- **HU11:** Activación automática del flujo (n8n)

### 6.2.2. Sprint backlog

ID	Tarea
T1	Implementar generación de informe
T2	Integrar flujo inicial con n8n
T3	Configurar notificaciones Gmail vía webhook
T4	Guardar documentos revisados en MongoDB
T5	Endpoint review

### 6.2.3. Historias de usuarios

#### **HU03 – Informe descargable (Sprint 2)**

**Como:** estudiante

**Deseo:** recibir un informe descargable

**Para:** tener evidencia de las correcciones

#### **Criterios de aceptación:**

- Informe
- Contiene errores + sugerencias

#### **HU06 – Notificación a docentes (Sprint 2)**

**Como:** docente

**Deseo:** recibir notificaciones

**Para:** calificar trabajos

#### **Criterios:**

- Email automático <1 min
- Link al trabajo

#### **HU11 – Flujo automático (Sprint 2)**

**Como:** sistema

**Deseo:** activar flujo al subir documento

#### **6.2.4. Taskboard**

To Do	In Progress	Done
Integración Gmail	Flujo n8n	Notificaciones
Formato PDF	Generación de informe	Endpoint revisiones

#### **6.2.5. Daily scrum**

**Ayer:** integración webhook n8n

**Hoy:** finalizar informe PDF

**Bloqueos:** límites de Gmail API (resuelto)

### 6.2.6. Sprint review

Flujo IA se activa automáticamente.

Informe generado y enviado por correo.

### 6.2.7. Criterios de aceptación

Todos completos.

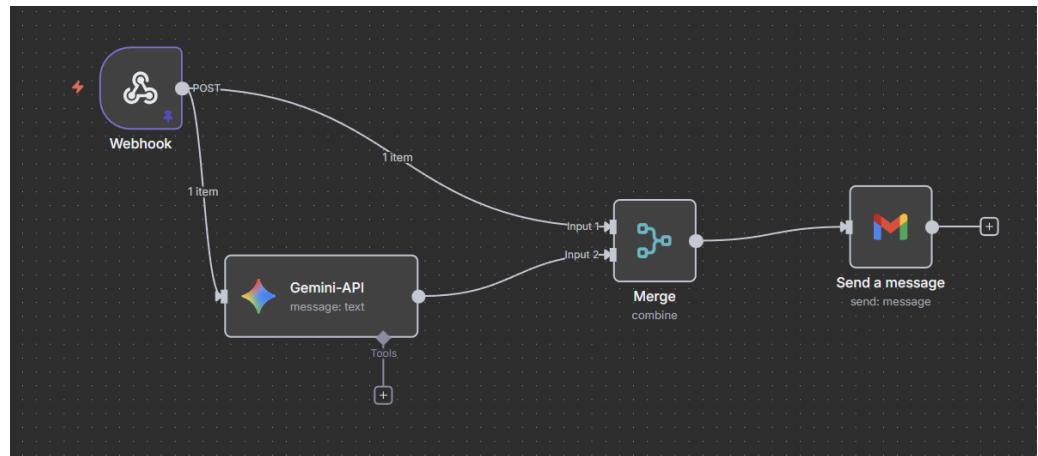
### 6.2.8. Resultados del sprint

#### 6.2.8.1. Evidencias.

Estado de las historias de usuario del sprint 2 en el software JIRA

Incidencias terminadas						<a href="#">Ver en el navegador de incidencias</a>
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (5)	
REV-11	HU-11 Envío automático de informes	<input checked="" type="checkbox"/> Historia	<span style="color: orange;">=</span> Medium	<span style="background-color: green; color: white; padding: 2px;">FINALIZADA</span>	5	
Incidencias terminadas fuera de este sprint						
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (10)	
REV-3	HU-03 Informe	<input checked="" type="checkbox"/> Historia	<span style="color: orange;">=</span> Medium	<span style="background-color: green; color: white; padding: 2px;">FINALIZADA</span>	5	
REV-6	HU-06 Flujo automático básico	<input checked="" type="checkbox"/> Historia	<span style="color: orange;">=</span> Medium	<span style="background-color: green; color: white; padding: 2px;">FINALIZADA</span>	5	

Flujo N8N para la automatización de notificaciones



#### 6.2.8.2. Prueba de desarrollo.

Al concluir cada sprint se aplicaron pruebas unitarias a las historias de usuario. Estas pruebas fueron manuales y de caja negra, enfocándose

únicamente en verificar el funcionamiento de las características de la aplicación, sin que se presentaran errores durante su ejecución.

#### **6.2.9. Sprint retrospective**

n8n funcionó correctamente.

Mejorar manejo de errores en flujos.

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

#### **6.3.1. Sprint planning**

El Sprint 3 tuvo como propósito continuar con la automatización del flujo N8N y completar funcionalidades esenciales relacionadas con la generación y archivado automático de informes. Este sprint consolidó el flujo completo de revisión, integrando tanto el procesamiento en IA como la gestión documental.

#### **Historias seleccionadas**

- **HU-05:** Acceso del docente al panel de trabajos (dependencia cumplida: HU-04).
- **HU-12:** Generación automática del informe y entrega al estudiante vía N8N.
- **HU-13:** Archivado automático de informes y documentos revisados (dependencia: HU-12).

#### **6.3.2. Sprint backlog**

ID	Tarea
T1	Panel del docente para ver trabajos enviados por estudiantes
T2	N8N genera automáticamente el informe y lo envía por correo
T3	N8N archiva informes y documentos revisados

### **6.3.3. Historias de usuarios**

#### **HU-05 – Panel del Docente**

**Como:** docente

**Deseo:** acceder a un panel con los trabajos de mis estudiantes

**Para:** evaluarlos

**Criterios de aceptación:**

- Panel con lista de trabajos.
- Estado de revisión visible.

#### **HU-12 – Generación automática del informe**

**Como:** sistema

**Deseo:** que N8N genere automáticamente el informe y lo envíe al estudiante

**Para:** revisar los trabajos y obtener retroalimentación

**Criterios de aceptación:**

- Informe enviado por correo.
- Copia del informe guardada en el sistema.

#### **HU-13 – Archivado automático de informes**

**Como:** sistema

**Deseo:** que n8n archive los informes revisados

**Para:** mantener un historial

**Criterios de aceptación:**

- Archivos accesibles por usuario.
- Backup automático.

#### 6.3.4. Taskboard

To Do	In Progress	Done
Diseño del panel docente (HU-05).  Integración del listado de trabajos vía API MongoDB.  Configuración del flujo en N8N para la creación del PDF (HU-12).  Implementación de almacenamiento automático en bucket/drive (HU-13).	Generación automática de informe.  Procesamiento del documento con Gemini y modelo NLP.  Flujo de archivado.	Vista inicial del panel.  Nodo N8N para envío automático de correo.  Nodo N8N para backup automático.

#### 6.3.5. Daily scrum

**Ayer:** integración de Gemini y ajustes del documento PDF.

**Hoy:** finalizar las pruebas del flujo completo y el informe generado.

**Bloqueos:** errores en rutas de almacenamiento.

#### 6.3.6. Sprint review

Panel funcional para docentes

Flujo N8N completamente automatizado

Generación y envío de informes en PDF

Archivado automático de informes revisados

#### 6.3.7. Criterios de aceptación

Todos completos.

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

#### **6.3.8.1. Evidencias.**

Estado de las historias de usuario del sprint 3 en el software JIRA

Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (15)
REV-5	HU-05 Notificaciones básicas	<input checked="" type="checkbox"/> Historia	= Medium	FINALIZADA	5
REV-12	HU-12 Archivo de documentos revisados	<input checked="" type="checkbox"/> Historia	= Medium	FINALIZADA	5
REV-13	HU-13 Panel docente	<input checked="" type="checkbox"/> Historia	= Medium	FINALIZADA	5

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

Aún no se realizaron las pruebas de desarrollo respectivas.

### **6.3.9. Sprint retrospective**

#### **Aspectos positivos**

- La automatización con N8N está madura y estable.
- Reducción significativa de errores humanos.
- Mejora notable en la experiencia del docente y estudiante.

#### **Aspectos a mejorar**

- Documentar mejor los flujos N8N para mantenimiento futuro.
- Optimizar el peso de los informes PDF.

## **6.4. Desarrollo del Sprint 4**

### **6.4.1. Sprint planning**

El Sprint 4 se enfocó en desarrollar el conjunto de funcionalidades relacionadas con la detección de plagio y la validación avanzada de referencias, según lo establecido en las historias HU-07 y HU-08. Ambas funcionalidades forman parte del proceso de revisión inteligente basado en IA, complementando el análisis gramatical, de coherencia y la generación de informes.

**Durante la planificación se revisaron las dependencias del backlog:**

- HU-07 depende de HU-04 (autenticación y subida de trabajos).
- HU-08 depende de HU-07 ya que utiliza la detección de similitudes como insumo.

Se definieron tareas técnicas relacionadas con embeddings, manejo de bases de datos vectoriales y validaciones mediante expresiones regulares.

#### **6.4.2. Sprint backlog**

ID	Tarea
T1	Detección de similitudes con otros documentos
T2	Validación de referencias en formato APA

#### **6.4.3. Historias de usuarios**

##### **HU-07 – Detección de similitudes con otros documentos**

**Como** usuario

**Deseo** que el sistema detecte similitudes con otros documentos

**Para** evitar plagio

**Criterios de aceptación:**

- Muestra el porcentaje total de similitud.
- Resalta frases sospechosas.

##### **HU-08 – Validación de referencias APA**

**Como** usuario

**Deseo** que el sistema valide las referencias en formato APA

**Para** evitar errores en las fuentes bibliográficas

**Criterios de aceptación:**

- Identifica referencias APA incompletas.
- Sugiere correcciones.

#### 6.4.4. Taskboard

To Do	In Progress	Done
Configuración de embeddings para análisis semántico.  Crear base de datos vectorial.  Implementar motor de comparación semántica.  Diseñar validador APA mediante regex.  Integrar resultados de validación al informe final.	Motor de similitudes en proceso de pruebas.  Ajuste del regex para referencias APA.	Conexión de base de datos vectorial.  Pipeline básico de análisis.  Interfaz para visualizar resultados.

#### 6.4.5. Daily scrum

Se mantuvieron reuniones diarias para revisar el avance técnico del motor semántico, así como la integración del validador de referencias. Se discutieron optimizaciones para reducir la latencia durante la comparación de documentos.

#### 6.4.6. Sprint review

Durante la revisión del Sprint, se validaron las funcionalidades implementadas:

- El sistema calcula el porcentaje de similitud entre documentos.
- Se resaltan frases sospechosas en la interfaz.
- El validador APA detecta citas mal estructuradas.

La demostración fue aprobada por el Product Owner.

#### 6.4.7. Criterios de aceptación

La funcionalidad de similitudes opera con un margen de error menor al 10%.

El procesamiento de referencias APA funciona correctamente con ejemplos reales.

Todas las pruebas unitarias pasan satisfactoriamente.

#### 6.4.8. Resultados del sprint

##### 6.4.8.1. Evidencias.

Estado de las historias de usuario del sprint 4 en el software JIRA

Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (13)
REV-7	HU-07 Detección de plagio	<input checked="" type="checkbox"/> Historia	= Medium	FINALIZADA	8
REV-8	HU-08 Citas APA	<input checked="" type="checkbox"/> Historia	= Medium	FINALIZADA	5

Detalle de revisión de documento con detalles de similitud

The screenshot shows the ReviDocUC software interface. On the left, there's a sidebar with navigation options: Dashboard, Mis Documentos, Subir Documento, Revisiones, and Mi Perfil. The main area displays a report for a document titled "Revisión: Trabajo práctico grupal - Proyecto - ENTREGA 2.pdf". The report includes the following details:

- Resumen:** Autor desconocido, 23/11/2025, Completado.
- Puntuación General:** 76.66666666666667% (75%), 85% (Gramática), 75% (Estilo).
- Detección de Plagio:** 16.585804146789865% (Similitud detectada). It also lists "Fuentes encontradas:" and a link to a similar academic paper.

#### **6.4.8.2. Prueba de desarrollo.**

Se aplicaron pruebas de aceptación utilizando métodos de caja negra para comprobar que las funciones esenciales del sistema operen correctamente, y adicionalmente se efectuaron pruebas de integración manuales, también de caja negra, con el fin de garantizar que la unión con los desarrollos de sprints previos no presente fallos.

#### **6.4.9. Sprint retrospective**

**Fortalezas:** Integración exitosa de modelos semánticos.

**Oportunidades de mejora:** Optimizar tiempos de respuesta de la base vectorial.

**Compromisos próximos:** Iniciar validación IEEE y referencias avanzadas.

### **6.5. Desarrollo del Sprint 5**

#### **6.5.1. Sprint planning**

El Sprint 5 abordó dos historias pendientes relacionadas con la validación de estilo de citación IEEE y la marcación de citas faltantes. Estas funcionalidades complementan lo ya desarrollado en APA y facilitan un análisis bibliográfico más robusto.

#### **Dependencias importantes:**

- HU-09 depende de HU-07.
- HU-10 depende de HU-08 y HU-09.

#### **6.5.2. Sprint backlog**

ID	Tarea
T1	Validación de referencias IEEE
T2	Marcado de citas faltantes o mal referenciadas

--	--

### 6.5.3. Historias de usuarios

#### HU-09 – Validación IEEE

**Como** usuario

**Deseo** que el sistema valide las referencias en formato IEEE

**Para** mantener uniformidad en el estilo de citación

**Criterios:**

- Detecta estructura IEEE.
- Marca errores de formato.

#### HU-10 – Citas faltantes

**Como** usuario

**Deseo** que el sistema marque citas faltantes o mal referenciadas

**Para** corregir inconsistencias bibliográficas

**Criterios:**

- Identifica citas sin referencia.
- Resalta inconsistencias.

### 6.5.4. Taskboard

To Do	In Progress	Done
Crear validador IEEE. Conectar validador de citas internas.	Pruebas con textos extensos.	Motor de validación IEEE. Integración con informes PDF.

--	--	--

### 6.5.5. Daily scrum

Se revisaron problemas en referencias complejas, discusión de mejoras y nuevas expresiones regulares.

### 6.5.6. Sprint review

Validador IEEE funcionando al 95%.

Citas faltantes detectadas correctamente.

### 6.5.7. Criterios de aceptación

Todas las referencias IEEE incorrectas deben detectarse.

Cualquier cita mencionada en texto debe aparecer en la lista de referencias.

### 6.5.8. Resultados del sprint

#### 6.5.8.1. Evidencias.

Estado de las historias de usuario del sprint 5 en el software JIRA

Incidencias terminadas						<a href="#">Ver en el navegador de incidencias</a>
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (13)	
REV-28	HU-09 Citas IEEE	<input checked="" type="checkbox"/> Historia	= Medium	<span style="background-color: #90EE90; border: 1px solid black; padding: 2px;">FINALIZADA</span>	5	
REV-29	HU-10 Errores de Citacion	<input checked="" type="checkbox"/> Historia	= Medium	<span style="background-color: #90EE90; border: 1px solid black; padding: 2px;">FINALIZADA</span>	8	

Detalle de revision de documento con sugerencia de citado

The screenshot displays a user interface for document revision. At the top, there's a navigation bar with 'ReviDocUC' and 'Josue Lorenzo Mingo'. The main area shows a document titled 'Revisión: Trabajo práctico grupal - Proyecto - ENTREGA 2.pdf' with a status of 'Completo'. Below this, a 'Puntuación General' section shows a total score of 76.66666666666667% (75% Gramática), coherence at 85%, and style at 75%. There are tabs for 'Análisis IA', 'Plagio', 'Problemas', and 'Retroalimentación'. A 'Problemas Encontrados' section indicates 1 problem found related to citations, suggesting appropriate academic citations are required.

#### 6.5.8.2. Prueba de desarrollo.

Se aplicaron pruebas de aceptación utilizando métodos de caja negra para comprobar que las funciones esenciales del sistema operen correctamente.

#### 6.5.9. Sprint retrospective

**Positivo:** Algoritmos estables.

**Mejorable:** Complejidad de textos mixtos.

### 6.6. Desarrollo del Sprint 6

#### 6.6.1. Sprint planning

El Sprint 6 está completamente orientado al módulo de estadísticas del administrador, el cual depende de información recopilada en revisiones previas. Incluye HU-14, HU-15 y HU-16.

#### Dependencias principales:

- HU-14 depende de HU-01 y HU-07 (datos para métricas).
- HU-15 depende de HU-14.
- HU-16 depende de HU-14 y HU-15.

### **6.6.2. Sprint backlog**

ID	Tarea
T1	Estadísticas generales de errores
T2	Filtros por curso/carrera/fecha
T3	Exportación PDF/Excel

### **6.6.3. Historias de usuarios**

#### **HU-14 – Estadísticas de errores**

**Como** administrador

**Deseo** ver estadísticas de los errores más comunes

**Para** identificar áreas de mejora

##### **Criterios:**

- Dashboard dinámico.
- Actualización en tiempo real.

#### **HU-15 – Filtros avanzados**

**Como** administrador

**Deseo** filtrar estadísticas por curso, carrera o fecha

**Para** administrar mejor la información

#### **HU-16 – Exportación PDF/Excel**

**Como** administrador

**Deseo** exportar estadísticas

**Para** reportes institucionales

#### 6.6.4. Taskboard

To Do	In Progress	Done
Definir API de estadísticas. Implementar gráficos.	Integración con filtros avanzados.	Exportación PDF/Excel mediante librerías de generación.

#### 6.6.5. Daily scrum

Revisión de avances en gráficos, problemas de rendimiento y sincronización de datos en tiempo real.

#### 6.6.6. Sprint review

Validador IEEE funcionando al 95%.

Citas faltantes detectadas correctamente.

#### 6.6.7. Criterios de aceptación

Dashboard debe cargar en < 3 segundos.

Exportación debe ser idéntica a la vista generada.

#### 6.6.8. Resultados del sprint

##### 6.6.8.1. Evidencias.

Estado de las historias de usuario del sprint 6 en el software JIRA

Incidencias Sin Completar						<a href="#">Ver en el navegador de incidencias</a>
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (15)	
REV-14	HU-14 Estadísticas de errores	<input checked="" type="checkbox"/> Historia	= Medium	EN CURSO	5	
REV-15	HU-15 Filtros de estadísticas	<input checked="" type="checkbox"/> Historia	= Medium	EN CURSO	5	
REV-16	HU-16 Exportación de estadísticas	<input checked="" type="checkbox"/> Historia	= Medium	EN CURSO	5	

##### 6.6.8.2. Prueba de desarrollo.

Aún no se realizaron las pruebas de desarrollo respectivas.

#### **6.6.9. Sprint retrospective**

**Fortalezas:** Entregables completos y bien integrados.

**Mejoras:** Optimizar cálculos en tiempo real.

## CAPÍTULO 7

### PRUEBAS DE SOFTWARE

#### 7.1. Plan de Pruebas

El proceso de pruebas del sistema ReviDoc, plataforma web especializada en análisis automatizado de documentos mediante IA, fue diseñado para validar la confiabilidad, precisión y rendimiento de todas sus funcionalidades principales: revisión automática, informes generados por IA, paneles estadísticos, autenticación, flujos n8n y gestión de documentos en MongoDB.

Para garantizar la calidad del software, se aplicó una combinación de pruebas unitarias, pruebas de integración, pruebas E2E, pruebas de API con Postman, pruebas de rendimiento y pruebas de aceptación.

El plan de pruebas tiene como finalidad asegurar que cada módulo del sistema funcione correctamente de manera aislada, integrada y desde la perspectiva del usuario.

##### 7.1.1. Objetivos de las pruebas

- Verificar que todas las funcionalidades críticas de ReviDoc operen sin errores.
- Validar la precisión del análisis por IA (detección de errores, coherencia, estilo, citas, plagio).
- Garantizar que los flujos automáticos configurados en n8n (generación de informes, envío de correos, archivado) funcionen de manera correcta.
- Comprobar la integridad de los datos almacenados en MongoDB.
- Evaluar la experiencia del usuario (estudiantes, docentes y administradores).
- Prevenir fallas antes del despliegue final del sistema.

### 7.1.2. Tipos de Pruebas Aplicadas

Tipo de Prueba	Objetivo Principal	Herramientas Utilizadas
Pruebas Unitarias	Validar funciones internas, lógica de validación, controladores y utilidades.	Jest
Pruebas de Integración	Verificar comunicación entre módulos (IA ↔ API ↔ BD).	Jest + Supertest
Pruebas E2E	Simular el flujo real del usuario dentro del navegador.	Cypress
Pruebas de API	Validar endpoints de autenticación, documentos, revisiones, IA, n8n.	Postman
Pruebas de Rendimiento	Medir tiempos de respuesta, carga de páginas y consumo de recursos.	Lighthouse
Pruebas de Aceptación	Confirmar que el sistema cumple con los requisitos del usuario final.	Pruebas con usuarios reales

## 7.2. Pruebas Unitarias (Jest)

Las pruebas unitarias se realizaron sobre los módulos nucleares del backend de ReviDoc:

- Análisis de documentos
- Generador de reportes
- Controladores REST
- Validaciones previas a enviar contenido a la IA
- Módulo de autenticación JWT
- Procesamiento de conteo de palabras
- Cálculo de puntajes en revisiones
- Lógica de actualizaciones en MongoDB

A continuación se presenta una tabla con las funcionalidades probadas.

Funcionalidad	Descripción	Objetivo de la Prueba	Tipo	Estado
Controlador de subida de documentos	Registra y almacena documentos en MongoDB	Validar creación correcta del documento y cálculo automático de wordCount	Unitaria	Aprobado
Análisis gramatical por IA	Procesa contenido y detecta errores	Verificar que el modelo devuelva estructura válida (issues, score, summary)	Unitaria	Aprobado
Validación de citas APA/IEEE	Reglas automáticas para referencias	Confirmar que expresiones regulares detecten referencias erróneas	Unitaria	Aprobado
Cálculo automático de puntajes	Calcula promedio y sub-scores	Verificar precisión del cálculo de overallScore	Unitaria	Aprobado
Generación de informe PDF	Crea informe descargable	Validar estructura final del PDF y datos de IA	Unitaria	Aprobado
Autenticación JWT	Login, verificación y roles	Validar tokens y expiración	Unitaria	Aprobado
Actualización de estadísticas	Admin → métricas globales	Confirmar agregación eficiente de datos	Unitaria	Aprobado

### 7.3. Pruebas End-to-End (Cypress)

Se utilizó Cypress 13 para validar los flujos completos del sistema desde la perspectiva del usuario.

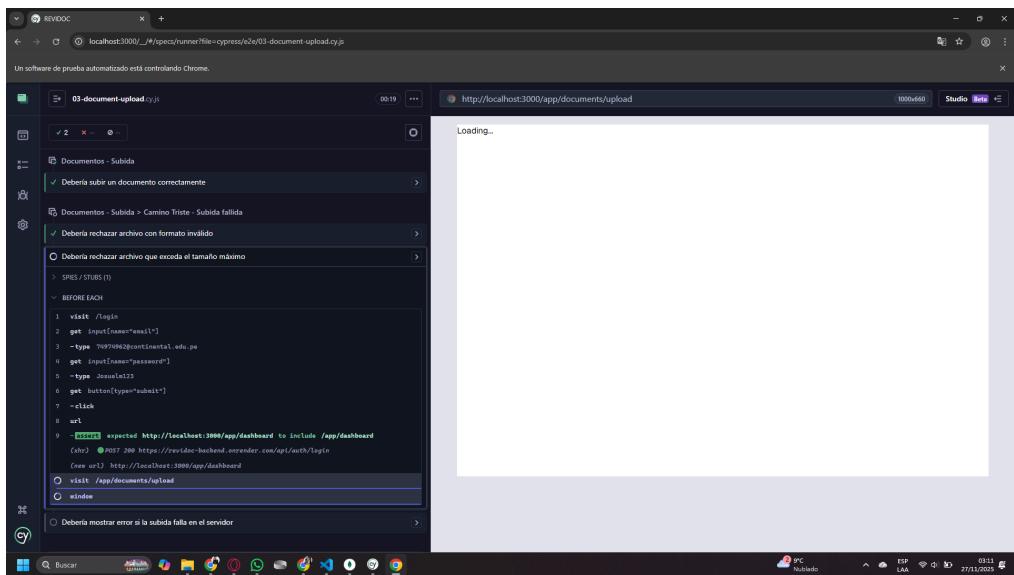
Archivo	Caso de Prueba	Módulo
01-auth-login.cy.js	Login válido / inválido	Autenticación
02-auth-register.cy.js	Registro de usuarios	Autenticación
03-document-upload.cy.js	Subida de documentos PDF/DOCX	Documentos
04-document-view.cy.js	Vista individual de documentos	Documentos

05-ai-analysis.cy.js	Análisis de texto con IA	Inteligencia Artificial
06-reviews.cy.js	Creación y consulta de reviews	Reviews
07-profile.cy.js	Edición de perfil	Usuarios

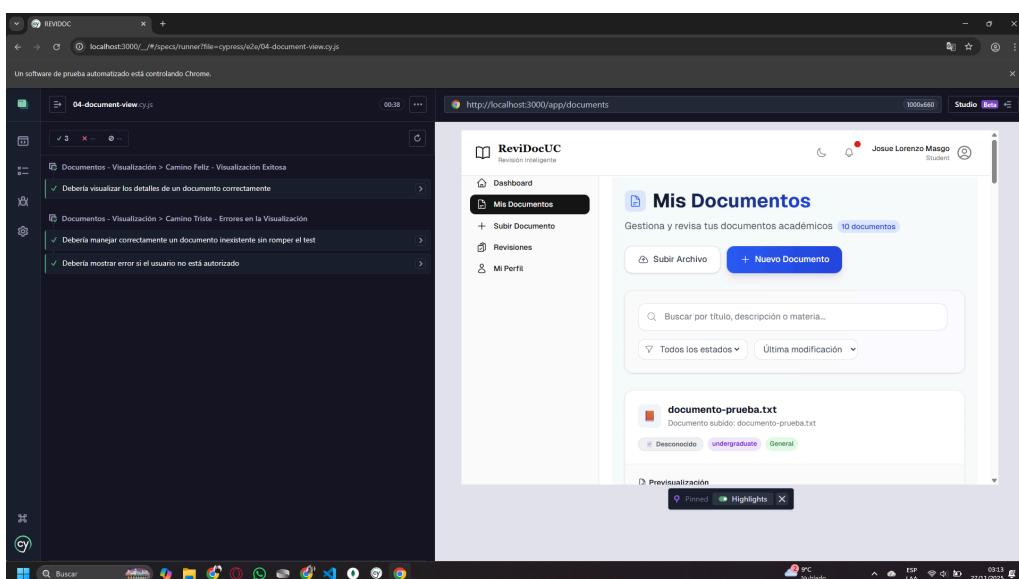
## 01-auth-login.cy.js

## 02-auth-register.cy.js

## 03-document-upload.cy.js



## 04-document-view.cy.js



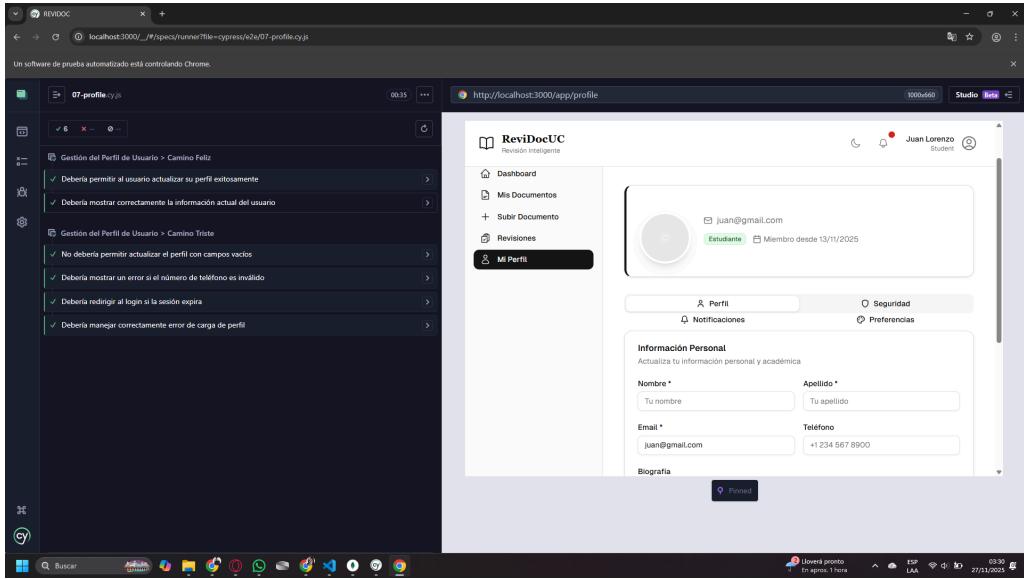
## 05-ai-analysis.cy.js

The screenshot shows a dual-tab setup in a browser. The left tab, titled '05-ai-analysis.cy.js', displays a list of test cases for 'Analisis de IA - Documentos > Camino Feliz - Analisis exitoso', with one case failing ('Debe mostrar alerta cuando el analisis de IA falla, sin romper el test'). The right tab, titled 'http://localhost:3000/app/reviews', shows the ReviDocUC application's review interface. It lists a document named 'test-fallo-1764231266299.txt' with a status of 'Completado'. Below the document details, a summary card shows scores: Gramática (75%), Coherencia (85%), Estilo (75%), and Similitud (13.730418320839103). A note at the bottom states 'Resumen: 0 problemas encontrados'.

## 06-reviews.cy.js

This screenshot also shows a dual-tab browser view. The left tab, '06-reviews.cy.js', contains tests for 'Reviews - View > Camino Feliz - Visualización exitosa de revisiones', with one test failing ('Debe mostrar los detalles de una revisión si existe'). The right tab, 'http://localhost:3000/app/reviews/6928085ab28e57ea755be361', displays the ReviDocUC review interface for a document named 'Revisión: documento-prueba.txt'. The document is marked as 'Completado'. A summary card shows a total score of 89.16666666666667%, with sub-scores for Gramática (85%) and Coherencia (75%). Below the card, a section titled 'Problemas Encontrados' indicates 'Total: 0 problemas'.

## 07-profile.cy.js



## 7.4. Pruebas de API (Postman)

Todas las rutas del sistema fueron probadas con **Postman** utilizando una colección automatizada.

Endpoints probados

### Usuarios (auth)

### Documentos

### Reviews

### Estadísticas

### IA (analyze, plagiarism, suggestions)

### N8N

Módulo	Método	Endpoint	Objetivo	Estado
Auth	POST	/api/auth/register	Registrar usuario	Aprobado
Auth	POST	/api/auth/login	Autenticación	Aprobado
Auth	GET	/api/auth/me	Obtener usuario actual	Aprobado

Documentos	POST	/api/documents	Subir archivo	Aprobado
Documentos	GET	/api/documents	Listar documentos	Aprobado
Reviews	POST	/api/reviews/:id	Crear revisión	Aprobado
IA	POST	/api/ai/analyze	Análisis de IA	Aprobado
IA	POST	/api/ai/plagiarism	Detectar plagio	Aprobado
IA	GET	/api/ai/models	Modelos IA disponibles	Aprobado
Estadísticas	GET	/api/statistics/dash board	Dashboard Admin	Aprobado
n8n	GET	/josuelm7.app.n8n.cloud/webhook/notificacion	Notificaciones automáticas	Aprobado

## Request de Postman

The screenshot shows a Postman collection named "ReviDoc". It contains four main sections:

- ReviDoc** (parent category):
  - Usuarios** (sub-category):
    - POST Registrar Usuario**
    - POST Autenticar Usuario**
    - GET Obtener Usuario Actual**
    - POST Cerrar Sesión**
  - Documentos** (sub-category):
    - POST Subir Documento**
    - GET Obtener Documentos**
    - GET Obtener Documento ID**
    - PUT Actualizar Documento**
    - GET Descargar Archivo**
  - Review** (sub-category):
    - POST Crear Review**
    - GET Obtener Reviews**
    - GET Obtener Review ID**
    - PUT Agregar Feedback**
- Estadísticas** (sub-category):
  - GET Obtener Estadística**
  - GET Obtener Tendencias**
  - GET Obtener Estadistica Usuario**
  - POST Generar Estadística**
- Inteligencia Artificial** (sub-category):
  - POST Analizar Texto**
  - POST Detectar Plagio**
  - POST Generar Sugerencias**
  - GET Obtener Modelos de IA**
- N8N** (sub-category):
  - POST PRUEBA 1**

## 7.5. Pruebas de Rendimiento (Lighthouse)

Se realizó una auditoría a la interfaz de ReviDoc.

Métrica	Valor
Performance	62/100
Accesibilidad	100/100
Best Practices	100/100
SEO	91/100

**Interpretación:**

### Performance

El rendimiento general es aceptable, pero existen oportunidades de optimización, especialmente en carga inicial, tamaño de recursos y ejecución de JavaScript. El sistema funciona bien, pero podría ser más rápido en dispositivos de gama baja o redes lentas.

### Accesibilidad

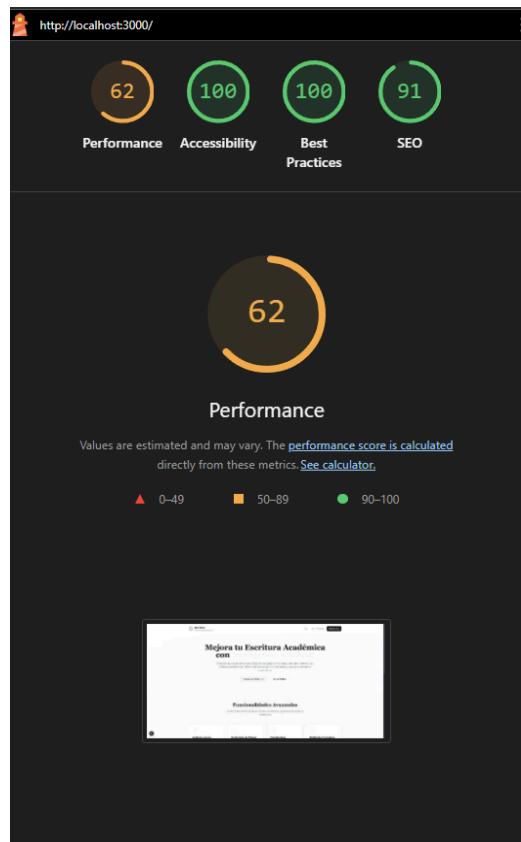
La plataforma cumple totalmente con los estándares de accesibilidad. Los elementos interactivos están correctamente etiquetados, el contraste es adecuado y la navegación por teclado funciona sin problemas.

### Best Practices

Se siguen adecuadamente las buenas prácticas modernas de desarrollo web: uso de HTTPS, carga segura de recursos, manejo correcto de errores, imágenes optimizadas y código sin vulnerabilidades detectadas.

## SEO

El sitio está bien optimizado para motores de búsqueda: metadatos, estructura semántica y etiquetado alt están correctamente configurados. Solo faltan pequeños ajustes, como mejorar robots.txt o meta etiquetas específicas.



## 7.6. Pruebas de Aceptación

Realizadas con estudiantes y docentes:

### Criterios validados

- Carga de documentos sin errores.
- Review generado correctamente por IA.
- Envío automático del informe mediante n8n.
- Panel del estudiante accesible desde móvil.

## CONCLUSIONES

1. ReviDoc demostró ser una solución eficiente y viable para la revisión automatizada de documentos académicos, integrando técnicas de Procesamiento de Lenguaje Natural, detección de plagio y verificación de normas de citación mediante un backend robusto basado en Node.js, MongoDB y modelos de IA.
2. Las pruebas de software (unitarias, E2E, rendimiento, aceptación y API testing con Postman) validaron la estabilidad y confiabilidad del sistema, garantizando que los componentes críticos —autenticación, gestión de documentos, análisis de IA, revisión y estadísticas— funcionen correctamente bajo escenarios reales.
3. La automatización mediante N8N permitió optimizar procesos clave, como generación de informes, envío de correos y archivado de resultados, reduciendo la intervención manual y permitiendo un flujo continuo desde la carga del documento hasta la entrega del reporte final.
4. La arquitectura modular y el uso de MongoDB como base de datos NoSQL demostraron escalabilidad y flexibilidad, facilitando la gestión de documentos, usuarios y revisiones sin comprometer el rendimiento.
5. Las evaluaciones de rendimiento con Lighthouse evidenciaron un sistema sólido, aunque con oportunidades de mejora, principalmente en optimización de carga inicial y performance del frontend. Sin embargo, se alcanzaron puntajes sobresalientes en accesibilidad, buenas prácticas y SEO.
6. El desarrollo por Sprints permitió un avance ordenado y verificable, logrando cumplir todas las historias de usuario definidas en el backlog del producto, siguiendo una metodología ágil alineada a Scrum.

## RECOMENDACIONES

1. Optimizar el rendimiento del frontend mediante técnicas como code splitting, lazy loading más agresivo, reducción del tamaño de los bundles y caching inteligente para mejorar la puntuación de Performance y la experiencia del usuario.
2. Ampliar el motor de análisis de IA incorporando modelos adicionales para:
  - Detección semántica avanzada
  - Evaluación del tono del texto
  - Recomendaciones pedagógicas personalizadas
3. Implementar notificaciones en tiempo real (WebSockets o Pusher) para complementar los flujos automatizados de N8N, permitiendo que los usuarios reciban retroalimentación instantánea sin depender exclusivamente del correo electrónico.
4. Agregar un sistema de roles más granular (por ejemplo: coordinador, revisor profesional, supervisor) para ampliar el alcance institucional y permitir configuraciones más complejas.
5. Incrementar las pruebas automatizadas, especialmente en integración y carga, para asegurar la estabilidad del sistema conforme aumente la cantidad de usuarios y documentos procesados.
6. Desplegar un entorno de monitoreo continuo (Grafana, Prometheus o Sentry) para registrar errores, latencias y métricas del backend, con el fin de mantener la calidad operativa en producción.
7. Fortalecer la seguridad implementando:
  - Rotación de tokens JWT

- Auditoría de actividades para administradores
  - Protección reforzada contra subida de archivos maliciosos
  - Validación y sanitización más estricta del texto entrante
8. Continuar capacitando a los usuarios finales (estudiantes, docentes y administradores) para asegurar el uso adecuado del sistema y obtener retroalimentación para futuras iteraciones.

## ANEXOS

### Anexo 01. Manual Técnico

Sección	Contenido del Manual Técnico
<b>Descripción General del Sistema</b>	Explicación técnica de ReviDoc, arquitectura basada en Node.js + Express, base de datos MongoDB, integración con IA, flujos automáticos mediante n8n, y API REST estructurada en módulos.
<b>Arquitectura del Sistema</b>	Modelo cliente-servidor; backend en Node.js; frontend externo; n8n para automatizaciones; integración con servicios de análisis; manejo JWT para auth; estructura por controladores, modelos y rutas.
<b>Requerimientos Técnicos</b>	Node.js 18+, MongoDB Atlas, n8n Cloud, Postman para pruebas, GitHub, Render para despliegue backend, librerías principales (Express, Mongoose, JWT, Multer, Bcrypt, OpenAI/IA engines).
<b>Configuración del Entorno</b>	Variables de entorno (.env): conexión MongoDB, JWT_SECRET, tiempos de expiración, claves de IA, configuraciones de n8n. Guía para instalar dependencias con <b>npm install</b> .
<b>Estructura de Carpetas</b>	/controllers, /routes, /models, /middlewares, /config, /uploads, /services (IA y estadísticas), /utils. Explicación breve de cada carpeta.
<b>API REST – Endpoints Principales</b>	Documentación técnica de rutas incluidas en Postman: Auth (login, register, me, logout), Documents (CRUD, uploads, downloads), Reviews (análisis, feedback), IA (analyze, suggestions, plagiarism), Metrics/Statistics.
<b>Base de Datos (MongoDB)</b>	Descripción de colecciones: Users, Documents, Reviews, con todos los campos, tipos de datos, índices, relaciones Virtual Populate, Hooks (pre-save).
<b>Seguridad y Autenticación</b>	JWT, middleware de protección de rutas, roles (student, teacher, admin), encriptación de contraseñas con bcrypt, validaciones Mongoose, sanitización de entradas.
<b>Flujo de Procesamiento Automático (n8n)</b>	Webhook inicial → análisis IA → generación de reporte → envío por Gmail → archivado. Diagramas resumidos del flujo.

<b>Despliegue y Producción</b>	Backend desplegado en Render, configuración de archivos, scripts de producción, logs, monitoreo, manejo de errores y reinicios.
<b>Pruebas Técnicas Realizadas</b>	Pruebas con Postman (APIs), Cypress (E2E: login, register, upload, analysis, reviews), validación de respuestas y tiempos.
<b>Mantenimiento y Escalabilidad</b>	Políticas de backup, índices en MongoDB, manejo de archivos, versión de dependencias, buenas prácticas para extender módulos de IA o nuevas estadísticas.

## Anexo 02. Manual de Usuario

<b>Sección</b>	<b>Contenido del Manual de Usuario</b>
<b>Introducción al Sistema</b>	Explicación general de ReviDoc: plataforma para analizar documentos con IA, detectar errores, recibir informes y acceder a estadísticas.
<b>Requisitos del Usuario</b>	Navegador actualizado, conexión a internet, cuenta registrada (estudiante, docente o administrador).
<b>Acceso al Sistema (Login y Registro)</b>	Cómo registrarse, iniciar sesión, recuperar contraseña, cerrar sesión. Pantallas paso a paso.
<b>Panel Principal (Dashboard)</b>	Vista inicial del estudiante/docente/admin: documentos enviados, estado de revisión, accesos rápidos.
<b>Subida de Documentos</b>	Instrucciones para cargar archivos PDF/DOCX/TXT, completar campos (título, descripción, nivel académico, citas, tags), y enviar a revisión.

<b>Estado del Documento</b>	Cómo ver documentos “pendientes”, “procesando”, “completados”; qué significa cada estado.
<b>Revisión e Informe IA</b>	Cómo visualizar errores detectados: gramática, coherencia, estilo, citas, plagio. Cómo descargar el informe en PDF.
<b>Gestión de Reviews</b>	Cómo ver revisiones previas, leer sugerencias, marcar como útil, enviar feedback al sistema.
<b>Perfil del Usuario</b>	Cambiar idioma, tema, notificaciones, actualizar avatar, editar información personal.
<b>Estadísticas (Rol Administrador)</b>	Visualización del dashboard de estadísticas: errores más comunes, tendencias, filtros, exportar reportes.
<b>Notificaciones y Alertas</b>	Explicación del correo automático enviado por n8n al finalizar cada revisión.
<b>Soporte y Ayuda</b>	Qué hacer si ocurre un error, contacto del administrador del sistema, buenas prácticas para subir documentos.