# UNIVERSIDAD TÉCNICA DEL NORTE

## FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

## CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

### TRABAJO DE INTEGRACIÓN CURRICULAR

**TEMA:**

"DESARROLLO DE UN SOFTWARE PARA LA EXTRACCIÓN Y ANÁLISIS DE TEXTO DE DOCUMENTOS PDF CON EL USO DE IA GENERATIVA PARA OPTIMIZAR LA GESTIÓN DOCUMENTAL"

Trabajo de titulación previo a la obtención del título de Ingeniera en Tecnologías de la Información.

**Línea de investigación:** Desarrollo, aplicación de software y cyber security (seguridad cibernética).


**AUTOR:**

Johanna Avigail Guamán Granda

**DIRECTOR:**

Msc. Pablo Andrés Landeta López


**Ibarra – Ecuador 2026**

# UNIVERSIDAD TÉCNICA DEL NORTE
## DIRECCIÓN DE BIBLIOTECA

## AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

### 1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

| DATOS DE CONTACTO | |
|---|---|
| **APELLIDOS Y NOMBRES:** | Guamán Granda Johanna Avigail |

| DATOS DE LA OBRA | |
|---|---|
| **TÍTULO:** | DESARROLLO DE UN SOFTWARE PARA LA EXTRACCIÓN Y ANÁLISIS DE TEXTO DE DOCUMENTOS PDF CON EL USO DE IA GENERATIVA PARA OPTIMIZAR LA GESTIÓN DOCUMENTAL |
| **AUTOR:** | Guamán Granda Johanna Avigail |
| **FECHA:** | 25/08/2026 |
| SOLO PARA TRABAJOS DE GRADO | |
| **PROGRAMA:** | ☒ **PREGRADO** ☐ **POSGRADO** |
| **TITULO POR EL QUE OPTA:** | Ingeniera en Tecnologías de la Información |
| **ASESOR /DIRECTOR:** | Msc. Caranqui Sánchez Víctor Manuel / Msc. Landeta López Pablo Andrés |

### 2. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 25 días del mes de agosto de 2026.

**EL AUTOR:**

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
JOHANNA AVIGAIL GUAMAN GRANDA

……
Johanna Avigail Guamán Granda

# CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 25 de agosto de 2026.

Msc. Pablo Landeta
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:
Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
PABLO ANDRES LANDETA LOPEZ

*Msc. Pablo Landeta*

*C.C: 1002161055*

# APROBACIÓN DEL COMITÉ CALIFICADOR

El Comité Calificado del trabajo de Integración Curricular “ Desarrollo de un software para la extracción y análisis de texto de documentos PDF con el uso de IA generativa para optimizar la gestión documental ” elaborado por Johanna Avigail Guamán Granda , previo a la obtención del título del INGENIERA EN TECNOLOGÍAS DE LA INFORMACIÓN , aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
PABLO ANDRES
LANDETA LOPEZ

*Msc. Pablo Landeta*

*C.C: 1002161055*

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
VICTOR MANUEL
CARANQUI SANCHEZ

*Msc. Víctor Caranqui*

*C.C: 1002006508*

## DEDICATORIA

A mi esposo, por ser siempre mi mayor apoyo, por acompañarme durante todo este proceso con paciencia, comprensión y mucho amor. Gracias por estar a mi lado en cada desvelo, por animarme cuando sentía cansancio y por confiar en mí incluso cuando yo dudaba de mis propias fuerzas. Este logro también es tuyo, porque tu apoyo fue fundamental para llegar hasta aquí.

A mi madre y a mi abuelita, por creer en mí incluso en los momentos en los que el camino parecía complicado. Gracias por cada consejo, por sus palabras de aliento y por enseñarme, con su ejemplo, que el esfuerzo y la perseverancia pueden llevarnos muy lejos. Su amor incondicional fue la fuerza que me motivó a seguir adelante y a no rendirme nunca.

A mi hermano, mi compañero de clases y de tantas experiencias universitarias, gracias por compartir conmigo esta etapa llena de aprendizajes, retos y recuerdos inolvidables. Tu compañía y apoyo hicieron que cada dificultad fuera más fácil de enfrentar y cada meta alcanzada tuviera un significado aún más especial.

Y a todos ustedes, gracias por convertirse en mi inspiración y en la razón para seguir luchando por mis sueños. Este trabajo está dedicado con todo mi cariño y gratitud, porque cada logro que alcanzo también les pertenece.

Johanna Avigail Guamán Granda

## AGRADECIMIENTO

En primer lugar, agradezco a Dios por brindarme salud, fortaleza, sabiduría y perseverancia para culminar esta importante etapa de mi vida académica. Su guía y bendición fueron fundamentales para superar cada desafío presentado durante el desarrollo de esta investigación.

Expreso mi más sincero agradecimiento a mi familia, quienes han sido mi principal fuente de apoyo, motivación y confianza. Gracias por acompañarme en cada momento, por su comprensión durante las largas jornadas de trabajo y por impulsarme siempre a seguir adelante con esfuerzo y dedicación.

De manera especial, extiendo mi profundo reconocimiento al Msc. Landeta López Pablo Andrés, director de tesis, por su orientación, conocimientos, compromiso y valioso acompañamiento durante todo el proceso investigativo. Sus aportes y recomendaciones fueron fundamentales para el desarrollo y culminación de este trabajo.

Asimismo, agradezco a mi asesor de tesis, Msc. Caranqui Sánchez Víctor Manuel, por su apoyo académico, disposición y colaboración constante, contribuyendo significativamente al fortalecimiento de esta investigación.

A los docentes de la carrera, gracias por compartir sus conocimientos, experiencias y enseñanzas a lo largo de mi formación universitaria. Cada aprendizaje adquirido ha sido esencial para mi crecimiento profesional y personal.

Finalmente, expreso mi gratitud a la institución universitaria y a todas las personas que, de manera directa o indirecta, contribuyeron con su apoyo académico, profesional y humano para la realización de esta tesis. Cada aporte recibido fue valioso para alcanzar esta meta tan importante en mi vida.

Johanna Avigail Guamán Granda

# ÍNDICE DE CONTENIDOS

DEDICATORIA . . . . . II

AGRADECIMIENTO . . . . . III

ÍNDICE DE FIGURAS . . . . . X

ÍNDICE DE TABLAS . . . . . XII

RESUMEN . . . . . XIII

ABSTRACT . . . . . XIV

CAPÍTULO I

1.1 Planteamiento del Problema . . . . . 1

1.2 Formulación del problema . . . . . 2

1.3 Objetivos . . . . . 3

1.3.1 Objetivo General . . . . . 3

1.3.2 Objetivos Específicos . . . . . 3

1.4 Alcance y delimitación . . . . . 4

1.5 Justificación . . . . . 5

CAPÍTULO II

2.1 Gestión documental digital . . . . . 8

2.2 Importancia en entornos organizacionales . . . . . 9

2.3 Normativas ISO aplicables . . . . . 9

2.4 Desafíos en digitalización de documentos escaneados (enfocado en PDF) . . . 11

2.5 Requisitos funcionales para sistemas de gestión documental . . . . . 12

2.6 Documentos PDF como objeto de procesamiento . . . . . 14
2.6.1 Características técnicas del formato PDF . . . . . 14
2.6.2 Creación de archivos PDF . . . . . 16
2.7 Tipos de PDF: nativos vs. escaneados . . . . . 17
2.7.1 PDF Nativo . . . . . 17
2.7.2 PDF escaneado . . . . . 17
2.8 Limitaciones para extracción de texto . . . . . 18
2.9 Inteligencia Artificial aplicada a documentos . . . . . 19
2.10 Evolución de modelos para procesamiento documental . . . . . 20
2.11 Aprendizaje Automático (Machine Learning) en análisis de texto . . . . . 22
2.11.1 Tipos de aprendizaje automático . . . . . 23
2.12 Procesamiento de Lenguaje Natural (NLP) para gestión semántica . . . . . 25
2.13 Inteligencia Artificial Generativa (GenAI) . . . . . 26
2.14 Fundamentos y arquitecturas de modelos generativos . . . . . 27
2.15 Grandes Modelos de Lenguaje (LLMs) aplicados a documentos . . . . . 28
2.16 Casos de uso en extracción y estructuración de datos . . . . . 30
2.17 Tecnologías de extracción de texto . . . . . 31
2.17.1 Reconocimiento Óptico de Caracteres (OCR) avanzado . . . . . 31
2.17.2 Tesseract . . . . . 32
2.17.3 ABBYY FineReader . . . . . 32
2.17.4 OCRopy . . . . . 33
2.17.5 Calamari . . . . . 33
2.17.6 Cloud Vision . . . . . 34
2.18 Técnicas híbridas: OCR + IA Generativa . . . . . 34
2.19 Métricas de evaluación de precisión . . . . . 35

2.19.1 F1-score . . . . . 35
2.19.2 Word Error Rate - WER . . . . . 36
2.19.3 Character Error Rate - CER . . . . . 37
2.20 Desarrollo de software para gestión documental . . . . . 37
2.20.1 Arquitectura de sistemas IA para procesamiento de PDF . . . . . 38
2.21 Backend . . . . . 41
2.22 Django (Python) . . . . . 41
2.23 Application Programming Interfaces - APIs . . . . . 42
2.23.1 FastAPI . . . . . 43
2.23.2 Flask . . . . . 44
2.23.3 Almacenamiento (vectores) . . . . . 45
2.24 Diseño de interfaz web para carga/visualización/búsqueda (UI/UX) . . . . . 46
2.24.1 Búsqueda semántica . . . . . 46
2.24.2 La experiencia de usuario UX . . . . . 47
2.24.3 Interfaz de usuario UI . . . . . 47
2.25 Frameworks . . . . . 47
2.25.1 LangChain/LLamaIndex . . . . . 48
2.25.2 Streamlit/Gradio (UI) . . . . . 49
2.26 PyPDF/PdfPlumber . . . . . 50
2.27 Evaluación de sistemas basados en IA . . . . . 51
2.28 Métodos para medir precisión en extracción de texto . . . . . 51
2.29 Sistema de reconocimiento de texto . . . . . 53
2.30 Pruebas de eficiencia (tiempo de respuesta, escalabilidad) . . . . . 56
2.31 Benchmarks en documentos de calidad variable . . . . . 57

CAPÍTULO III

3.1 Tipo de Investigación . . . . . 59
3.1.1 Enfoque Cuantitativo. . . . . 59
3.2 Metodología de Desarrollo . . . . . 60
3.2.1 Justificación de la Metodología en Cascada . . . . . 61
3.2.2 Fases del modelo en cascada. . . . . 62
3.3 Variables de Estudio . . . . . 67
3.3.1 Variable Independiente . . . . . 67
3.3.2 Variables Dependientes . . . . . 68
3.4 Materiales, Equipos y Recursos Tecnológicos . . . . . 68
3.4.1 Materiales . . . . . 68
3.4.2 Equipos . . . . . 69
3.4.3 Software utilizado . . . . . 70
3.4.4 Justificación del uso de recursos tecnológicos . . . . . 71
3.5 Documento de Requerimientos del Proyecto . . . . . 72
3.5.1 Requerimientos funcionales . . . . . 73
3.5.2 Requerimientos no funcionales . . . . . 75
3.6 Arquitectura del Sistema . . . . . 75
3.6.1 Componentes principales de la arquitectura . . . . . 76
3.6.2 Arquitectura cliente-servidor. . . . . 77
3.6.3 Flujo general de procesamiento documental. . . . . 78
3.7 Organización Modular del Sistema . . . . . 78

3.7.1 Fase de análisis . . . . . 79
3.7.2 Fase de Diseño . . . . . 80
3.7.3 Fase de Implementación . . . . . 81
3.7.4 Fase de Verificación y Evaluación . . . . . 82
3.8 Estructura del Software . . . . . 83
3.9 Seguridad del Sistema . . . . . 85
3.9.1 Validación de Archivos PDF . . . . . 85
3.9.2 Protección de Credenciales y Configuración Sensible . . . . . 87
3.9.3 Seguridad del Almacenamiento Documental . . . . . 88
3.9.4 Seguridad de la API y Control de Acceso . . . . . 88
3.9.5 Auditoría de Eventos de Seguridad . . . . . 89
3.10 Sistema de Evaluación Experimental . . . . . 89
3.10.1 Evaluación Objetiva mediante CER y WER . . . . . 89
3.10.2 Evaluación mediante Character Error Rate (CER) . . . . . 90
3.10.3 Evaluación mediante Word Error Rate (WER) . . . . . 90
3.10.4 Benchmarks de Procesamiento . . . . . 91
3.10.5 Evaluación Experimental mediante G-Eval . . . . . 94
3.10.6 Evaluación Humana . . . . . 94
3.11 Cronograma de Desarrollo . . . . . 96
CAPÍTULO IV
4.1 Arquitectura Implementada . . . . . 101
4.1.1 Componentes Principales . . . . . 101
4.2 Resultados de funcionamiento del sistema . . . . . 103

4.3 Evaluación de calidad mediante WER y CER . . . . . 104
4.3.1 Interpretación de las métricas . . . . . 106
4.3.2 Resultados de WER y CER . . . . . 107
4.4 Evaluación mediante G-Eval . . . . . 109
4.4.1 Evaluación G-Eval y Validación Humana . . . . . 112
4.5 Evaluación de rendimiento mediante benchmarks . . . . . 115
4.5.1 Análisis de tiempos de procesamiento del sistema . . . . . 117
4.6 Discusión de Resultados . . . . . 118
4.6.1 Interpretación de Métricas . . . . . 120
4.6.2 Factores de Éxito . . . . . 121
CONCLUSIONES . . . . . 122
RECOMENDACIONES . . . . . 124
BIBLIOGRAFÍA . . . . . 126
ANEXOS . . . . . 150

## ÍNDICE DE FIGURAS


Figura 1 Proceso de extracción automatizada de datos de PDF . . . . . 4

Figura 2 Visión del cambio . . . . . 10
Figura 3 Flujo de trabajo típico para la creación de un archivo PDF . . . . . 16
Figura 4 Muestra la correlación entre la inteligencia artificial, el aprendizaje automático y el aprendizaje profundo . . . . . 21
Figura 5 Diseño de la solución . . . . . 22
Figura 6 Midjourney imagen producida de IA generativa personificada . . . . . 27
Figura 7 Una visión de la IA generativa desde el punto de vista de los modelos, los sistemas y las aplicaciones . . . . . 28
Figura 8 Resumen del marco LitAI . . . . . 39
Figura 9 Arquitectura Interacción Inteligente de Documentos . . . . . 40
Figura 10 Arquitectura del sistema . . . . . 41
Figura 11 Arquitectura de Django . . . . . 43
Figura 12 Una aplicación puede tener múltiples interfaces, como una API web, una CLI, una interfaz web y una interfaz de escritorio . . . . . 44
Figura 13 Estructura de cómo funciona FastAPI . . . . . 45
Figura 14 Arquitectura de reconocimiento de texto . . . . . 53
Figura 15 Proceso de binarización en el reconocimiento de texto . . . . . 54

Figura 16 Modelo de desarrollo en cascada . . . . . 60
Figura 17 Estructura general del sistema . . . . . 73
Figura 18 Arquitectura general del sistema . . . . . 76
Figura 19 Flujo general de procesamiento documental . . . . . 79
Figura 20 Estructura de directorios del proyecto . . . . . 86
Figura 21 Ciclo de vida de la evaluación comparativa . . . . . 92

Figura 22 Arquitectura implementada del sistema . . . . . 102
Figura 23 Distribución de métricas CER y WER . . . . . 108
Figura 24 Matriz de correlación entre criterios de evaluación G-Eval aplicados al sistema de extracción documental. . . . . 111
Figura 25 Comparación entre evaluación humana normalizada y G-Eval. . . . . 114
Figura 26 Distribución estadística de tiempos de procesamiento entre el método baseline y el método propuesto. . . . . 117
Figura 27 Distribución de los tiempos del sistema . . . . . 118
Figura 28 Arquitectura del sistema . . . . . 150
Figura 29 Estructura de Directorios del Proyecto . . . . . 151
Figura 30 Inicialización del Sistema en Entorno Local . . . . . 151
Figura 31 Interfaz de Usuario: Pantalla Principal . . . . . 152
Figura 32 Ventana de carga de archivo . . . . . 152
Figura 33 Ventana emergente de procesamiento . . . . . 153

Figura 34 Visualización del resumen del documento procesado . . . . . 154
Figura 35 Interfaz de visualización del texto extraído . . . . . 154
Figura 36 Búsqueda de documentos por texto . . . . . 155
Figura 37 Búsqueda filtro por fecha del sistema . . . . . 156
Figura 38 Descarga desde la interfaz en formatos PDF, Word o txt . . . . . 156
Figura 39 Documentación interactiva Swagger/FastAPI . . . . . 157
Figura 40 Evaluación experimental G-Eval . . . . . 157
Figura 41 Almacenamiento local de documentos PDF . . . . . 158
Figura 42 Resultados de evaluación humana por criterio . . . . . 159
Figura 43 Comparación entre PDF original y texto extraído . . . . . 159
Figura 44 Configuración de variables de entorno del sistema . . . . . 160

# ÍNDICE DE TABLAS


| | | |
|---|---|---|
| Tabla I | Tipos de aprendizaje automático | 24 |
| Tabla II | Comparación de modelos | 30 |
| Tabla III | Tipos documentales y sus características | 69 |
| Tabla IV | Recursos de hardware utilizados en el desarrollo del sistema | 70 |
| Tabla V | Herramientas y tecnologías utilizadas en el desarrollo del sistema | 70 |
| Tabla VI | Requisitos Funcionales | 73 |
| Tabla VII | Requisitos No Funcionales | 75 |
| Tabla VIII | Estructura inicial de archivos y directorios del proyecto | 80 |
| Tabla IX | Estructura de módulos durante la fase de diseño | 80 |
| Tabla X | Servicios implementados durante la fase de implementación | 81 |
| Tabla XI | Estructura del frontend del sistema | 82 |
| Tabla XII | Scripts y directorios utilizados durante la fase de evaluación | 82 |
| Tabla XIII | Estructura general de directorios y componentes del sistema | 83 |
| Tabla XIV | Etapas del ciclo de vida de un benchmark | 93 |
| Tabla XV | Criterios utilizados en la evaluación G-Eval | 95 |
| Tabla XVI | Cronograma de actividades - Fase 1 | 97 |
| Tabla XVII | Cronograma de actividades - Fase 2 | 97 |
| Tabla XVIII | Cronograma de actividades - Fase 3 | 98 |
| Tabla XIX | Cronograma de actividades - Fase 4 | 99 |
| Tabla XX | Cronograma de actividades - Fase 5 | 100 |
| Tabla XXI | Procesos de normalización y limpieza aplicados durante la evaluación | 105 |
| Tabla XXII | Interpretación de resultados CER y WER | 106 |
| Tabla XXIII | Resultados estadísticos de evaluación CER y WER ( %) | 108 |
| Tabla XXIV | Resultados de evaluación mediante G-Eval | 109 |
| Tabla XXV | Comparación entre evaluación humana normalizada y G-Eval | 113 |
| Tabla XXVI | Métodos utilizados en la evaluación experimental | 115 |
| Tabla XXVII | Resultados estadísticos de tiempos de procesamiento | 116 |
| Tabla XXVIII | Resultados de evaluación CER y WER por documento | 160 |

# RESUMEN

Muchas organizaciones enfrentan una dificultad considerable cuando tienen que lidiar con gestión documental, particularmente cuando se trata de PDFs escaneados, que complican significativamente las tareas de búsqueda, organización y recuperación de datos relevantes. Ante esta situación, el proyecto buscó desarrollar una herramienta informática que permitiera procesar archivos PDF para identificar, extraer y comprender su contenido de texto mediante el uso de tecnología OCR e inteligencia artificial generativa. Para este proyecto, se decidió levantar una plataforma web que descansara sobre FastAPI en la parte del servidor, complementada con PostgreSQL para gestionar toda la información. En cuanto a la experiencia del usuario, se optó por Next.js, mientras que los archivos se guardaron directamente en servidores locales. La extracción de texto a partir de imágenes corrió a cargo de Mistral OCR, y finalmente Gemini se encargó de generar resúmenes de forma automática además de organizar y clasificar los documentos procesados. Esta plataforma permite subir archivos en PDF, verificar que sean válidos, procesar y limpiar el texto, visualizar página por página, generar resúmenes de los documentos, identificar su categoría, hacer búsquedas dentro del contenido y descargar reportes en varios formatos. Para evaluar el desempeño de la solución se aplicaron métricas como WER y CER, además de evaluación G-Eval, revisión humana y pruebas de rendimiento llevadas a cabo sobre 26 documentos. Los hallazgos indicaron un CER medio de 4.18 % y un WER medio de 7.83 %, acompañados de una puntuación G-Eval general de 0.7681, que superó el umbral establecido de 0.7. Se deduce que el sistema facilita optimizar la administración de documentos por medio de extracción y procesamiento automático, si bien es aconsejable potenciar la forma en que se representan las tablas de mayor complejidad.

**Palabras clave:** OCR, Inteligencia Artificial Generativa, OpenAI, G-Eval, WER, CER, PostgreSQL, SQLAlchemy.

## ABSTRACT

In their daily operations, many organizations manage scanned PDF documents that prevent users from searching, organizing, and analyzing the information contained within. To meet this need, the current study sought to develop software capable of extracting and analyzing text within PDF documents using OCR and generative artificial intelligence. Our approach involved building a web application with FastAPI as backend, PostgreSQL for database, Next.js as frontend framework, with Local storage, Mistral OCR (Optical Character Recognition) and Gemini. Mistral was used for extracting text from uploaded files. Gemini was used for generating summary of the document as well as identifying the type of document. Features of our application include: Uploading of PDF files, Validating the pdf files, Extracting text and cleaning it, Visualising the document page by page, Generating summary, Searching for content and Downloading the report. WER and CER scores, G-Eval metric, human metric, and test sets consisting of 26 samples were used to measure how good our solution is. We achieved a CER of 4.18 % on average and a WER of 7.83 % on average. We also have an overall G-Eval score of 0.7681 which is greater than our goal of >0.7. It was concluded that through automation of extracting and analysis, document management can be optimized; for further improvement, complex table representation should be addressed.



**Keywords:** OCR, Generative Artificial Intelligence, OpenAI, G-Eval, WER, CER, PostgreSQL, SQLAlchemy.

# CAPÍTULO I

# INTRODUCCIÓN

## 1.1 Planteamiento del Problema

En la actualidad muchas organizaciones enfrentan el desafío creciente de manejar grandes volúmenes de información contenida en documentos, como señala J. Smith et al. la cantidad de documentos disponibles en la actualidad es ingente, ya sea de documentos estructurados (tablas), semiestructurados (XML) o no estructurados (texto plano). Este volumen de documentos hace bastante ineficiente un procesamiento manual, debido al coste/persona y al tiempo [1].

Empresas grandes manejan diferentes tipos de documentos tanto impresos como digitales:

> ...la información contenida de estos documentos se presentan en múltiples formatos como ser archivos digitalizados tipo PDF o imágenes jpg, png, jpeg etc., pero también en papel. El origen de estos documentos es entre otros: archivos originales generados por entidades públicas o empresas terceras privadas, archivos emitidos in-house en empresas privadas, copias de archivos escaneados, fotocopias o incluso fotos tomadas con dispositivos móviles, etc. La diversidad de diseños y extensiones en los documentos hace que la extracción de información automática de forma genérica sea una tarea compleja [2] - [3]

.

La gestión documental, que históricamente había dependido de procesos manuales y trabajo humano intensivo, está viviendo ahora una transformación fundamental gracias al avance de las tecnologías digitales. Tal como se señala en [4], “la evolución tecnológica genera consecuencias directas en las labores del Archivo General corporativo, desde la clasificación y catalogación hasta la eliminación de documentos obsoletos”. Es por esto que la Inteligencia Artificial,

particularmente en su variante generativa, constituye un paso lógico hacia la automatización de funciones como la extracción, interpretación y clasificación documental, incrementando de este modo el potencial informativo de las instituciones.

En el informe "Chequeo Digital 2022–2023" da a conocer que el 78 % de las empresas ecuatorianas utilizan al menos un programa especializado, lo que sugiere un avance significativo en la adopción de tecnologías digitales. Como también, más del 50 % de las empresas aún se encuentran en un nivel inicial de madurez digital, especialmente en dimensiones como la analítica de datos y la automatización de procesos [5]. En base a los porcentajes antes mencionados se evidencia una oportunidad crítica para implementar soluciones avanzadas que optimicen procesos clave como la extracción y gestión de información documental.

Las instituciones se encuentran en necesidad de modernizarse en la gestión documental esto con el fin que puedan mejorar en la eficiencia y la accesibilidad a sus procesos internos, por lo que se propone abordar este problema mediante el desarrollo de un software basado en inteligencia artificial generativa usando herramientas de extracción de texto. Como bien dicen A. Churo et al., estos componentes permiten al modelo de predicción recibir documentos dados por el usuario y poder realizar la detección de los datos de interés para realizar la lectura de las imágenes por medio de tecnología OCR [6]. Esta solución permitirá convertir archivos de imagen en texto digital editable de forma automática, segura y precisa, mejorando notablemente la eficiencia de la gestión documental en las empresas.

## 1.2 Formulación del problema

¿Cómo podría un software de extracción de texto de documentos PDF basado en inteligencia artificial generativa optimizar la gestión documental?

Este proyecto de investigación se inscribe dentro del campo de Desarrollo, Aplicación de Software y Ciberseguridad, específicamente en la sublínea de Inteligencia Artificial. Nuestro objetivo central es aprovechar métodos y técnicas de inteligencia artificial generativa, conjuga-

dos con herramientas especializadas para extraer información de documentos en PDF, lo que permite transformar así archivos escaneados en datos estructurados y reutilizables. De este modo, buscamos optimizar los procesos de búsqueda, análisis y control de información digital, mejorando significativamente la eficiencia, precisión y autonomía en diversos escenarios y disciplinas académicas.

## 1.3 Objetivos

### 1.3.1 Objetivo General

Desarrollar un software basado en Inteligencia Artificial Generativa para la extracción, análisis y gestión automatizada de documentos en formato PDF, con el propósito de optimizar la organización, búsqueda y procesamiento de información en entornos de gestión documental.

### 1.3.2 Objetivos Específicos

- Analizar tecnologías y enfoques actuales basados en inteligencia artificial generativa (GenAI) aplicables a la extracción automatizada de información desde documentos PDF escaneados en el contexto de la gestión documental.

- Desarrollar un software que utilice técnicas de GenAI para extraer información no estructurada de documentos PDF escaneados y organizarla en un formato estructurado para facilitar su gestión.

- Evaluar la precisión, eficiencia y rendimiento del software desarrollado mediante métricas e indicadores específicos aplicables a sistemas basados en inteligencia artificial generativa.

## 1.4 Alcance y delimitación

Este trabajo se propone crear una aplicación web equipada con tecnología de inteligencia artificial generativa, capaz de extraer texto de documentos PDF que han sido escaneados. El objetivo fundamental es potenciar la accesibilidad y facilitar tanto la búsqueda como la recuperación de la información almacenada en estos archivos.

La plataforma incorporará herramientas específicas para recuperar texto proveniente de documentos escaneados, permitiendo que dicho contenido sea fácilmente copiado, editado o buscado sin complicaciones. De esta forma, los usuarios tendrán la oportunidad de trabajar con el texto de sus documentos en un formato que resulta práctico de manejar y que incrementa considerablemente su funcionalidad y rendimiento.

El diseño de la plataforma incluirá una interfaz intuitiva que hará que subir archivos, procesar la información y mostrar los resultados sea sencillo y accesible. Además, los usuarios podrán realizar búsquedas ágiles en los documentos para encontrar exactamente lo que necesitan, garantizando así que el trabajo sea cómodo y eficiente desde principio a fin. La exactitud y la capacidad de funcionamiento se evaluarán empleando documentos que presenten distintos niveles de calidad visual, garantizando así que el sistema responda adecuadamente en escenarios auténticos de gestión documental

A continuación en la Figura 1, se describe el proceso de cómo se extrae los datos de archivos PDF:

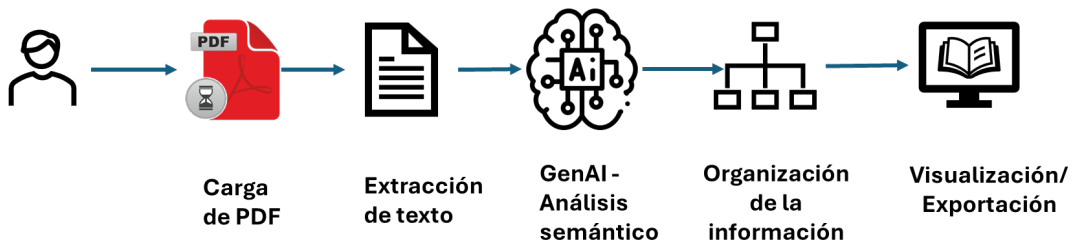


**Figura 1**. Proceso de extracción automatizada de datos de PDF

Para extraer los datos contenidos en un documento PDF, el usuario cargará el archivo al

software, luego la herramienta de extracción de texto analizará el contenido, lo convertirá automáticamente en datos estructurados y los procesará mediante inteligencia artificial generativa, permitiendo realizar análisis semántico, clasificación o generación de resúmenes automáticos. El objetivo es lograr que una organización lleve una óptima información digital donde permita una búsqueda de documentos segura, rápida, eficiente y escalable, pues este software mostrará el documento junto con el texto extraído, ofreciendo al usuario la posibilidad de copiar o editar la información obtenida.

## 1.5 Justificación

La gestión documental continúa siendo un reto importante en las organizaciones públicas y privadas. En [7] menciona que:

> En la actualidad gran parte de la información se encuentra almacenada en forma de texto, ya sea en papel o en formato digital. El texto impreso presenta limitantes como un mayor costo de producción, menor velocidad de difusión y menor alcance en relación con el texto en formato digital.

Los documentos en papel han logrado llegar a la digitalización mediante el escaneo y almacenamiento en formato PDF, gran parte de esta documentación permanece como imágenes no editables y su búsqueda del documento en papel resulta complicado, pues esto impide realizar consultas, verificaciones, ediciones y análisis automatizados.

A pesar de que la digitalización trae consigo ventajas ampliamente conocidas, la automatización en la gestión de documentos continúa siendo un terreno al que apenas se le presta la debida atención, obstaculizando así las posibilidades reales de concretar una verdadera transformación digital en toda su extensión [8]. Los trámites administrativos, las gestiones legales y los procesos operativos que dependen de los datos almacenados en expedientes (ya sean en formato papel o electrónico) se ven seriamente afectados por esta situación, generando pérdida de tiempo, costos evitables y una merma notable en la eficiencia de la organización.

En este escenario, la Inteligencia Artificial Generativa (GenAI) es una tecnología transformadora, con el potencial de mejorar los procesos de gestión documental, la IA generativa está impactando de manera directa el campo de la documentación, especialmente en tareas como la recuperación de información, la catalogación, la minería de datos y el análisis automatizado de documentos, gracias a su capacidad de interpretar texto, generar resúmenes y estructurar contenido no estructurado [9].

Este proyecto nace como una solución directa a esa necesidad específica, planteando la creación de un software basado en inteligencia artificial generativa capaz de realizar análisis semántico, clasificación y síntesis de contenidos. Además, incorpora tecnología OCR para extraer texto de documentos PDF escaneados, con el propósito de recuperar la información que contienen y transformarla en un formato reutilizable y editable, facilitando así su integración en sistemas de gestión documental.

Este proyecto aportará una solución práctica y real a la necesidad frecuente que se vive en instituciones, pues transformar documentos que hasta ahora solo podían ser leídos visualmente en archivos de texto que puedan ser consultados, editados o procesados automáticamente. Esta herramienta reducirá significativamente la dependencia de la transcripción manual, acelerando la búsqueda de los procesos administrativos, reduciendo tediosos procesos manuales y mejorando la productividad de los equipos de trabajo.

Las empresas públicas y particulares gozaran de este gran beneficio, especialmente aquellas que manejan grandes volúmenes de documentos físicos transformados a PDF. Esto incluye ministerios, municipios, entidades educativas públicas, hospitales y organismos de control, entre otros. Al implementar esta solución, podrán mejorar su eficiencia operativa, facilitar el acceso a la información institucional, cumplir con políticas de “Cero Papel” y avanzar en sus procesos de transformación digital [10].

Desde una perspectiva técnica y conceptual, esta iniciativa representa un aporte considerable para la ingeniería informática, pues gracias a la implementación de tecnologías de extracción de texto e inteligencia artificial generativa podremos automatizar tareas que actualmente se realizan

manualmente. Además, se trata de un enfoque moderno y consciente a la hora de administrar documentos, garantizando que estén protegidos y permitiendo que los registros se mantengan actualizados.

En el área académico e investigativo, la solución propuesta se puede utilizar como referencia para trabajos futuros que busquen ampliar las capacidades del sistema hacia la clasificación de documentos, integración con bases de datos documentales, entre otros. Además, demuestra que gracias a los saberes adquiridos durante la formación profesional pueden convertirse en proyectos aplicables, escalables y con impacto directo en empresas públicas o privadas.

Mirando esto desde el punto de vista social, la iniciativa brinda una claridad mucho mayor y hace que sea más sencillo para los ciudadanos acceder al contenido de los documentos, lo que les permite encontrar, verificar y revisar información con rapidez. Desde la óptica económica, reduce significativamente los costos que surgen de actividades manuales como la digitación, la impresión de documentos en papel, la búsqueda de expedientes en archivo físico y la administración de almacenes.

Políticamente, esto coloca a las instituciones bajo el cumplimiento de las leyes y reglamentaciones actuales que exigen la transformación digital y permitir el acceso a distancia de la información. En el ámbito educativo, puede funcionar como un instrumento en universidades para convertir a formato digital los documentos de los estudiantes, proyectos de grado, informes y archivos de la institución.

# CAPÍTULO II

# MARCO TEÓRICO

## 2.1 Gestión documental digital

Toda organización, ya sea pública o privada, contará con documentación como respaldo para evidenciar las actividades que en ella se desarrollan, ya que dicha documentación conservará información relevante y fundamental ante cualquier situación legal, administrativa o de control. Por eso, es fundamental que toda institución cuente con un adecuado manejo del almacenamiento y acceso a la información.

Según [11], la gestión documental se define como "la gestión de la información creada y recibida por una organización en el transcurso de sus actividades y el desarrollo de sus funciones". Así mismo Zambrano *et al.* [12], señala que "se considera responsable del control efectivo y sistemático, de la creación, recepción, mantenimiento, uso y disposición de documentos del área de gestión, incluido el proceso de fusión y mantenimiento de información y evidencia de documentos, como una operación organizacional".

Al mismo tiempo, es importante destacar que el valor documental contribuye significativamente a la gestión organizacional, al garantizar la calidad en los procesos internos, pues al contener un almacenamiento físico adecuado se obtendrá administrativamente un flujo documental durante todo su ciclo de vida. Cabe señalar que ningún documento oficial puede ser descartado, ya que estos son fundamentales para auditorías, análisis de procesos y toma de decisiones dentro de la organización [13].

En otras palabras, la gestión documental representa un gran desafío para las organizaciones, ya que el volumen de documentación aumenta progresivamente conforme crece la institución.

Ante esta situación, muchas organizaciones optaron por incorporar herramientas que simplificaran dicha tarea. Con el auge de la computación y el desarrollo de soluciones tecnológicas, los sistemas digitales de gestión documental se consolidaron rápidamente como la alternativa más eficiente. A partir de entonces, se comenzaron a integrar nuevas funcionalidades orientadas a optimizar el manejo, la organización y la protección de los documentos, en respuesta a las crecientes demandas del entorno actual [14].

## 2.2 Importancia en entornos organizacionales

Para gestionar eficientemente una gran cantidad de documentos e información, es fundamental contar con un sistema de gestión documental que facilite el almacenamiento, permita el acceso rápido a los documentos mediante búsquedas y mantenga la documentación organizada, significa que la implementación de este tipo de sistema resulta especialmente útil en organizaciones que operan en múltiples provincias, ya que permite centralizar tareas, almacenar información y clasificarla adecuadamente. Esto incluye documentos de salida como facturas electrónicas, formularios y otros tipos de registros, manteniendo todo integrado en una sola plataforma y gestionando tanto los documentos de entrada como los de salida de la institución [15].

Cabe destacar que la información y el conocimiento en la sociedad actual se sustentan en el crecimiento de las Tecnologías de la Información y la Comunicación (TIC), lo cual ha generado un impacto sin precedentes en el ámbito documental. Por esta razón, es fundamental que las organizaciones presten mayor atención a la forma en que organizan, administran y controlan sus fuentes documentales, sus flujos de información, así como a su recurso más valioso: el capital humano [16].

## 2.3 Normativas ISO aplicables

En el año 2001 se publicaron las primeras normas ISO relacionadas con la gestión documental, específicamente la ISO 15489-1 y la ISO 15489-2 [17]. A partir de ese momento, se han

desarrollado múltiples normas e informes técnicos que, tomando como base los conceptos iniciales, han contribuido a establecer un conjunto sólido de buenas prácticas para la implementación de la gestión de documentos electrónicos.

En [18] La Norma ISO 15489 Information and documentation Records Management (Información y Documentación ─ Gestión de Documentos), "es el resultado de un esfuerzo por parte de la comunidad archivística internacional por estandarizar la praxis archivística, en un momento de incertidumbre y crisis en cuanto a la capacidad de gestionar información veraz en las organizaciones".

Esta norma establece que los documentos son creados, recibidos y gestionados por las organizaciones en el desarrollo de sus actividades. Por ello, toda institución debe generar documentos originales, fiables y utilizables, que sirvan como respaldo de sus acciones, con el fin de cumplir con los requisitos legales y garantizar la debida rendición de cuentas [19].

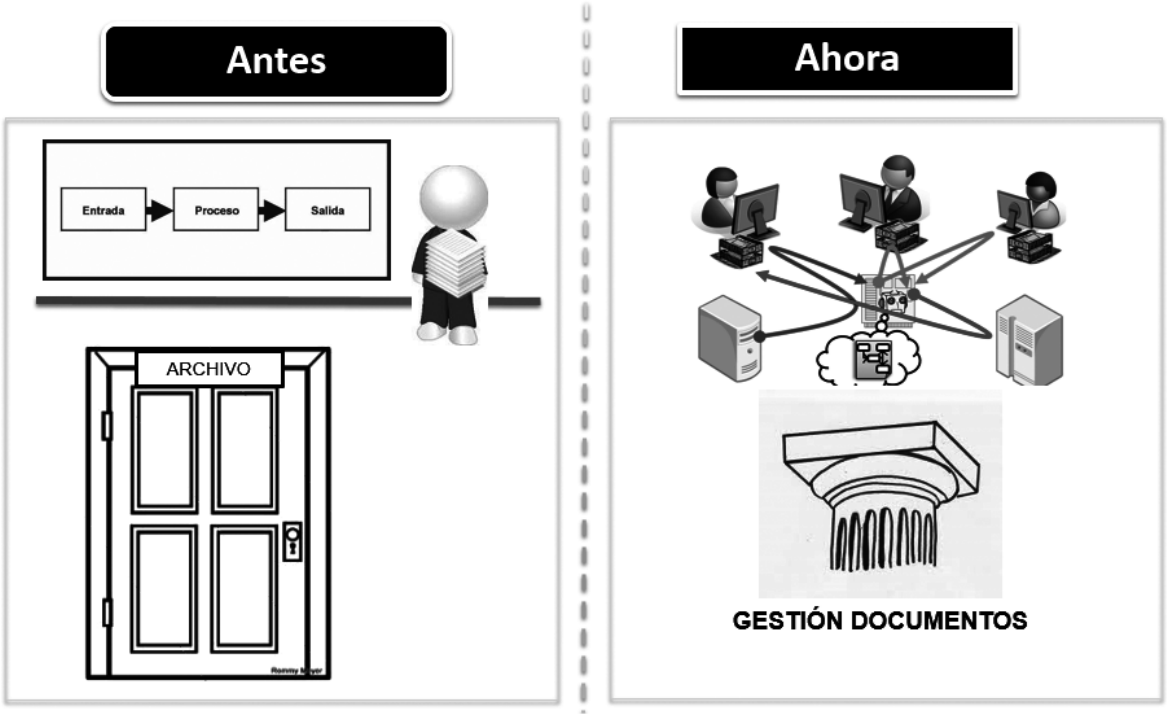


**Figura 2**. Visión del cambio [20].

En [20], se destaca la importancia de la implementación de la Norma ISO 15489 en los procesos de gestión documental, ya que esta norma proporciona los lineamientos necesarios para alcanzar una adecuada transformación digital, garantizando así una administración eficiente,

segura y sistemática de los documentos:

Las directrices de la ISO 15489 no se aplican bien en un entorno donde las responsabilidades de la gestión documental quedan en manos de los archiveros sólo cuando se adquiere la custodia de los documentos. En la figura 2 se representa esta situación con una puerta con la etiqueta "archivo", que se abre de vez en cuando para recibir documentos y otras veces para servirlos cuando los piden. Detrás de la puerta los archiveros, de forma totalmente profesional, actúan sobre los documentos aplicando los procesos e instrumentos documentales, pero sin mucha interconexión con los procesos de trabajo que han producido los documentos.

Las directrices de la ISO 15489 se aplican bien, sin embargo, en entornos en los que antes de poner en marcha los procesos (por ejemplo, por encontrase en la transformación digital de los mismos) hay que realizar un análisis de los mismos y el componente documental es esencial [20, p. 22].

Además, la norma está orientada al trabajo en entornos digitales, aunque se mantiene independiente de cualquier tecnología específica, pues esto se debe a que fue diseñada deliberadamente de esta manera para evitar conflictos con la evolución tecnológica, considerando que constantemente surgen nuevas herramientas que superan a las anteriores. Por esta razón, el comité editorial procuró redactar la norma de forma que dejara abiertas todas las posibilidades del mundo digital y evitara estructuras limitadas únicamente a la gestión documental en papel [21].

### 2.4 Desafíos en digitalización de documentos escaneados (enfocado en PDF)

Según Azim *et al.*, señalan que la transformación digital está representando nuevos desafíos en todas las organizaciones, obligadas a adaptarse a las nuevas tecnologías y formas de comunicación. Este cambio es especialmente acusado en los procesos relacionados con la gestión de la información, donde la masificación de los contenidos digitales, su diversidad de fuentes y formatos y la convivencia obligada con soportes de papel, obliga a establecer nuevas políticas de gestión electrónica de documentos [22].

Hay que tener en cuenta que la tecnología evoluciona a velocidades nunca vistas, algo que nos exige a todos, instituciones, empresas y ciudadanía en general, a mantenernos en permanente renovación ante los cambios que se generan. En la actualidad, observamos que los registros en formato digital se van integrando poco a poco en multitud de sectores.

Claro que sí, pero la realidad es que este camino conlleva obstáculos significativos, en particular cuando hay que pasar toda la documentación al formato digital. Las empresas necesitan estar listas para hacer una inversión fuerte, pues poner en marcha sistemas novedosos para administrar documentos requiere tanto contratar especialistas como entrenar al equipo interno. Por otro lado, hay que pensar en cómo mantener estos sistemas activos a lo largo del tiempo, considerando que los avances tecnológicos suceden constantemente y es imperativo actualizarse conforme aparecen nuevas formas de almacenamiento. Igualmente, cuando una compañía se lanza a un proyecto completo de digitalización, resulta fundamental que defina normas sólidas de cuidado de información, dado que el manejo electrónico de datos acarrea mayor vulnerabilidad ante posibles fugas si no se implementan controles de seguridad suficientes [23].

## 2.5 Requisitos funcionales para sistemas de gestión documental

Los requisitos principales para los sistemas de gestión de documentos electrónicos son el cumplimiento de la disponibilidad, integridad y confidencialidad de los documentos [24], ya que debe permitir el acceso a los archivos tanto para un pequeño grupo como para miles de usuarios, sin que esto afecte su rendimiento o fiabilidad. Además, deben facilitar el trabajo colaborativo, asegurando que los documentos se mantengan íntegros y protegidos contra accesos no autorizados. También es importante que estos sistemas estén preparados para ser utilizados por empleados que trabajan de forma remota [25].

Como también, es fundamental que se adapte a las necesidades específicas de cada institución, ya que esto permite una comunicación más fluida y una mejor coordinación dentro de la organización, pues estos funcionan como sistemas distribuidos, formados por distintos módulos que trabajan en conjunto. Al mismo tiempo, pueden operar como una parte integrada dentro

de un sistema de información más amplio de la empresa. Para poder conectarse con otras plataformas o herramientas, es fundamental que cuenten con interfaces abiertas que permitan esa integración [26].

Según lo expuesto en [25], los sistemas de gestión de documentos electrónicos deben cumplir con requisitos básicos, entre ellos, garantizar la confidencialidad de los datos. Para ello, utilizan dos tipos de autorizaciones:

**Privilegios:** Son permisos a nivel del sistema, como crear o eliminar usuarios o acceder a documentos de otros en caso de problemas. Estos privilegios están reservados para los administradores del sistema.

**Permisos:** Los permisos funcionan a nivel de elementos específicos como carpetas, documentos y similares. Definen exactamente qué acciones tiene permitidas cada usuario sobre esos recursos: desde leerlos y editarlos, hasta borrarlos, cambiar sus propiedades o administrar quién más puede acceder a ellos.

Para asegurar un funcionamiento estable, estos sistemas requieren un hardware confiable. Por ejemplo, si muchos usuarios acceden a documentos al mismo tiempo, se necesitan discos duros de alta velocidad y varios servidores para mantener el sistema operativo sin interrupciones [25], como también es fundamental disponer de una herramienta de búsqueda que sea eficaz, sencilla y flexible. Según lo indica [25], existen dos formas principales de realizar búsquedas de documentos:

- **Por atributos.-** como el autor, la fecha de creación o modificación, y quién lo editó por última vez.
- **Por contenido.-** buscando frases específicas dentro del texto. Es útil si también permite búsquedas por sinónimos, antónimos, palabras parecidas en escritura o pronunciación, y otras variantes.

Además, estos sistemas deben incluir herramientas potentes para automatizar procesos de

negocio y ofrecer una interfaz gráfica amigable para los usuarios, pues para tener un sistema bien estructurado es importante justificar la inversión en este tipo de sistemas. Dado que su implementación implica un alto costo, se espera que su uso permita ahorrar dinero a largo plazo y mejore la eficiencia de la empresa, como ocurre con otros sistemas de información.

## 2.6 Documentos PDF como objeto de procesamiento

El documento PDF, cuyas siglas significan Portable Document Format, fue creado por Adobe en 1985 con el propósito de garantizar una gestión estable de la información. Este formato permite a los usuarios visualizar correctamente la estructura y contenido del archivo, independientemente de la plataforma informática utilizada, lo que lo convierte en una herramienta eficaz y ampliamente compatible [27]. Cabe resaltar que el formato PDF es de gran ayuda para cualquier tipo de visualización que se requiera revisar, ya que permite obtener de manera inmediata la información necesaria para su consulta.

### 2.6.1 Características técnicas del formato PDF

El formato PDF puede contener texto, gráficos, tablas e incluso animaciones, permitiendo su representación mediante medios digitales. Adobe Acrobat, el programa desarrollado por Adobe, convierte documentos a este formato, lo que permite a los usuarios visualizar su contenido en diversas plataformas informáticas. Además, una de sus principales ventajas es que los archivos pueden comprimirse significativamente, facilitando su almacenamiento y transferencia [28].

También es importante mencionar cómo está compuesto Adobe Acrobat, el cual, según [28], se estructura en las siguientes seis partes:

- *Acrobat Reader* es un software gratuito que se puede descargar desde el sitio web de Adobe (http://www.adobe.com). También se distribuye junto con otras aplicaciones de Adobe. Es un programa de visualización, por lo que los usuarios pueden abrir un archivo PDF, navegar por las

páginas del documento e incluso imprimir copias en papel.

- *Acrobat Distiller* es un programa que convierte un archivo PostScript en un archivo PDF. Después de que un documento se crea con una aplicación nativa y se guarda como archivo PostScript, Distiller lo convierte al formato de archivo PDF. Muchas de las aplicaciones más recientes de Adobe, como Photoshop, Illustrator y PageMaker, están equipadas con una función de conversión integrada que permite una conversión directa en un solo paso desde el documento mismo a PDF.

  En [29] menciona que PostScript, es un tipo de archivo gráfico vectorial. Su punto fuerte reside en su capacidad para preparar gráficos y texto digitales para imprimir. Se puede enviar un archivo PS (PostScript) directamente a una impresora sin necesidad de abrirlo en una aplicación .

- *Acrobat* es un programa tanto de visualización como de edición. Acrobat ofrece muchas funciones útiles de edición para un documento PDF ya existente. Por ejemplo, insertar páginas, reemplazar una página con otra, recortar una página, eliminar páginas, crear una nota, editar texto y agregar anotaciones son funciones disponibles para que los usuarios personalicen documentos PDF. También existen muchos complementos de terceros que se pueden usar con Acrobat para mejorar las funciones de edición. Sin embargo, muchos programas que no son de Adobe aún necesitan pasar por el proceso de conversión PostScript.

- *PDF Writer* es un controlador de impresora que convierte documentos nativos en archivos PDF. En lugar de enviar el documento a una impresora física para obtener una copia en papel, al seleccionar PDF Writer como dispositivo de impresión se crea una copia digital del archivo PDF. Este método de creación de archivos PDF es el más sencillo. Aun así, debe señalarse que PDF Writer tiene menos opciones que Acrobat Distiller.

- *Acrobat Capture* es un complemento que se puede ejecutar dentro de Acrobat. Vale la pena destacar que Capture tiene una característica muy útil: convierte esas imágenes de texto que escaneas en auténticos PDFs con caracteres editables. En otras palabras, una vez procesada la imagen, podrás modificar el contenido textual del documento sin problema alguno.

- *Acrobat Catalog* es un programa que indexa grandes volúmenes de archivos PDF para facilitar el acceso posterior. Similar al sistema de catálogo de fichas de una biblioteca, permite al usuario localizar fácilmente un documento PDF específico mediante el motor de búsqueda en Acrobat Reader o Acrobat. Catalog es una herramienta esencial para archivar archivos PDF de manera

estructurada.

### 2.6.2 Creación de archivos PDF

En [27], explica que hay tres métodos para generar archivos PDF:

- Generación directamente desde la aplicación,
- Creación a partir de un archivo PostScript, y
- Impresión mediante un controlador de impresora PDF.

Algunos de los programas necesarios para producir archivos PDF, como Acrobat Distiller y PDF Writer, deben estar instalados en el disco duro del computador. También es posible generar archivos PDF desde navegadores web como Microsoft Internet Explorer y Netscape, software de Diseño asistido por computadora (Computer Aided Design CAD), así como aplicaciones de procesamiento de texto e imágenes. Cualquier programa que incluya un comando de impresión puede generar un archivo PDF. A continuación, en la Figura 3 se ilustran los tres métodos anteriormente mencionados para la generación de archivos PDF:

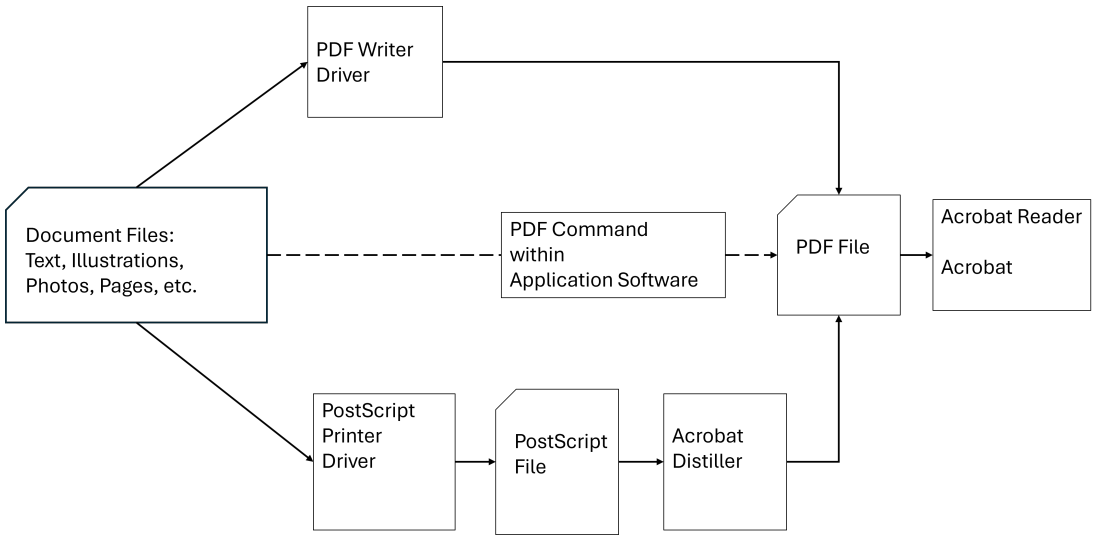


**Figura 3**. Flujo de trabajo típico para la creación de un archivo PDF [28].

## 2.7 Tipos de PDF: nativos vs. escaneados

Hay dos tipos principales de documentos PDF: los PDF nativos y los PDF escaneados. La cual al transformar en formato PDF debemos comprender que hay dos maneras de diferenciar este formato, ya que cada uno almacena la información de manera distinta y presenta características únicas. A continuación, se explican ambos tipos de PDF y sus principales diferencias.

### 2.7.1 PDF Nativo

El PDF nativo se genera directamente desde un software que permite exportar documentos a formato PDF, como Microsoft Word. En estos casos, el contenido se conserva en formato digital, lo que facilita su procesamiento y búsqueda. Según lo señalado en [30], es posible extraer texto manualmente de este tipo de documentos siguiendo los siguientes pasos:

- Abrir el archivo en un lector o editor de PDF.
- Seleccionar y copiar el texto deseado.

Este procedimiento es eficaz en la mayoría de los documentos conocidos como “PDF con función de búsqueda”.

### 2.7.2 PDF escaneado

El PDF escaneado, como su nombre lo indica, se genera al digitalizar un documento impreso mediante un escáner, el cual lo guarda como una imagen en un archivo PDF. Según lo señalado en [31], “no permite realizar búsquedas en el documento ni extraer texto, ya que el texto solo está codificado como una imagen gráfica. Un documento PDF de imagen es similar a un documento en papel en cuanto a su nivel de accesibilidad.”

## 2.8 Limitaciones para extracción de texto

Según lo expuesto en [32], los textos en imágenes suelen presentar una apariencia variada debido a diferencias en aspectos como la fuente, el tamaño, el estilo, la orientación, la alineación, la textura, el color, el contraste y el fondo. El texto en imágenes presenta algunas variaciones debido a la diferencia en las siguientes propiedades [33]:

• Tamaño: El tamaño del texto puede variar mucho.

• Alineación: El texto en escenas puede estar alineado en cualquier dirección y tener distorsiones geométricas, mientras que el texto de subtítulos suele estar alineado horizontalmente y, a veces, puede aparecer como texto no plano.

• Color: Generalmente los caracteres mantienen tonalidades iguales o parecidas; sin embargo, cuando hay poco contraste entre lo que se ve escrito y la superficie donde está, resulta bastante complicado sacar el texto de allí.

• Borde: La mayoría de los textos de subtítulos y de escenas están diseñados para ser fácilmente legibles, lo que da como resultado bordes marcados en los límites del texto y el fondo.

• Compresión: Muchas imágenes se graban, transfieren y procesan en formato comprimido. Por lo tanto, se puede lograr un sistema de extracción de texto más rápido si se puede extraer texto sin descompresión.

• Distorsión: La perspectiva desde la cual se captura una imagen provoca que algunos textos sufran distorsiones visuales, afectando directamente el desempeño del procedimiento de extracción.

## 2.9 Inteligencia Artificial aplicada a documentos

Para comprender mejor el concepto de Inteligencia Artificial (IA), es importante señalar que esta tecnología ha logrado integrarse en muchas de nuestras actividades diarias, convirtiéndose en una herramienta presente en múltiples campos del quehacer humano, en [34] se explica cómo:

> La inteligencia artificial, también conocida como IA, es una nueva ciencia técnica que estudia y desarrolla teorías, métodos, tecnologías y sistemas de aplicación para simular, extender y expandir la inteligencia humana. La inteligencia artificial puede estudiar las leyes de las actividades inteligentes humanas, simular el modo de pensamiento del cerebro humano, procesar datos y tomar decisiones. La inteligencia artificial representa una evolución de la tecnología informática, que responde imitando la inteligencia humana. El campo de la inteligencia artificial incluye robótica, reconocimiento facial, reconocimiento de imágenes, procesamiento del lenguaje natural, sistemas expertos, entre otros. La tecnología de inteligencia artificial se considera una de las tres tecnologías prácticas de vanguardia del siglo XXI. En los últimos años, la inteligencia artificial ha sido ampliamente utilizada en muchos campos y ha ido independizándose progresivamente de la tecnología informática, formando así una rama y un sistema autónomos.

El auge de la Inteligencia Artificial se remonta, en parte, al trabajo del destacado matemático inglés Alan Turing, quien fue clave en la ruptura del código Enigma durante la Segunda Guerra Mundial, una hazaña que resultó inalcanzable para muchos de sus contemporáneos. Esta experiencia llevó a Turing a reflexionar profundamente sobre la posibilidad de que las máquinas pudieran pensar. Como resultado, propuso lo que hoy se conoce como la Prueba de Turing, un referente fundamental en el campo de la IA, dicha prueba establece que, si un humano interactúa con otro humano y con una máquina y no es capaz de distinguir la máquina del humano, se dice que la máquina es inteligente [35].

La inteligencia artificial ha ido ganando presencia en distintos ámbitos que responden a las demandas de las personas, posicionándose como un recurso fundamental para agilizar y perfeccionar el funcionamiento de los procesos [36]. Destaca especialmente su aplicación en el terreno de la gestión documental [37]. Conforme crece el volumen de información que generan a diario

las instituciones, la urgencia por contar con soluciones tecnológicas que permitan administrar archivos y documentos con eficiencia resulta más acuciante [38].

La tecnología de IA (Inteligencia Artificial), mediante la implementación de algoritmos avanzados y aprendizaje automático (machine learning), ha demostrado su potencial para identificar, clasificar y almacenar datos de forma más eficiente [39]. Cuando se aplica la IA en la administración de archivos y documentos, consigue aliviar considerablemente el trabajo manual que demanda este tipo de tareas, mientras que disminuye de manera notable las posibilidades de que ocurran fallos derivados del factor humano.

También es necesario mencionar que la inteligencia artificial (IA) se organiza jerárquicamente en varias ramas, lo cual se representa en la Figura 4 en el nivel más amplio se encuentra AI, que abarca todas las técnicas orientadas a simular la inteligencia humana, dentro de AI se encuentra el aprendizaje automático (Machine Learning, ML), que permite a las máquinas aprender a partir de datos sin ser programadas explícitamente, como también, dentro de ML están las redes neuronales (Neural Networks, NN), las cuales simulan el comportamiento de las neuronas biológicas. El aprendizaje profundo (Deep Learning, DL) es un subconjunto de NN, que utiliza arquitecturas profundas para tareas complejas como visión por computadora o procesamiento del lenguaje natural, también incluye el enfoque “Brain-Inspired”, que agrupa métodos inspirados en el cerebro humano, como las Redes Neuronales de Picos (Spiking Neural Networks, SNN). Estas últimas simulan la comunicación neuronal basada en impulsos eléctricos y, aunque no están contenidas dentro de ML o DL, representan un campo emergente de investigación con gran potencial [40].

## 2.10 Evolución de modelos para procesamiento documental

El texto es el portador de la civilización humana. Antes de la era digital, el texto se escribía a mano o se imprimía en papel. Con el desarrollo continuo y la amplia aplicación de la tecnología informática, la forma en que las personas guardan texto ha pasado gradualmente de documentos en papel a documentos digitales. El proceso de almacenamiento de estos documentos durante un

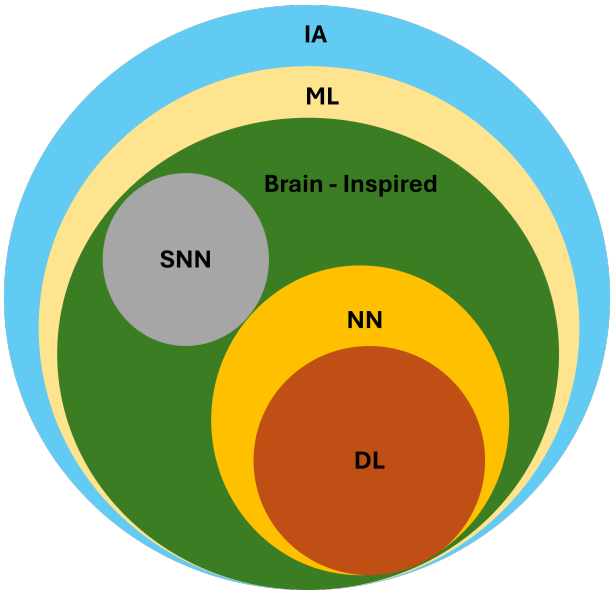


**Figura 4**. Muestra la correlación entre la inteligencia artificial (IA), el aprendizaje automático (AM) y el aprendizaje profundo (AD) [41].

período prolongado puede hacer que su calidad comience a deteriorarse. Por lo tanto, para mantener la calidad del contenido original, es necesario convertir estos documentos en documentos digitales [42].

La tecnología digital, el software y los dispositivos móviles han avanzado significativamente en los últimos años y ahora desempeñan un papel importante en la vida moderna [43], [44]. La gestión documental es un sistema fundamental para mejorar considerablemente los procesos empresariales de una organización.

Según lo señalado en [45], la integración de soluciones y técnicas basadas en inteligencia artificial permite alcanzar diversos beneficios, entre ellos:

> El uso de nuevos y potentes algoritmos de IA para el procesamiento de imágenes o el aprendizaje automático abre la puerta al análisis y "comprensión" del contenido de los documentos y, por ende, a la generación de nueva y relevante información contenida en los mismos. En consecuencia, se abre la capacidad para reconocer texto, imágenes, tablas de datos o, incluso, la estructura de un documento de manera automatizada y precisa.

Debido al enorme aumento del flujo de documentos electrónicos provocado por la pandemia, se ha incrementado el trabajo mecánico y repetitivo del personal, directivos y servicios

documentales de las empresas, quienes registran y responden a mil o más documentos al día. Como resultado, los costos laborales y de tiempo se han disparado [46]. La relevancia de este problema en la actualidad radica en mejorar los sistemas tradicionales de gestión electrónica de documentos mediante el uso de análisis de datos y métodos de aprendizaje automático para optimizar el trabajo de los empleados de la organización y garantizar un tránsito de calidad de todo el ciclo de vida del documento electrónico con una intervención humana mínima [47].

En la figura 5 se presenta un ejemplo de el software δαρχείο [48], cuyo propósito es convertir un archivo físico en uno digital, facilitando una mejor organización y control de la información digitalizada, todo ello conforme a normas y estándares de gestión documental. Un archivo físico está compuesto por documentos de distintos tipos, organizados en carpetas con un orden lógico. Por tanto, al digitalizarlo, es esencial conservar esa estructura. El software desarrollado permite replicar esta organización mediante la creación automática de carpetas dentro de un servidor FTP previamente configurado.

FTP, que significa File Transfer Protocol (Protocolo de Transferencia de Archivos), es un protocolo de la red Internet diseñado específicamente para facilitar la transferencia de grandes volúmenes de datos. Su funcionamiento está descrito en la especificación técnica RFC 959 [49].

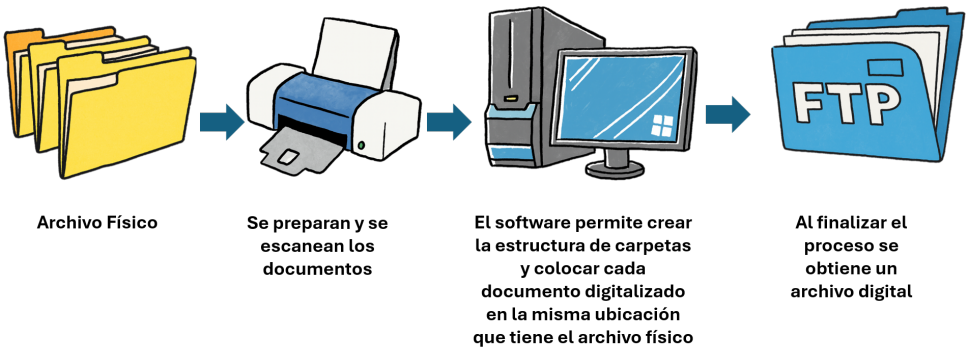


**Figura 5**. Diseño de la solución [48].

## 2.11 Aprendizaje Automático (Machine Learning) en análisis de texto

El aprendizaje automático (Machine Learning, ML) se basa en el estudio científico de algoritmos y modelos estadísticos diseñados para aprender de manera no explícita, es decir que

no se le indica directamente lo que debe hacer, sino que lo hace a partir de los datos, ya que esta descubre patrones, relaciones o reglas por sí mismo. Estos algoritmos se aplican en múltiples herramientas que utilizamos en la vida cotidiana, por ejemplo, al realizar una búsqueda en Google, los algoritmos de aprendizaje automático clasifican y priorizan las páginas web según su relevancia. El Machine Learning se emplea en diversos ámbitos como la minería de datos, el procesamiento de imágenes, el análisis predictivo, entre otros. En esencia, el algoritmo aprende qué hacer con los datos para ejecutar tareas de forma automática y eficiente [50].

Actualmente, la cantidad de datos en formato texto crece a un ritmo exponencial, lo que hace que clasificarlos manualmente sea cada vez más difícil y poco práctico. Por ello, resulta fundamental encontrar métodos eficientes que permitan organizar grandes volúmenes de información en poco tiempo. Una vez clasificados, estos datos se transforman en información valiosa, que luego puede utilizarse para planificar y tomar decisiones estratégicas en contextos empresariales e industriales [51].

El aprendizaje automático abarca una amplia variedad de métodos, entre los cuales el aprendizaje profundo mediante redes neuronales ha demostrado ser el más exitoso en el procesamiento de texto. Además, en la literatura se mencionan otros enfoques, como el enfoque basado en palabras clave, el enfoque basado en corpus (que, en la práctica, es un método supervisado de aprendizaje automático), el enfoque basado en reglas (donde las reglas emocionales se derivan a partir de métodos estadísticos, lingüísticos y computacionales), y el enfoque híbrido [52].

### 2.11.1 Tipos de aprendizaje automático

Como se muestra en la Tabla I se detallan los tres tipos de aprendizaje automático (Machine Learning) [53], donde se describe detalladamente en qué consiste cada uno y cuál es su función dentro del aprendizaje profundo:

**Tabla I**.
Tipos de aprendizaje automático

| Tipo | Descripción |
|---|---|
| *Aprendizaje supervisado* | *El aprendizaje supervisado* requiere de un conjunto de datos de entrenamiento, por ejemplo la predicción del número de usuarios activos suscritos a una plataforma, donde toma variable el número de productos vendidos o las reseñas positivas de los usuarios, la cual se utilizan para calibrar los parámetros ajustables del modelo de aprendizaje automático. Una vez que el modelo ha sido entrenado con éxito, se puede utilizar para predecir la variable objetivo y a partir de nuevos puntos de datos no vistos correspondientes a las características de entrada x. En cuanto al tipo de aprendizaje supervisado, se puede distinguir entre problemas de regresión, donde se predice un valor numérico (por ejemplo, número de usuarios), y problemas de clasificación, donde el resultado de la predicción es una categoría o clase, como "observadores" o "compradores". |
| *Aprendizaje no supervisado* | *El aprendizaje no supervisado* ocurre cuando el sistema de aprendizaje debe detectar patrones sin contar con etiquetas o especificaciones preexistentes. Es decir, los datos de entrenamiento solo consisten en variables x, con el objetivo de encontrar información estructural relevante, como grupos de elementos que comparten propiedades comunes (conocido como agrupamiento o clustering) o representaciones de datos que se proyectan desde un espacio de alta dimensión a uno de menor dimensión (conocido como reducción de dimensionalidad) [54]. Un ejemplo destacado de aprendizaje no supervisado en los mercados electrónicos es la aplicación de técnicas de agrupamiento para clasificar clientes o mercados en segmentos, con el fin de lograr una comunicación más específica y dirigida a cada grupo objetivo. |
| *Aprendizaje de refuerzo* | En un *aprendizaje de refuerzo*, en lugar de proporcionar pares de entrada y salida, se describe el estado actual del sistema, se especifica un objetivo, se proporciona una lista de acciones permitidas junto con sus restricciones ambientales, y se permite que el modelo de aprendizaje automático experimente el proceso de alcanzar el objetivo por sí mismo utilizando el principio de prueba y error para maximizar una recompensa. Los modelos de aprendizaje por refuerzo se han aplicado con gran éxito en entornos cerrados, como los videojuegos [55], pero también son relevantes en sistemas multiagente, como los mercados electrónicos [56]. |

## 2.12 Procesamiento de Lenguaje Natural (NLP) para gestión semántica

Según lo expuesto en [57], la ingeniería de software, la inteligencia artificial y la semántica se fusionan en el área del Procesamiento del Lenguaje Natural (Natural Language Processing, NLP), que es una disciplina potente. Su objetivo principal es lograr que las computadoras y el lenguaje humano trabajen en conjunto, permitiendo que las máquinas comprendan, interpreten y generen texto o habla de manera similar a como lo haría un ser humano.

Asimismo, en [57] se menciona que las técnicas de Procesamiento del Lenguaje Natural (Natural Language Processing, NLP) han experimentado un rápido desarrollo y hoy son fundamentales en una amplia gama de aplicaciones, especialmente en aquellas que buscan mejorar la interacción entre los seres humanos y las computadoras personales (PC). Los avances en inteligencia artificial como la robótica y la arquitectura cerebral, que emula los miles de millones de conexiones neuronales del cerebro humano han sido clave en esta evolución.

- **Tokenización:** El proceso de separar un texto en tokens o frases distintas para su análisis. Usando un tokenizador, los "datos no estructurados y el texto en lenguaje natural" se dividen en fragmentos de datos que pueden considerarse piezas distintas [58]. Se puede crear una matriz que represente el mensaje de un texto utilizando tokens que reflejan frecuencias en el texto.
- **Etiquetado de Partes del Habla (Part-of-Speech Tagging):** Clasificación de cada palabra según su función gramatical.
- **Análisis de Sentimientos:** Al fusionar datos, Procesamiento del Lenguaje Natural (Natural Language Processing, NLP) y aprendizaje automático, el análisis de sentimientos es un método introspectivo que identifica los momentos clave en las interacciones. Se utiliza para evaluar contactos con clientes, interacciones en centros de atención, reseñas en internet, publicaciones en foros en línea y diversos otros datos.
- **Traducción Automática:** Una técnica importante del Procesamiento del Lenguaje Natural (Natural Language Processing, NLP) es la traducción automática. El lenguaje se

produce y se analiza usando métodos que se engloban dentro del ámbito de la traducción automática [59]. Las principales empresas utilizan sistemas sofisticados de traducción automática.

- **Resumen de Texto:** El método de extraer un resumen de un texto extenso sin sacrificar información crucial se conoce como resumen de texto. Es esencial que el resumen transmita la información con claridad y fluidez.

## 2.13 Inteligencia Artificial Generativa (GenAI)

La inteligencia artificial generativa (GenAI) es capaz de "ir más allá de lo que ha visto antes y crear algo nuevo", dijo Ian Goodfellow, otorgando así a GenAI la generatividad este término significa: la capacidad de producir o crear algo nuevo. Además, con esta afirmación, Goodfellow posicionó a GenAI en contraste con la inteligencia artificial tradicional, que se consideraba principalmente por su capacidad de clasificar y encontrar orden en la información existente. Desde la aparición de ChatGPT, GenAI se ha convertido en el centro de atención, y ahora están disponibles numerosos modelos de este tipo que son capaces de generar diversos tipos de contenido, desde textos hasta videos, pasando por imágenes [60].

Según Feuerriegel *et al.*, el término inteligencia artificial generativa se refiere a técnicas computacionales capaces de generar contenido aparentemente nuevo y significativo, como texto, imágenes o audio, a partir de datos de entrenamiento. La amplia difusión de esta tecnología con ejemplos como DALL·E 2, GPT-4 y Copilot está revolucionando actualmente la forma en que trabajamos y nos comunicamos [61].

La inteligencia artificial generativa (GenAI) hace referencia a soluciones de aprendizaje automático (Machine Learning) entrenadas con grandes volúmenes de datos para generar contenido a partir de instrucciones proporcionadas por el usuario. Por ejemplo, un comando como: "persona con forma de IA desbordando el mundo con textos, imágenes y otros medios, futurista, alta resolución, oscuro", puede utilizarse en el generador de imágenes Midjourney para produ-

cir [62] ilustraciones como la que se muestra en la Figura 6 este generador está disponible en la plataforma oficial en línea [63].



**Figura 6**. Midjourney imagen producida de IA generativa personificada [62].

## 2.14 Fundamentos y arquitecturas de modelos generativos

La Figura 7 presenta una visión estructurada de la IA generativa desde tres niveles complementarios: modelo, sistema y aplicación, organizados en función de la modalidad de salida (texto, imagen/video, voz/música y código). Esta categorización permite comprender cómo la IA generativa evoluciona desde su arquitectura técnica hasta su implementación práctica en contextos reales [61].

En definitiva, la inteligencia artificial generativa (IA Gen) se sirve de enormes cantidades de datos organizados, ya sean textos, imágenes o cualquier otra forma de información, para producir, conforme a lo que el usuario le pide, contenidos inéditos tales como texto, imágenes o datos anticipados. Esta clase de tecnología no guarda información de manera exacta, sino que capta regularidades en el lenguaje, los tonos, los modos de expresión y los temas partiendo de un extenso “corpus” de datos, parecido a lo que sería una colección de libros. De esta manera, cuando recibe una instrucción, la IA no devuelve palabra por palabra aquello que aprendió, sino que lo toma como punto de partida para generar contestaciones novedosas y bien articuladas.

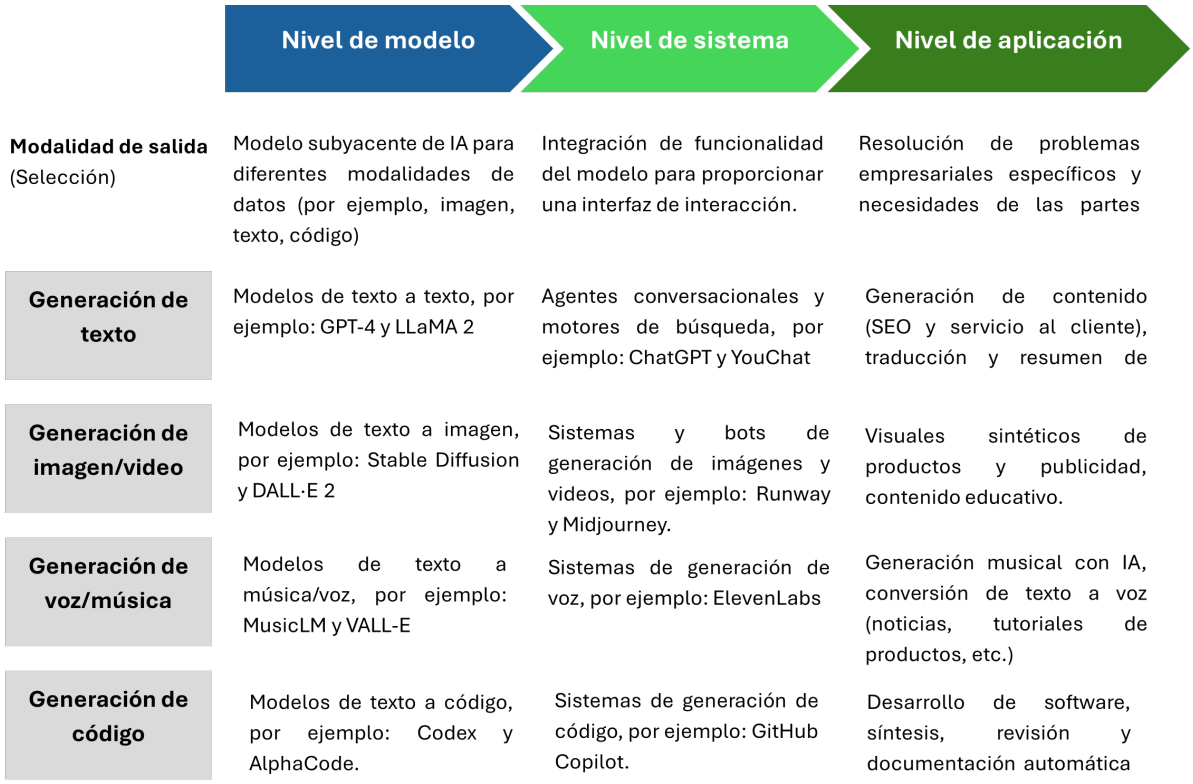


**Figura 7**. Una visión de la IA generativa desde el punto de vista de los modelos, los sistemas y las aplicaciones [61].

## 2.15 Grandes Modelos de Lenguaje (LLMs) aplicados a documentos

Los grandes modelos de lenguaje (LLM) son algoritmos de aprendizaje profundo capaces de identificar, extraer, resumir, anticipar y generar texto a partir del conocimiento que han adquirido al ser entrenados con conjuntos de datos muy masivos. Todos los modelos de lenguaje comparten la característica de encargarse de tareas relacionadas con el procesamiento del lenguaje natural (NLP). Se les llama "grandes" porque están formados por una cantidad muy elevada de parámetros (los pesos y sesgos de las redes neuronales que los componen), que pueden llegar a ser miles de millones, y se entrenan con enormes volúmenes de datos. Esta combinación les permite ofrecer respuestas más exactas y complejas en comparación con modelos de estructuras similares, pero con una menor cantidad de parámetros. La efectividad de un modelo no solo depende de su tamaño y de la cantidad de datos empleados en su entrenamiento, sino también de la calidad de dichos datos [64].

El surgimiento de los Modelos de Lenguaje de Gran Escala (LLMs, por sus siglas en inglés) como ChatGPT y Gemini ha provocado un cambio en el ámbito del Procesamiento del Lenguaje Natural (PLN). El uso de estas tecnologías nuevas y emergentes ha permitido a las personas aumentar la productividad y mejorar el rendimiento [65].

Los modelos que han sido entrenados con grandes volúmenes de texto e imágenes pueden entender mejor el lenguaje y también generar textos que parecen escritos por personas. Implementar el poder de los LLMs representa una vía atractiva para abordar los desafíos que plantea la información no estructurada [66].

La mayoría de sistemas de gestión documental tradicionales dependen de búsquedas mediante palabras clave o del etiquetado manual para encontrar y recuperar archivos. Estas metodologías resultan efectivas siempre que los documentos posean metadatos claramente definidos y se encuentren organizados de forma coherente, aunque presentan complicaciones notables cuando se trata de información desorganizada, particularmente en PDFs que mezclan múltiples tipos de contenido, incluyendo texto, imágenes y tablas.

Para ir más allá de estas limitaciones, hace falta una metodología mucho más elaborada que sea capaz de comprender verdaderamente el contexto y los vínculos que existen dentro del texto. Es precisamente aquí donde entran en juego los grandes modelos de lenguaje (LLMs), que se posicionan como una alternativa transformadora, puesto que poseen la habilidad de identificar patrones complejos y asimilar los sentidos más profundos que encierra el lenguaje humano, superando así las restricciones que marcan los métodos convencionales [67].

El potencial transformador de los LLMs (Large Language Model) radica en su capacidad para codificar la información en un sentido contextual y captar relaciones dentro de los documentos. Esto facilita una recuperación de documentos y un procesamiento de consultas más precisos, mejorando la eficiencia y eficacia de la productividad y el tiempo de los empleados. Sin técnicas y mecanismos efectivos para procesar y consultar estos datos, valiosos conocimientos permanecen enterrados dentro de los documentos, lo que los convierte en recursos infrautilizados [67].

En [68], expone una visión general de los modelos existentes, a continuación, en la Tabla II, se muestra una comparación de diversos modelos de lenguaje basados en Transformers, destacando sus principales ventajas, aplicaciones en tareas de procesamiento de lenguaje natural y las limitaciones asociadas a su uso.

**Tabla II**.
Comparación de modelos de lenguaje basados en Transformers

| **Modelo** | **Ventajas** | **Aplicaciones** | **Limitaciones** |
|---|---|---|---|
| Serie GPT | Genera texto similar al humano y es versátil en tareas de PLN. | Completado de texto, resumen, traducción y respuesta a preguntas. | Dificultad para adaptarse a dominios específicos y alto consumo computacional. |
| BERT | Analiza el contexto circundante, tanto lo anterior como lo posterior, facilitando una comprensión más profunda del lenguaje considerado como un todo. | Clasificación de documentos, identificación de entidades nombradas y generación de respuestas a consultas. | Presenta una desventaja importante: requiere modelos muy grandes y mucho poder de procesamiento. |
| Transformer XL, XLNet, RoBERTa, ALBERT | Se desempeñan bien manejando textos largos, reducen el tamaño del modelo y mejoran notablemente los resultados. | Consiguen un rendimiento excepcional en múltiples aplicaciones de procesamiento del lenguaje natural. | Requieren enormes cantidades de poder computacional e inversión significativa de tiempo para adaptarse a dominios específicos. |

## 2.16 Casos de uso en extracción y estructuración de datos

En los últimos años, el valor y los beneficios que las empresas obtienen de los datos digitales han cambiado significativamente. Esto ha llevado a muchas organizaciones a recopilar y almacenar grandes volúmenes de información. Sin embargo, guardar los datos no es suficiente ya que necesitan extraer la información importante esto con el fin de convertirlos en un recurso

útil para la organización. Por eso, cada vez es más común que las empresas se enfoquen no solo en almacenar datos de forma eficiente, sino también en procesarlos y utilizarlos con un propósito claro. Estos objetivos pueden variar según el tipo de empresa o industria. Por ejemplo, en el sector de la producción, algunas metas comunes incluyen ofrecer nuevos servicios, anticiparse a fallas mediante mantenimiento predictivo, mejorar la calidad o lograr procesos de fabricación más sostenibles y eficientes [69].

Desafortunadamente, la mayoría de los métodos que se aplican para alcanzar estos objetivos se enfocan únicamente en la integración y el análisis de fuentes de datos estructuradas y semiestructuradas [69]. Sin embargo, en [70], aproximadamente el 80 % de los datos disponibles a nivel mundial son datos no estructurados, es decir, documentos de texto, imágenes, videos, archivos de audio u otros datos codificados en formato binario, como los documentos PDF. Especialmente en las empresas, estos documentos PDF desempeñan un papel importante porque se utilizan como formato de intercambio para datos textuales y tabulares.

## 2.17 Tecnologías de extracción de texto

La extracción de texto es un proceso esencial en los sistemas de gestión de documentos, especialmente cuando se trabaja con documentos no estructurados, como archivos PDF escaneados. Estas tecnologías permiten convertir la información contenida en imágenes o documentos digitales en texto editable, facilitando su análisis, almacenamiento y reutilización.

### 2.17.1 Reconocimiento Óptico de Caracteres (OCR) avanzado

OCR conocida como reconocimiento óptico de caracteres, es un campo de la informática que se encarga de reconocer texto basado en imágenes a partir de fotografías y convertirlo en caracteres digitales reales. El OCR (Optical Character Recognition) funciona de manera similar a la capacidad del cerebro humano para reconocer letras, números y símbolos. Puede leer tanto texto manuscrito como impreso. El rendimiento del OCR está directamente relacionado con la

calidad de los documentos e imágenes de entrada [71].

Según en [72], dice que el uso inicial de la tecnología OCR surgió en 1929, cuando Gustav Tauschek la patentó en Alemania. Posteriormente, Handel también obtuvo una patente en Estados Unidos. El primer producto comercial de OCR fue lanzado por Kurzweil Computer Products en 1978, y el primer cliente fue LexisNexis [73]. Después de eso, OCR se convirtió en una tecnología líder en el mercado de software, con altas ventas y ganancias.

### 2.17.2 Tesseract

Tesseract es una de las herramientas OCR de código abierto para el reconocimiento de texto impreso en documentos escaneados. El motor procesa documentos de texto en cuatro pasos principales: (1) preprocesamiento de las imágenes de entrada utilizando técnicas de preprocesamiento como la binarización o el umbral de Otsu. (2) análisis de componentes conectados para identificar regiones de texto (blobs) en las imágenes. (3) identificación de líneas/palabras para separar palabras en los blobs identificados, y (4) reconocimiento de texto para las palabras separadas. Las versiones más recientes de Tesseract (4.0 y posteriores) emplean una red neuronal profunda basada en memoria a largo corto plazo (LSTM) en el paso 4 para el reconocimiento de texto, reemplazando los algoritmos tradicionales de coincidencia de patrones. Tesseract logró resultados sobresalientes en documentos bien estructurados y limpios (por ejemplo, libros, artículos impresos, recibos, etc.). Sin embargo, su rendimiento disminuye significativamente en documentos de baja calidad o cuando los documentos escaneados incluyen elementos no textuales, por ejemplo, elementos gráficos, figuras, tablas y otros [74].

### 2.17.3 ABBYY FineReader

ABBYY FineReader es un software que utiliza tecnología de Reconocimiento Óptico de Caracteres (OCR) para analizar la estructura de la imagen de un documento. El programa divide la página en distintos elementos, como bloques de texto, tablas e imágenes para luego, segmentar

las líneas en palabras y estas en caracteres individuales. Una vez identificados, los caracteres se comparan con un conjunto de patrones predefinidos y a partir de esta comparación, el sistema genera múltiples hipótesis sobre la naturaleza de cada carácter y evalúa diversas posibilidades de segmentación de líneas y palabras, esto con el fin de tener una buena interpretación más precisa para presentar el texto reconocido [75].

### 2.17.4 OCRopy

OCRopy [76] es una canalización OCR para el reconocimiento de texto impreso. La arquitectura del sistema incluye tres módulos principales: (1) análisis de diseño, (2) reconocimiento de líneas de texto y (3) modelo de lenguaje estadístico. En el módulo de análisis de diseño, el sistema identifica columnas y bloques de texto en el documento de entrada (por ejemplo, líneas de texto). El módulo de reconocimiento de texto reconoce los textos en las líneas de texto identificadas. Implementa una red neuronal LSTM bidireccional de una sola capa en combinación con la función de pérdida de clasificación temporal conexionista (CTC) para predecir la secuencia de caracteres en las líneas de entrada. El modelo de lenguaje estadístico integra conocimiento del dominio de aplicación de la tarea OCR para mejorar el rendimiento [74].

### 2.17.5 Calamari

Calamari [77] es un módulo de reconocimiento de texto que puede integrarse en cualquier canalización OCR, por ejemplo, OCRopy. Utiliza líneas de texto completas como entradas y emplea la arquitectura de red neuronal recurrente convolucional / clasificación temporal conexionista (CRNN-CTC) propuesta en [78]. El marco CRNN-CTC es el modelo de aprendizaje profundo más avanzado para aplicaciones de reconocimiento de secuencias basadas en imágenes. El marco integra diferentes componentes es decir, una red neuronal convolucional profunda (CNN) para la extracción de características y una red neuronal recurrente (RNN) para el modelado de secuencias entrenados conjuntamente de manera de extremo a extremo. Se agrega una capa de transcripción en la parte superior del modelo RNN para traducir las predicciones de la

RNN en secuencias de etiquetas utilizando un decodificador de clasificación temporal conexionista (CTC) [74].

### 2.17.6 Cloud Vision

Google Cloud lleva el reconocimiento óptico de caracteres (OCR) a otro nivel gracias a su avanzada inteligencia artificial. No solo se limita a identificar texto, sino que también es capaz de comprenderlo, organizarlo y enriquecerlo, transformando esos datos en información útil y valiosa para las empresas. Ofrece la flexibilidad de utilizar las herramientas de OCR como una solución integral como es el caso de Document AI para optimizar la eficiencia, o bien acceder directamente a las APIs disponibles desde la consola de Google Cloud y así integrar las funciones de OCR en tus propias aplicaciones [79].

## 2.18 Técnicas híbridas: OCR + IA Generativa

La inteligencia artificial generativa (GenAI) ha avanzado significativamente el campo de la inteligencia artificial al permitir que las máquinas produzcan respuestas similares a las humanas y generen contenido a partir de grandes cantidades de datos. Una de sus aplicaciones transformadoras es la mejora en la forma en que los usuarios interactúan con documentos estáticos como los PDF, tradicionalmente se realiza búsquedas manuales como palabras claves y haciendo uso del desplazamiento de búsqueda por paginas [80]. Estos métodos consumen mucho tiempo, especialmente al tratar con grandes volúmenes de información en ámbitos como los sectores legal, académico y sanitario [81].

El Reconocimiento Óptico de Caracteres (OCR) funciona como una de las tecnologías centrales utilizadas para este propósito [82]. Utilizar OCR para procesar información textual, como títulos de artículos, resúmenes y el texto principal, es relativamente sencillo. Sin embargo, la efectividad de los resultados de OCR puede verse comprometida por la calidad de los artículos científicos presentados en formatos de imagen y PDF. Para abordar elementos más complejos

que el texto plano, como tablas y referencias, también se emplea OCR, aunque a menudo tiene dificultades para reconocer y procesar la estructura y la lógica inherente a estos componentes [83].

Si bien el OCR puede extraer e interpretar texto incrustado dentro de estas figuras, puede enfrentar desafíos como baja resolución, colores o fondos que dificultan la lectura, u otros objetos visuales cerca del texto que confunden al sistema. Han surgido técnicas como la generación de subtítulos para imágenes como métodos potenciales para analizar figuras. Sin embargo, muchas de estas soluciones pasan por alto la información contextual de las figuras, específicamente la descripción textual que se proporcionada en el cuerpo principal del documento [84].

## 2.19 Métricas de evaluación de precisión

La evaluación de la precisión en los sistemas de extracción de textos y análisis de documentos es un aspecto esencial para determinar la calidad y fiabilidad de los resultados obtenidos. En este trabajo se utilizan métricas cuantitativas y cualitativas para medir tanto la precisión del texto extraído como la calidad semántica de los resúmenes generados.

### 2.19.1 F1-score

Sensitivity or recall (sensibilidad o recuperación) en la ecuación (2.1) es la fracción de casos positivos reales que se predicen correctamente como positivos o pertenecientes a la clase. También se conoce como tasa de verdaderos positivos [85], [86], [87]. Precision (precisión) en la ecuación (2.2) es la fracción de casos positivos correctamente predichos del total de casos predichos como positivos [85], [86], [87]. F1-score (Puntuación F1), también conocida como coeficiente de Dice, es la media armónica entre la precisión (Precision) y la exhaustividad (recall), proporcionando un equilibrio entre ambas (Ecuación (2.3)) [85], [86], [88].

$$Recall = \frac{TP}{TP + FN} \tag{2.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

Donde TP son los verdaderos positivos, FP son los falsos positivos y FN representa los falsos negativos. F1-score: Esta métrica considera todas las clases utilizando el total de verdaderos positivos, falsos negativos y falsos positivos [89]:

$$Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.3}$$

### 2.19.2 Word Error Rate - WER

La tasa de error de palabras (Word Error Rate - WER, por sus siglas en inglés) es más adecuada para evaluar elementos semánticos más amplios, como la coherencia de las oraciones en un chatbot o la fluidez de las frases en la traducción automática [90]. Según lo expuesto en [91], la Tasa de Error de Palabras (Word Error Rate, WER) se calcula mediante la siguiente ecuación (2.4):

$$WER = \frac{I + S + D}{C + S + D} \tag{2.4}$$

donde I son inserciones, S: sustituciones, D: eliminaciones y C son palabras correctas.

- Una **sustitución** ocurre cuando una palabra es reemplazada (por ejemplo, “noose” se transcribe como “moose”).
- Una **inserción** es cuando se añade una palabra que no fue dicha (por ejemplo, “SAT” se convierte en “essay tea”).

- Una **eliminación** ocurre cuando una palabra se omite completamente en la transcripción (por ejemplo, “turn it around” se convierte en “turn around”) [90].

### 2.19.3 Character Error Rate - CER

La Tasa de Error de Caracteres (Character Error Rate - CER, por sus siglas en inglés) es una métrica utilizada para medir la salida de un texto generada por un sistema ya sea una transcripción de voz o texto extraído de imágenes, esta métrica calcula el número mínimo de operaciones necesarias (inserciones, eliminaciones y sustituciones de caracteres) para transformar la salida generada por el sistema en el texto correcto. Cuanto menor sea este valor, mayor será la precisión del sistema [92].

La fórmula básica para la métrica de Tasa de Error de Caracteres (CER) es la ecuación (2.5):

$$CER = \frac{I + D + S}{N} \tag{2.5}$$

Donde:

- **I** = Número de inserciones de caracteres
- **D** = Número de eliminaciones de caracteres
- **S** = Número de sustituciones de caracteres
- **N** = Número total de caracteres en el texto de referencia

## 2.20 Desarrollo de software para gestión documental

El desarrollo de software de gestión documental implica la creación de sistemas informáticos diseñados para organizar, almacenar, consultar, procesar y recuperar documentos de manera

eficiente. Estos sistemas permiten transformar procesos manuales en flujos digitales, reduciendo el uso de papel, mejorando la disponibilidad de la información y facilitando la trazabilidad documental.

### 2.20.1 Arquitectura de sistemas IA para procesamiento de PDF

El diseño de arquitectura se refiere a la actividad de definir y desarrollar conceptos, estructuras y relaciones durante todo el ciclo de vida del sistema de interés para asegurar una materialización exitosa [93], [94]. Generalmente, de acuerdo con los principios básicos de la ingeniería de sistemas (SE) o con normas internacionales autorizadas como ISO/IEC/IEEE 15288:2015, ISO/IEC/IEEE 29148:2018, el diseño de arquitectura se realiza a través del proceso de definición del problema y definición de la solución del sistema [93], [94].

De manera similar, el diseño de la arquitectura de un sistema basado en inteligencia artificial requiere, en primer lugar, la definición del problema que se desea resolver y, posteriormente, la formulación de la solución del sistema. No obstante, para lograr una implementación efectiva, es importante elegir una tecnología IA que permita que el sistema sea eficiente, escalable y eficaz frente al problema planteado, pues hay que considerar que tecnologías de IA tienen diferentes capacidades, limitaciones y requisitos [95].

Aunque existen diversos procedimientos para aplicar esto en el campo que nos ocupa, resulta conveniente revisar algunos trabajos destacados llevados a cabo por investigadores de distintas especialidades:

Medisetti *et al.* [84], presentan LitAI, una propuesta novedosa que fusiona instrumentos de inteligencia artificial generativa disponibles sin costo alguno con metodologías convencionales, con el objetivo de optimizar la localización de información en múltiples formatos dentro de textos literarios. El sistema incorpora tecnologías como el reconocimiento automático de caracteres (OCR) junto con plataformas de IA generativa, facilitando la extracción eficiente de contenido textual, datos tabulares e imágenes almacenados en documentos PDF. Con este fin, los autores

diseñaron indicaciones personalizadas (prompts) que se valen del reconocimiento de patrones contextuales y métodos avanzados de formulación de consultas, consiguiendo así una captura rigurosa de los datos.

Tal como se aprecia en la Figura 8, LitAI ha sido concebido para agilizar el análisis de trabajos académicos siguiendo una metodología ordenada que comienza con la aplicación de OCR (pytesseract), el cual traduce documentos escaneados en formato de texto procesable. Utiliza indicaciones personalizadas para identificar y extraer de manera automática apartados fundamentales como el Resumen, Introducción, Metodología y Conclusión, permitiendo así un examen detallado de cada componente de la literatura. Por otra parte, convierte datos tabulares en representaciones textuales y formatos CSV para simplificar su manipulación, e integra GPT-4 Vision con el propósito de recuperar imágenes, crear leyendas descriptivas y formular interpretaciones detalladas. El resultado es un documento consolidado y estructurado del material procesado, lo que redunda en un aumento considerable en la disponibilidad y relevancia práctica de los documentos de carácter científico [84].

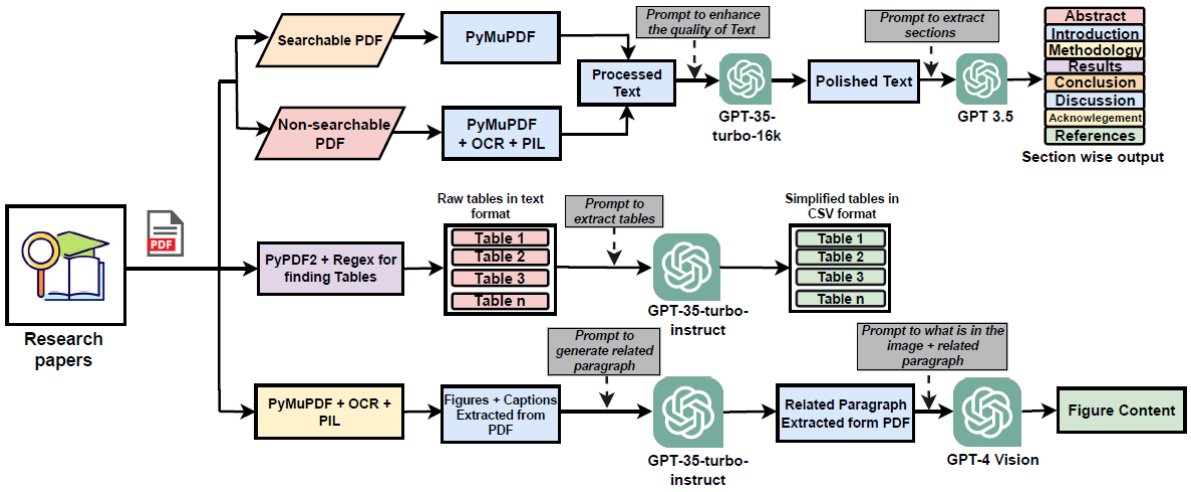


**Figura 8**. Resumen del marco LitAI [84].

Sriman *et al.* [96], presenta un modelo conversacional de inteligencia artificial (IA) que pueda utilizarse para comunicarse de manera eficiente con múltiples documentos PDF, en la figura 9 indican la arquitectura de Interacción Inteligente de Documentos (Intelligent Document Interaction -IDI) comienza con una interfaz web intuitiva, creada con la biblioteca Streamlit, donde los usuarios pueden cargar documentos PDF. Para la extracción de texto se utiliza PyPDF2, lo

cual constituye el primer paso en la cadena de procesamiento del lenguaje natural (NLP). El texto extraído se transforma luego en vectores numéricos mediante un módulo de incrustación semántica, lo que permite su compatibilidad con técnicas de aprendizaje automático. FAISS [97] se emplea para indexar estas incrustaciones, lo que permite realizar búsquedas por similitud de manera eficiente.

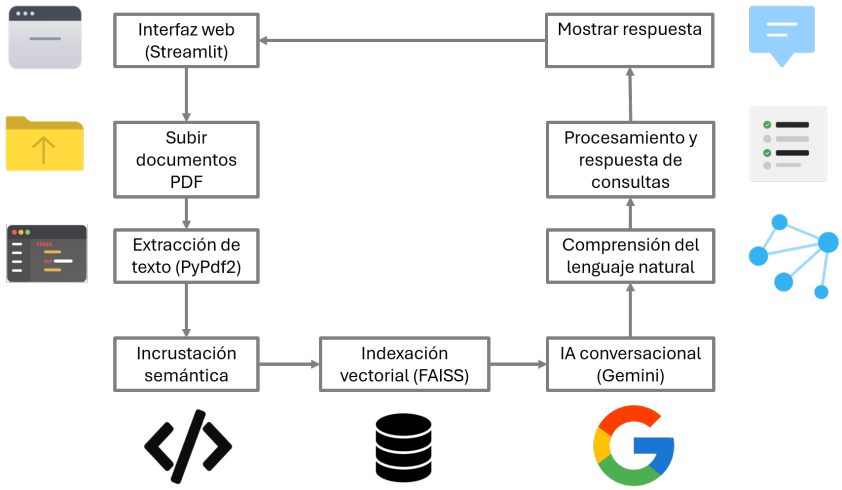


**Figura 9**. Arquitectura Interacción Inteligente de Documentos (Intelligent Document Interaction -IDI) [96].

Patil *et al.* [98], manifiestan que la construcción de un sistema basado en inteligencia artificial ayuda a los lectores a interactuar con archivos PDF a través de consultas y respuestas en lenguaje natural, dicho sistema usa técnicas avanzadas de procesamiento de lenguaje natural, como representaciones contextualizadas a nivel de documento, para facilitar un análisis efectivo de los documentos, de modo que los usuarios puedan localizar información de manera fácil y precisa. El uso de este tipo de análisis documental y mecanismos de almacenamiento vectorial, combinado con modelos de IA generativa y una arquitectura basada en transformadores, ofrece un enfoque innovador para consultar el contenido de documentos PDF, superando las limitaciones de los sistemas convencionales de recuperación de documentos.

En la figura 10 se muestra la estructura arquitectónica de un marco basado en inteligencia artificial para documentos PDF, así como el mecanismo de consulta y respuesta, el cual se divide en los siguientes componentes para garantizar interfaces de usuario naturales:

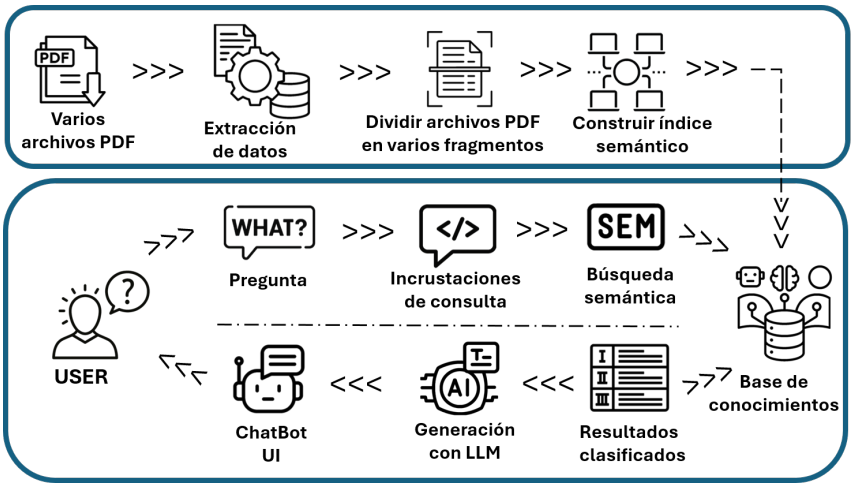


**Figura 10**. Arquitectura del sistema [98].

## 2.21 Backend

El BackEnd es la parte del software encargada del acceso a los datos y de la lógica interna de la aplicación, y no es visible ni accesible directamente para el usuario final. Esta capa gestiona cómo se procesan y almacenan los datos, y se encarga de que todo funcione correctamente detrás de escena [99]. Tener en cuenta que los datos de una aplicación se almacenan en una base de datos dentro de un servidor [100].

## 2.22 Django (Python)

**Python** es de código abierto con "un lenguaje de programación que te permite trabajar rápidamente e integrar sistemas de manera más efectiva" [101]. Además, Toal *et al.*, dicen que ofrecen una descripción más detallada del lenguaje, destacando sus características clave y su aplicabilidad en diversos contextos de desarrollo:

> Guido van Rossum comenzó a trabajar en Python en 1989 y ha supervisado gran parte del desarrollo de este popular lenguaje durante tres décadas. La comunidad de Python le otorgó el título de Dictador Benevolente de por Vida, o BDFL, título que mantuvo hasta 2018. Python es conocido por ser fácil de leer y escribir. Su sintaxis tiene un estilo ligero y limpio. Un gran número de tipos de datos, incluyendo tuplas, listas, conjuntos y diccionarios, están integrados con formas

> sintácticas convenientes. Aunque es particularmente poderoso como lenguaje de scripting, es un lenguaje verdaderamente de propósito general, habiéndose utilizado en dominios que van desde videojuegos y entretenimiento hasta aplicaciones web, estadísticas y ciencia de datos. El lenguaje también es notable por su cultura. Los aficionados de Python, conocidos como Pythonistas, se enorgullecen de escribir código elegante, expresivo y "pythónico". Aprenden y practican El Zen de Python [102], una colección de buenas prácticas y filosofías que ayudan a dar a las aplicaciones una sensación característica [103, p. 83].

**Django** es un framework web de alto nivel, de código abierto, desarrollado en Python. Su principal ventaja es que permite crear aplicaciones de manera rápida, segura y eficiente, utilizando menos código que otros frameworks similares. Además, cuenta con una amplia comunidad de desarrolladores a nivel global, que contribuye constantemente con mejoras y actualizaciones tanto del framework como de su documentación [104], [105].

**Arquitectura de Django** se apoya en una arquitectura similar al modelo MVC (Modelo – Vista – Controlador), aunque la adapta a su propia estructura conocida como MVT (Modelo "Model" – Vista "View"– Plantilla "Template"). En este enfoque, la capa Modelo (Model) contiene toda la información necesaria para definir, estructurar y gestionar los datos de la aplicación web; de hecho, también se le conoce como la capa de acceso a la base de datos. Por su parte, la capa Plantilla (Template) se encarga de la presentación, es decir, define cómo se muestran los datos al usuario. Finalmente, la capa Vista (View) contiene la lógica de la aplicación: accede a los datos del modelo y los pasa a la plantilla adecuada para generar la respuesta que verá el usuario [105].

La Figura 11 ilustra cómo está estructurada la arquitectura del framework Django [106].

## 2.23 Application Programming Interfaces - APIs

Las interfaces de programación de aplicaciones, en ingles Application Programming Interfaces (en adelante referidas como APIs) son una de esas innovaciones en el ámbito del desarrollo de software. Las APIs forman un componente integral del ecosistema de software [107], es de-

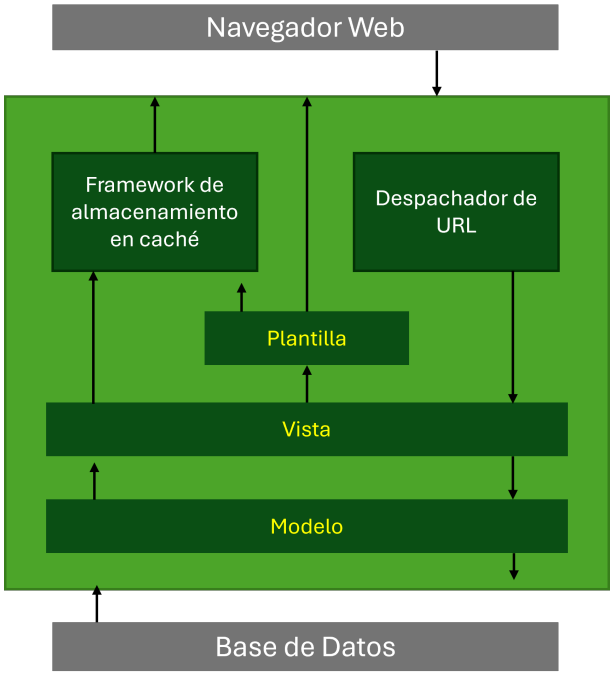


**Figura 11**. Arquitectura de Django [106].

cir que se han convertido en una forma ideal de construir soluciones de software de gran escala sobre una plataforma tecnológica común [108].

Una API es una interfaz que nos permite que diferentes aplicaciones se comuniquen y compartan información entre sí de manera programada, es decir, mediante código o desde la terminal, sin necesidad de usar una interfaz gráfica. A diferencia de las interfaces visuales, que requieren interacción directa del usuario, las interfaces programadas automatizan esa comunicación del propio software [109]. Para usar las APIs, es necesario mencionar que "existen múltiples tipos de interfaces de aplicación, como las interfaces de línea de comandos (CLI; interfaces que te permiten usar una aplicación desde una terminal), interfaces de escritorio, interfaces web o interfaces de API web" [109]. Como se puede ver en la figura 12, una aplicación puede tener una o más de estas interfaces.

### 2.23.1 FastAPI

FastAPI es un framework de API web construido sobre Starlette. Starlette es un framework web ASGI (interfaz de puerta de enlace de servidor asincrónica) de alto rendimiento, ligero y

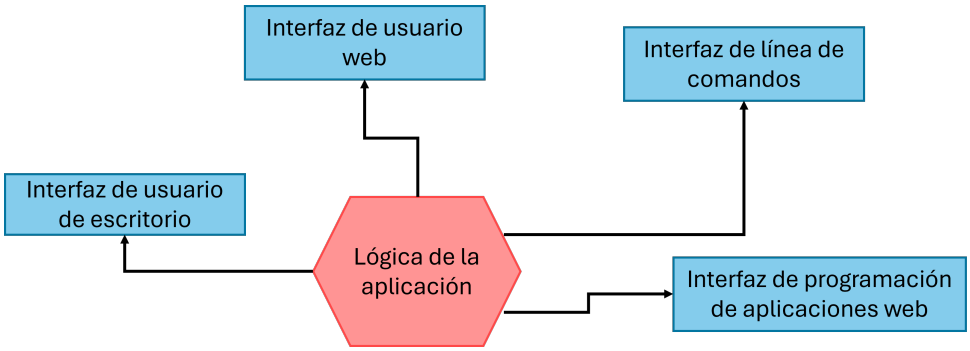


**Figura 12**. Una aplicación puede tener múltiples interfaces, como una API web, una CLI, una interfaz web y una interfaz de escritorio [109].

asincrónico, lo que significa que podemos implementar nuestros servicios como una colección de tareas asincrónicas para ganar rendimiento en nuestras aplicaciones. Además, fastAPI utiliza pydantic para la validación de datos. La siguiente figura 13 ilustra cómo se integran todas estas tecnologías diferentes [109].

Uvicorn es un servidor web asincrónico que se usa comúnmente para ejecutar aplicaciones de Starlette. Uvicorn maneja las solicitudes HTTP y las pasa a Starlette, que funciona dentro de tu aplicación para responder cuando llega una solicitud al servidor. FastAPI está construido sobre Starlette, y mejora las rutas de Starlette con funcionalidad de validación de datos y documentación de API [109].

### 2.23.2 Flask

Flask es un framework web ligero escrito en Python, se convirtió en un competidor muy popular en el mundo de los frameworks web de Python. Ahora es uno de los frameworks web de Python más ampliamente utilizados por start-ups, y está siendo comúnmente aceptado como la herramienta perfecta para soluciones rápidas y simples en la mayoría de las empresas. En su núcleo, proporciona un conjunto de bibliotecas potentes para manejar las tareas más comunes del desarrollo web, tales como [110]:

- Enrutamiento de URL que facilita mapear URLs a tu código.
- Plantillas con Jinja2 y su proceso de renderización, es de los motores más potentes de

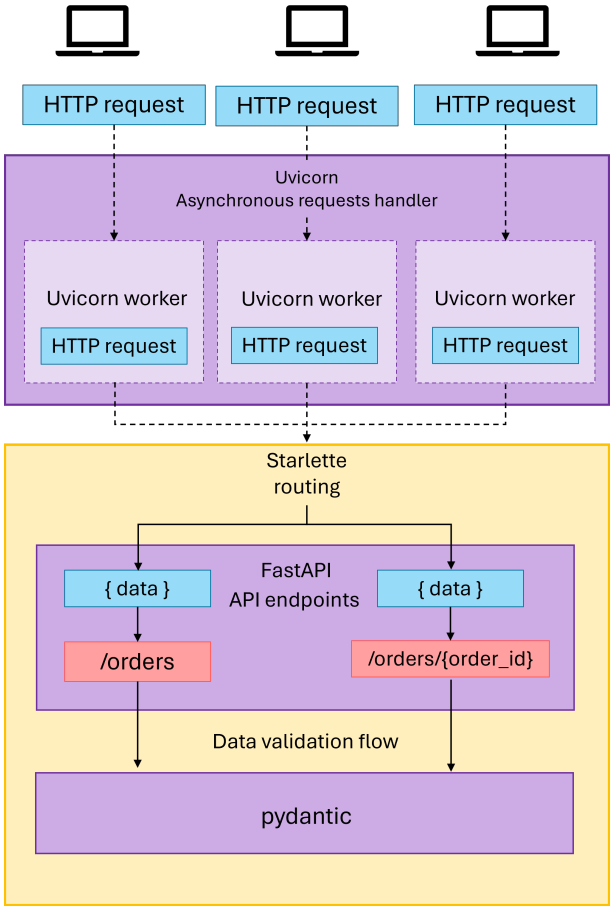


**Figura 13**. Estructura de cómo funciona FastAPI [109].

Python para plantillas.

- Gestión de sesiones y protección de cookies.
- Solicitudes y análisis en HTTP con el manejo de respuestas flexibles.
- Depurador interactivo basado en la web
- Gestión de configuración de aplicaciones fácil de usar y flexible [110].

### 2.23.3 Almacenamiento (vectores)

Almacenamiento de vectores se refiere a la base de datos vectoriales que es un tipo de base de datos que almacena objetos junto con su representación en forma de vectores. Esto permite no solo buscar por un identificador, como ocurre en una base de datos relacional, sino también generar un embedding de la consulta para realizar una búsqueda semántica [111].

Es importante destacar la definición de embeddings, ya que resulta fundamental para comprender el funcionamiento de las bases de datos vectoriales, ya que son representaciones vectoriales densas de palabras o frases en un espacio continuo de baja dimensión, es decir que en lugar de tratar cada palabra como una entidad independiente, los embeddings permiten a los modelos entender las similitudes y diferencias entre las palabras en función de su contexto [112].

Para este tipo de búsqueda, normalmente se utilizan estructuras de datos probabilísticas que reducen la necesidad de explorar todo el espacio vectorial donde se encuentran los embeddings. Sobre estas estructuras se aplican diferentes versiones del algoritmo kNN aproximado [111].

## 2.24 Diseño de interfaz web para carga/visualización/búsqueda (UI/UX)

El diseño web es el proceso de crear los elementos visuales y funcionales con los que los usuarios interactúan en un sitio, incluyendo la estructura, los colores y la tipografía. Pero no se trata solo de apariencia: un buen diseño también debe centrarse en la experiencia del usuario y en la usabilidad, asegurando que la navegación sea intuitiva y que las tareas puedan realizarse sin dificultad. Para que una web sea efectiva, debe combinar tres aspectos clave: ser visualmente atractiva, fácil de usar y cumplir con los objetivos funcionales para los que fue creada [113].

### 2.24.1 Búsqueda semántica

Los motores de búsqueda son herramientas informáticas diseñadas para localizar archivos alojados en servidores web. Muchos de ellos se enfocan en encontrar páginas web, siendo Google [114] el ejemplo más reconocido.

En los últimos años, estas búsquedas han mejorado notablemente en su capacidad para ofrecer a los usuarios resultados de búsqueda más precisos y relevantes, pues gracias a la búsqueda impulsada por la inteligencia artificial ahora van más allá de las coincidencias exactas de palabras o sus variantes, enfocándose en entender la intención detrás de la consulta y su similitud con fragmentos de texto. Ya que esto permite encontrar información más relevante de manera

más rápida y precisa que con los métodos estadísticos tradicionales [115], [116].

### 2.24.2 La experiencia de usuario UX

La experiencia de usuario (User eXperience, UX por sus siglas en inglés) se refiere a cómo se siente e interactúa un usuario al utilizar un producto tecnológico [117]. En este contexto, el papel del usuario es fundamental, ya que su percepción, satisfacción y retroalimentación directa permiten valorar la calidad de la experiencia ofrecida por un sistema. Como señala Hassenzahl, la interacción con la tecnología desde la perspectiva del usuario es clave para comprender y mejorar dicha experiencia [118].

### 2.24.3 Interfaz de usuario UI

El diseño de interfaz se enfoca en la estructura y apariencia visual de un software, sitio web o aplicación, buscando crear un entorno digital estéticamente agradable y funcional para el usuario. Este tipo de diseño, conocido como UI (User Interface), integra elementos del diseño gráfico como la tipografía, los colores, las imágenes vectoriales y las ilustraciones, combinándolos de manera armoniosa para lograr una presentación visual atractiva que favorezca la interacción y mejore la experiencia del usuario [119].

## 2.25 Frameworks

Los frameworks tienen como objetivo organizar y estructurar el código de un sistema, proporcionando una base o esquema como un patrón o esqueleto que guía el desarrollo e implementación de aplicaciones. Utilizar un framework en cualquier tipo de proyecto permite reducir significativamente el tiempo de desarrollo, facilita la implementación y contribuye a crear soluciones más mantenibles y escalables, adaptándose a las necesidades y características específicas de cada aplicación [120].

Permite que cualquier lenguaje de programación se adapte a la estructura del framework, facilitando así una mayor comprensión y mantenibilidad del código. Además, promueve la separación de la aplicación en capas, lo que mejora su organización y escalabilidad.

En [120], se indica que, por lo general, una aplicación se divide en tres capas principales:

- La lógica de presentación que administra las interacciones entre el usuario y el software.
- La Lógica de datos que permite el acceso a un agente de almacenamiento persistente u otros.
- La lógica de dominio o de negocio, que manipula los modelos de datos de acuerdo a los comandos recibidos desde la presentación.

### 2.25.1 LangChain/LLamaIndex

- **LangChain.-** fue lanzado en octubre de 2022 como un proyecto de código abierto por Harrison Chase, quien lo desarrolló mientras trabajaba en la startup de machine learning Robust Intelligence. Este framework nació ante la necesidad de contar con herramientas de orquestación para aplicaciones basadas en modelos de lenguaje (LLM) y, en poco tiempo, se convirtió en una de las soluciones más populares entre la comunidad de desarrolladores [121].

  De igual manera, conviene mencionar los usos más destacados de LangChain, que según [121] incluyen múltiples aplicaciones destinadas a conectar modelos de lenguaje con herramientas de terceros y procesos de trabajo elaborados.

  - *Orquestación de componentes LLM:* Estos permiten la coordinación e integración de varios componentes para aplicaciones de IA.
  - *Desarrollo de aplicaciones conversacionales:* Son herramientas para creación de chatbots, así como de asistentes virtuales.

- *Gestión de cadenas de procesamiento:* Permite crear flujos complejos de procesamiento de lenguaje natural.
- *Integración con fuentes de datos externas:* Conecta LLMs con bases de datos, APIs y otros sistemas.

- **LlamaIndex.-** fue creado por Jerry Liu como un framework avanzado de gestión de datos para aplicaciones basadas en modelos de lenguaje (LLM). Surgió ante la necesidad de conectar de forma eficiente estos modelos con datos empresariales y personales, facilitando así el acceso y la integración de información relevante en los procesos de inteligencia artificial [121].

  Usos Principales:

  - *Indexación y recuperación de información:* Tiene como fuerte la conexión entre modelos de lenguaje grandes y todo tipo de bases de datos, sean organizadas o completamente desestructuradas.
  - *Asistentes corporativos impulsados por conocimiento:* Permite desarrollar soluciones que pueden analizar y hacer consultas sobre los datos de la empresa.
  - *Manejo de archivos documentales:* Permite la incorporación fluida de documentos en formato PDF, hojas de cálculo Excel y otros tipos de archivos dentro de ecosistemas LLM.
  - *Framework de datos para LLMs:* Ofrece recursos fundamentales para administrar y acceder a información en proyectos de inteligencia artificial [121].

### 2.25.2 Streamlit/Gradio (UI)

- **Streamlit.-** Es un marco de desarrollo de código abierto diseñado para facilitar la creación de aplicaciones web interactivas, especialmente útil para presentar de forma clara y accesible los datos obtenidos a partir del análisis del tráfico de red [122].

  García Espinosa *et al.*, mencionan que Streamlit es un framework de código abierto en Python diseñado para crear de forma rápida y sencilla aplicaciones web orientadas a la

ciencia de datos y al aprendizaje automático. Su principal ventaja es que permite a los desarrolladores construir e implementar aplicaciones en cuestión de horas, usando únicamente Python, lo que antes podía llevar semanas. Gracias a su curva de aprendizaje muy accesible y su compatibilidad con las principales bibliotecas del ecosistema Python "como pandas, matplotlib, seaborn, plotly, Keras, PyTorch o SymPy (con soporte para LaTeX)", ha ganado gran popularidad y respaldo dentro de la comunidad [123].

- **Gradio.-** es una librería de Python que funciona perfectamente con los principales frameworks de aprendizaje automático, incluyendo PyTorch y TensorFlow, permitiendo desarrollar interfaces de usuario de manera rápida y directa mediante código Python estándar. Se ha convertido en una herramienta fundamental tanto en contextos académicos como en centros de investigación de empresas tecnológicas de envergadura, tales como Cisco y Amazon. Lo que distingue a Gradio es su catálogo de componentes de interfaz totalmente adaptables, pensados para acoplarse sin complicaciones a modelos de machine learning, además de su enfoque minimalista en cuanto a configuración, ya que basta con escribir muy pocas líneas de código. Puede ejecutarse directamente desde cuadernos Jupyter, Google Colab o scripts en Python [123].

## 2.26 PyPDF/PdfPlumber

- **Pypdf.-** es una biblioteca gratuita y de código abierto, desarrollada íntegramente en Python, que permite manipular archivos PDF de diversas formas, como dividir, combinar, recortar y transformar sus páginas. Además, ofrece la posibilidad de añadir datos personalizados, opciones de visualización y contraseñas a los documentos. También es capaz de extraer texto y metadatos, lo que la convierte en una herramienta versátil para trabajar con archivos PDF desde Python [124].

- **PdfPlumber.-** es una biblioteca de extracción de texto de archivos PDF basada en PDFMiner, que permite acceder a la información carácter por carácter, incluyendo detalles como la fuente, el tamaño de letra y la posición del texto. Además, ofrece herramientas

para extraer tablas contenidas en los documentos [125]. Aunque los datos necesarios para reconstruir la estructura del texto son similares a los que se pueden obtener con PDFMiner en su modo de extracción a XML (aunque en ese caso se requiere un procesamiento adicional), para acceder a información detallada como fuente y tamaño de letra es imprescindible analizar el contenido carácter por carácter. Por ello, al igual que ocurre con la extracción de PDFMiner en formato XML, utilizar la salida en formato HTML de PDFPlumber resulta una opción más adecuada para ese tipo de análisis [126].

## 2.27 Evaluación de sistemas basados en IA

La evaluación rigurosa de los sistemas de inteligencia artificial es fundamental para garantizar su eficacia, equidad y seguridad, sin descuidar principios clave como la integridad, disponibilidad y confidencialidad de los datos. Si bien la IA ha transformado sectores como la salud, la banca, la educación y la industria, su implementación debe analizarse con cuidado para asegurar que cumpla con estándares éticos y técnicos. En particular, en esta región, la evaluación debe abarcar no solo aspectos técnicos, sino también considerar los posibles impactos sociales y éticos [127]. La falta de infraestructura tecnológica en algunos países representa una barrera importante para su adopción efectiva, y las diferencias en políticas públicas y marcos regulatorios dificultan la creación de estándares uniformes [128].

## 2.28 Métodos para medir precisión en extracción de texto

Según lo expuesto en [129], los métodos y técnicas de extracción de texto comprenden los siguientes enfoques:

***Reconocimiento Óptico de Caracteres (OCR):*** implica el sofisticado proceso de convertir documentos escaneados en texto personalizable mediante el uso de algoritmos avanzados de reconocimiento de patrones. Reconocimiento de Escritura Manual: identificación y transcripción de texto escrito a mano en manuscritos o documentos históricos.

***Métodos Basados en Características:*** análisis de características específicas como estilos de fuente, tamaños y disposiciones espaciales para extraer texto.

***Modelos de Aprendizaje Automático:*** entrenamiento de modelos para reconocer y extraer patrones de texto en documentos históricos.

***Procesamiento de Imágenes:*** se dedican esfuerzos a mejorar la claridad de lo que se captura, eliminando ruido visual y acentuando las diferencias de tonalidad, lo que permite que la lectura de caracteres sea considerablemente más precisa.

***Procesamiento de Lenguaje Natural (PLN):*** después de obtener el texto, este pasa por un análisis exhaustivo que facilita su correcta interpretación y, cuando sea requerido, su conversión a otros idiomas.

***Enfoques Híbridos:*** cuando se combinan diversas metodologías técnicas en un enfoque integral, se logran resultados considerablemente más exactos y fiables en el proceso de identificación y extracción de datos textuales.

Los textos en imágenes contienen información útil e informativa que puede utilizarse para comprender claramente el contenido de las imágenes. Una imagen de documento contiene información diversa como textos, imágenes y gráficos, es decir, dibujos lineales y bocetos. Estas imágenes suelen obtenerse al escanear revistas, documentos antiguos o deteriorados, textos manuscritos, libros impresos, portadas de libros a color, periódicos, entre otros [130].

Ciertamente, cuando nos enfrentamos a documentos escaneados, aparecen toda clase de dificultades: la mala calidad del contraste, una resolución insuficiente, y además problemas como el desplazamiento de colores, fondos excesivamente complejos a nivel visual, o matices del texto que se mimetizan con lo que los rodea.

## 2.29 Sistema de reconocimiento de texto

Esta sección [131], presenta una descripción general y simplificada de cómo funciona un sistema de reconocimiento de texto, tal como se ilustra en la Figura 14. En términos básicos, el sistema recibe como entrada una imagen que contiene texto, y produce como salida ese contenido textual en formato digital, es decir, listo para ser leído y procesado por una computadora. El sistema se compone de tres módulos principales:

- Preprocesamiento
- Reconocimiento de texto
- Posprocesamiento

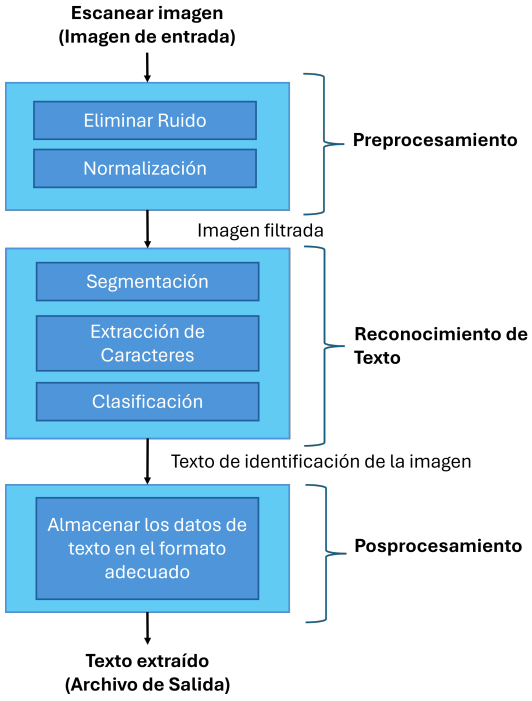


**Figura 14**. Arquitectura de reconocimiento de texto [131].

A continuación, se explica brevemente el rol de cada uno:

- **Módulo de Preprocesamiento.-** Todo comienza con la digitalización de un documento en papel, normalmente a través de un escáner óptico. El resultado es una imagen compuesta

por píxeles. Esta imagen debe ser optimizada antes de poder extraer información útil, por lo que en esta fase se aplican técnicas de mejora que permiten obtener mejores resultados más adelante.

Entre las operaciones más comunes se encuentran:

- **Eliminación de ruido.-** Este paso es crucial. Al reducir o eliminar el "ruido" visual (pequeñas imperfecciones o distorsiones), se mejora la claridad de la imagen, lo que facilita una detección de texto mucho más precisa. Se utilizan filtros como el de media, min-max o gaussiano, entre otros.
- **Normalización.-** Permite estandarizar el tamaño, la inclinación y la orientación de los caracteres para que el sistema los procese de manera uniforme. Es una etapa fundamental para asegurar la consistencia del reconocimiento.
- **Binarización.-** Aquí, la imagen en escala de grises se convierte en una imagen en blanco y negro (binaria). Esta transformación permite separar claramente el texto del fondo, lo que es esencial para procesos posteriores como la segmentación. La Figura 15 muestra una comparación entre la imagen original a color, su versión en escala de grises y el resultado final en binario.

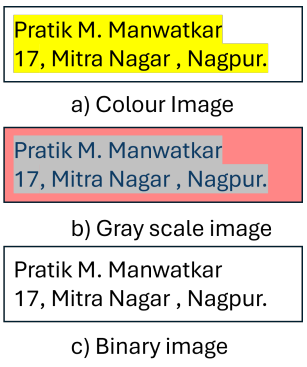


**Figura 15**. Proceso de binarización en el reconocimiento de texto [131].

- **Módulo de Reconocimiento de Texto.-** Toma como punto de partida la imagen que ha sido sometida a preprocesamiento y se dedica a detectar e interpretar digitalmente el contenido textual presente en ella. Con este propósito, se implementan tres metodologías fundamentales:

- **Segmentación.-** Se trata de dividir la imagen en caracteres individuales, permitiendo así su análisis independiente.
- **Extracción de características.-** En esta etapa se localizan y eligen los atributos más significativos presentes en cada carácter, los cuales funcionarán posteriormente como fundamento para su identificación precisa. Entre las metodologías más comunes encontramos:
    - Principal Component Analysis (PCA)
    - Linear Discriminant Analysis (LDA)
    - Independient Component Analysis (ICA)
    - String code (CC)
    - Gestión de zonas
    - Características realizadas en gradientes
    - Histogramas, entre otros.
- **Clasificación.-** Después de haber sacado las características, llega el momento de categorizar cada uno de los caracteres, lo que significa comparar su aspecto con modelos que ya se conocen para poder determinar si se trata de una letra, un número o algún símbolo en particular. Entre las técnicas más utilizadas para realizar esta etapa encontramos:
    - Artificial Neural Networks (ANN)
    - Plantillas comparadas (Template Matching)
    - Support Vector Machines (SVM)

- **Módulo de Posprocesamiento.-** Finalmente, el texto reconocido se guarda en un archivo utilizable de formato digital, como un archivo TXT o documento editable (como un documento Word). Esto hace que el contenido extraído sea editable, buscable o accionable en otros sistemas.

## 2.30 Pruebas de eficiencia (tiempo de respuesta, escalabilidad)

La eficiencia de los sistemas basados en inteligencia artificial (IA) se ha vuelto un criterio esencial para su evaluación, especialmente en entornos que exigen respuestas rápidas y un procesamiento intensivo de datos. En este contexto, dos métricas clave son el tiempo de respuesta, que mide la latencia entre la entrada del usuario y la salida generada por el sistema, y la escalabilidad, que determina la capacidad del sistema para mantener un rendimiento óptimo incluso cuando aumenta la carga de trabajo o el número de usuarios simultáneos [132].

Evaluar la eficiencia de los sistemas de IA debe ser analizado por dos métricas clave: el tiempo de respuesta conocida como latencia y la escalabilidad, pues ambos factores son fundamentales para garantizar un buen desempeño en entornos de producción. La latencia se refiere al tiempo que demora el sistema en responder después de recibir una orden, algo especialmente relevante en aplicaciones interactivas como asistentes virtuales o herramientas de análisis en tiempo real. Por otro lado, la escalabilidad mide qué tan bien se mantiene el rendimiento cuando aumenta la carga de trabajo o el número de usuarios. Según estudios sobre el rendimiento en sistemas de IA, se utilizan pruebas de benchmarking que evalúan aspectos como el tiempo de entrenamiento, el uso de memoria y la capacidad de respuesta ante distintos volúmenes de datos. Estas pruebas permiten simular escenarios de alta demanda, identificar cuellos de botella y mejorar el rendimiento del sistema [133].

El tiempo de respuesta es una métrica crítica en las pruebas de rendimiento, especialmente en sistemas de IA y gestión documental, ya que mide el intervalo entre la solicitud del usuario y la respuesta del sistema, también conocido como “round-trip time” [134]. Esta métrica es esencial para garantizar una experiencia fluida, evitando demoras que puedan afectar la satisfacción o provocar la pérdida de usuarios [135], [136].

En un entorno de IA que automatiza procesos como clasificación, OCR o búsqueda de documentos, el tiempo de respuesta debe mantenerse dentro de límites aceptables por lo general por debajo de un segundo según estándares de usabilidad [134].

## 2.31 Benchmarks en documentos de calidad variable

En [137], expone que una forma de traducir este término al español podría ser “análisis técnicos competitivos”. Sin embargo, como ocurre en muchos otros campos, se ha adoptado el término en inglés. Esto ayuda a distinguir este tipo de estudios de otros parecidos, como las comparaciones entre productos o los análisis de impacto.

El benchmarking, según la definición del American Productivity & Quality Center (APQC) [138], es un proceso continuo y metódico de medición y comparación de los procesos empresariales de una organización con los de las empresas líderes a nivel mundial, siempre que exista compatibilidad entre ellas. Su objetivo es obtener información valiosa que permita identificar buenas prácticas y aplicar mejoras que contribuyan al rendimiento y desempeño de la organización [139].

Una vez analizado el termino de benchmark, puede afirmarse que esta metodología permite a las organizaciones avanzar de manera más eficiente en el desarrollo de productos, tecnologías o sistemas, al basarse en la comparación con prácticas exitosas ya implementadas. Como señala Solfa [140], “es una técnica muy eficiente para introducir mejoras en las organizaciones, ya que pueden incorporarse y adaptarse procesos cuya efectividad ya ha sido probada por otras organizaciones. Por esta razón, ayuda a las organizaciones a introducir mejoras rápidamente”.

Los benchmarks se revelan como una herramienta tremendamente valiosa en aquellos escenarios donde la variabilidad en la calidad documental es una realidad, particularmente en empresas que se enfrentan a la administración de volúmenes considerables de documentos. A través de esta metodología es posible fijar parámetros de referencia sólidos y medibles que catalicen la mejora de manera sostenida. Asimismo, posibilita cotejar nuestros procedimientos, ofertas o prestaciones frente a los competidores o frente a aquello que hacen las grandes referencias del mercado, circunstancia que resulta esencial para descubrir espacios de optimización y establecer objetivos tanto específicos como realistas.

También es importante destacar que, para desarrollar un buen plan de benchmarking, es

necesario seguir algunas etapas. A continuación, se presentan estas según criterios propuestos por algunos autores. Según Sánchez citado por Sánchez y Douglas [141] las etapas son:

- **Planificación**, en la cual se identifica el objeto a ser estudiado, las empresas a comparar y los métodos de recolección de datos.
- **Análisis**, donde se mide el desempeño que tienen las organizaciones a comparar y su desempeño futuro.
- **Integración**, se comunican los resultados de las comparaciones realizadas a las empresas sometidas a estudio, y se establecen los objetivos a perseguir con la implementación del benchmarking.
- **Acción**, se desarrollan procedimientos de mejora continua, e implementación de planes para monitorear los resultados, y se desarrollan planes de acción.
- **Madurez**, en ella se escoge de todas las organizaciones comparadas, la organización que va a ser sometida a estudio.

Es fundamental que una organización antes de tomar una decisión conozca y comprenda bien cada una de las etapas antes mencionadas como la planificación, el análisis, la integración, la acción y la madurez ya que están estrechamente conectadas entre sí. Seguirlas al pie de la letra puede ser clave para lograr comparaciones efectivas que conduzcan al éxito o, en caso contrario, al fracaso de la estrategia.

# CAPÍTULO III

# MATERIALES Y MÉTODOS

## 3.1 Tipo de Investigación

La presente investigación se enmarca en la línea de Desarrollo y Aplicación de Software, con enfoque en Inteligencia Artificial y Seguridad Cibernética. La presente investigación corresponde a un estudio de tipo aplicada, debido a que busca resolver un problema real relacionado con la gestión documental mediante el desarrollo de un sistema inteligente para la extracción y análisis automatizado de información contenida en documentos PDF escaneados. El propósito principal consiste en optimizar los procesos de organización, recuperación y análisis documental utilizando tecnologías de inteligencia artificial generativa y OCR estructurado.

### 3.1.1 Enfoque Cuantitativo

El enfoque de la investigación es cuantitativo, ya que se emplean métricas objetivas para evaluar el rendimiento y precisión del sistema propuesto, tales como Character Error Rate (CER), Word Error Rate (WER), tiempos de procesamiento y benchmarks comparativos. Adicionalmente, la investigación posee un diseño experimental, debido a que se comparan distintas configuraciones de procesamiento OCR y análisis documental, evaluando el comportamiento del sistema propuesto frente a métodos baseline tradicionales.

## 3.2 Metodología de Desarrollo

Para el desarrollo del sistema se utilizó la metodología en cascada, debido a que proporciona una estructura secuencial y organizada para la planificación, diseño, implementación y evaluación de proyectos de software. En este sentido, dicha metodología permitió definir claramente cada una de las etapas del desarrollo, facilitando el control y seguimiento de las actividades ejecutadas durante el proyecto.

Según Milena Velásquez Restrepo et al., en el desarrollo de software suelen emplearse metodologías tradicionales, destacando el modelo en cascada por su carácter ordenado y progresivo [142], tal como se muestra en la Figura 16, el modelo de desarrollo en cascada se estructura en las siguientes etapas:

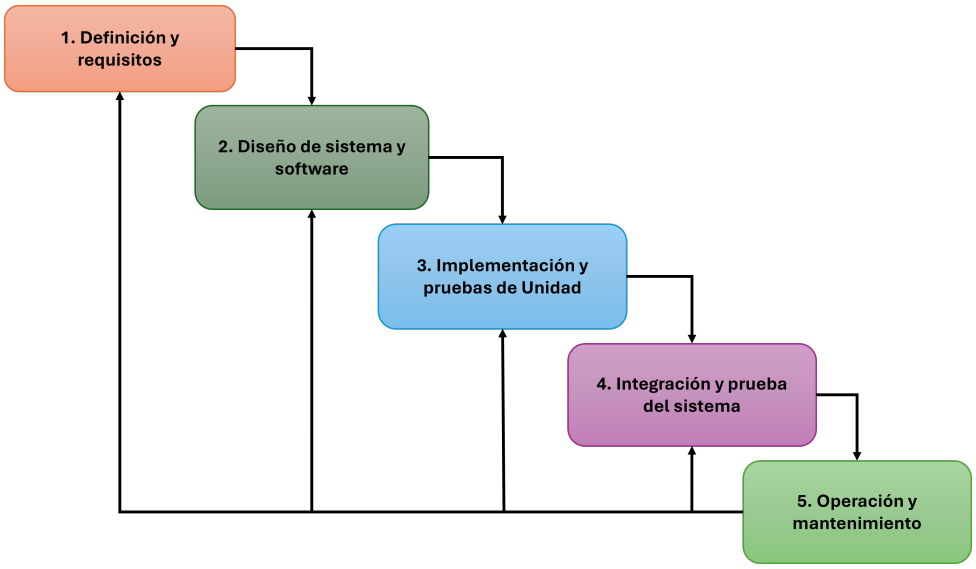


**Figura 16**. Modelo de desarrollo en cascada.

- ***Definición y requisitos:*** Se determina los servicios que ofrecerá el sistema, así como sus restricciones y metas, esto se lo hace juntamente con el usuario, con el fin de llegar a detallar cada una de las funciones que requiera el sistema.

- ***Diseño de sistema y software:*** Se divide el proyecto en componentes, ya sean de hardware o software identificando las principales abstracciones llegando a que se relacionen sus elementos entre sí, y se define la arquitectura global del sistema.

- ***Implementación y pruebas unitarias:*** Se desarrollan los módulos del software y se verifican por separado cada módulo, para que cada uno funcione correctamente.
- ***Integración y pruebas del sistema:*** Una vez desarrolladas todas las partes o módulos, se evalúa el funcionamiento del sistema, a fin de entregar validado y completo al cliente.
- ***Operación y mantenimiento:*** Se da seguimiento al sistema para detectar posibles errores que surjan mediante su uso, para luego incorporar mejoras, dando soporte en su funcionamiento [143].

### 3.2.1 Justificación de la Metodología en Cascada

Se optó por la metodología en cascada porque ofrece un enfoque organizado, paso a paso y bien documentado para construir sistemas de software. De esta manera, fue posible dividir el desarrollo del sistema inteligente de gestión documental en etapas muy precisas, lo que simplificó tanto la planificación como la ejecución y posterior evaluación de cada elemento tecnológico que formaba parte del proyecto de investigación.

Cabe indicar que el proyecto presenta requerimientos funcionales y no funcionales que pudieron ser identificados desde las etapas iniciales del desarrollo, aspecto que favorece la aplicación del modelo en cascada. Entre estos requerimientos se encuentran la carga de documentos PDF, extracción textual mediante OCR estructurado, recuperación tabular, visualización documental, clasificación automática, generación de resúmenes y evaluación experimental mediante métricas objetivas.

La implementación de la metodología en cascada permitió organizar el proceso de desarrollo de manera lógica y ordenada. Primero se examinaron los problemas documentales que existían, después se definió la arquitectura del sistema, se construyeron los diferentes módulos y, al final, se evaluó el funcionamiento de la solución que se había propuesto. Gracias a este enfoque, fue posible integrar paulatinamente soluciones modernas impulsadas por inteligencia artificial y procedimientos sofisticados para la administración de documentos, siempre manteniendo un

entorno de trabajo robusto y totalmente verificable.

De igual forma, el enfoque en cascada posibilitó organizar de manera apropiada la construcción de la arquitectura cliente-servidor que se planteó en este estudio. Al establecer de antemano los componentes necesarios, interfaz de usuario, servidor de aplicaciones, almacenamiento de datos, reconocimiento óptico de caracteres configurado y elementos de IA generativa, se logró desplegar el sistema de forma gradual, preservando la solidez integral de toda la infraestructura tecnológica desarrollada.

### 3.2.2 Fases del modelo en cascada

El desarrollo del software inteligente para la extracción y análisis de texto en documentos PDF escaneados se estructuró mediante el modelo en cascada, debido a que esta metodología permite organizar el ciclo de vida del software en etapas secuenciales, controladas y progresivas. En este sentido, cada fase del desarrollo fue ejecutada de manera ordenada, permitiendo establecer una relación lógica entre el análisis de requerimientos, el diseño de la arquitectura, la implementación de funcionalidades y la evaluación experimental del sistema.

**a) Fase de Análisis y Planificación**

La fase de análisis y planificación constituyó el punto de partida para el desarrollo del sistema inteligente de gestión documental. En esta etapa se realizó el estudio de la problemática existente en los procesos tradicionales de extracción y análisis de documentos PDF escaneados, identificando limitaciones relacionadas con:

- Pérdida de estructura documental.
- Se reconocen errores en textos.
- La recuperación de tablas es ineficiente.
- Existe ruido visual en los documentos legales y administrativos.

A lo largo de este trabajo, se revisaron diversos tipos de documentos que formaron parte

del proceso investigativo, incluyendo facturas, resoluciones, memorandos, oficios, solicitudes, certificados, formularios, contratos y ordenanzas. Es importante señalar que estos documentos cuentan con características bastante complejas tales como sellos institucionales, firmas, tablas, encabezados, pies de página e inconsistencias en la calidad de los escaneos, lo que posibilitó reconocer los retos más relevantes vinculados con la automatización del procesamiento de documentos.

**b) Fase de Diseño**

La fase de diseño tuvo como finalidad definir la estructura arquitectónica y organizacional del sistema antes de iniciar la implementación de los componentes funcionales. En este sentido, se diseñó una arquitectura cliente-servidor modular orientada al procesamiento automatizado de documentos PDF escaneados mediante OCR estructurado e inteligencia artificial generativa.

Cabe indicar que durante esta etapa se definió la arquitectura general del sistema, estableciendo la interacción entre frontend, backend, base de datos, almacenamiento documental y servicios de inteligencia artificial:

- **Frontend.-** aquí cabe mencionar que la implementación se llevó a cabo utilizando Next.js, ejecutándose en el puerto 3001 y ofreciendo una interfaz web moderna mediante la cual los usuarios pueden interactuar de manera fluida con la plataforma.
- **Backend.-** componente importante de la arquitectura, fue desarrollado con FastAPI y funcionó en el puerto 8001, donde se llevó a cabo toda la gestión del procesamiento documental y la comunicación con los servicios externos.
- **Base de datos y almacenamiento documental.-** La gestión de documentos y su archivo se realiza de la siguiente manera: cuando un usuario carga un fichero PDF desde la plataforma web, el sistema backend captura ese archivo, lo preserva en el almacenamiento local del servidor y registra la información relacionada en la base de datos PostgreSQL.
- **Servicios de inteligencia artificial.-** ejecuta el OCR estructurado mediante Mistral OCR, una vez procesado el documento, el sistema realiza limpieza conservadora,

normalización de tablas y análisis mediante Gemini para generar resumen y clasificación documental.

De acuerdo con lo anterior, también se diseñó el flujo de visualización por página, permitiendo mostrar simultáneamente la imagen original del PDF y el texto extraído correspondiente a cada página. En este sentido, se incorporó PyMuPDF como herramienta de renderizado de páginas PDF para facilitar la comparación visual entre el documento original y el resultado OCR.

**c) Fase de Implementación**

La fase de implementación correspondió al desarrollo de los componentes funcionales y tecnológicos definidos previamente durante la etapa de diseño. En este sentido, se desarrolló:

- **El frontend:** utilizando Next.js con el propósito de construir una interfaz web dinámica orientada a la carga, visualización y gestión de documentos PDF escaneados. Cabe indicar que la interfaz implementada permite al usuario subir documentos PDF, consultar documentos procesados, realizar búsquedas textuales, aplicar filtros por fechas, visualizar el PDF original por página y descargar resultados en formatos MD, TXT y DOCX. Asimismo, se incorporaron mecanismos visuales de carga y seguimiento del estado de procesamiento documental.
- **El backend:** fue implementado mediante FastAPI, proporcionando servicios REST encargados de gestionar el procesamiento OCR, almacenamiento documental y análisis mediante inteligencia artificial generativa. En este sentido, el backend recibe los documentos enviados desde el frontend, almacena los archivos en rutas locales y registra los metadatos correspondientes dentro de PostgreSQL.
- **Mistral OCR:** como motor principal de reconocimiento óptico de caracteres. Esta integración permitió ejecutar OCR estructurado por página, detectar tablas en formato HTML, recuperar encabezados y pies de página y conservar estructuras documentales complejas presentes dentro de los archivos PDF escaneados, también se implementó un mecanismo de fallback OCR por imagen para resolver casos donde

el OCR principal devuelve tablas incompletas o pérdida parcial de información. En estos escenarios, el sistema renderiza la página afectada como imagen mediante PyMuPDF y posteriormente la envía nuevamente a Mistral OCR con el propósito de recuperar información más precisa.

- **Sistema de limpieza documental:** se implementaron reglas orientadas a eliminar únicamente ruido visual evidente, preservando contenido relevante como títulos institucionales, tablas, identificadores documentales, referencias administrativas y datos contextuales importantes.
- **Gemini:** fue utilizado como una herramienta de inteligencia artificial generativa con el propósito de ejecutar tareas de clasificación documental, generar resúmenes de manera automática y efectuar análisis contextual sobre contenido textual que había sido previamente depurado y estructurado. Consecuentemente, esta integración facilitó notablemente la potenciación y estructuración del OCR mediante estas capacidades especiales y su análisis documental.
- **Docker Compose:** para desplegar PostgreSQL dentro de un entorno contenedorizado, facilitando la configuración reproducible del sistema y la administración de la base de datos durante el desarrollo experimental.

**d) Fase de Verificación y Evaluación**

La fase de verificación y evaluación tuvo como propósito validar el funcionamiento técnico del sistema y medir el rendimiento alcanzado durante el procesamiento documental automatizado. En este sentido, se ejecutaron pruebas funcionales, pruebas de integración, pruebas de rendimiento y evaluaciones experimentales orientadas a determinar la precisión del sistema propuesto.

Cabe indicar que las pruebas funcionales permitieron validar procesos relacionados con carga documental, extracción OCR, visualización por página, generación de resúmenes, clasificación documental, búsqueda textual y descarga de resultados. Asimismo, Se comprobó que los estados documentales (pending, processing, completed y failed) se gestionaban correctamente a lo largo de todo el flujo de procesamiento.

Por otra parte, mediante las pruebas de integración fue posible validar que la comunicación entre los distintos componentes tecnológicos del sistema funcionaba de manera adecuada. Se analizó específicamente cómo interactuaban el frontend, backend, PostgreSQL, el almacenamiento local, Mistral OCR y Gemini, lo que permitió asegurar que el sistema mantuviera su estabilidad durante el procesamiento de documentos.

Tomando en cuenta lo mencionado anteriormente, el desempeño del sistema OCR fue analizado mediante las métricas CER y WER, las cuales son frecuentemente utilizadas en estudios relacionados con reconocimiento óptico de caracteres y procesamiento de lenguaje natural [144], [145]. En particular, CER permite determinar qué tan preciso es el reconocimiento a nivel de caracteres, identificando los errores presentes respecto al texto original. Por otro lado, WER se enfoca en la precisión a nivel de palabras, calculando el porcentaje de palabras que no fueron reconocidas correctamente en comparación con el texto de referencia [146].

$$\text{Tasa de error de caracteres (CER)} = \frac{\text{Número de caracteres incorrectos}}{\text{Número total de caracteres en el texto de referencia}} \times 100 \tag{3.1}$$

$$\text{Tasa de error de palabras (WER)} = \frac{\text{Número de palabras incorrectas}}{\text{Número total de palabras en el texto de referencia}} \times 100 \tag{3.2}$$

En ambos indicadores, el numerador representa la cantidad de caracteres o palabras que presentan diferencias entre el texto generado por el OCR y el texto original de referencia. Mientras tanto, el denominador corresponde al total de caracteres o palabras que contiene dicho texto de referencia. Tanto CER como WER se basan en la distancia de Levenshtein, una técnica que permite calcular el número mínimo de cambios necesarios como inserciones, eliminaciones o sustituciones para transformar una cadena de texto en otra. Gracias a este método, es posible comparar de manera más precisa los resultados obtenidos por el

OCR con el contenido real y, de esta forma, determinar con mayor exactitud las tasas de error [34].

Asimismo, se realizaron benchmarks comparativos entre el método baseline basado en OCR plano y el método propuesto basado en OCR estructurado, recuperación tabular, limpieza conservadora y análisis con inteligencia artificial generativa. Estos benchmarks permitieron medir tiempos de procesamiento, calidad estructural y recuperación de información documental.

Finalmente, se optó por incorporar G-Eval como instrumento complementario destinado a efectuar un examen de naturaleza cualitativa, con el propósito de evaluar cuestiones relacionadas con el mantenimiento de la estructura, la fidelidad del material documental, el orden coherente de la información y la correspondencia existente entre lo que se extrajo en formato textual y cómo aparecía representado visualmente en el documento original.

Sin embargo, este tipo de evaluación terminó ocupando un lugar marginal y no llegó a convertirse en un verdadero indicador del rendimiento, en gran medida debido a que los modelos del lenguaje utilizados durante la etapa experimental mostraban de forma inherente un funcionamiento de carácter probabilístico.

## 3.3 Variables de Estudio

### 3.3.1 Variable Independiente

La variable independiente corresponde a la implementación de tecnologías basadas en inteligencia artificial generativa y OCR estructurado aplicadas al procesamiento automatizado de documentos PDF escaneados. Cabe indicar que esta variable involucra procesos relacionados con la extracción inteligente de texto, recuperación tabular, limpieza documental y análisis contextual de la información.

X = Uso de IA generativa y OCR estructurado

En este sentido, el sistema incorpora servicios avanzados de OCR estructurado mediante Mistral OCR y análisis documental mediante Gemini, permitiendo preservar la estructura del documento original y mejorar la recuperación de información relevante. Por consiguiente, esta variable constituye el eje principal sobre el cual se desarrolla la propuesta tecnológica de la investigación.

### 3.3.2 Variables Dependientes

El nivel de optimización que se logra en los procesos de gestión documental al implementar el sistema inteligente constituye nuestra variable dependiente. Desde esta perspectiva, hablamos de optimización cuando mejoramos la extracción de texto, organizamos mejor nuestros documentos, recuperamos la información de forma más eficiente y disminuimos el ruido que típicamente aparece en documentos que han sido escaneados.

$$Y = \text{Optimización de la gestión documental}$$

Cabe indicar que esta variable fue evaluada mediante métricas objetivas relacionadas con la precisión OCR, fidelidad textual, tiempos de procesamiento y calidad estructural de los documentos procesados. De acuerdo con lo anterior, la investigación busca demostrar que la implementación de inteligencia artificial generativa y OCR estructurado contribuye significativamente a mejorar la eficiencia de los procesos documentales.

## 3.4 Materiales, Equipos y Recursos Tecnológicos

### 3.4.1 Materiales

Los materiales utilizados en la investigación corresponden principalmente a documentos PDF escaneados utilizados como dataset experimental. Cabe indicar que el dataset documental estuvo conformado por un total de 25 documentos PDF administrativos y legales escaneados,

pertenecientes a diferentes tipos documentales, los cuales se detallan en la Tabla III, entre estos se incluyen facturas, resoluciones, memorandos, oficios, solicitudes, certificados, formularios, contratos y ordenanzas. Dichos documentos presentan características complejas tales como ruido visual, sellos institucionales, firmas, tablas, encabezados y pies de página, lo cual permitió evaluar el comportamiento del sistema en escenarios documentales reales.

**Tabla III**.
Tipos documentales y sus características

| **Tipo documental** | **Descripción** | **Características principales** |
|---|---|---|
| Facturas | Documentos tributarios | Tablas, códigos, valores monetarios |
| Resoluciones | Documentos administrativos | Encabezados oficiales y firmas |
| Memorandos | Comunicación institucional | Texto estructurado y referencias |
| Oficios | Comunicaciones formales | Sellos y formatos institucionales |
| Solicitudes | Formularios administrativos | Datos variables y campos escritos |
| Certificados | Documentos de validación | Firmas y datos personales |
| Formularios | Registros administrativos | Estructura tabular |
| Ordenanzas | Normativas institucionales | Texto extenso y artículos |
| Contratos | Acuerdos legales y administrativos | Cláusulas, firmas y datos jurídicos |

### 3.4.2 Equipos

El sistema fue desarrollado y probado en un entorno de hardware, en la Tabla IV se menciona las características que se necesito para ejecutar procesos de extracción de texto, análisis semántico y evaluación automática de documentos.

**Tabla IV**.
Recursos de hardware utilizados en el desarrollo del sistema

| Categoría | Elemento | Descripción |
|---|---|---|
| Hardware | Intel Core i5-10400F @ 2.90 GHz, Memoria RAM: 8 GB | Procesador y memoria utilizados para el desarrollo y ejecución del sistema, permitiendo un rendimiento adecuado en tareas de procesamiento y análisis de datos. |
| Hardware | Unidad de Procesamiento Gráfico (GPU) - NVIDIA GT 710 con mínimo 2 GB VRAM | Permite acelerar el procesamiento de documentos, especialmente en tareas relacionadas con inteligencia artificial y análisis de datos. |
| Hardware | Almacenamiento SSD de 2.00 TB | Facilita el acceso rápido a los documentos y mejora la velocidad de lectura y escritura del sistema. |

### 3.4.3 Software utilizado

Las herramientas y tecnológicos utilizados en la presente investigación fueron seleccionados considerando criterios relacionados con rendimiento, compatibilidad, escalabilidad y capacidad de integración con tecnologías de inteligencia artificial generativa y procesamiento OCR estructurado. En este sentido, la arquitectura del sistema fue diseñada bajo un enfoque cliente-servidor desacoplado, permitiendo separar las responsabilidades relacionadas con visualización documental, procesamiento OCR, análisis inteligente y almacenamiento de información.

**Tabla V**.
Herramientas y tecnologías utilizadas en el desarrollo del sistema

| Componente | Herramienta | Versión | Función |
|---|---|---|---|
| Frontend | Next.js | 16.1.6 | Desarrollo de la interfaz web para carga, búsqueda y visualización documental |
| Backend | FastAPI | 0.135.1 | Implementación de servicios REST y procesamiento documental |
| Base de datos | PostgreSQL | 18.2 | Almacenamiento estructurado de documentos y resultados OCR |

*Continúa en la siguiente página*

| Componente | Herramienta | Versión | Función |
|---|---|---|---|
| ORM | SQLAlchemy | 2.0.48 | Gestión de modelos y persistencia de datos |
| OCR estructurado | mistral-ocr-latest | 1.12.4 | Extracción textual estructurada desde documentos PDF |
| Inteligencia artificial generativa | gemini-2.5-flash | 1.66.0 | Generación de resúmenes y clasificación documental |
| Renderizado de páginas PDF | PyMuPDF | 1.27.2.2 | Conversión de páginas PDF a imágenes para reprocesamiento OCR |
| Contenedorización | Docker | 28.0.4 | Aislamiento y despliegue reproducible del entorno de base de datos |
| Orquestación de servicios | Docker Compose | 2.34.0 | Gestión del entorno local y contenedores |
| Lenguaje de programación | Python | 3.13.2 | Desarrollo de lógica backend y scripts experimentales |
| Almacenamiento documental | Storage local | N/A | Conservación de documentos PDF originales e imágenes procesadas |
| Evaluación experimental | Scripts Python | N/A | Ejecución de métricas CER, WER y benchmarks |
| Documentación API | Swagger-/OpenAPI | Integrado en FastAPI | Visualización y prueba de endpoints REST |

### 3.4.4 Justificación del uso de recursos tecnológicos

La selección de FastAPI como framework backend respondió a su capacidad para desarrollar APIs de alto rendimiento basadas en programación asíncrona, permitiendo optimizar el procesamiento concurrente de documentos PDF. Asimismo, PostgreSQL fue utilizado debido a su robustez y capacidad para gestionar información estructurada relacionada con documentos, estados de procesamiento y resultados OCR.

Cuando se trataba de decidir qué herramienta utilizar, Mistral OCR se posicionó como la

alternativa ideal, principalmente por su competencia en el reconocimiento óptico de caracteres de manera estructurada y su capacidad para mantener las tablas sin alteraciones, factores fundamentales para asegurar que los documentos escaneados conservaran tanto su estructura como su formato inicial.

Gemini, por su lado, se encargó de reforzar todo el proceso brindando capacidades sofisticadas para analizar el significado de los textos, reconocer de manera automática qué clase de archivo se trataba y resumir la información utilizando tecnología de IA generativa. Vale la pena remarcar que Docker y Docker Compose resultaron herramientas clave para trabajar con contenedores y administrar el espacio de desarrollo, garantizando así que cada servicio operara de forma reproducible y en completo aislamiento.

## 3.5 Documento de Requerimientos del Proyecto

El documento de requerimientos constituye una etapa fundamental dentro del desarrollo del sistema, debido a que permite identificar, analizar y definir las necesidades funcionales y no funcionales necesarias para el correcto funcionamiento de la solución tecnológica propuesta. En este sentido, el levantamiento de requerimientos permitió establecer las características técnicas, operativas y estructurales del sistema inteligente orientado a la extracción y análisis automatizado de documentos PDF escaneados.

Vale la pena señalar que durante la etapa de establecimiento de requerimientos, se consideraron cuidadosamente las dificultades que atraviesan los métodos tradicionales de administración documental, especialmente las relacionadas con el deterioro de la organización en documentos digitalizados, los obstáculos para encontrar datos concretos, la aparición de ruido visual y las limitaciones propias de los OCR convencionales. Así pues, los requisitos que se fijaron funcionaron a modo de guía direccional para encaminar el desarrollo de una arquitectura mixta que integra tecnología OCR convencional con las posibilidades que ofrece la inteligencia artificial generativa.

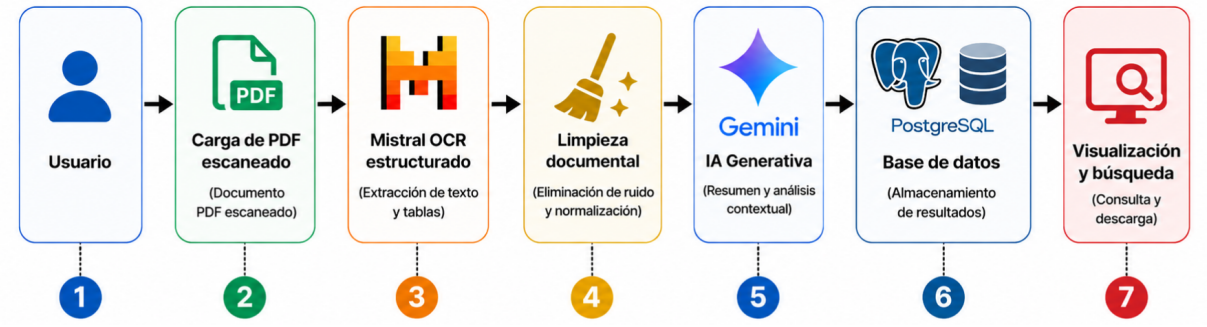


**Figura 17**. Estructura general del sistema inteligente de gestión documental.

Para ilustrar cómo opera en términos generales el sistema inteligente de gestión documental, la Figura 17 muestra el flujo principal que sigue el procesamiento de documentos en esta investigación. De esta manera, el diagrama facilita la visualización de cada una de las etapas involucradas: desde la carga de documentos, pasando por la extracción OCR estructurada, la limpieza contextual, el análisis con inteligencia artificial generativa, hasta finalmente el almacenamiento de los resultados obtenidos.

### 3.5.1 Requerimientos funcionales

**Tabla VI**.
Requisitos Funcionales

| Código | Requisito funcional | Descripción |
|---|---|---|
| RF-01 | Carga de documentos PDF escaneados | El sistema debe permitir al usuario cargar documentos PDF escaneados desde la interfaz web para su posterior procesamiento documental. |
| RF-02 | Validación de archivo | El sistema debe verificar que el archivo cargado corresponda al formato PDF antes de iniciar el procesamiento. |
| RF-03 | Extracción OCR estructurada | El sistema debe ejecutar procesos de OCR estructurado utilizando Mistral OCR para extraer texto, tablas y estructura documental. |

*Continúa en la siguiente página*

| Código | Requisito funcional | Descripción |
|---|---|---|
| RF-04 | Recuperación de tablas | El sistema debe detectar y preservar tablas estructuradas presentes en los documentos PDF procesados. |
| RF-05 | Limpieza documental inteligente | El sistema debe eliminar ruido visual y preservar contenido documental relevante mediante limpieza conservadora. |
| RF-06 | Reprocesamiento OCR por imagen | El sistema debe reprocesar páginas específicas como imágenes cuando se detecten tablas incompletas o pérdida de información. |
| RF-07 | Generar automáticamente resúmenes | El sistema se encargará de generar resúmenes de forma automática a través de IA generativa, por lo tanto usa Gemini. |
| RF-08 | Clasificación documental | La clasificación estará a cargo del sistema de manera automática y del tipo de documento que se está procesando. |
| RF-09 | Visualización de la relación página-texto | La presentación de la imagen de la página original del PDF y su equivalente en texto extraído estará a cargo del sistema. |
| RF-10 | Búsqueda documental | Los usuarios podrán buscar en los documentos mediante palabras clave y filtros de texto. Esta actividad será validada por el sistema |
| RF-11 | Filtrado por fechas | La filtración de intervalos y fechas estará a cargo del sistema. |
| RF-12 | Descarga de resultados | La descarga de los resultados de búsqueda procesados en formato TXT, MD y DOCX estará habilitada por el sistema. |
| RF-13 | Almacenamiento documental | El almacenamiento de los documentos procesados y los resultados del OCR en PostgreSQL y en el almacenamiento local, serán una acción del sistema. |
| RF-14 | Gestión de estados de procesamiento | La gestión de los estados de los documentos como pendiente, en trámite, completado y fallido serán procesados y presentados por el sistema. |

*Continúa en la siguiente página*

| Código | Requisito funcional | Descripción |
|---|---|---|
| RF-15 | Evaluación experimental | El sistema debe ejecutar métricas experimentales CER y WER para evaluar precisión OCR. |

### 3.5.2 Requerimientos no funcionales

**Tabla VII**.
Requisitos No Funcionales

| Código | Requisito no funcional | Descripción |
|---|---|---|
| RNF-01 | Seguridad documental | El sistema debe proteger credenciales y validar documentos PDF antes de su procesamiento. |
| RNF-02 | Escalabilidad | El sistema debe permitir futuras ampliaciones modulares y procesamiento de nuevos tipos documentales. |
| RNF-03 | Disponibilidad | El sistema debe garantizar estabilidad operativa durante el procesamiento documental. |
| RNF-04 | Rendimiento | El sistema debe mantener tiempos de respuesta adecuados durante la extracción OCR y análisis documental. |
| RNF-05 | Compatibilidad | El sistema debe funcionar correctamente en entornos modernos compatibles con tecnologías web. |

## 3.6 Arquitectura del Sistema

La arquitectura del sistema fue diseñada bajo un enfoque cliente-servidor, con separación de responsabilidades entre la interfaz de usuario, la lógica de procesamiento, el almacenamiento de información y los servicios externos de inteligencia artificial. En este sentido, la solución propuesta integra tecnologías modernas para el desarrollo de aplicaciones web, procesamiento OCR estructurado, análisis mediante inteligencia artificial generativa y gestión de información documental.

Cabe indicar que el sistema está compuesto por un frontend desarrollado en Next.js, un backend implementado en FastAPI, una base de datos relacional PostgreSQL, almacenamiento local para documentos PDF y servicios de inteligencia artificial como Mistral OCR y Gemini. De acuerdo con lo anterior, esta arquitectura permite procesar documentos PDF escaneados, extraer información textual, limpiar ruido documental, generar resúmenes automáticos, clasificar documentos y visualizar la correspondencia entre cada página del PDF y su texto extraído, en la Figura 18 se visualiza la arquitectura general del sistema propuesto.

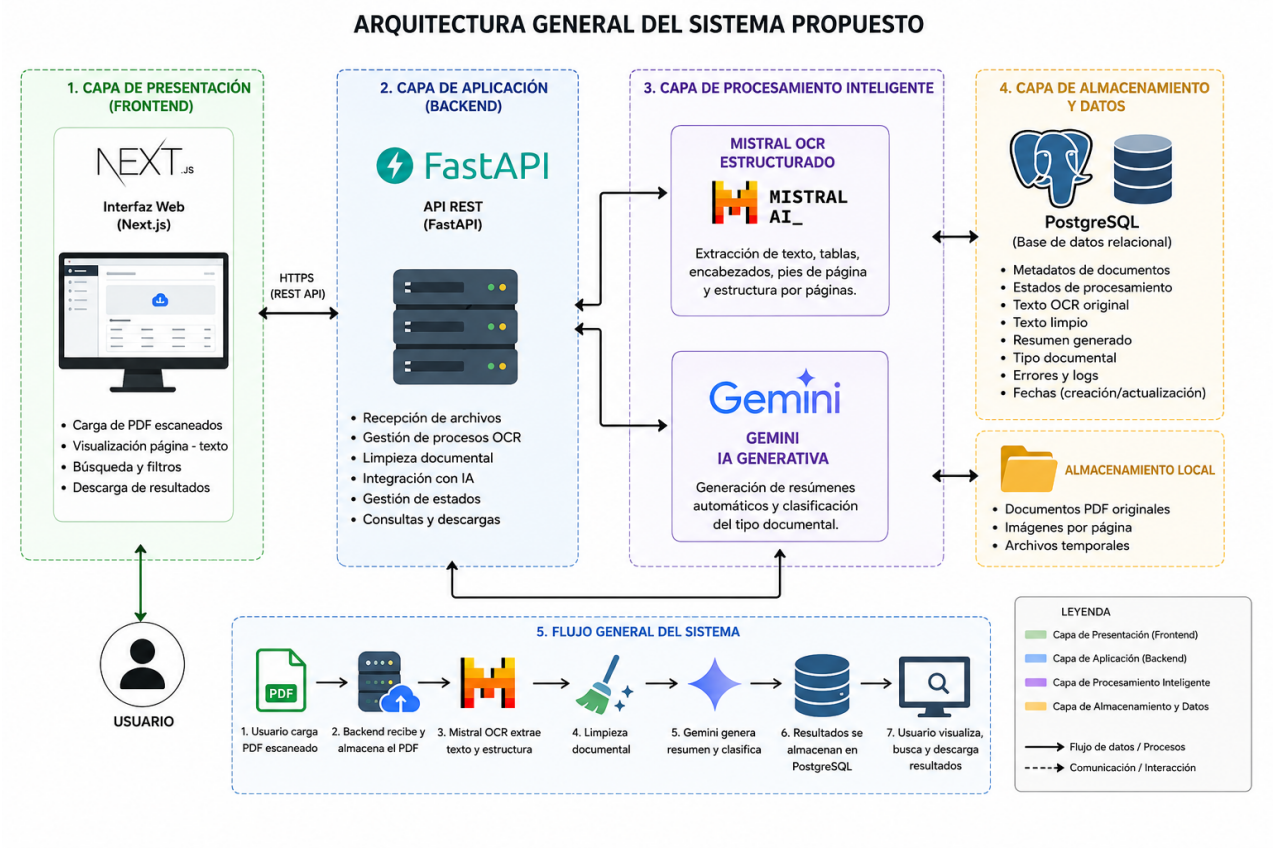


**Figura 18**. Arquitectura general del sistema propuesto.

### 3.6.1 Componentes principales de la arquitectura

- **Frontend.-** fue desarrollado con Next.js y constituye el punto de interacción entre el usuario y el sistema. Desde esta interfaz, el usuario puede cargar documentos PDF escaneados, consultar documentos procesados, realizar búsquedas, aplicar filtros por fechas, visualizar el PDF original por página y descargar los resultados en diferentes formatos.

- **Backend.-** implementado con FastAPI, cumple la función de coordinar todo el flujo de procesamiento documental. Este componente recibe los archivos PDF, los almacena localmente, registra sus metadatos en PostgreSQL, ejecuta el proceso OCR estructurado, aplica limpieza documental, solicita el análisis mediante inteligencia artificial generativa y expone los endpoints necesarios para la consulta y descarga de información.
- **Base de datos PostgreSQL.-** permite almacenar la información estructurada del sistema. En ella se registran datos como el nombre del archivo, tipo de contenido, ruta de almacenamiento, estado del procesamiento, texto OCR original, texto limpio, resumen, tipo documental, errores de procesamiento y fechas de creación o actualización.
- **Mistral OCR estructurado.-** se trata de una herramienta diseñada para extraer y recuperar el contenido textual de archivos PDF provenientes de documentos escaneados. Cabe destacar que esta solución permite preservar los elementos estructurales más relevantes del documento, tales como tablas, encabezados, referencias de pie de página y la distribución del contenido en el conjunto de páginas.
- **Gemini IA Generativa.-** con esta herramienta es posible llevar a cabo un análisis contextual profundo del texto una vez que ha sido limpiado y estructurado adecuadamente. De esta manera, facilita la creación de resúmenes de forma automática y permite identificar la categoría documental correspondiente, lo que redunda en una mejor organización y acceso a la información disponible.

### 3.6.2 Arquitectura cliente-servidor

La arquitectura cliente-servidor permite distribuir las responsabilidades del sistema entre la interfaz de usuario y el servidor de procesamiento. En este sentido, el cliente, desarrollado en Next.js, se encarga de presentar la información de manera visual e interactiva, mientras que el servidor, implementado en FastAPI, ejecuta las operaciones de procesamiento, análisis y almacenamiento documental.

Cabe indicar que la comunicación entre frontend y backend se realiza mediante endpoints

REST, lo cual facilita la integración entre componentes y permite mantener una estructura modular. Por consiguiente, esta arquitectura favorece el mantenimiento del sistema, la escalabilidad futura y la incorporación de nuevos módulos, como búsqueda semántica, chatbot documental o procesamiento por lotes.

### 3.6.3 Flujo general de procesamiento documental

Cuando un usuario sube un PDF escaneado a través de la plataforma web, comienza un flujo de procesamiento que funciona de la siguiente manera. El frontend transmite ese archivo hacia el backend, donde se guarda la copia original del documento y se genera un primer registro en la base de datos marcado como pendiente. Seguidamente, el sistema backend pone en marcha el reconocimiento óptico de caracteres usando Mistral OCR, consiguiendo de esta forma extraer el texto, las tablas y toda la información estructurada página por página.

De acuerdo con lo anterior, el sistema aplica una limpieza documental conservadora con el propósito de eliminar ruido visual sin afectar el contenido relevante del documento. Posteriormente, el texto limpio es enviado a Gemini para generar un resumen automático y determinar el tipo documental. Finalmente, los resultados son almacenados en PostgreSQL y quedan disponibles para su visualización, búsqueda, filtrado y descarga, en la Figura 19 se muestra el flujo general.

## 3.7 Organización Modular del Sistema

La organización modular del sistema fue estructurada considerando las distintas fases del modelo en cascada implementado durante el desarrollo de la investigación. En este sentido, la estructura de directorios y componentes del proyecto responde directamente a las necesidades funcionales, arquitectónicas y experimentales definidas durante las etapas de análisis, diseño, implementación y evaluación.

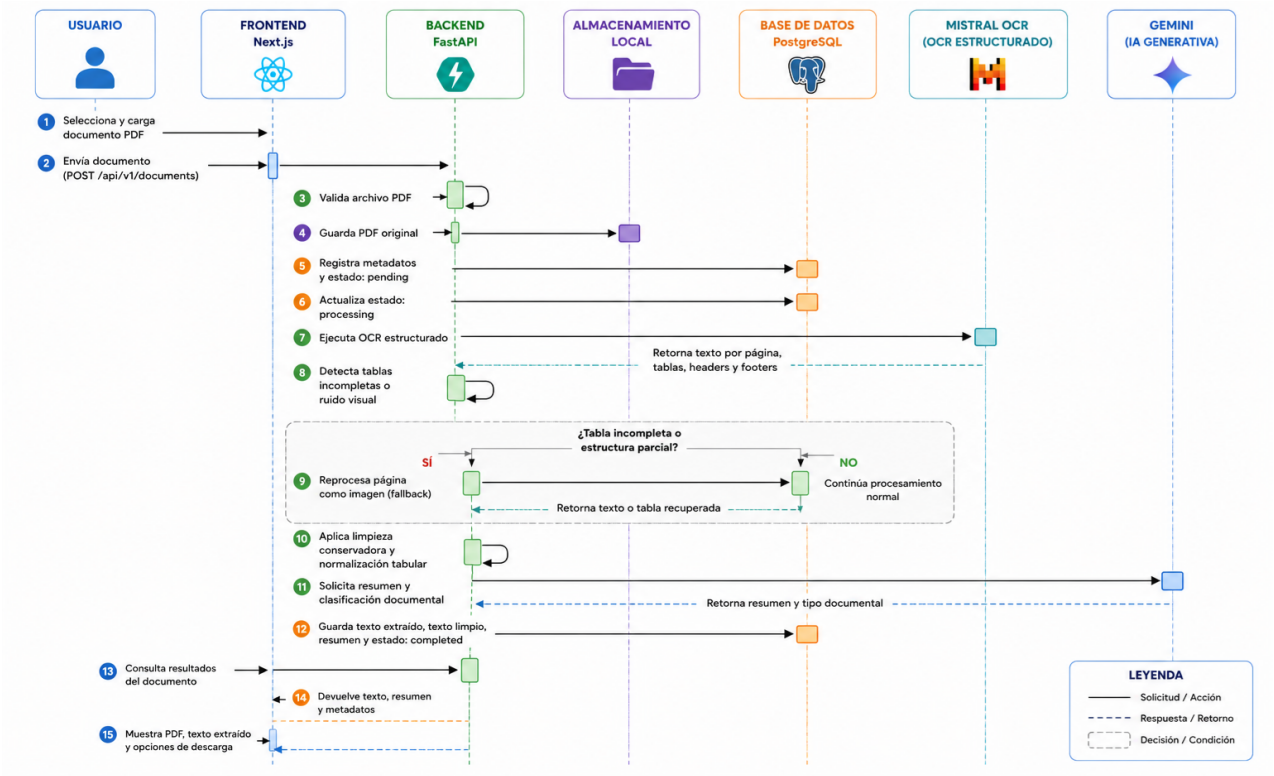


**Figura 19**. Flujo general de procesamiento documental del sistema.

### 3.7.1 Fase de análisis

La fase de análisis permitió definir las necesidades funcionales, estructurales y arquitectónicas del sistema inteligente de gestión documental. En este sentido, durante esta etapa se identificaron las problemáticas relacionadas con extracción textual deficiente, pérdida de estructura documental, recuperación incompleta de tablas y presencia de ruido visual en documentos PDF escaneados.

Cabe indicar que esta etapa permitió establecer los requerimientos funcionales relacionados con carga de documentos PDF, OCR estructurado, visualización documental por página, búsqueda textual, filtrado por fechas, descarga de resultados y generación automática de resúmenes y clasificación documental. Entre los principales elementos documentales definidos durante esta etapa se encuentran:

**Tabla VIII**.
Estructura inicial de archivos y directorios del proyecto

| **Directorio / Archivo** | **Función** |
|---|---|
| `docs/` | Contiene documentación técnica y descriptiva del proyecto |
| `architecture.md` | Define la arquitectura general del sistema |
| `requirements/` | Contiene requerimientos funcionales y no funcionales |
| `docker-compose.yml` | Define servicios e infraestructura inicial |
| `README.md` | Describe configuración y ejecución del proyecto |

### 3.7.2 Fase de Diseño

La fase de diseño permitió definir la arquitectura cliente-servidor y la distribución modular de responsabilidades dentro del sistema. En este sentido, durante esta etapa se diseñó la interacción entre frontend, backend, base de datos, OCR estructurado, almacenamiento documental y servicios de inteligencia artificial generativa. En la Tabla IX, se menciona los componentes estructurales:

**Tabla IX**.
Estructura de módulos durante la fase de diseño

| **Directorio / Módulo** | **Función dentro de la fase de diseño** |
|---|---|
| `frontend/` | Define la estructura visual del sistema y la interacción con el usuario |
| `backend/app/api/` | Diseña endpoints REST y flujo de comunicación |
| `backend/app/models/` | Define entidades y estructura de base de datos |
| `backend/app/schemas/` | Establece esquemas de entrada y salida para la API |
| `backend/app/core/` | Centraliza configuración general y variables de entorno |
| `backend/app/db/` | Diseña conexión y persistencia con PostgreSQL |
| `storage/documents/` | Define almacenamiento documental persistente |

Asimismo, durante esta etapa se diseñó el pipeline híbrido de procesamiento documental, el cual integra OCR estructurado, fallback OCR por imagen, limpieza conservadora, normalización de tablas y análisis mediante inteligencia artificial generativa.

### 3.7.3 Fase de Implementación

La fase de implementación correspondió al desarrollo de los módulos funcionales definidos durante la etapa de diseño, es decir, el sistema fue implementado mediante componentes especializados encargados de ejecutar procesamiento documental automatizado, OCR estructurado, almacenamiento de información y análisis contextual mediante inteligencia artificial generativa. En la Tabla X se describe los principales módulos implementados fueron los siguientes:

**Tabla X**.
Servicios implementados durante la fase de implementación

| **Directorio / Servicio** | **Función dentro de la fase de implementación** |
|---|---|
| `backend/app/services/document_service.py` | Gestiona subida, registro y procesamiento documental |
| `backend/app/services/ocr_service.py` | Ejecuta OCR estructurado y fallback OCR por imagen |
| `backend/app/services/text_service.py` | Aplica limpieza conservadora y normalización tabular |
| `backend/app/services/ai_service.py` | Genera resumen y clasificación documental mediante Gemini |
| `backend/app/services/pdf_page_service.py` | Renderiza páginas PDF para visualización |
| `backend/app/services/report_service.py` | Genera descargas en formatos MD, TXT, DOCX y PDF |
| `backend/app/services/evaluation_service.py` | Ejecuta evaluaciones WER/CER individuales |

Por otra parte, el frontend fue implementado mediante componentes reutilizables organiza-

dos, en la Tabla XI se menciona los directorios específicos, ya que es la parte del desarrollo web que se ocupa de la interfaz de usuario, es decir es de todo lo que se ve y con lo que interactúan en una página web:

**Tabla XI**.
Estructura del frontend del sistema

| **Directorio** | **Función** |
|---|---|
| `frontend/app/` | Contiene páginas y rutas principales |
| `frontend/components/` | Implementa componentes reutilizables de interfaz |
| `frontend/lib/` | Centraliza funciones auxiliares y comunicación API |

### 3.7.4 Fase de Verificación y Evaluación

La fase de verificación y evaluación permitió validar el comportamiento técnico y experimental del sistema mediante pruebas funcionales, métricas objetivas y benchmarks comparativos, esta etapa fue organizada mediante scripts independientes ejecutados por consola, evitando afectar el funcionamiento normal del frontend y backend durante las pruebas experimentales. En este sentido, se implementaron módulos especializados orientados a medir fidelidad textual, rendimiento OCR y calidad estructural del procesamiento documental, en la Tabla XII se detalla los principales componentes.

**Tabla XII**.
Scripts y directorios utilizados durante la fase de evaluación

| **Directorio / Script** | **Función dentro de la fase de evaluación** |
|---|---|
| `backend/scripts/batch_wer_cer.py` | Ejecuta evaluaciones masivas CER/WER |
| `backend/scripts/evaluate_document.py` | Evalúa documentos individuales |

*Continúa en la siguiente página*

| Directorio / Script | Función dentro de la fase de evaluación |
|---|---|
| backend/scripts/benchmark_baseline_vs_structured.py | Compara baseline vs método propuesto |
| backend/scripts/benchmark_hierarchical_summary.py | Evalúa resumen jerárquico |
| backend/scripts/benchmark_processing.py | Mide tiempos de procesamiento |
| backend/scripts/benchmark_answer_time_by_type.py | Evalúa tiempos de respuesta por tipo documental |
| backend/evaluation_reports/ | Almacena reportes CSV y resultados experimentales |

## 3.8 Estructura del Software

La organización estructural del Software fue definida bajo un enfoque modular orientado a desacoplar las diferentes capas funcionales del sistema, ya que permite separar adecuadamente las responsabilidades relacionadas con interfaz gráfica, procesamiento OCR, inteligencia artificial generativa, almacenamiento documental y evaluación experimental, en la Tabla XIII se describen los directorios y módulos más relevantes implementados dentro de la arquitectura del software desarrollado.

**Tabla XIII**.
Estructura general de directorios y componentes del sistema

| Directorio / Componente | Propósito | Módulos clave |
|---|---|---|
| app/api/ | Gestiona rutas y endpoints REST del sistema. | routes.py |
| app/core/ | Administra configuración global y variables de entorno. | config.py |

*Continúa en la siguiente página*

| Directorio / Componente | Propósito | Módulos clave |
|---|---|---|
| app/db/ | Gestiona conexión y sesiones de base de datos. | session.py, base.py |
| app/models/ | Define modelos y estructuras relacionales. | document.py |
| app/schemas/ | Valida y serializa datos de la API. | Schemas Pydantic |
| app/services/ | Implementa la lógica principal del sistema. | ocr_service.py, text_service.py, ai_service.py, document_service.py, report_service.py, pdf_page_service.py, evaluation_service.py, openai_transform_service.py, security_service.py, rate_limit_service.py, audit_service.py |
| backend/scripts/ | Ejecuta evaluaciones y benchmarks experimentales. | batch_wer_cer.py, evaluate_document.py, g_eval_openai.py, benchmark_answer_time_by_type.py, benchmark_baseline_vs_structured.py, benchmark_hierarchical_summary.py |
| backend/evaluation_reports/ | Almacena reportes y resultados experimentales. | Reportes CSV |
| frontend/app/ | Contiene la estructura principal del frontend. | Layouts, páginas y estilos |
| references/ | Contiene documentos de referencia OCR. | Ground truth |

*Continúa en la siguiente página*

| Directorio / Componente | Propósito | Módulos clave |
|---|---|---|
| `storage/documents/` | Almacena documentos y recursos procesados. | PDFs, imágenes y auditoría |
| `docker-compose.yml` | Configura servicios Docker del entorno local. | PostgreSQL |

Una vez conocido cada uno de los directorios y módulos que contiene el sistema, en la Figura 20 se muestra como está diseñado internamente la estructura del software, pues cada modulo tiene su propio color que sirve como evidencia de los módulos que están dentro de cada directorio.

## 3.9 Seguridad del Sistema

La seguridad del sistema constituye un componente fundamental dentro de la arquitectura propuesta, debido a que el procesamiento documental involucra almacenamiento, manipulación y análisis de documentos administrativos y legales en formato PDF. En este sentido, fue necesario implementar mecanismos orientados a reducir riesgos relacionados con procesamiento de archivos maliciosos, exposición de credenciales, uso no autorizado de recursos y acceso indebido a información documental.

### 3.9.1 Validación de Archivos PDF

Uno de los principales mecanismos de seguridad implementados dentro del sistema corresponde a la validación de archivos PDF antes de iniciar cualquier proceso de almacenamiento o extracción OCR. En este sentido, el backend desarrollado en FastAPI verifica múltiples características del documento con el propósito de evitar el procesamiento de archivos inválidos o potencialmente maliciosos.

Vale la pena señalar que el sistema cuenta con una validación preliminar que se concen-

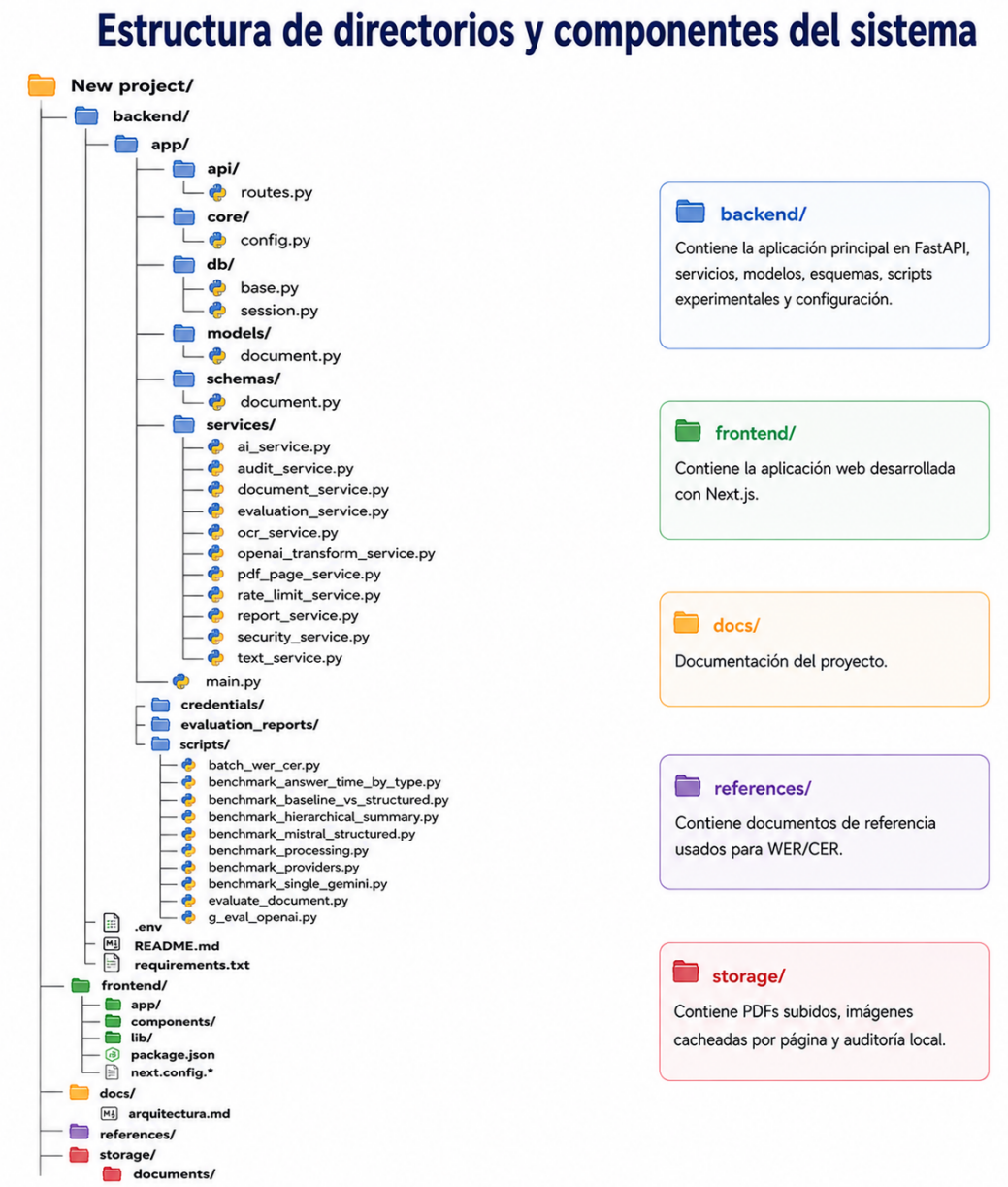


**Figura 20**. Estructura de directorios del proyecto.

tra en revisar la extensión del archivo, lo que admite exclusivamente documentos con formato .pdf. Posteriormente, se verifica el tipo MIME que llega cuando se sube el archivo, lo que permite reconocer solamente aquellos formatos vinculados a documentos PDF, tales como `application/pdf` y `application/x-pdf`.

El sistema lleva a cabo verificaciones adicionales sobre la estructura binaria del documento, lo que facilita controlar que contenga la firma %PDF y valida que se trata de un fichero PDF auténtico. De la misma forma, el servidor utiliza PyMuPDF para intentar abrir el archivo y

garantizar que posee como mínimo una página susceptible de ser tratada.

Por otra parte, la plataforma cuenta con un tope máximo ajustable respecto al volumen de ficheros que se suben, gestionado a través de parámetros ambientales. Con esto en mente, se fijó un umbral de partida de 25 MB utilizando el parámetro MAX_UPLOAD_SIZE_MB, lo que previene un gasto desmesurado de recursos mientras se realizan tareas de subida y tratamiento de documentos.

### 3.9.2 Protección de Credenciales y Configuración Sensible

La protección de credenciales constituye un aspecto crítico dentro de la arquitectura del sistema, debido a que el procesamiento documental depende de servicios externos de inteligencia artificial y almacenamiento de datos. En este sentido, todas las claves API y configuraciones sensibles fueron desacopladas del código fuente principal mediante el uso de variables de entorno.

Cabe mencionar que las credenciales correspondientes a Mistral OCR, Gemini, OpenAI y PostgreSQL son gestionadas utilizando archivos .env y cargadas dinámicamente mediante la librería pydanticsettings. Por consiguiente, esta estrategia evita la exposición directa de información sensible dentro del repositorio del proyecto o del código fuente del sistema. De acuerdo con lo anterior, las variables consideradas sensibles dentro del entorno de desarrollo incluyen:

- `MISTRAL_API_KEY`
- `GEMINI_API_KEY`
- `OPENAI_API_KEY`
- `DATABASE_URL`

### 3.9.3 Seguridad del Almacenamiento Documental

El sistema incorpora mecanismos orientados a proteger los documentos almacenados localmente dentro del entorno de procesamiento documental. En este sentido, los archivos PDF cargados por el usuario no se almacenan utilizando su nombre original, sino mediante identificadores internos basados en UUID. Ademas las descargas y previsualizaciones de documentos originales son gestionadas mediante endpoints controlados por el backend, evitando acceso directo a rutas internas del almacenamiento local. De acuerdo con lo anterior, esta arquitectura mejora el control sobre acceso documental y permite incorporar futuras políticas de autenticación y autorización.

### 3.9.4 Seguridad de la API y Control de Acceso

La API desarrollada mediante FastAPI incorpora mecanismos básicos de seguridad orientados a proteger el procesamiento documental y reducir riesgos asociados a uso masivo de recursos. En este sentido, el sistema implementa validaciones robustas antes de iniciar cualquier proceso OCR o generación de contenido mediante inteligencia artificial.

El backend incorpora limitación básica de solicitudes mediante rate limiting en memoria para operaciones relacionadas con carga documental, descargas y previsualizaciones. Esta estrategia permite controlar temporalmente la cantidad de solicitudes realizadas por dirección IP dentro de intervalos específicos de tiempo, como tambien utiliza variables configurables para definir límites máximos de solicitudes por minuto. En este sentido, se establecieron controles diferenciados para cargas y descargas documentales mediante las variables `UPLOAD_RATE_LIMIT_PER_MINUTE` y `DOWNLOAD_RATE_LIMIT_PER_MINUTE`.

### 3.9.5 Auditoría de Eventos de Seguridad

El sistema incorpora un mecanismo básico de auditoría orientado al registro de eventos relacionados con acciones críticas ejecutadas durante el procesamiento documental. En este sentido, las operaciones relevantes son registradas en archivos de auditoría en formato JSONL dentro del almacenamiento local del sistema, es decir que los eventos registrados incluyen cargas de documentos, descargas de reportes, descargas de PDFs originales y solicitudes de previsualización documental. Por consiguiente, este mecanismo permite mantener trazabilidad sobre acciones ejecutadas dentro del sistema, los registros de auditoría son almacenados en la ruta:

`storage/documents/_audit/audit.log`

## 3.10 Sistema de Evaluación Experimental

El sistema incorpora una capa experimental de evaluación orientada a medir objetivamente la calidad del procesamiento documental, la fidelidad textual del OCR y el rendimiento general implementado. En este sentido, la evaluación experimental fue diseñada como un componente independiente del flujo principal del sistema, ejecutándose mediante scripts especializados por consola con el propósito de evitar afectar el funcionamiento normal de la aplicación web, usa métricas cuantitativas tradicionales como Character Error Rate (CER) y Word Error Rate (WER), complementadas mediante evaluaciones cualitativas basadas en Large Language Models utilizando el enfoque G-Eval.

### 3.10.1 Evaluación Objetiva mediante CER y WER

La evaluación objetiva del sistema fue realizada utilizando las métricas Character Error Rate (CER) y Word Error Rate (WER), ampliamente utilizadas dentro del área de reconocimiento óptico de caracteres para medir fidelidad textual. En este sentido, CER permite evaluar errores a nivel de caracteres, mientras que WER mide errores relacionados con palabras completas.

Ambas métricas requieren un texto de referencia correcto o ground truth con el propósito de comparar el resultado OCR generado por el sistema frente al contenido original esperado, de las cuales se usaron 25 documentos para realizar la evaluación de las dos métricas.

### 3.10.2 Evaluación mediante Character Error Rate (CER)

La métrica Character Error Rate (CER) en español Tasa de Error de Caracteres, fue utilizada para medir el nivel de error textual a nivel de caracteres entre el texto extraído por el sistema OCR y un texto de referencia o ground truth. En este sentido, CER constituye una de las métricas más utilizadas dentro del área de reconocimiento óptico de caracteres debido a su capacidad para identificar errores finos relacionados con sustituciones, eliminaciones e inserciones de caracteres. Según [146] explica que CER es la tasa de error a nivel de carácter, definida como:

$$CER = \frac{i + s + d}{n} \tag{3.3}$$

Donde ***n*** es el número total de caracteres, ***i*** las inserciones, ***s*** las sustituciones y ***d*** las eliminaciones necesarias para convertir el texto de referencia en el resultado OCR. Un valor de CER igual a cero indica coincidencia perfecta entre el texto extraído y el texto de referencia. En este sentido, mientras menor sea el valor obtenido, mayor será la precisión textual alcanzada por el sistema OCR.

### 3.10.3 Evaluación mediante Word Error Rate (WER)

La métrica Word Error Rate (WER) en español Tasa de Error de Palabras, fue utilizada para medir el nivel de error a nivel de palabras entre el texto OCR generado por el sistema y el texto de referencia correspondiente [147]. Esta métrica resulta importante debido a que la pérdida, sustitución o inserción incorrecta de palabras puede afectar significativamente la interpretación semántica de documentos administrativos y legales. La ecuación utilizada para calcular WER

corresponde a:

$$WER = \frac{s + d + i}{n} \tag{3.4}$$

Donde:

- ***s*** representa sustituciones de palabras.
- ***d*** representa eliminaciones de palabras.
- ***i*** representa inserciones de palabras.
- ***n*** corresponde al número total de palabras presentes en el texto de referencia.

Cuando el valor está próximo a cero, eso quiere decir que el documento se ha procesado con gran precisión y se ha conservado bien el significado del texto durante el OCR. Por otra parte, gracias a esta métrica fue posible detectar diversos inconvenientes como la omisión accidental de términos, la fragmentación del contenido y fallos en la segmentación del texto.

### 3.10.4 Benchmarks de Procesamiento

Con el propósito de evaluar el rendimiento y comportamiento del pipeline híbrido implementado dentro del sistema, se desarrollaron benchmarks experimentales orientados a comparar distintas variantes de procesamiento documental, donde las evaluaciones fueron ejecutadas mediante scripts independientes por consola, permitiendo realizar pruebas controladas sin afectar el funcionamiento normal del sistema.

El uso de benchmarks para realizar comparaciones, medir el rendimiento, monitorear avances e identificar debilidades se ha consolidado como una práctica común dentro del desarrollo de inteligencia artificial. Estos referentes son utilizados con frecuencia por los desarrolladores para

presentar el desempeño de sus modelos y compararlos con otros al momento de su lanzamiento [148], [149].

Un benchmark de calidad debe ser comprensible, tener claramente definido su propósito y alcance, y además resultar práctico para su aplicación, aunque hasta la fecha no se ha publicado una evaluación estructurada sobre la calidad de los benchmarks de inteligencia artificial, tanto para modelos fundacionales como para otros tipos de sistemas, ni se han realizado análisis comparativos que permitan identificar diferencias de calidad entre los benchmarks más utilizados actualmente [150].

Ademas es importante mencionar que benchmark tiene un ciclo de vida que permite ser usado para las evaluaciones que se vayan aplicar, esto se muestra en la Figura 21.

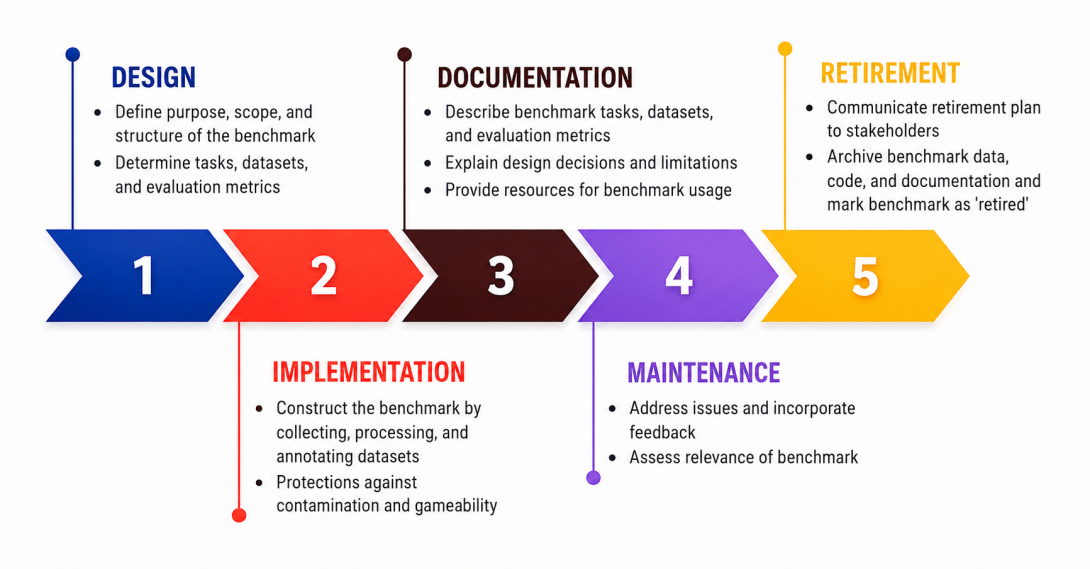


**Figura 21**. Cinco etapas del ciclo de vida de la evaluación comparativa [150].

Con lo que se muestra en la Figura 21 es necesario mencionar que realiza cada etapa, según [150], nos dice que el ciclo de vida de un benchmark comprende varias etapas: diseño, implementación, documentación, mantenimiento y retiro, en la Tabla XIV se menciona su definición de cada etapa:

**Tabla XIV**.
Etapas del ciclo de vida de un benchmark

| **Etapa** | **Descripción** |
|---|---|
| Diseño | En esta etapa se definen los objetivos del benchmark, así como las tareas, métricas y conjuntos de datos que serán utilizados para evaluar el sistema. |
| Implementación | Consiste en construir el benchmark mediante la recopilación y preparación de datos, además de desarrollar herramientas que permitan medir el rendimiento de los modelos. |
| Documentación | Se elaboran guías y descripciones detalladas para facilitar la comprensión, el acceso y el uso adecuado del benchmark. |
| Mantenimiento | Implica actualizar y supervisar continuamente el benchmark para corregir errores, incorporar mejoras y mantener su utilidad. |
| Retiro | Ocurre cuando el benchmark deja de ser relevante o surgen alternativas más eficientes, por lo que se archiva y se reemplaza gradualmente. |

Los benchmarks implementados comparan dos variantes principales las cuales son: método baseline utiliza únicamente texto OCR plano sin transformación estructural, recuperación tabular ni limpieza documental avanzada y el método propuesto implementa OCR estructurado, recuperación de tablas en formato HTML, detección de encabezados y pies de página, limpieza conservadora y normalización documental.

Entonces se puede decir que se evalúa con benchmarks como pruebas experimentales por consola que registran los tiempos de procesamiento del sistema. Estas pruebas miden el tiempo de OCR, transformación, generación de resumen y extracción de respuestas, permitiendo comparar el método baseline, que usa texto OCR plano, frente al método propuesto, que usa texto estructurado en formato tabular.

### 3.10.5 Evaluación Experimental mediante G-Eval

El sistema incorpora una evaluación experimental complementaria basada en el enfoque G-Eval utilizando modelos de lenguaje de OpenAI bajo el paradigma LLM-as-a-judge. En este sentido, esta estrategia fue utilizada para evaluar aspectos cualitativos relacionados con fidelidad estructural y utilidad documental cuando no existe un texto de referencia exacto. Es necesario mencionar que, a diferencia de CER y WER, G-Eval no requiere ground truth para realizar la evaluación. Por consiguiente, este enfoque permite analizar criterios semánticos y estructurales difíciles de medir mediante métricas tradicionales.

La calidad del contenido limpio y bien organizado (clean_text) que genera el pipeline OCR se mide a través de puntajes que oscilan entre 0 y 1, lo que procura emplear criterios similares a los que implementa DeepEval. Dicho de otra forma:

- Cuando los valores son cercanos a 0 significa un desempeño pobre y,
- Cuanto los valores son próximos a 1 describen que los documentos son de excelente calidad.

Asimismo, el sistema incorpora un umbral mínimo (threshold) establecido en 0.7 para determinar si una métrica es considerada satisfactoria durante la evaluación experimental, cabe señalar que se uso únicamente como mecanismo complementario de análisis cualitativo y no como métrica principal de validación científica, debido a la naturaleza probabilística de los modelos de lenguaje utilizados durante el proceso experimental.

### 3.10.6 Evaluación Humana

La evaluación humana fue implementada como mecanismo complementario de validación cualitativa con el propósito de analizar la calidad estructural y utilidad documental del texto extraído por el sistema inteligente de gestión documental. En este sentido, la evaluación permitió

medir aspectos relacionados con fidelidad textual, organización estructural y limpieza documental que no pueden ser evaluados completamente mediante métricas automáticas como Character Error Rate (CER) y Word Error Rate (WER).

Cabe indicar que la evaluación fue realizada mediante comparación visual entre el documento PDF original y el texto extraído generado por el pipeline híbrido basado en OCR estructurado e inteligencia artificial generativa. Por consiguiente, los evaluadores analizaron diferentes tipos documentales procesados por el sistema, incluyendo memorandos, resoluciones, formularios, certificados y facturas.

De acuerdo a [151], menciona que Likert es una herramienta de medición usada para recoger las opiniones o evaluaciones de las personas sobre un tema específico de manera cuantitativa, donde la puntuación se realiza usando una escala de Likert habitual que va del 1 al 5, donde 1 significa la peor calidad y 5 la mejor.

Se definieron cinco criterios principales de evaluación: fidelidad, completitud, orden de lectura, calidad de tablas y limpieza de ruido. Cada criterio fue evaluado mediante una escala ordinal de cinco niveles, permitiendo medir la calidad percibida del procesamiento documental realizado por el sistema. En la Tabla XV se menciona los criterios usados para obtener un resultado por cada criterio y compararlo con G-Eval.

**Tabla XV**.
Criterios utilizados en la evaluación G-Eval

| **Criterio** | **Descripción** |
|---|---|
| Fidelidad | Nivel de correspondencia entre el texto extraído y el contenido original del PDF. |
| Completitud | Grado en que el sistema recupera toda la información relevante del documento. |
| Orden de lectura | Correcta secuencia y organización del texto extraído. |
| Calidad de tablas | Preservación estructural y legibilidad de tablas recuperadas. |

**Tabla XV – Continuación**

| Criterio | Descripción |
|---|---|
| Limpieza de ruido | Capacidad del sistema para eliminar elementos irrelevantes sin afectar contenido útil. |

## 3.11 Cronograma de Desarrollo

En concordancia con la metodología en cascada adoptada para el desarrollo del Software, el cronograma del proyecto se estructura en fases secuenciales que permiten garantizar un proceso ordenado, controlado y alineado con los objetivos de la investigación. Este enfoque facilita la planificación detallada de las actividades, asegurando que cada etapa se complete antes de avanzar a la siguiente, lo cual es fundamental en proyectos que integran tecnologías complejas como la inteligencia artificial generativa.

El proyecto se estructura a través de cinco etapas bien definidas: primero viene el análisis exhaustivo de los requisitos, seguido del diseño de la arquitectura del sistema, luego la fase de codificación e implementación, después las pruebas exhaustivas y finalmente el lanzamiento en producción. En cada una de estas etapas se desarrollan actividades concretas dirigidas específicamente hacia la creación de una herramienta capaz de extraer información y analizar contenido de documentos PDF que han sido digitalizados, aprovechando para ello tecnologías avanzadas de inteligencia artificial tales como Mistral OCR y Gemini.

**Fase 1: Análisis y definición de requisitos**

**Tabla XVI**.
Cronograma de actividades - Fase 1

| Semana | Actividad | Descripción | Horas |
|---|---|---|---|
| 1 | Identificación del problema | Análisis del contexto de la gestión documental institucional y detección de limitaciones en el procesamiento de documentos PDF escaneados. | 8 |
| 1 | Revisión bibliográfica | Estudio de enfoques actuales en inteligencia artificial generativa, OCR y procesamiento de documentos. | 10 |
| 1 | Análisis de tecnologías | Evaluación de herramientas como FastAPI, PostgreSQL y modelos de IA generativa. | 8 |
| 2 | Definición de requisitos funcionales | Identificación de funcionalidades del sistema como carga, extracción, análisis y clasificación. | 10 |
| 2 | Definición de requisitos no funcionales | Definición de rendimiento, seguridad y escalabilidad. | 6 |
| 2 | Modelado de procesos | Diseño del flujo de procesamiento del sistema. | 8 |
| 2 | Elaboración de documentación | Redacción formal del documento de requisitos. | 10 |

## Fase 2: Diseño del sistema

**Tabla XVII**.
Cronograma de actividades - Fase 2

| Semana | Actividad | Descripción | Horas |
|---|---|---|---|
| 3 | Diseño de arquitectura del sistema | Diseño de la arquitectura modular frontend-backend-base de datos-servicios IA. | 14 |
| 3 | Diseño de base de datos PostgreSQL | Modelado de tablas para documentos, páginas, OCR y reportes experimentales. | 10 |
| 4 | Diseño de endpoints API | Diseño de rutas REST para procesamiento documental y recuperación de información. | 10 |

*Continúa en la siguiente página*

| Semana | Actividad | Descripción | Horas |
|---|---|---|---|
| 4 | Diseño de almacenamiento documental | Organización de directorios para PDFs, imágenes y archivos experimentales. | 8 |
| 5 | Diseño de interfaz frontend | Diseño de interfaces para carga, visualización y búsqueda documental. | 14 |
| 5 | Diseño de scripts experimentales | Planificación de benchmarks CER, WER y G-Eval. | 12 |

## Fase 3: Implementación

**Tabla XVIII**.
Cronograma de actividades - Fase 3

| Semana | Actividad | Descripción | Horas |
|---|---|---|---|
| 6 | Configuración del entorno backend | Configuración de FastAPI, entorno virtual y dependencias principales. | 8 |
| 6 | Configuración PostgreSQL y Docker | Implementación de PostgreSQL mediante Docker Compose. | 8 |
| 6 | Desarrollo de endpoints REST | Implementación de endpoints para carga, consulta y descarga documental. | 16 |
| 6 | Integración con Mistral OCR | Implementación del servicio para extracción de texto desde documentos escaneados. | 10 |
| 7 | Desarrollo de almacenamiento documental | Implementación de almacenamiento local y manejo de UUID. | 10 |
| 7 | Integración de Mistral OCR | Desarrollo del procesamiento OCR estructurado por páginas. | 18 |
| 7 | Implementación fallback OCR | Desarrollo del reprocesamiento OCR mediante imágenes con PyMuPDF. | 14 |
| 7 | Desarrollo de limpieza documental | Implementación de limpieza conservadora y normalización textual. | 12 |

*Continúa en la siguiente página*

| Semana | Actividad | Descripción | Horas |
|---|---|---|---|
| 8 | Integración de Gemini | Desarrollo de clasificación documental y generación de resúmenes. | 14 |
| 8 | Desarrollo del frontend Next.js | Implementación de interfaces para carga y visualización documental. | 18 |
| 8 | Implementación de búsqueda y filtros | Desarrollo de búsqueda textual y filtros por fechas. | 10 |
| 8 | Implementación visualización por página | Integración de imágenes PDF y texto OCR correspondiente. | 16 |
| 9 | Implementación de descargas | Desarrollo de exportación TXT, MD y PDF original. | 8 |
| 9 | Implementación de seguridad | Validación PDF, rate limiting y auditoría básica. | 12 |
| 9 | Configuración de variables de entorno | Integración de .env y configuración centralizada. | 6 |

### Fase 4: Verificación y pruebas

**Tabla XIX**.
Cronograma de actividades - Fase 4

| Semana | Actividad | Descripción | Horas |
|---|---|---|---|
| 10 | Pruebas funcionales | Verificación del correcto funcionamiento de los módulos del sistema. | 8 |
| 10 | Pruebas de integración | Evaluación de la interacción entre OCR, IA generativa y base de datos. | 8 |
| 10 | Evaluación CER | Ejecución de métricas Character Error Rate. | 10 |
| 11 | Evaluación WER | Ejecución de métricas Word Error Rate. | 10 |
| 11 | Desarrollo de benchmarks OCR | Comparación entre baseline y OCR estructurado. | 14 |

*Continúa en la siguiente página*

| Semana | Actividad | Descripción | Horas |
|---|---|---|---|
| 11 | Evaluación G-Eval | Evaluación cualitativa mediante IA juez con OpenAI. | 12 |

## Fase 5: Implementación y mantenimiento

**Tabla XX**.

Cronograma de actividades - Fase 5

| Semana | Actividad | Descripción | Horas |
|---|---|---|---|
| 12 | Optimización del pipeline OCR | Mejoras de rendimiento y estabilidad del procesamiento documental. | 12 |
| 12 | Optimización de consultas y almacenamiento | Ajustes sobre PostgreSQL y almacenamiento documental. | 10 |
| 12 | Documentación técnica del sistema | Elaboración de documentación arquitectónica y técnica. | 14 |
| 13 | Elaboración de diagramas y reportes | Generación de diagramas de arquitectura y flujo experimental. | 10 |
| 13 | Redacción de resultados experimentales | Desarrollo de análisis y discusión de resultados CER, WER y G-Eval. | 16 |
| 13 | Correcciones finales | Ajustes técnicos y validación final del sistema. | 8 |
| 13 | Elaboración de conclusiones y recomendaciones | Redacción final de tesis y cierre del proyecto. | 12 |

# CAPÍTULO IV

# RESULTADOS Y ANÁLISIS

El presente capítulo describe las pruebas realizadas al sistema desarrollado, así como los resultados obtenidos durante el proceso de validación experimental de la propuesta tecnológica, ya que se enfocó en verificar el correcto funcionamiento de los módulos implementados, como medir la calidad del procesamiento OCR estructurado y analizar el desempeño general del sistema en diferentes escenarios documentales. Las pruebas ejecutadas permitieron validar tanto aspectos funcionales como métricas técnicas relacionadas con precisión OCR, recuperación estructural, limpieza documental y rendimiento del procesamiento inteligente de documentos PDF.

## 4.1 Arquitectura Implementada

El sistema desarrollado implementa una arquitectura multicapa con separación lógica entre presentación, servicios, procesamiento y persistencia de datos. En la Figura 22 se muestra la capa de presentación que corresponde al frontend desarrollado en Next.js, encargado de la interacción con el usuario. La capa API implementada en FastAPI administra las solicitudes del sistema y coordina los procesos internos. La capa de servicios contiene la lógica de procesamiento OCR, limpieza documental, análisis mediante IA y generación de reportes. Finalmente, la capa de persistencia administra el almacenamiento estructurado de información mediante PostgreSQL.

### 4.1.1 Componentes Principales

Los componentes principales que contiene el sistema son:

- **Frontend web:** Desarrollado con Next.js. Permite al usuario subir documentos PDF, vi-

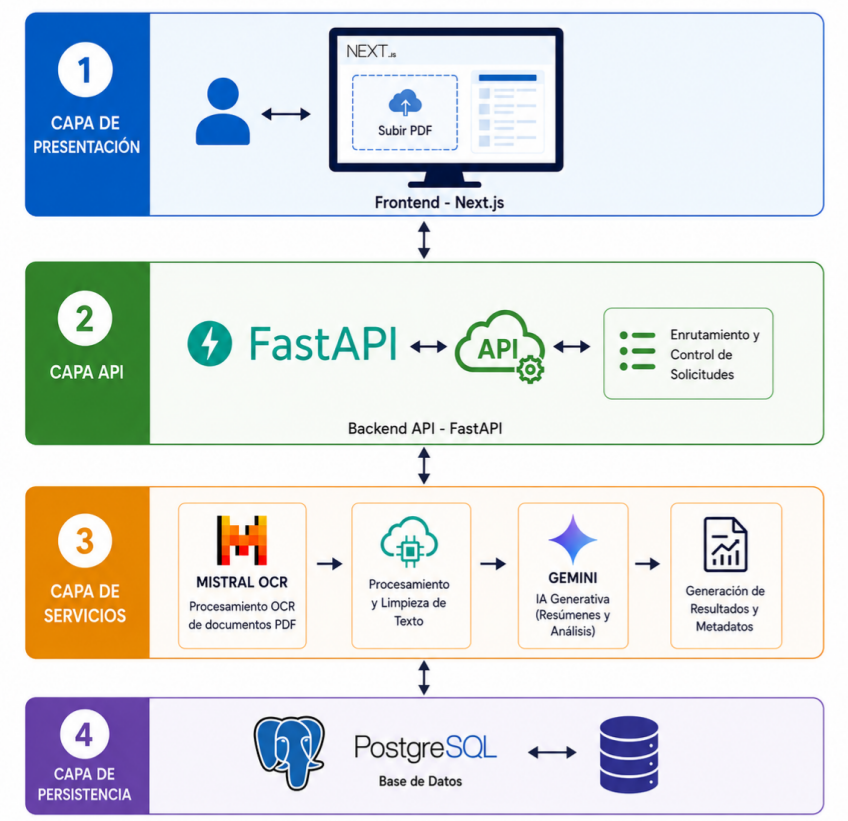


**Figura 22**. Arquitectura implementada del sistema.

sualizar el PDF original, revisar el texto extraído, buscar documentos, filtrar por fechas y descargar resultados.

- **Backend API:** Desarrollado con FastAPI. Recibe los documentos, coordina el procesamiento, expone endpoints, gestiona búsquedas, descargas, visualización y comunicación con los servicios de OCR e IA.
- **Base de datos:** PostgreSQL + SQLAlchemy. Almacena metadatos del documento, estado de procesamiento, texto OCR, texto limpio, resumen, tipo documental y errores.
- **Almacenamiento local:** Los archivos PDF originales, imágenes procesadas página a página, caché de visualización y registros de auditoría se guardan en almacenamiento local.
- **Motor OCR:** El motor OCR utiliza Mistral OCR para extraer texto de documentos PDF, dividir contenido según páginas, identificar tablas, encabezados y pies de página. Cuando una tabla resulta incompleta, se activa un mecanismo de respaldo basado en análisis por imagen.

- **Módulo de limpieza y normalización:** Este componente se dedica a suprimir interferencias visuales, erradicar logos y marcas que no pertenecen al material documental, conservar los datos de interés, unificar el formato de las tablas en HTML y respetar la división original de las páginas.
- **Módulo de IA generativa:** Este componente se dedica a suprimir interferencias visuales, erradicar logos y marcas que no pertenecen al material documental, conservar los datos de interés, unificar el formato de las tablas en HTML y respetar la división original de las páginas.
- **Módulo de seguridad:** Los PDF que llegan se someten a validación, se fijan restricciones en cuanto al tamaño de los documentos, se aplican controles básicos de velocidad de acceso, se registran las actividades en los logs locales y se ofrecen medidas de protección durante la descarga de archivos.
- **Sistema de evaluación experimental:** Se ha implementado un conjunto de scripts que pueden ejecutarse directamente desde la terminal, lo que permite calcular métricas como WER, CER y G-Eval utilizando la API de OpenAI, además de realizar análisis comparativos entre la estrategia original y el enfoque mejorado que se presenta.
- **Docker / entorno local:** Docker Compose permite levantar PostgreSQL de manera aislada y reproducible, facilitando el entorno de desarrollo en máquinas locales.

## 4.2 Resultados de funcionamiento del sistema

El sistema desarrollado permite cargar documentos PDF, procesarlos mediante OCR, limpiar el texto extraído, conservar tablas en formato HTML, generar un resumen automático y clasificar el tipo documental. Además, permite visualizar de forma paralela la página original del PDF y el texto extraído correspondiente. Durante las pruebas se verificó que el sistema puede procesar distintos tipos de documentos, entre ellos:

- Facturas

- Resoluciones
- Memorandos
- Oficios
- Solicitudes
- Certificados
- Ordenanzas
- Contratos

## 4.3 Evaluación de calidad mediante WER y CER

Para evaluar la precisión del texto extraído se utilizaron las métricas WER y CER, la cual CER mide el error a nivel de caracteres, mientras que el WER mide el error a nivel de palabras. Ambas métricas comparan el texto extraído por el sistema contra el texto de referencia elaborado a partir del documento original editable en formato Word.

Según Song [152], menciona que CER permite determinar qué tan preciso es el reconocimiento a nivel de caracteres, identificando los errores presentes respecto al texto original. Por otro lado, WER se enfoca en la precisión a nivel de palabras, calculando el porcentaje de palabras que no fueron reconocidas correctamente en comparación con el texto de referencia [146].

$$\text{Tasa de error de caracteres (CER)} = \frac{\text{Número de caracteres incorrectos}}{\text{Número total de caracteres en el texto de referencia}} \times 100 \quad (4.1)$$

$$\text{Tasa de error de palabras (WER)} = \frac{\text{Número de palabras incorrectas}}{\text{Número total de palabras en el texto de referencia}} \times 100 \quad (4.2)$$

Antes de calcular WER y CER, se aplicó una normalización controlada para que la comparación sea justa entre el texto de referencia y el texto extraído por el sistema, en la Tabla XXI se menciona lo que se hizo principalmente.

**Tabla XXI**.
Procesos de normalización y limpieza aplicados durante la evaluación

| **Proceso** | **Descripción** |
|---|---|
| Uso del texto limpio del sistema | La evaluación no se realizó sobre el OCR crudo. Se utilizó el campo `clean_text`, correspondiente al texto obtenido después del proceso de limpieza y preprocesamiento documental. |
| Limpieza del texto de referencia | El archivo Word de referencia fue convertido a texto plano y posteriormente sometido a reglas similares de normalización para permitir una comparación consistente con el texto extraído. |
| Conversión a minúsculas | Es una decisión que tomamos para que el sistema no interpretara "Certificado" y "certificado" como errores distintos durante la evaluación. De esta manera, logramos simplificar el proceso evitando que variaciones por mayúsculas afectaran los resultados. |
| Normalización de espacios | Nos encargamos de quitar aquellos espacios que estaban repetidos, homogeneizamos los saltos de línea y reacomodamos el material con el propósito de lograr una estructura más consistente que permitiera efectuar comparaciones de forma automatizada. |
| Normalización de caracteres | Se redujeron diferencias menores relacionadas con formato, signos especiales, comillas y separadores presentes en los documentos procesados. |

*Continúa en la siguiente página*

| Proceso | Descripción |
|---|---|
| Eliminación de ruido | Se removieron elementos que no aportan al contenido principal, tales como logos, marcas visuales, encabezados repetitivos, pies de página, URLs innecesarias y artefactos generados por OCR. |

### 4.3.1 Interpretación de las métricas

Los niveles de aceptabilidad en una transcripción automática no son fijos, ya que dependen del contexto en el que se aplique el sistema y de los objetivos planteados en cada proyecto. En la Tabla XXII se muestra una clasificación referencial de los valores CER y WER, elaborada a partir de la experiencia y los criterios utilizados por la comunidad investigadora durante los últimos años [153], [154]. Es importante considerar que estos rangos funcionan como una orientación general y no deben interpretarse como estándares estrictos.

**Tabla XXII**.
Interpretación de resultados CER y WER

| Métrica | Excelente | Muy bueno | Bueno | Mejorable | Deficiente |
|---|---|---|---|---|---|
| CER | $< 2{,}5\,\%$ | $2{,}5 - 5\,\%$ | $5 - 10\,\%$ | $10 - 20\,\%$ | $> 20\,\%$ |
| WER | $< 5\,\%$ | $5 - 15\,\%$ | $15 - 30\,\%$ | $30 - 50\,\%$ | $> 50\,\%$ |

Generalmente, un valor de CER menor al 5 % se interpreta como un desempeño excelente, debido a que la necesidad de realizar correcciones manuales posteriores disminuye considerablemente [154]. Cuando el CER se encuentra entre el 5 % y el 10 %, el resultado suele catalogarse como bueno o aceptable, ya que corregir el texto continúa siendo más rápido y práctico que transcribirlo completamente desde el inicio [153]. En cambio, valores superiores al 20 % normalmente se consideran poco satisfactorios para tareas donde se requiere una lectura clara y directa del contenido.

### 4.3.2 Resultados de WER y CER

El sistema alcanzó un CER promedio de 4.17 % y un WER promedio de 7.83 %, resultados que lo ubican dentro de un nivel de desempeño considerado muy bueno, de acuerdo con lo establecido en la Tabla XXII para aplicaciones de reconocimiento óptico de caracteres. De igual manera, se obtuvo una mediana de 1.74 % en CER y de 2.34 % en WER, cifras notablemente menores a los valores promedio. Esto sugiere que la mayoría de los documentos fueron reconocidos con muy pocos errores y que solo ciertos casos específicos influyeron en el incremento de las medias generales.

Algunos documentos lograron un 0 % en ambos indicadores, indicando que fueron reconocidos de manera perfecta al compararlos con el contenido original. Por otro lado, los peores resultados mostraron tasas que llegaron hasta 25.57 % en CER y 68.75 % en WER, esto nos posibilita detectar y poner de manifiesto la existencia de archivos cuya estructura resulta especialmente intrincada o que presentan una calidad visual comprometida, circunstancias que obstaculizaron de forma considerable el desempeño del reconocimiento óptico de caracteres.

En relación con la dispersión estadística, la desviación estándar de 6.22 % para CER y 14.34 % para WER refleja variabilidad en los resultados entre documentos. En particular, el WER presentó una dispersión superior, lo cual sugiere que ciertos errores afectaron principalmente la correcta identificación de palabras completas más que caracteres individuales. A continuación en la Tabla XXIII se describe los resultados obtenidos de WER y CER:

**Tabla XXIII**.

Resultados estadísticos de evaluación CER y WER ( %)

| **Métrica** | **CER** | **WER** |
|---|---|---|
| Media | 4.17 % | 7.83 % |
| Mediana | 1.74 % | 2.34 % |
| Mínimo | 0 % | 0 % |
| Máximo | 25.57 % | 68.75 % |
| Desviación estándar | 6.22 % | 14.34 % |

En la Figura 23 se observa que la mayor concentración de documentos se ubica en rangos bajos de error, lo que evidencia un desempeño favorable del sistema en la extracción textual, en la cual CER se obtuvo una media de 4.18 % y una mediana de 1.75 %, mientras que para WER se obtuvo una media de 7.83 % y una mediana de 2.34 %. La diferencia entre media y mediana indica la presencia de algunos documentos con errores más altos que incrementan el promedio general. Asimismo, la desviación estándar de WER fue mayor que la de CER, lo que refleja una mayor variabilidad en los errores a nivel de palabras. En general, los resultados muestran que el sistema mantiene una buena precisión en la mayoría de documentos evaluados, aunque existen casos específicos que requieren revisión por presentar errores superiores al comportamiento promedio.

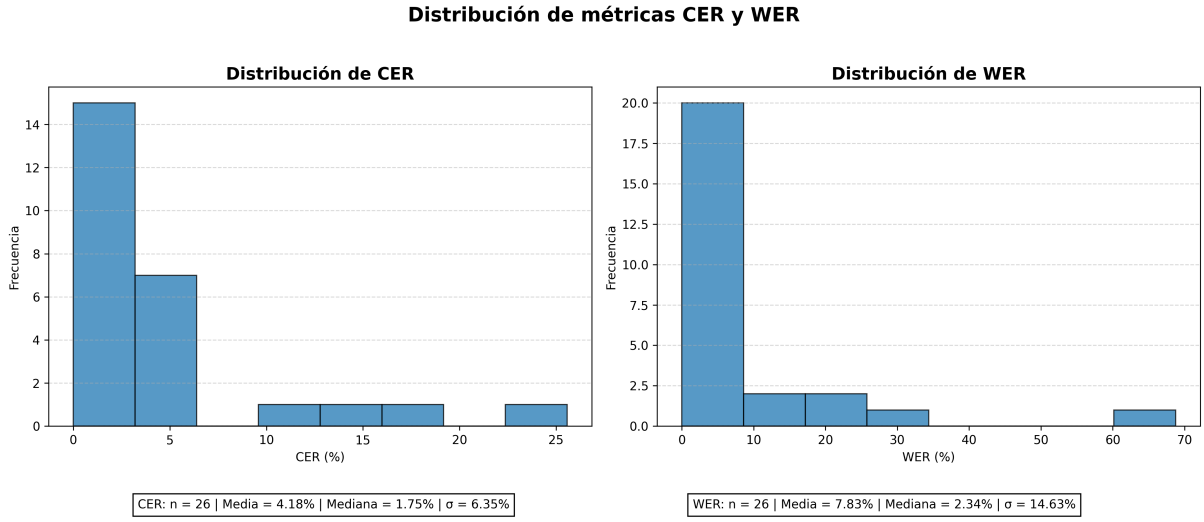


**Figura 23**. Distribución de métricas CER y WER

## 4.4 Evaluación mediante G-Eval

Con el fin de complementar la evaluación cuantitativa realizada mediante WER y CER, se aplicó una evaluación experimental basada en G-Eval utilizando un modelo de lenguaje de OpenAI [155]. Esta evaluación permitió analizar aspectos cualitativos del texto extraído que no siempre pueden ser medidos únicamente mediante comparación exacta de palabras o caracteres.

En esta evaluación se tomaron los 26 documentos que conformaban el conjunto de prueba. Para cada uno de ellos, se llevó a cabo una comparación entre el texto de referencia y el texto procesado que el sistema había generado (clean_text). La evaluación se ejecutó utilizando el modelo gpt-4o-mini, otorgando una puntuación que oscilaba entre 0 y 1 conforme a cada criterio evaluado, donde 0 indicaba un desempeño insuficiente y 1 significaba un desempeño excelente. Se fijó un umbral mínimo de aprobación en 0.7. De esta forma, un documento se consideró aprobado en una métrica determinada cuando alcanzaba una puntuación igual o mayor a ese umbral. El cálculo de la tasa de éxito se llevó a cabo aplicando la siguiente fórmula:

Tasa de éxito = Documentos con puntuación superior a 0.7 / Cantidad total de documentos analizados.

En la evaluación se tomaron en cuenta estas métricas: fidelidad, completitud, secuencia de lectura, calidad de las tablas, eliminación de ruido y puntuación final.

**Tabla XXIV**.
Resultados de evaluación mediante G-Eval

| **Métrica G-Eval** | **Descripción** | **Promedio** | **Tasa de éxito** |
|---|---|---|---|
| Fidelidad | Evalúa si el texto extraído conserva el contenido original sin alterar su significado. | 0.8769 | 0.9615 |
| Completitud | Evalúa si el texto extraído contiene la información relevante del documento. | 0.8038 | 0.9615 |

*Continúa en la siguiente página*

| Métrica G-Eval | Descripción | Promedio | Tasa de éxito |
|---|---|---|---|
| Orden de lectura | Evalúa si el contenido extraído mantiene una secuencia lógica y natural de lectura. | 0.8173 | 0.9615 |
| Calidad de tablas | Evalúa si las tablas fueron representadas correctamente en formato estructurado. | 0.5923 | 0.5000 |
| Limpieza de ruido | Evalúa si se eliminaron elementos no relevantes como logos, URLs, encabezados o pies innecesarios. | 0.7442 | 0.8846 |
| Puntaje general | Resume el desempeño global del texto extraído. | 0.7681 | 0.7692 |

Los resultados muestran que el sistema obtuvo un buen desempeño en fidelidad, completitud y orden de lectura, con puntajes promedio superiores a 0.8. Esto indica que el texto extraído conserva en gran medida el contenido del documento original y mantiene una estructura de lectura comprensible.

Respecto a la limpieza de ruido, se obtuvieron resultados que resultaron bastante satisfactorios, con una media de 0.7442, cifra que logró superar el umbral mínimo establecido. No obstante, existen documentos puntuales en los que persisten todavía algunos elementos visuales superfluos que merecerían un refinamiento adicional.

En cuanto a las tablas, este fue justamente el aspecto donde el sistema enfrentó sus mayores problemas, debido a que se logró apenas 0.5923 de promedio y una tasa de éxito de 0.5000. Lo anterior pone de manifiesto que, si bien el sistema logra procesar y organizar cierta porción del contenido en formato tabular, estas estructuras siguen representando el talón de Aquiles dentro del proceso completo de extracción de documentos.

La puntuación general que se obtuvo fue de 0.7681, demostrando que se logró rebasar con comodidad el umbral mínimo que se había fijado. Por lo tanto, según lo que muestra G-Eval, se puede decir que el sistema opera de manera satisfactoria en lo que respecta a la extracción y

organización del contenido, si bien claramente hay aspectos del manejo de tablas que requieren atención y mejora.

La Figura 24 muestra la matriz de correlaciones que obtuvimos comparando los diversos parámetros que evaluamos con G-Eval durante el análisis de qué tan bien el sistema OCR estructurado extrae el texto. Estos valores de correlación nos permiten ver cuán relacionadas están entre sí las métricas que miden fidelidad, completitud, secuencia de lectura, calidad de tablas y eliminación de interferencias.

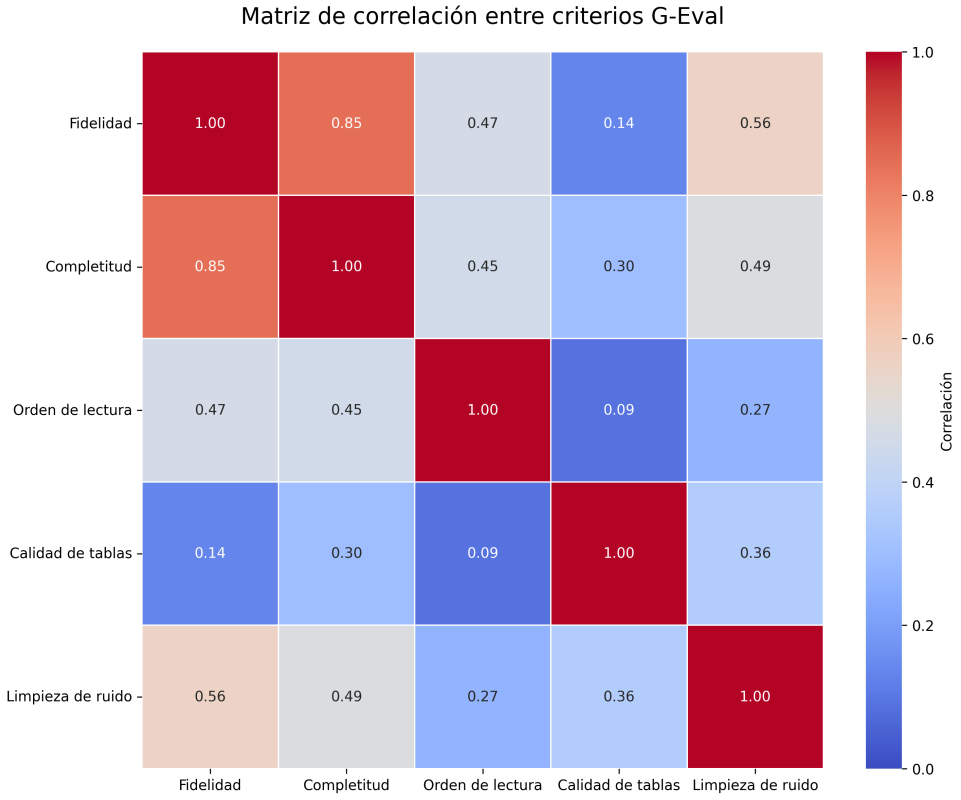


**Figura 24**. Matriz de correlación entre criterios de evaluación G-Eval aplicados al sistema de extracción documental.

Lo que llama especialmente la atención es la relación particularmente sólida que se observa entre la fidelidad y la completitud, con una correlación que alcanza 0.85, lo cual evidencia que los documentos que mejor preservaron el significado original fueron también quienes mantuvieron una mayor cantidad de información relevante.

Asimismo, se destaca que el proceso de eliminación de ruido presentaba correlaciones mo-

deradas respecto a la fidelidad (0.56) y la completitud (0.49), lo que evidencia de qué forma la supresión efectiva de componentes interferentes, encabezados repetidos, logos, enlaces web, errores de reconocimiento óptico, contribuye favorablemente a incrementar la calidad general del texto recuperado.

Respecto a cómo se organiza la lectura, se encuentra que este aspecto mantiene una relación moderada con la fidelidad (0.47) y la completitud (0.45), lo que sugiere que mantener el contenido en un orden lógico y coherente juega un papel importante, aunque no determinante, en mejorar la capacidad de asimilar completamente la información que se procesa.

La relación entre la calidad de las tablas y el orden de lectura resultó ser la más tenue (0.09), lo que muestra que ambos criterios evalúan aspectos completamente diferentes en la extracción de documentos. Este hallazgo detalla que contar con tablas adecuadamente organizadas no garantiza necesariamente que el contenido extraído presente una secuencia lógica y coherente.

A su vez, los resultados sugieren que los parámetros relacionados con mantener la esencia del contenido y la depuración de los archivos presentan asociaciones notables, lo que resalta que tales factores ejercen una influencia determinante en la eficacia integral del sistema de recuperación de documentos que se plantea.

### 4.4.1 Evaluación G-Eval y Validación Humana

Con el propósito de complementar la evaluación automática del sistema, se realizó una comparación entre los resultados obtenidos mediante G-Eval y una evaluación humana basada en los mismos criterios de calidad [151]. Esta comparación permitió analizar si la valoración generada por el modelo de lenguaje se aproxima a la percepción de usuarios humanos al revisar el texto extraído por el sistema.

A fin de llevar a cabo la evaluación automática, se hace uso de G-Eval [155] se utiliza el modelo gpt-4o-mini, se tiene en cuenta criterios tales como la fidelidad, la exhaustividad, el orden de lectura, la calidad de las tablas, la eliminación de ruido y la puntuación total. Cada uno

de estos indicadores se calificó en una escala del 0 al 1, siendo 0 una nota suspensa y 1 una nota excelente. Establecimos el umbral de aceptación en una nota mínima de 0,7.

En la evaluación humana se aplicó una rúbrica basada en escala Likert de 1 a 5 [151], y fue realizada a tres evaluadores que corresponden a personal administrativo de la Empresa Pública de Movilidad, quienes revisaron los documentos procesados por el sistema y calificaron los mismos criterios utilizados en G-Eval, las calificaciones originales asignadas por los evaluadores humanos se presentan en los Anexos, dichos puntajes humanos fueron normalizados a una escala de 0 a 1 mediante la siguiente fórmula:

Puntaje humano normalizado = Puntaje humano / 5

**Tabla XXV**.
Comparación entre evaluación humana normalizada y G-Eval

| **Criterio** | **Evaluación Humana** | **G-Eval** | **Diferencia absoluta** |
|---|---|---|---|
| Fidelidad | 0.6667 | 0.8769 | 0.2102 |
| Completitud | 0.6667 | 0.8038 | 0.1371 |
| Orden de lectura | 0.7667 | 0.8173 | 0.0506 |
| Calidad de tablas | 0.4667 | 0.5923 | 0.1256 |
| Limpieza de ruido | 0.5333 | 0.7442 | 0.2109 |

Los resultados muestran que G-Eval asignó puntajes superiores a los evaluadores humanos en la mayoría de criterios. La mayor cercanía se observa en el orden de lectura, con una diferencia absoluta de 0.0506, lo que indica una coincidencia relativamente alta entre la percepción humana y la evaluación automática para este criterio. En cambio, las mayores diferencias se presentan en fidelidad y limpieza de ruido, donde los evaluadores humanos fueron más estrictos que G-Eval.

En la Figura 25 permite observar de qué manera se alinean las valoraciones de expertos humanos con los resultados que proporciona G-Eval. Lo notable aquí es que tanto un método como el otro convergen hacia conclusiones similares: la secuencia de lectura, la fidelidad al contenido

original y la exhaustividad resultan ser los elementos que reciben mayores calificaciones, aunque las tablas se encuentran notablemente por debajo con los resultados más bajos. Esto pone de manifiesto que, tanto si se habla de evaluadores humanos como del sistema G-Eval, existe consenso a la hora de reconocer las tablas como la zona más crítica que demanda intervención urgente en el sistema.

Lo curioso es que G-Eval tiende a asignar puntuaciones superiores en prácticamente todos los aspectos evaluados. Esto podría deberse a que los evaluadores automáticos generalmente priorizan la consistencia general del contenido por encima de otras consideraciones, dado que los especialistas en la materia tienden a ser más exigentes en sus valoraciones, pues su conocimiento profundo les permite identificar con mayor facilidad detalles visuales sutiles, información ausente, rastros de imperfecciones o problemas en cómo se presenta el material.

Respecto al orden de lectura, los dos métodos arrojaron resultados muy parecidos, alcanzando una puntuación de 0.77 en la evaluación humana y de 0.82 con G-Eval. Esto pone de manifiesto un consenso general en torno a que el sistema logra mantener adecuadamente la secuencia de lectura. En cambio, las discrepancias más notables surgen en lo referente a la fidelidad y la eliminación de ruido, revelando que los evaluadores humanos detectaron deficiencias o artefactos que el sistema automático no penalizó de manera igualmente severa.

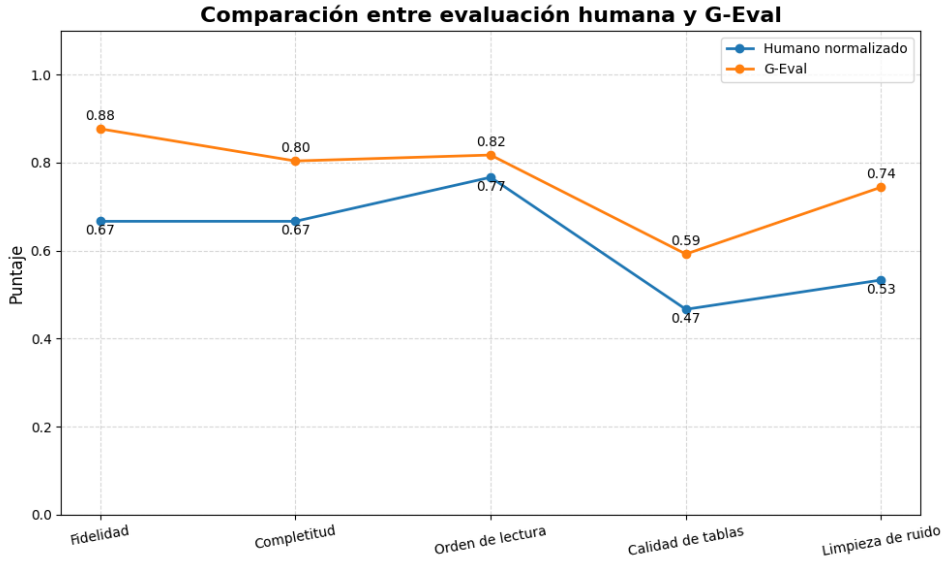


**Figura 25**. Comparación entre evaluación humana normalizada y G-Eval.

## 4.5 Evaluación de rendimiento mediante benchmarks

Con el propósito de analizar el rendimiento del sistema, se realizaron pruebas mediante benchmarks, según [156], un benchmark puede entenderse como la combinación de uno o varios conjuntos de datos junto con una métrica de evaluación, diseñada para representar tareas o capacidades específicas y utilizada por la comunidad investigadora como un marco común para comparar diferentes métodos. El uso de benchmarks se ha convertido en una práctica habitual, ya que permite facilitar comparaciones, evaluar el rendimiento, monitorear avances e identificar limitaciones en los modelos desarrollados [150]. Esta evaluación permitió identificar cuánto tarda el sistema en extraer, procesar y analizar documentos PDF mediante OCR e inteligencia artificial generativa. En este caso, se compararon dos enfoques los que se mencionan en la Tabla XXVI:

**Tabla XXVI**.
Métodos utilizados en la evaluación experimental

| **Método** | **Descripción** |
|---|---|
| Baseline | Utiliza la salida OCR en texto plano sin transformación estructural. |
| Propuesto | Utiliza OCR estructurado, limpieza técnica y representación de tablas en formato HTML. |

La evaluación se realizó midiendo los tiempos en segundos para cada documento, en la Tabla XXVII presenta las métricas promedio obtenidas para los métodos baseline y propuesto, permitiendo comparar el impacto temporal asociado a las mejoras estructurales implementadas en el sistema.

**Tabla XXVII**.
Resultados estadísticos de tiempos de procesamiento

| **Métrica** | **Baseline (s)** | **Propuesto (s)** |
|---|---|---|
| Tiempo mínimo | 5.93 | 5.30 |
| Tiempo máximo | 15.84 | 16.92 |
| Tiempo promedio | 8.61 | 9.37 |
| Mediana | 7.60 | 8.19 |
| Desviación estándar | 2.60 | 3.01 |

En la Tabla XXVII se observa que el método propuesto presenta un tiempo promedio de procesamiento ligeramente superior al baseline, con 9.37 s frente a 8.61 s. Esta diferencia corresponde a un incremento aproximado de 0.76 s por documento, lo cual se considera aceptable debido a que el enfoque propuesto no solo extrae texto plano, sino que también genera una salida más estructurada, conserva tablas en formato HTML, aplica una limpieza conservadora del contenido y permite obtener un resumen automático del documento. Por tanto, el aumento en el tiempo de procesamiento se compensa con una mejora en la organización, legibilidad y utilidad del texto extraído para la gestión documental.

La Figura 26 se presentan diagramas de cajas y bigotes que ilustran cómo se distribuyen los tiempos de procesamiento entre el enfoque baseline y el que proponemos. De acuerdo con lo que muestran los datos, nuestro método resulta con una mediana un poco más elevada frente al baseline, lo cual refleja un aumento moderado en cuanto al tiempo que tarda en ejecutarse. Esto es comprensible considerando que incluimos pasos suplementarios como la depuración de documentos, la organización de los datos extraídos por OCR y la creación de estructuras tabulares.

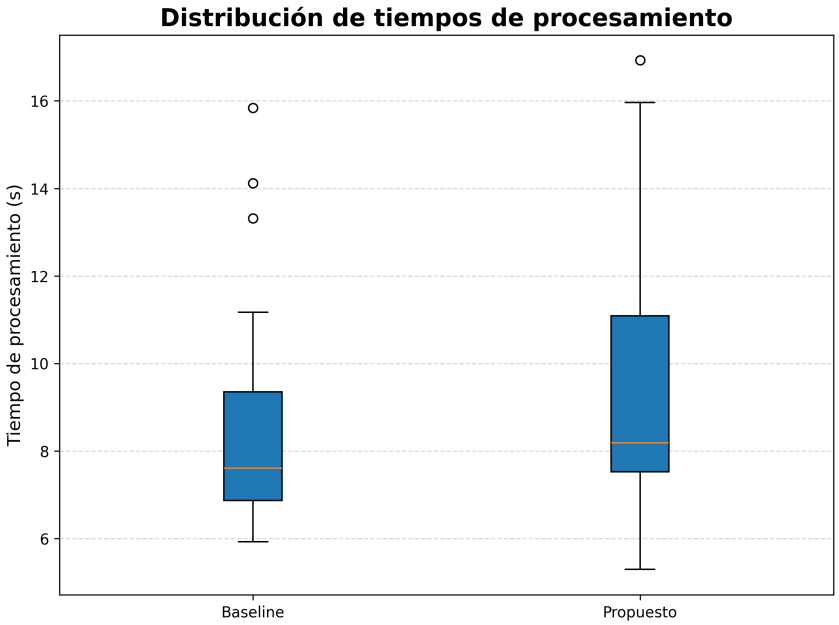


**Figura 26**. Distribución estadística de tiempos de procesamiento entre el método baseline y el método propuesto.

### 4.5.1 Análisis de tiempos de procesamiento del sistema

En la Figura 27 se presenta la distribución de los tiempos de procesamiento del sistema para los 26 documentos evaluados. Se observa que la mayor concentración de documentos se encuentra entre los 7 y 8 segundos, lo que evidencia que la mayoría de archivos fueron procesados en tiempos moderados. El tiempo mínimo registrado fue de 5.30 segundos y el máximo de 16.92 segundos.

La mediana de 8.19 segundos contrasta notablemente con la media de 9.37 segundos, lo que evidencia un sesgo marcado hacia la derecha en la distribución. Esto sucede porque algunos documentos particulares requirieron un tiempo significativamente mayor en su procesamiento probablemente se trate de expedientes más voluminosos, con varias páginas, tablas incrustadas o estructuras documentales bastante complejas.

En líneas generales, los hallazgos evidencian que el sistema ofrece tiempos de respuesta completamente satisfactorios en el contexto de la gestión de documentos, teniendo en cuenta que todo el flujo involucra reconocimiento óptico de caracteres, depuración textual, organización

estructurada y elaboración automática de síntesis mediante modelos de inteligencia artificial generativa.

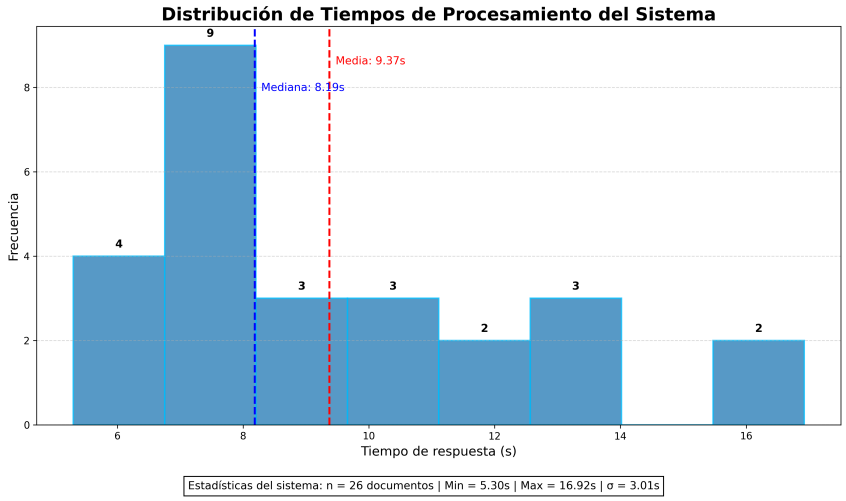


**Figura 27**. Distribución de los tiempos del sistema

## 4.6 Discusión de Resultados

Los resultados obtenidos permiten analizar el desempeño del sistema desde tres perspectivas: calidad de extracción textual, calidad semántica y estructural, y rendimiento del procesamiento documental. Para ello se aplicaron métricas WER y CER, evaluación G-Eval, evaluación humana y benchmarks de rendimiento sobre un conjunto de 26 documentos.

Los datos de WER/CER muestran un rendimiento bastante satisfactorio: el CER se situó en un promedio de 4.18 % mientras que el WER alcanzó un 7.83 % en promedio, lo que indica que los errores fueron relativamente contenidos en la mayoría de los archivos evaluados. Es interesante observar cómo en estas dos métricas la mediana quedó por debajo de la media, un patrón que indica que la mayoría de los documentos logró alcanzar grados de precisión aceptables; sin embargo, algunos archivos especialmente difíciles de procesar terminaron elevando considerablemente los valores promedio al final.

La evaluación mediante G-Eval alcanzó una puntuación de 0.7681, lo que obviamente supera con éxito el umbral mínimo fijado en 0.7. Los resultados más sobresalientes aparecieron

en rubros como fidelidad, integridad y secuencia de lectura, lo que demuestra que el sistema mantiene fielmente la información fuente y conserva una estructura coherente y sin problemas. Sin embargo, fue la evaluación de las tablas la que obtuvo los resultados más pobres, lo que confirma que estos elementos de alto nivel de complejidad representan una de las barreras más críticas que el sistema debe superar.

En lo que respecta al desempeño, nuestro método tardó en promedio 9.37 segundos en procesar los datos, superado ligeramente por el baseline que completó la tarea en 8.61 segundos. Si bien la propuesta resultó un poco más lenta, hay que considerar que incluye fases extras de limpieza, normalización y estructuración del contenido textual. De esta manera, la diferencia temporal resulta perfectamente justificable si se toman en cuenta los beneficios derivados: una salida significativamente más ordenada y aprovechable para cualquier sistema de gestión documental.

Las conversaciones directas con evaluadores permitió comprobar cuán confiables resultaban ser los veredictos generados por G-Eval. Lo que se observó fue que los evaluadores humanos mostraban una severidad bastante mayor que la del sistema automatizado, sobre todo a la hora de juzgar la fidelidad de los datos y el nivel de ruido que contenían. A pesar de ello, los resultados de la evaluación tanto manual como automatizada llegaron a la misma conclusión fundamental: las tablas fueron el aspecto con peor rendimiento entre todos los que se examinaron, lo que pone de relieve la necesidad imperante de enfocarse en potenciar este componente en versiones futuras.

Los hallazgos demuestran que la plataforma logra satisfactoriamente su propósito de recuperar información textual de archivos PDF, estructurarla adecuadamente y procesarla con técnicas de OCR e inteligencia artificial generativa. Dicho esto, quedó clara la necesidad de refinar el funcionamiento cuando se trata de archivos que contienen cuadros voluminosos, composiciones visuales intrincadas o interferencia visual de cualquier tipo.

### 4.6.1 Interpretación de Métricas

La evaluación mediante métricas del sistema permitieron su observación desde diferentes dimensiones:

- **CER:** se midió el error tomando en cuenta el nivel de caracteres. Esta métrica fue útil para identificar errores pequeños como letras omitidas, caracteres incorrectos o diferencias mínimas en palabras. El CER promedio de 4.18 % indica que el sistema mantuvo una alta precisión textual a nivel de caracteres.
- **WER:** permitía calcular el error a nivel de palabra. Esta métrica fue más sensible que CER, ya que si se omitió, añadió o sustituyó una palabra el resultado cambia. El promedio de WER obtenido fue de 7.83 % lo que indica que si bien el sistema conserva la mayor información, en algunos documentos hubo errores al omitir o añadir palabras.
- **G-Eval:** resultó fundamental para valorar dimensiones cualitativas que frecuentemente escapan a las métricas convencionales como WER y CER. Mediante indicadores específicos como fidelidad, completitud, secuencia de lectura, integridad de tablas y eliminación de artefactos, fue posible determinar si el texto recuperado mantenía coherencia, estaba íntegro, era legible y presentaba una estructura adecuada. Con un puntaje de 0.7681, el sistema demostró haber trascendido el nivel mínimo de desempeño que se había establecido como aceptable.
- **Evaluación humana:** la validación por parte de usuarios: esta etapa fue importante para contrastar cómo percibían los usuarios las respuestas que producía el sistema. Cuando se pusieron en paralelo las valoraciones humanas con las de G-Eval, quedó evidente que el sistema automático tendía a otorgar calificaciones superiores, en tanto que los evaluadores de carne y hueso mostraban mayor rigor al detectar defectos de visualización, artefactos que permanecían en la imagen o desajustes dentro de los datos tabulares.
- **Benchmarks:** de rendimiento permitieron medir los tiempos de procesamiento del sistema. Estas métricas fueron útiles para analizar el costo temporal del método propuesto

frente al baseline. Aunque el método propuesto fue ligeramente más lento, entregó una salida más estructurada y adecuada para el análisis documental.

### 4.6.2 Factores de Éxito

Los resultados obtenidos permiten identificar varios factores que contribuyeron al desempeño positivo del sistema:

- Uso de Mistral OCR estructurado, permitió extraer texto de documentos PDF y representar tablas en formato HTML. Esto facilitó una salida más organizada y cercana a la lectura natural del documento.
- Implementación de una limpieza conservadora del texto, orientada a eliminar ruido sin alterar el contenido relevante. Este proceso permitió reducir elementos innecesarios como logos, URLs, encabezados, pies de página y artefactos visuales.
- Separación del contenido por páginas, lo que permitió visualizar el PDF original junto al texto extraído correspondiente. Esto mejoró la revisión del documento y facilitó la validación del resultado.
- Uso de Gemini para la generación de resúmenes y clasificación documental, permitiendo obtener una interpretación automática del contenido extraído.
- Incorporación de métricas experimentales de evaluación, como WER, CER, G-Eval, evaluación humana y benchmarks. Esto permitió medir el sistema desde una perspectiva objetiva y no limitarse únicamente a una revisión visual.
- El enfoque modular del sistema también fue un factor importante, ya que permitió separar responsabilidades entre OCR, limpieza, análisis con IA, almacenamiento, visualización y evaluación. Esta organización facilitó la implementación de mejoras sin afectar el flujo principal del sistema.

# CONCLUSIONES

La puesta en marcha de este sistema permitió cumplir exitosamente con el objetivo principal que originó esta investigación: crear una herramienta capaz de extraer y analizar contenido textual desde documentos PDF empleando tecnología OCR e inteligencia artificial generativa. La solución logró incorporar de manera armónica un conjunto amplio de características, que abarcaban desde la recepción y validación de documentos hasta la extracción y procesamiento de datos textuales, pasando por una organización estructurada en apartados, la generación automática de síntesis documentales, la clasificación y gestión de ficheros, complementadas con herramientas que permitían consultar información de forma ágil y generar informes personalizados según las necesidades. Todo esto representó un avance notable en la optimización de las actividades relacionadas con la gestión y administración documental.

Mistral OCR resultó ser una solución bastante efectiva para extraer información textual de la gran mayoría de los documentos que sometimos a prueba. La configuración que pusimos en marcha logró generar un resultado bien organizado y estructurado, con los datos separados por páginas y las tablas convertidas a formato HTML, aspecto que sin duda hizo más fácil tanto la lectura como el análisis y el uso posterior del material obtenido.

Respecto a la exactitud del sistema, los números que arrojaron las métricas WER y CER mostraron un comportamiento positivo en lo que respecta a la extracción de texto. El CER normalizado promedió un 4.18 %, en tanto que el WER se situó en un promedio del 7.83 %, lo que pone de manifiesto que en los documentos analizados se registraron errores mínimos frente a los textos que sirvieron como referencia en el proceso de evaluación.

De otra forma, mediante el método G-Eval fue posible examinar tanto la calidad semántica como la estructural de lo que el sistema había generado. Con un puntaje medio de 0.7681, los resultados superaron ampliamente el estándar mínimo de 0.7 que se había fijado. Las evaluaciones más altas correspondieron a aspectos vinculados con la fidelidad, la cobertura completa de contenidos y el flujo natural de la lectura, aunque quedó claro que la forma en que se representaban las tablas más elaboradas era donde aún había espacio para avanzar.

Por otra parte, la revisión llevada a cabo por personas permitió expandir y contrastar los resultados que proporcionó la herramienta automática G-Eval. Los especialistas efectuaron un examen mucho más profundo, enfocándose especialmente en aspectos como la exactitud de la información, la eliminación de componentes superfluos y la calidad de los cuadros presentados. A pesar de que los métodos automático y manual llegaron a conclusiones distintas en varios aspectos, ambos coincidieron en un punto fundamental: el verdadero problema que tiene el sistema radica en cómo gestiona y presenta aquellas estructuras de datos tabulares particularmente complejas.

En cuanto al desempeño, los benchmarks realizados mostraron que el enfoque propuesto demanda un poco más de tiempo computacional comparado con la solución de referencia. Mientras que el baseline se ubicó en 8.61 segundos en promedio, nuestro sistema logró 9.37 segundos. A pesar de ello, los 0.76 segundos adicionales resultan completamente tolerables, pues el método que desarrollamos entrega resultados mucho más ordenados, claros y provechosos para quienes necesitan analizar y gestionar documentos.

Indicar al final que, el mecanismo que hemos desarrollado actúa como un instrumento útil y seguro para potenciar cómo se administran los documentos, transformando archivos PDF en contenido textual procesable, lo que permite realizar más rápida la localización de datos específicos, creando resúmenes de manera automática y organizando la información obtenida para que sea más simple revisarla, comprenderla y utilizarla dentro de plataformas digitales.

# RECOMENDACIONES

Habría que trabajar más a fondo en cómo se procesan esas tablas intrincadas, sobre todo las que tienen muchas columnas, se extienden por varias filas o están repartidas en diferentes páginas. Los resultados lo dejan claro: tanto en los análisis automáticos con G-Eval como en las evaluaciones hechas por personas, fue justamente aquí donde se vieron los peores rendimientos. Si logramos mejorar este aspecto, podríamos conseguir que la información que extraemos sea mucho más exacta y de mejor calidad.

También convendría enriquecer la colección de archivos que se emplean en las fases de prueba, integrando documentación heterogénea y distintas calidades en los procesos de digitalización. Resultaría valioso contar con materiales que incluyan formularios, rúbricas manuscritas, sellos institucionales, cuadros amplios y composiciones gráficas más intrincadas, para así robustecimiento la solidez del proceso de validación y examinar cómo reacciona el sistema ante casos más semejantes a la realidad operativa.

La seguridad es algo que no podemos pasar por alto, especialmente cuando se trata de proteger los documentos que maneja nuestro sistema. Por eso resulta imprescindible implementar múltiples niveles de autenticación y autorización que garanticen una defensa robusta. Esto cobra mayor relevancia todavía en entornos institucionales o cuando el sistema se encuentra en funcionamiento real. Entre los cambios que habría que considerar están la adopción de HTTPS, un mecanismo de permisos que se adapte a los diferentes roles dentro de la organización, controles de auditoría más exhaustivos y medidas de seguridad más firmes en las zonas donde se descargan archivos, con el propósito fundamental de garantizar tanto la privacidad como la confiabilidad de la información guardada.

Dado que el alcance de este proyecto tiene límites bien definidos, la solución que se desarrolló deliberadamente no incluye funcionalidades para autenticar usuarios ni administrar sus perfiles, ya que la prioridad estaba puesta en optimizar todo lo relacionado con cargar, extraer, analizar, buscar y evaluar archivos PDF. No obstante, es importante señalar que si este sistema llegara a implementarse en un contexto real o dentro de una institución, sería imprescindible

agregar sistemas de autenticación, controles de acceso con permisos específicos por rol y restricciones en el manejo de documentos, todo esto para proteger adecuadamente los datos que guarda el sistema y asegurar que solo personal autorizado pueda acceder y usarlo.

Cuando te enfrentas a documentos extensos, es útil pensar en estrategias que hagan más eficiente la creación de resúmenes. Puedes intentar fragmentar el material en secciones, usar una estructura de niveles para procesarlo o reducir de antemano cuánto texto le entregas al modelo de IA generativa.

De igual forma, es importante que las pruebas experimentales funcionen de manera separada respecto a los componentes principales del sistema, lo que permitirá que métricas de rendimiento como WER, CER, G-Eval y otros estándares de comparación se puedan ejecutar tanto desde la línea de comandos como por medio de módulos independientes, lo que permitió además simplificar el testing y el análisis posterior sin afectar la velocidad del sistema ni la satisfacción de los usuarios.

Lo recomendable es trabajar junto con especialistas que se desenvuelven directamente en la gestión de documentos para realizar pruebas que permitan evaluar aspectos como si la plataforma resulta accesible, si el diseño es comprensible y si las herramientas implementadas realmente funcionan según lo esperado. Una estrategia de este tipo proporcionaría datos valiosos para mejorar y adaptar el software en función de las demandas que puedan surgir más adelante.

# BIBLIOGRAFÍA


[1] I. Gil-Leiva, M. S. L. Fujita, F. M. Redigolo y J. F. Saran, "Extracción de información de documentos PDF para su uso en la indización automática de e-books," *Transinformação*, vol. 34, e210069, 2022, [En línea]. Disponible: https://www.scielo.br/j/tinf/a/GGTg9JHsPhvch8qgcRbCVQN/?lang=es, ISSN: 0103-3786. DOI: 10.1590/2318-0889202234e210069.

[2] M. Maciel y C. F. Pons, "Extracción de información de facturas (ar) agrupada en jerarquías de negocios," en *Simposio Argentino de Inteligencia Artificial (ASAI 2022)-JAIIO 51 (Modalidad virtual y presencial (UAI), octubre 2022)*, [En línea]. Disponible: https://sedici.unlp.edu.ar/handle/10915/151615, 2022, págs. 7-21.

[3] H. Motahari, N. Duffy, P. Bennett y T. Bedrax-Weiss, "A report on the first workshop on document intelligence (DI) at NeurIPS 2019," *ACM SIGKDD Explorations Newsletter*, vol. 22, n.º 2, págs. 8-11, 2021, [En línea]. Disponible: https://dl.acm.org/doi/abs/10.1145/3447556.3447563. DOI: 10.1145/3447556.3447563.

[4] E. D. González et al., "La Gestión Documental empresarial ante el desafío de la transición digital," *CLIP de SEDIC: Revista de la Sociedad Española de Documentación e Información Científica*, n.º 85, págs. 5-10, 2022, [En línea]. Disponible: https://edicionsedic.es/clip/article/view/69/59, ISSN: 2659-2983. DOI: 10.47251/clip.n85.69.

[5] Escuela Superior Politécnica del Litoral y Banco Interamericano de Desarrollo, "Reporte Chequeo Digital 2022–2023: El aprendizaje en la madurez digital en Ecuador," Ministerio de Telecomunicaciones y de la Sociedad de la Información, Quito, Ecuador, inf. téc., 2024. visitado 17 de ago. de 2025. dirección: https://observatorioecuadordigital.mintel.gob.ec/wp-content/uploads/2024/09/CHEQUEO-DIGITAL.pdf.

[6] E. A. A. Churo, J. P. C. Tapia y G. A. Cisneros, "Detección y digitalización de datos de interés en documentos de identificación," *Dominio de las Ciencias*, vol. 8, n.º 3, pág. 10, 2022, [En línea]. Disponible: https://dialnet.unirioja.es/servlet/articulo?codigo=8635187, ISSN: 2477-8818. DOI: 10.23857/dc.v8i3.

[7] D. C. Rodríguez y J. E. P. La, "Sistema OCR para la extracción de información digitalizada proveniente de máquinas de escribir," *Serie Científica de la Universidad de las Ciencias Informáticas*, vol. 10, n.º 11, págs. 24-31, 2017, [En línea]. Disponible: https://dialnet.unirioja.es/servlet/articulo?codigo=8589933, ISSN: 2306-2495.

[8] W. Araque-Jaramillo, E. Suquillo-Guijarro, A. Arguello-Salazar, J. Cadena y G. Duque Espinoza, *Digitalización y el desarrollo sostenible de la MIPYME en Ecuador*. Universidad del Azuay Casa Editora, mar. de 2024, pág. 90, ISBN: 9789942645647. DOI: 10.33324/ceuazuay.323.702. visitado 17 de ago. de 2025. dirección: https://publicaciones.uazuay.edu.ec/index.php/ceuazuay/catalog/book/323.

[9] M. Blázquez-Ochando y P. Lázaro-Rodríguez, "Debates, desafíos y oportunidades de la inteligencia artificial en Documentación: El seminario ConocimIA," *Métodos de información*, vol. 15, n.º 28, págs. 52-83, 2024, [En línea]. Disponible: https://www.metodosdeinformacion.es/mei/index.php/mei/article/view/IIMEI15-N28-052083. DOI: 10.5557/IIMEI15-N28-052083.

[10] United Nations Department of Economic and Social Affairs, *Informe de los Objetivos de Desarrollo Sostenible 2023: Edición especial*. United Nations, oct. de 2023, ISBN: 9789210024938. DOI: 10.18356/9789210024938. visitado 17 de ago. de 2025. dirección: https://doi.org/10.18356/9789210024938.

[11] C. Bustelo-Ruesta, "Los grandes temas relacionados con la gestión de documentos: desafíos y oportunidades," *Profesional de la información*, vol. 20, págs. 129-134, 2 2011, [En línea]. Disponible: https://revista.profesionaldelainformacion.com/index.php/EPI/article/view/epi.2011.mar.01, ISSN: 1699-2407. DOI: 10.3145/EPI.2011.MAR.01.

[12] I. E. Z. Plúa, E. M. Q. Torrijo y F. R. León, "Gestión documental en universidades: Una mirada desde Latinoamérica," *ReHuSo: Revista de Ciencias Humanísticas y Sociales*, vol. 6, págs. 108-119, Especial ago. de 2021, [En línea]. Disponible: https:

//revistas.utm.edu.ec/index.php/Rehuso/article/view/3779, ISSN: 2550-6587. DOI: 10.33936/REHUSO.V6IESPECIAL.3779.

[13] C. S. S. Arteaga y L. A. G. Salmon, "El nivel de eficacia y eficiencia como principio fundamental de la gestión documental," *Revista de Ciencias Humanísticas y Sociales (ReHuSo)*, vol. 6, págs. 87-101, 1 2021, [En línea]. Disponible: http://scielo.senescyt.gob.ec/scielo.php?script=sci_arttext&pid=S2550-65872021000100087, ISSN: 2550-6587. DOI: 10.5281/ZENODO.5513107.

[14] R. Y. Flores Huicho, "Implementación del modelo de gestión documental digital para mejorar la calidad de servicio en la empresa ENTER SAC," [En línea]. Disponible: https://repositorio.upci.edu.pe/handle/upci/235, Tesis de pregrado, Universidad Peruana de Ciencias e Informática, dic. de 2020.

[15] Á. M. Ledesma-Ramos, "La Gestión Documental en la Administración Pública. Recurso estratégico para el logro de los objetivos," *Revista Estudios del Desarrollo Social: Cuba y América Latina*, vol. 10, 3 2022, [En línea]. Disponible: http://scielo.sld.cu/scielo.php?pid=S2308-01322022000300014&script=sci_arttext&tlng=en, ISSN: 2308-0132.

[16] Y. C. Montejo y H. P. Sousa, "Gestión documental, Gestión de información y Gestión del conocimiento: nociones e interrelaciones," *Bibliotecas. Anales de investigación*, págs. 222-227, 0 abr. de 2021, [En línea]. Disponible: https://dialnet.unirioja.es/servlet/articulo?codigo=5704452, ISSN: 0006-176X.

[17] International Organization for Standardization, "ISO/TR 15489-2:2001: Information and documentation – Records management – Part 2: Guidelines," International Organization for Standardization, Technical Report ISO/TR 15489-2:2001, sep. de 2001. visitado 17 de ago. de 2025. dirección: https://www.iso.org/standard/35845.html.

[18] J. del Castillo Guevara y M. M. Mena Mugica, "La gestión de documentos de archivo en el actual contexto organizacional y la introducción a la Norma ISO 15489," *ACIMED*, vol. 22, págs. 47-59, 2011, [En línea]. Disponible: http://scielo.sld.cu/scielo.

php?script=sci_arttext&pid=S1024-94352011000100005&nrm=iso, ISSN: 1024-9435.

[19] L. Camilo-Momblanc y H. Y. Castro-Milán, "La gestión documental y el control interno: un binomio indispensable," *Revista del Archivo Nacional*, vol. 84, n.º 1–12, págs. 9-26, dic. de 2020. dirección: https://www.dgan.go.cr/ran/index.php/RAN/article/view/481.

[20] C. B. Ruesta, "La trasformación digital y las normas internacionales para gestionar documentos Digital transformation and international standard on records management," *Revista TRIA, ISSN 1134-1602, Nro. 21, 2017, págs. 17–29*, págs. 17-29, 21 2017, [En línea]. Disponible: https://dialnet.unirioja.es/servlet/articulo?codigo=6902035, ISSN: 1134-1602.

[21] C. Findlay, *Updated for the Digital Age ISO 15489*, [En línea]. Disponible: https://magazine.arma.org/wp-content/uploads/simple-file-list/2017_03_IM_ISO15489_updated_findlay.pdf, mayo de 2017.

[22] L. Guzmán y R. Lafuente, *Gestión Documental Electrónica: Retos y desafíos para un nuevo aprendizaje*, [En línea]. Disponible: https://www.adapting.com/wp-content/uploads/2018/11/22_compendio_retos_era_transf_digital_latam.pdf, 2016.

[23] N. A. M. Azim, S. F. M. Yatin, R. C. A. Jensonray y S. A. Mansor, "Digitization of Records and Archives: Issues and Concerns," *International Journal of Academic Research in Business and Social Sciences*, vol. 8, 9 sep. de 2018, [En línea]. Disponible: https://hrmars.com/IJARBSS/article/view/4582/Digitization-of-Records-and-Archives-Issues-and-Concerns. DOI: 10.6007/IJARBSS/V8-I9/4582.

[24] T. Bogatova, *Kompanii Parus i BIG vooruzhayut finansistov*, oct. de 1999. visitado 17 de ago. de 2025. dirección: https://www.itweek.ru/idea/article/detail.php?ID=52387.

[25] R. N. Ali, V. H. Abdullayev y V. S. Abbasova, "Analysis of Main Requirements for Electronic Document Management Systems," *ScienceRise*, vol. 1, págs. 28-31, 1 feb. de

2020, [En línea]. Disponible: https://journal.eu-jr.eu/sciencerise/article/view/1373, ISSN: 2313-8416. DOI: 10.21303/SR.V0I1.1148.

[26] P. E. Vyacheslavovich, "Análisis de los sistemas de gestión documental electrónica en los órganos de gobierno municipal," *Ciencias de la computación y la información*, vol. 133, 8 2012, ISSN: 1999-9429. dirección: https://cyberleninka.ru/article/n/analiz-sistem-elektronnogo-dokumentooborota-v-organah-munitsipalnogo-upravleniya.

[27] J. R. Craft, "Portable Document Format (PDF): Standardizing document files for publication," *DESIDOC Bulletin of Information Technology*, vol. 23, págs. 19-24, 1 ene. de 2003, [En línea]. Disponible: https://publicationsdrdo.in/index.php/djlit/article/view/3587, ISSN: 09714383. DOI: 10.14429/DBIT.23.1.3587.

[28] W.-L. Cheng, "Portable Document Format (PDF)—Finally, a Universal Document Exchange Technology," *Journal of Technology Studies*, vol. 28, págs. 59-63, 1 ene. de 2002, [En línea]. Disponible: https://jotsjournal.org/articles/10.21061/jots.v28i1.a.9, ISSN: 1071-6084. DOI: 10.21061/JOTS.V28I1.A.9.

[29] Adobe, *¿Qué son los archivos PS y cómo se abren?* s. f. visitado 17 de ago. de 2025. dirección: https://www.adobe.com/mx/creativecloud/file-types/image/vector/ps-file.html.

[30] N. Saydametov, "Segmentation of scanned PDF documents," Bachelor's thesis, Charles University, Faculty of Mathematics y Physics, jun. de 2024. dirección: http://hdl.handle.net/20.500.11956/192087.

[31] M. R. Turró, "Are PDF Documents Accessible?" *Information Technology and Libraries*, vol. 27, págs. 25-43, 3 sep. de 2008, [En línea]. Disponible: https://ital.corejournals.org/index.php/ital/article/view/3246, ISSN: 2163-5226. DOI: 10.6017/ITAL.V27I3.3246.

[32] D. Gera y N. Jain, "Comparison of Text Extraction Techniques- A Review," *International Journal of Innovative Research in Computer and Communication Engineering*,

vol. 3, n.º 2, págs. 621-626, feb. de 2015, ISSN: 2320-9801. DOI: 10.15680/IJIRCCE.2015.0302003. dirección: https://doi.org/10.15680/IJIRCCE.2015.0302003.

[33] D. Ghai y N. Jain, "Comparison of Various Text Extraction Techniques for Images-A Review," *International Journal of Graphics and Image Processing*, vol. 3, n.º 3, págs. 210-218, 2013.

[34] G. Chen y Q. Yuan, "Application and existing problems of computer network technology in the field of artificial intelligence," *Proceedings - 2021 2nd International Conference on Artificial Intelligence and Computer Engineering, ICAICE 2021*, págs. 139-142, 2021, [En línea]. Disponible: https://ieeexplore.ieee.org/document/9797546. DOI: 10.1109/ICAICE54393.2021.00035.

[35] M. Haenlein y A. Kaplan, "A brief history of artificial intelligence: On the past, present, and future of artificial intelligence," *California Management Review*, vol. 61, págs. 5-14, 4 ago. de 2019, [En línea]. Disponible: https://journals.sagepub.com/doi/abs/10.1177/0008125619864925, ISSN: 21628564. DOI: 10.1177/0008125619864925.

[36] E. Yulianto, T. Murdianto y A.-A. Al-Amin, "The Role of Artificial Intelligence (AI) in Records and Document Management," *COSMOS: Jurnal Ilmu Pendidikan, Ekonomi dan Teknologi*, vol. 1, págs. 484-499, 6 sep. de 2024, [En línea]. Disponible: https://cosmos.iaisambas.ac.id/index.php/cms/article/view/172, ISSN: 3046-4846. DOI: 10.37567/COSMOS.V1I5.172.

[37] S. Atmaja, "Pemanfaatan Artificial Intelligence (AI) Dalam Transformasi Digital Untuk Pelayanan Publik," *Jurnal Manajemen dan Bisnis*, vol. 6, págs. 9-21, 1 mar. de 2024, [En línea]. Disponible: https://ejournal.lppm-unbaja.ac.id/index.php/jmb/article/view/3233, ISSN: 2686-5939. DOI: 10.47080/JMB.V6I1.3233.

[38] S. Dzulhasni, D. Zakia, E. Y. Puspitasari, L. Rini y P. Wijaya, "Implikasi Etika pada Penggunaan Artificial Intelligence (AI) dalam Akuntansi Manajemen," *Jurnal Akuntansi, Manajemen, Bisnis dan Teknologi*, vol. 4, págs. 136-143, 1 feb. de 2024, [En línea]. Disponible: https://jurnalambitek.stie-mahaputra-riau.ac.id/

ojs/index.php/ambitek/article/view/136, ISSN: 2962-2042. DOI: 10.56870/AMBITEK.V4I1.136.

[39] A. T. Amalia, P. A. Perkantoran y F. E. dan Bisnis, "Sistem Informasi Manajemen Arsip Elektronik (E-Arsip) Berbasis Microsoft Access Terhadap Efektivitas Penemuan Kembali Arsip Pada SMKN 4 Surabaya," *Jurnal Pendidikan Administrasi Perkantoran (JPAP)*, vol. 10, págs. 195-210, 3 dic. de 2022, [En línea]. Disponible: https://journal.unesa.ac.id/index.php/jpap/article/view/19064/8869, ISSN: 2338-9621. DOI: 10.26740/JPAP.V10N3.P195-210.

[40] N. Sharma, R. Sharma y N. Jindal, "Machine Learning and Deep Learning Applications-A Vision," *Global Transitions Proceedings*, vol. 2, págs. 24-28, 1 jun. de 2021, [En línea]. Disponible: https://www.sciencedirect.com/science/article/pii/S2666285X21000042, ISSN: 2666-285X. DOI: 10.1016/J.GLTP.2021.01.004.

[41] P. P. Shinde y S. Shah, "A Review of Machine Learning and Deep Learning Applications," *Proceedings - 2018 4th International Conference on Computing, Communication Control and Automation, ICCUBEA 2018*, jul. de 2018, [En línea]. Disponible: https://ieeexplore.ieee.org/document/8697857. DOI: 10.1109/ICCUBEA.2018.8697857.

[42] J. Ho y M. Liu, "Research on Document Image Binarization: A Survey," *2024 IEEE 7th International Conference on Electronic Information and Communication Technology, ICEICT 2024*, págs. 457-462, 2024, [En línea]. Disponible: https://ieeexplore.ieee.org/document/9617487. DOI: 10.1109/ICEICT61637.2024.10671186.

[43] E. R. Villarreal, J. Garcia-Alonso, E. Moguel y J. A. H. Alegria, "Blockchain for Healthcare Management Systems: A Survey on Interoperability and Security," *IEEE Access*, vol. 11, págs. 5629-5652, 2023, [En línea]. Disponible: https://ieeexplore.ieee.org/document/10015729, ISSN: 21693536. DOI: 10.1109/ACCESS.2023.3236505.

[44] S. Nuanmeesri, "Mobile Application Development of Managing Elderly Household Accounts Using Speech Recognition," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 14, págs. 84-100, 02 feb. de 2020, [En línea]. Disponible: https:

//online-journals.org/index.php/i-jim/article/view/11651, ISSN: 1865-7923. DOI: 10.3991/IJIM.V14I02.11651.

[45] J. F. C. Cal y J. M. Torres, "La inteligencia artificial en la gestión del Patrimonio Documental," *RUIDERAe: Revista de Unidades de Información, ISSN-e 2254-7177, Nro. 23, 2024 (Ejemplar dedicado a: Inteligencia Artificial)*, pág. 2, 23 2024, [En línea]. Disponible: https://dialnet.unirioja.es/servlet/articulo?codigo=9892682, ISSN: 2254-7177.

[46] S. R. Hamid et al., "The Development of Smart Document Management System With Mobile Application Technology In Agricul-Tural Sector (Malaysia Sustainability Palm Oil)," *Journal of Technology Management and Technopreneurship (JTMT)*, vol. 11, págs. 28-43, 1 2023, [En línea]. Disponible: https://jtmt.utem.edu.my/jtmt/article/view/6067, ISSN: 2231-7996.

[47] M. Sambetbayeva, I. Kuspanova, A. Yerimbetova, S. Serikbayeva y S. Bauyrzhanova, "Development of intelligent electronic document management system model based on machine learning methods," *Eastern-European Journal of Enterprise Technologies*, vol. 1, págs. 68-76, 2(115) feb. de 2022, [En línea]. Disponible: https://journals.uran.ua/eejet/article/view/251689, ISSN: 1729-4061. DOI: 10.15587/1729-4061.2022.251689.

[48] H. M. Cano, "Sistema informático para la digitalización del expediente académico del archivo histórico de la secretaría docente.," *Serie Científica de la Universidad de las Ciencias Informáticas*, vol. 14, págs. 57-74, 9 ago. de 2021, [En línea]. Disponible: https://publicaciones.uci.cu/index.php/serie/article/view/933, ISSN: 2306-2495.

[49] J. de Dios Murillo Morera y S. C. Polini, "Implementación de un servidor FTP utilizando el modelo cliente/servidor mediante el uso de sockets en lenguaje c UNIX con el fin de mejorar los tiempos de respuesta en la red," *Uniciencia*, vol. 24, págs. 83-89, 1 2010, ISSN: 2215-3470. dirección: https://dialnet.unirioja.es/servlet/articulo?

codigo=5381353&info=resumen&idioma=SPA%20https://dialnet.unirioja.es/servlet/articulo?codigo=5381353.

[50] B. Mahesh, "Machine Learning Algorithms - A Review," *International Journal of Science and Research (IJSR)*, vol. 9, págs. 381-386, 1 ene. de 2020, [En línea]. Disponible: https://www.ijsr.net/getabstract.php?paperid=ART20203995. DOI: 10.21275/ART20203995.

[51] S. U. Hassan, J. Ahamed y K. Ahmad, "Analytics of machine learning-based algorithms for text classification," *Sustainable Operations and Computers*, vol. 3, págs. 238-248, ene. de 2022, ISSN: 2666-4127. DOI: 10.1016/J.SUSOC.2022.03.001. dirección: https://www.sciencedirect.com/science/article/pii/S2666412722000101?via%3Dihub.

[52] A. R. Murthy y A. K. M. Kumar, "A Review of Different Approaches for Detecting Emotion from Text," *IOP Conference Series: Materials Science and Engineering*, vol. 1110, pág. 012 009, 1 mar. de 2021, ISSN: 1757-899X. DOI: 10.1088/1757-899X/1110/1/012009. dirección: https://iopscience.iop.org/article/10.1088/1757-899X/1110/1/012009.

[53] C. Janiesch, P. Zschech y K. Heinrich, "Machine learning and deep learning," *Electronic Markets*, vol. 31, págs. 685-695, 3 sep. de 2021, ISSN: 14228890. DOI: 10.1007/S12525-021-00475-2. dirección: https://link.springer.com/article/10.1007/s12525-021-00475-2.

[54] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer Science, 2006, págs. 1-758, [En línea]. Disponible: https://www.microsoft.com/en-us/research/wp-content/uploads/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf.

[55] D. Silver et al., "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," *Science*, vol. 362, págs. 1140-1144, 6419 dic. de 2018, [En línea]. Disponible: https://www.science.org/doi/10.1126/science.aar6404, ISSN: 10959203. DOI: 10.1126/science.aar6404.

[56] M. Peters, W. Ketter, M. Saar-Tsechansky y J. Collins, “A reinforcement learning approach to autonomous decision-making in smart electricity markets,” *Machine Learning*, vol. 92, págs. 5-39, 1 jul. de 2013, [En línea]. Disponible: https://link.springer.com/article/10.1007/s10994-013-5340-0, ISSN: 08856125. DOI: 10.1007/s10994-013-5340-0.

[57] S. Das y D. Das, “Natural Language Processing (NLP) Techniques: Usability in Human-Computer Interactions,” *2024 6th International Conference on Natural Language Processing, ICNLP 2024*, págs. 783-787, 2024, [En línea]. Disponible: https://ieeexplore.ieee.org/document/10692776. DOI: 10.1109/ICNLP60986.2024.10692776.

[58] D. H. Maulud et al., “Review on Natural Language Processing Based on Different Techniques,” *Asian Journal of Research in Computer Science*, págs. 1-17, jun. de 2021, ISSN: 2581-8260. DOI: 10.9734/AJRCOS/2021/V10I130231. dirección: https://journalajrcos.com/index.php/AJRCOS/article/view/184.

[59] Z. Taskin y U. Al, “Natural language processing applications in library and information science,” *Online Information Review*, vol. 43, págs. 676-690, 4 jul. de 2019, [En línea]. Disponible: https://www.emerald.com/insight/content/doi/10.1108/oir-07-2018-0217/full/html, ISSN: 14684527. DOI: 10.1108/OIR-07-2018-0217.

[60] W. H. L. Pinaya et al., *Generative AI for Medical Imaging: extending the MONAI Framework*, jul. de 2023. DOI: 10.48550/arXiv.2307.15208. arXiv: 2307.15208. visitado 17 de mayo de 2025. dirección: https://arxiv.org/abs/2307.15208.

[61] S. Feuerriegel, J. Hartmann, C. Janiesch y P. Zschech, “Generative AI,” *Business and Information Systems Engineering*, vol. 66, págs. 111-126, 1 feb. de 2024, [En línea]. Disponible: https://link.springer.com/article/10.1007/s12599-023-00834-7, ISSN: 18670202. DOI: 10.1007/S12599-023-00834-7.

[62] H. S. Sætra, “Generative AI: Here to stay, but for good?” *Technology in Society*, vol. 75, pág. 102 372, nov. de 2023, [En línea]. Disponible: https://www.sciencedirect.com/science/article/pii/S0160791X2300177X, ISSN: 0160-791X. DOI: 10.1016/j.techsoc.2023.102372.

[63] Midjourney. “Midjourney,” visitado 10 de jun. de 2025. dirección: https://www.midjourney.com/home.

[64] A. García Martín, “Grandes modelos de lenguaje en sistemas de recomendación,” [En línea]. Disponible: https://gredos.usal.es/handle/10366/164903, Tesis de mtría., Universidad de Salamanca, 2024.

[65] A. Cambon et al., *Early LLM-based Tools for Enterprise Information Workers Likely Provide Meaningful Boosts to Productivity*, dic. de 2023. dirección: https://www.microsoft.com/en-us/research/publication/early-llm-based-tools-for-enterprise-information-workers-likely-provide-meaningful-boosts-to-productivity/.

[66] H. Naveed et al., “A Comprehensive Overview of Large Language Models,” *International Journal For Multidisciplinary Research*, vol. 7, 1 jul. de 2023. DOI: 10.36948/ijfmr.2025.v07i01.34609. dirección: https://arxiv.org/pdf/2307.06435.

[67] A. Rochan, P. Sowmya y D. A. J. Daniel, “Large Language Model Based Document Query Solution Using Vector Databases,” *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems, ADICS 2024*, 2024, [En línea]. Disponible: https://ieeexplore.ieee.org/document/10533636. DOI: 10.1109/ADICS58448.2024.10533636.

[68] V. N. Patil et al., “Integrating AI-Powered Chatbots and PDF Processing for Enhanced Document Management: A Full-Stack AI SaaS Approach,” *Library Progress International*, vol. 44, págs. 9982-9992, 3 oct. de 2024, [En línea]. Disponible: https://bpasjournals.com/library-science/index.php/journal/article/view/2218, ISSN: 2320-317X. DOI: 10.48165/BAPAS.2024.44.2.1.

[69] M. Hansen, A. Pomp, K. Erki y T. Meisen, “Data-Driven Recognition and Extraction of PDF Document Elements,” *Technologies*, vol. 7, n.º 3, pág. 65, 2019, ISSN: 2227-7080. DOI: 10.3390/technologies7030065. dirección: https://www.mdpi.com/2227-7080/7/3/65.

[70] J. Hurwitz, A. Nugent, F. Halper y M. Kaufman, *Big Data For Dummies® - Big Data For Dummies.pdf*. Wiley, 2013, pág. 472, ISBN: 1118644174. dirección: https://books.google.de/books?id=XPkAEFXo7VgC&printsec=frontcover&dq=Big+Data+for+Dummies&hl=de&sa=X&ved=0ahUKEwifg8zG5M_XAhUIJMAKHUowCRkQ6AEIKjAA#v=onepage&q=Big%20Data%20for%20Dummies&f=false%20www.it-ebooks.info.

[71] R. Mithe, S. Indalkar y N. Divekar, "Optical Character Recognition," *Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)*, vol. 2, págs. 1-4, 2013, [En línea]. Disponible: https://www.ijrte.org/portfolio-item/A0504032113/.

[72] K. Alkhalaf, "OCR-Based Electronic Documentation Management System," *International Journal of Innovation, Management and Technology*, vol. 5, 6 2014, [En línea]. Disponible: https://www.ijimt.org/index.php?m=content&c=index&a=show&catid=63&id=866, ISSN: 20100248. DOI: 10.7763/IJIMT.2014.V5.560.

[73] P. K. Charles, V. Harish, M. Swathi y C. H. Deepthi, "A Review on the Various Techniques used for Optical Character Recognition," *International Journal of Engineering Research and Applications*, vol. 2, n.º 1, págs. 659-662, 2012, ISSN: 2248-9622. visitado 17 de abr. de 2025. dirección: https://www.ijera.com/papers/Vol2_issue1/DB21659662.pdf.

[74] W. Khallouli, R. Pamie-George, S. Kovacic, A. Sousa-Poza, M. Canan y J. Li, "Leveraging Transfer Learning and GAN Models for OCR from Engineering Documents," *2022 IEEE World AI IoT Congress, AIIoT 2022*, págs. 15-21, 2022, [En línea]. Disponible: https://ieeexplore.ieee.org/document/9817319. DOI: 10.1109/AIIOT54504.2022.9817319.

[75] ABBYY, *What is AI OCR: How it differs from Traditional OCR*, s. f. visitado 17 de ago. de 2025. dirección: https://www.abbyy.com/blog/ai-ocr/.

[76] T. M. Breuel, "The OCRopus open source OCR system," en *Document Recognition and Retrieval XV*, ép. Proceedings of SPIE, vol. 6815, SPIE, ene. de 2008, 68150F. DOI: 10.1117/12.783598. dirección: https://doi.org/10.1117/12.783598.

[77] C. Wick, C. Reul y F. Puppe, “Calamari - A High-Performance Tensorflow-based Deep Learning Package for Optical Character Recognition,” *Digital Humanities Quarterly*, vol. 14, 2 jul. de 2018, ISSN: 19384122. dirección: https://arxiv.org/pdf/1807.02004.

[78] B. Shi, X. Bai y C. Yao, “An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, págs. 2298-2304, 11 jul. de 2015, ISSN: 01628828. DOI: 10.1109/TPAMI.2016.2646371. dirección: https://arxiv.org/pdf/1507.05717.

[79] Google Cloud, *OCR con Google AI | Google Cloud*, s. f. visitado 17 de ago. de 2025. dirección: https://cloud.google.com/use-cases/ocr?hl=es.

[80] Q. Wang et al., “Learning-based Artificial Intelligence Artwork: Methodology Taxonomy and Quality Evaluation,” *ACM Comput. Surv.*, vol. 57, n.º 3, nov. de 2024, ISSN: 0360-0300. DOI: 10.1145/3698105. dirección: https://doi.org/10.1145/3698105.

[81] M. P. Geetha, G. Thirukumaran, C. Pradakshana, B. Sudharsana y T. Ashwin, “Conversational AI Meets Documents Revolutionizing PDF Interaction with GenAI,” *2nd International Conference on Emerging Research in Computational Science, ICERCS 2024*, 2024, [En línea]. Disponible: https://ieeexplore.ieee.org/document/10895303. DOI: 10.1109/ICERCS63125.2024.10895303.

[82] A. Singh, K. Bacchuwar y A. Bhasin, “A Survey of OCR Applications,” *International Journal of Machine Learning and Computing*, págs. 314-318, 2012, [En línea]. Disponible: https://www.ijml.org/show-31-230-1.html, ISSN: 20103700. DOI: 10.7763/IJMLC.2012.V2.137.

[83] J. Huang, H. Chen, F. Yu y W. Lu, “From Detection to Application: Recent Advances in Understanding Scientific Tables and Figures,” *ACM Comput. Surv.*, vol. 56, n.º 10, jun. de 2024, ISSN: 0360-0300. DOI: 10.1145/3657285. dirección: https://doi.org/10.1145/3657285.

[84] G. Medisetti, Z. Compson, H. Fan, H. Yang e Y. Feng, "LitAI: Enhancing Multimodal Literature Understanding and Mining with Generative AI," *Proceedings of the International Conference on Multimedia Information Processing and Retrieval, MIPR*, págs. 471-476, 2024, [En línea]. Disponible: https://ieeexplore.ieee.org/abstract/document/10707959, ISSN: 27704319. DOI: 10.1109/MIPR62202.2024.00080.

[85] M. Olek, *About Evaluation of F1 Score for RECENT Relation Extraction System*, mayo de 2023. DOI: 10.48550/arXiv.2305.09410. arXiv: 2305.09410 [cs.CL]. dirección: https://arxiv.org/abs/2305.09410.

[86] A. Gupta, A. Anand e Y. Hasija, "Recall-based Machine Learning approach for early detection of Cervical Cancer," *2021 6th International Conference for Convergence in Technology, I2CT 2021*, abr. de 2021, [En línea]. Disponible: https://ieeexplore.ieee.org/document/9418099. DOI: 10.1109/I2CT51068.2021.9418099.

[87] B. J. Erickson y F. Kitamura, "Magician's corner: 9. performance metrics for machine learning models," *Radiology: Artificial Intelligence*, vol. 3, 3 mayo de 2021, ISSN: 26386100. DOI: 10.1148/ryai.2021200126. dirección: https://pubs.rsna.org/doi/10.1148/ryai.2021200126.

[88] J. S. Akosa, "Predictive Accuracy: A Misleading Performance Measure for Highly Imbalanced Data," en *Proceedings of the SAS Global Forum 2017 Conference*, Paper 0942-2017, SAS Institute Inc., Cary, North Carolina, 2017, págs. 1-12. dirección: https://support.sas.com/resources/papers/proceedings17/0942-2017.pdf.

[89] M. C. H. Lee, J. Braet y J. Springael, "Performance Metrics for Multilabel Emotion Classification: Comparing Micro, Macro, and Weighted F1-Scores," *Applied Sciences (Switzerland)*, vol. 14, pág. 9863, 21 nov. de 2024, ISSN: 20763417. DOI: 10.3390/app14219863. dirección: https://www.mdpi.com/2076-3417/14/21/9863.

[90] Rev, *What Is Word Error Rate (WER)?* s. f. visitado 17 de mayo de 2025. dirección: https://www.rev.com/resources/what-is-wer-what-does-word-error-rate-mean.

[91] M. Abbas y D. Megtouf, “Towards a practical and efficient WER Calculation for Arabic ASR systems,” *2nd International Conference on Natural Language and Speech Processing, ICNLSP 2018*, págs. 1-3, jun. de 2018, [En línea]. Disponible: https://ieeexplore.ieee.org/document/8374365. DOI: 10.1109/ICNLSP.2018.8374365.

[92] Galileo, *Understanding Character Error Rate (CER) for AI Accuracy*, s. f. visitado 17 de jun. de 2025. dirección: https://galileo.ai/blog/character-error-rate-cer-metric.

[93] D. M. Buede y W. D. Miller, *The Engineering Design of Systems: Models and Methods*. Wiley Series in Systems Engineering y Management, 2016, págs. 1-30. dirección: https://content.e-bookshelf.de/media/reading/L-7550133-bf419bf362.pdf.

[94] H. Jaakkola y B. Thalheim, “Architecture-Driven Modelling Methodologies,” *Frontiers in Artificial Intelligence and Applications*, vol. 225, págs. 97-116, 2011, ISSN: 18798314. DOI: 10.3233/978-1-60750-690-4-97. dirección: https://ebooks.iospress.nl/doi/10.3233/978-1-60750-690-4-97.

[95] S. Park, J. yoon Lee y J. Lee, “AI system architecture design methodology based on IMO (Input-AI Model-Output) structure for successful AI adoption in organizations,” *Data & Knowledge Engineering*, vol. 150, págs. 102-264, mar. de 2024, ISSN: 0169-023X. DOI: 10.1016/J.DATAK.2023.102264. dirección: https://www.sciencedirect.com/science/article/pii/S0169023X23001246?ref=pdf_download&fr=RR-2&rr=96177ec22c968258.

[96] B. Sriman, S. H. A. Silviya, Y. Mouleesh, S. Vinod, S. Nishanthini y P. Nikitha, “Intelligent Document Interaction with Advanced Vector Embeddings and FAISS-CPU Indexing,” *8th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2024 - Proceedings*, 2024, [En línea]. Disponible: https://ieeexplore.ieee.org/document/10800948. DOI: 10.1109/ICECA63461.2024.10800948.

[97] Meta Engineering, *Faiss: A library for efficient similarity search*, mar. de 2017. visitado 17 de ago. de 2025. dirección: https://engineering.fb.com/2017/03/29/data-infrastructure/faiss-a-library-for-efficient-similarity-search/.

[98] R. Patil, A. Nikam, T. Lichade, R. Patel, D. Maheshwari y A. Rajguru, "Intelligent PDF Query System for Document Analysis," *2025 International Conference on Cognitive Computing in Engineering, Communications, Sciences and Biomedical Health Informatics (IC3ECSBHI)*, págs. 665-669, ene. de 2025, [En línea]. Disponible: https://ieeexplore.ieee.org/document/10991119. DOI: 10.1109/IC3ECSBHI63591.2025.10991119.

[99] Platzi, *Conceptos Fundamentales de Back End en Arquitectura Web*, s. f. visitado 17 de ago. de 2025. dirección: https://platzi.com/cursos/practico-backend/que-es-el-backend/.

[100] S. Graciela, P. Ibarra, R. Quispe, F. F. Mullicundo, D. A. Lamas y L. Presente, "Herramientas y tecnologías para el desarrollo web desde el FrontEnd al BackEnd," *XXIII Workshop de Investigadores en Ciencias de la Computación*, págs. 963-968, December 2021. dirección: http://sedici.unlp.edu.ar/handle/10915/120476.

[101] Python Software Foundation, *Welcome to Python.org*, s. f. visitado 10 de mayo de 2025. dirección: https://www.python.org/.

[102] T. Peters, *PEP 20 – The Zen of Python*, ago. de 2004. visitado 17 de ago. de 2025. dirección: https://peps.python.org/pep-0020/.

[103] R. Toal, S. Strieker y M. Berardini, *Programming Language Explorations*. Chapman y Hall/CRC, ago. de 2024, [En línea]. Disponible: https://www.taylorfrancis.com/books/mono/10.1201/9781003222354/programming-language-explorations-ray-toal-sage-strieker-marco-berardini, ISBN: 9781003222354. DOI: 10.1201/9781003222354.

[104] Django Software Foundation, *Writing your first Django app, part 5*, Documentación de Django, versión 6.0, s. f. visitado 19 de ago. de 2026. dirección: https://docs.djangoproject.com/en/6.0/intro/overview/.

[105] R. Espinosa-Hurtado, "Análisis comparativo para la evaluación de frameworks usados en el desarrollo de aplicaciones web," *CEDAMAZ*, vol. 11, págs. 133-141, 2 dic. de 2021, ISSN: 1390-5880. DOI: 10.54753/cedamaz.v11i2.1182. dirección: https://revistas.unl.edu.ec/index.php/cedamaz/article/view/1182.

[106] R. Caldera Vergara, "Estudio del framework de desarrollo web Django," Trabajo de Fin de Grado, Universidad de Alcalá, Escuela Politécnica Superior, 2017. dirección: http://hdl.handle.net/10017/32018.

[107] K. Manikas, "Revisiting software ecosystems Research: A longitudinal literature study," *Journal of Systems and Software*, vol. 117, págs. 84-103, jul. de 2016, ISSN: 0164-1212. DOI: 10.1016/J.JSS.2016.02.003. dirección: https://www.sciencedirect.com/science/article/abs/pii/S0164121216000406.

[108] K. Manikas y K. M. Hansen, "Software ecosystems – A systematic literature review," *Journal of Systems and Software*, vol. 86, págs. 1294-1306, 5 mayo de 2013, ISSN: 0164-1212. DOI: 10.1016/J.JSS.2012.12.026. dirección: https://www.sciencedirect.com/science/article/abs/pii/S016412121200338X.

[109] J. H. Peralta, *Microservice APIs: Using Python, Flask, FastAPI, OpenAPI and more*. Manning, 2022, ISBN: 9781617298417. dirección: https://www.manning.com/books/microservice-apis.

[110] M. Copperwaite y C. Leifer, *Learning Flask Framework Build dynamic, data-driven websites and modern web applications with Flask Learning Flask Framework Credits*. Packt Publishing, 2015, ISBN: 978-1-78398-336-0. dirección: https://api.pageplace.de/preview/DT0400.9781783983377_A26214819/preview-9781783983377_A26214819.pdf.

[111] V. E. Lermanda Candia, "Sistema Graph RAG de Extracción de Información para la S.S.D.D.H.H de Chile.," Tesis de pregrado, Universidad de Concepción Facultad de Ingeniería Departamento de Ingeniería Informática Y Ciencias de La Computación, 2024. dirección: https://repositorio.udec.cl/handle/11594/3980.

[112] P. Huet, *Embeddings: Qué son y cómo transforman datos en información*, jul. de 2024. visitado 17 de ago. de 2025. dirección: https://openwebinars.net/blog/embeddings/.

[113] J. G. Delgado, *Diseño de interfaces para la web actual De la estructura de la interfaz a la usabilidad*. Esic, 2024. dirección: https://www.esic.edu/sites/default/files/2024-10/978-84-11920-96-4%20Dise%C3%B1o%20de%20interfaces%20para%20la%20web%20actual.pdf.

[114] R. Baeza-Yates y B. Ribeiro-Neto, *Modern information retrieval*. New York: ACM Press y Addison-Wesley, 1999, ISBN: 0-201-39829-X. dirección: https://web.cs.ucla.edu/~miodrag/cs259-security/baeza-yates99modern.pdf.

[115] H. Delgado y F. Uzcátegui, "Procesamiento de Lenguaje Natural para la Búsqueda Semántica en Repositorios Académicos," en *Memorias de la Novena Conferencia Nacional de Computación, Informática y Sistemas (CoNCISa 2022)*, Caracas, Venezuela: Universidad Central de Venezuela, nov. de 2022, págs. 20-23, ISBN: 9789807683074. dirección: https://concisa.net.ve/memorias/CoNCISa2022/CoNCISa2022-p020-023.pdf.

[116] H. Bast, B. Buchhold y E. Haussmann, "Semantic Search on Text and Knowledge Bases," *Foundations and Trends® in Information Retrieval*, vol. 10, págs. 119-271, 2-3 2016, ISSN: 15540677. DOI: 10.1561/1500000032. dirección: https://dl.acm.org/doi/10.1561/1500000032.

[117] L. Arhippainen y M. Tähti, "Empirical Evaluation of User Experience in two Adaptive Mobile Application Prototypes," en *Proceedings of the 2nd International Conference on Mobile and Ubiquitous Multimedia*, ép. Linköping Electronic Conference Proceedings, Linköping University Electronic Press, dic. de 2003, págs. 27-34. dirección: https://ep.liu.se/konferensartikel.aspx?series=&issue=11&Article_No=7.

[118] M. Hassenzahl y N. Tractinsky, "User experience - A research agenda," *Behaviour and Information Technology*, vol. 25, págs. 91-97, 2 mar. de 2006, ISSN: 0144929X. DOI: 10.1080/01449290500330331. dirección: https://www.tandfonline.com/doi/abs/10.1080/01449290500330331.

[119] M. R. P. Velastegui, *Análisis metodológico del diseño UX/UI en el sitio web e-commerce Almacenes Juan Eljuri.* Dic. de 2022. dirección: https://dspace.espoch.edu.ec/handle/123456789/20539.

[120] C. Larman, *UML y Patrones 2ED*. Prentice Hall, 2003, págs. 1-624, [En línea]. Disponible: https://catalogobiblioteca.puce.edu.ec/cgi-bin/koha/opac-detail.pl?biblionumber=269666, ISBN: 9788420534381.

[121] J. C. Díaz Ortega, *Selección de Frameworks para Agentes LLM: Un Análisis Comparativo*, Research article, Universidad de La Salle, jun. de 2025. dirección: https://www.researchgate.net/publication/392512266_Seleccion_de_Frameworks_para_Agentes_LLM_Un_Analisis_Comparativo.

[122] Streamlit, *Streamlit: A faster way to build and share data apps*, s. f. visitado 17 de sep. de 2025. dirección: https://streamlit.io/.

[123] G. García Espinosa, "Puesta en producción de modelos de aprendizaje automático y administración de su ciclo de vida (MLOps)," Tesis de maestría, Universidad Politécica de Cartagena, oct. de 2023. dirección: http://hdl.handle.net/10317/12847.

[124] M. Fenniak et al., *pypdf*, Version 6.16.1, ago. de 2026. visitado 14 de ago. de 2026. dirección: https://pypi.org/project/pypdf/.

[125] jsvine, *pdfplumber: Plumb a PDF for detailed information about each character, rectangle, line, and more*, s. f. visitado 17 de ago. de 2025. dirección: https://github.com/jsvine/pdfplumber.

[126] V. M. Peñasco Estívalez, "Pipeline for the extraction of structured text from Clinical Practice Guidelines in PDF format," [En línea]. Disponible: https://zaguan.unizar.es/record/96556/files/TAZ-TFG-2020-2189.pdf, Trabajo de Fin de Grado, Zaragoza, 2020.

[127] P. Jimbo Santana y M. Jimbo Santana, "Evaluación de sistemas de inteligencia artificial: métodos y prácticas en el contexto sociopolítico de américa latina y el caribe," *Journal of Science and Research*, vol. 10, n.º 2, págs. 85-96, abr. de 2025, ISSN: 2528-8083. dirección: https://revistas.utb.edu.ec/index.php/sr/article/view/3294.

[128] R. Zambrano y J. M. Sanchez-Torres, “AI Public Policies in Latin America: Disruption or more of the same?” *ACM International Conference Proceeding Series*, págs. 25-33, oct. de 2021. DOI: 10.1145/3494193.3494294. dirección: https://doi.org/10.1145/3494193.3494294.

[129] G. R. Y. Naik, B. Shashidhara y G. K. Amith, “A Review On Text Extraction Techniques For Degraded Historical Document Images,” *2nd IEEE International Conference on Advances in Information Technology, ICAIT 2024 - Proceedings*, 2024, [En línea]. Disponible: https://ieeexplore.ieee.org/abstract/document/10690761. DOI: 10.1109/ICAIT61638.2024.10690761.

[130] D. Ghai y N. Jain, “Text Extraction from Document Images-A Review,” *International Journal of Computer Applications*, vol. 84, págs. 975-8887, 3 2013, [En línea]. Disponible: https://research.ijcaonline.org/volume84/number3/pxc3892661.pdf. DOI: 10.5120/14559-2661.

[131] P. M. Manwatkar y K. R. Singh, “A technical review on text recognition from images,” *Proceedings of 2015 IEEE 9th International Conference on Intelligent Systems and Control, ISCO 2015*, sep. de 2015, [En línea]. Disponible: https://https://ieeexplore.ieee.org/document/7282362. DOI: 10.1109/ISCO.2015.7282362.

[132] Y. Jin y Z. Yang, “Scalability Optimization in Cloud-Based AI Inference Services: Strategies for Real-Time Load Balancing and Automated Scaling,” *Proceedings of 2025 4th International Conference on Big Data, Information and Computer Network, BDICN 2025*, vol. 1, págs. 266-270, abr. de 2025. DOI: 10.1145/3727353.3727398. dirección: https://arxiv.org/pdf/2504.15296.

[133] N. Mungoli, *Scalable, Distributed AI Frameworks: Leveraging Cloud Computing for Enhanced Deep Learning Performance and Efficiency*, abr. de 2023. DOI: 10.48550/arXiv.2304.13738. arXiv: 2304.13738. dirección: https://arxiv.org/abs/2304.13738.

[134] J. Colantonio, *What is Response Time Testing (Performance)*, mar. de 2024. visitado 17 de ago. de 2025. dirección: https://testguild.com/performance-testing-response-time/.

[135] R. Kumari, *Response Time Testing – What it is & How to Measure it?* Nov. de 2023. visitado 17 de ago. de 2025. dirección: https://testsigma.com/blog/response-time-testing/.

[136] A. Q. Khan, *Response Time Testing in Software Testing*, ago. de 2026. visitado 7 de ago. de 2026. dirección: https://www.browserstack.com/guide/response-time-test.

[137] J. Alonso-Arévalo y S. Martín Cerro, "Benchmarking : una herramienta para gestionar la excelencia en las bibliotecas y los servicios de información," en *III Jornadas de Bibliotecas Universitarias de Castilla y León*, Burgos, España, nov. de 2000. dirección: http://eprints.rclis.org/4963/.

[138] APQC, *Helping Organizations Work Smarter, Faster, and with Greater Confidence | APQC*, s. f. visitado 17 de mayo de 2025. dirección: https://www.apqc.org/.

[139] J. Alberto y A. Guzmán, "El Benchmarking como proceso de mejora continua," *Contribuciones a la Economía*, vol. 13, pág. 28, 1 2015, ISSN: 1696-8360. dirección: https://dialnet.unirioja.es/servlet/articulo?codigo=9163393&info=resumen&idioma=SPA.

[140] F. Del Giorgio Solfa, *Benchmarking en el sector público: Aportes y propuestas de implementación para la provincia de Buenos Aires*. Edición del autor, mayo de 2012, ISBN: 978-987-33-2236-5. dirección: http://sedici.unlp.edu.ar/handle/10915/25886.

[141] S. S. Pirela y D. Romero, "Benchmarking como estrategia gerencial en las organizaciones," *Revista Ethos venezolana*, vol. 7, págs. 73-82, 1 2015, ISSN: 1856-9862.

[142] S. M. Velásquez, J. D. Vahos Montoya, M. E. Gómez Adasme, E. J. Restrepo Zapata, A. A. Pino y S. Londoño Marín, "Una revisión comparativa de la literatura acerca de metodologías tradicionales y modernas de desarrollo de software," *Revista CINTEX*,

vol. 24, n.º 2, págs. 13-23, dic. de 2019, ISSN: 0122-350X. DOI: 10.33131/24222208.334. dirección: https://revistas.pascualbravo.edu.co/index.php/cintex/article/view/334.

[143] L. D. La, C. D. Olivera, L. Manuel y D. Alonso, "Modelos de Desarrollo de Software: Modelos de Desarrollo de Software," *Revista Cubana de Ciencias Informáticas*, vol. 15, 1 ene. de 2021, ISSN: 2227-1899. dirección: https://rcci.uci.cu/index.php/RCCI/article/view/1952.

[144] K. Satirapiwong y T. Siriborvornratanakul, "Information extraction for different layouts of invoice images," *The Imaging Science Journal*, vol. 69, págs. 417-429, 5-8 nov. de 2021, ISSN: 1743131X. DOI: 10.1080/13682199.2022.2157367. dirección: https://www.tandfonline.com/doi/abs/10.1080/13682199.2022.2157367.

[145] A. Tsimpiris, D. Varsamis y G. Pavlidis, "Tesseract OCR evaluation on Greek food menus datasets," *International Journal of Computing and Optimization*, vol. 9, págs. 13-32, 1 2022. DOI: 10.12988/IJCO.2022.9829. dirección: https://www.m-hikari.com/ijco/ijco2022/ijco1-2022/9829.html.

[146] C. Neudecker, K. Baierer, M. Gerber, C. Christian, A. Apostolos y P. Stefan, "A survey of OCR evaluation tools and metrics," *ACM International Conference Proceeding Series*, págs. 13-18, sep. de 2021. DOI: 10.1145/3476887.3476888. dirección: https://dl.acm.org/doi/10.1145/3476887.3476888.

[147] A. NithyaKalyani y S. Jothilakshmi, "Chapter 7 - Speech Summarization for Tamil Language," en *Intelligent Speech Signal Processing*, N. Dey, ed., Academic Press, 2019, págs. 113-138, ISBN: 978-0-12-818130-0. DOI: 10.1016/B978-0-12-818130-0.00007-6. dirección: https://www.sciencedirect.com/science/article/pii/B9780128181300000076.

[148] OpenAI et al., *GPT-4 Technical Report*, 2024. arXiv: 2303.08774 [cs.CL]. dirección: https://arxiv.org/abs/2303.08774.

[149] Anthropic, *Introducing the next generation of Claude*, mar. de 2024. visitado 17 de ago. de 2025. dirección: https://www.anthropic.com/news/claude-3-family.

[150] A. Reuel, A. Hardy, C. Smith, M. Lamparth, M. Hardy y M. J. Kochenderfer, "BetterBench: Assessing AI Benchmarks, Uncovering Issues, and Establishing Best Practices," *Advances in Neural Information Processing Systems*, vol. 37, nov. de 2024, ISSN: 10495258. DOI: 10.52202/079017-0685. dirección: https://arxiv.org/pdf/2411.12990.

[151] M. Gao, J. Ruan, R. Sun, X. Yin, S. Yang y X. Wan, *Human-like Summarization Evaluation with ChatGPT*, abr. de 2023. DOI: 10.48550/arXiv.2304.02554. arXiv: 2304.02554. dirección: https://arxiv.org/abs/2304.02554.

[152] M. Song, "Defining the problem: The impact of OCR quality on retrieval-augmented generation performance and strategies for improvement," *Information Processing & Management*, vol. 63, pág. 104 368, 1 ene. de 2026, ISSN: 0306-4573. DOI: 10.1016/J.IPM.2025.104368. dirección: https://www.sciencedirect.com/science/article/pii/S0306457325003097.

[153] T. Hodel, D. Schoch, C. Schneider y J. Purcell, "General Models for Handwritten Text Recognition: Feasibility and State-of-the Art. German Kurrent as an Example," *Journal of Open Humanities Data*, vol. 7, jul. de 2021. DOI: 10.5334/JOHD.46. dirección: https://openhumanitiesdata.metajnl.com/articles/10.5334/johd.46.

[154] G. Muehlberger et al., "Transforming scholarship in the archives through handwritten text recognitionTranskribus as a case study," *Journal of Documentation*, vol. 75, págs. 954-976, 5 sep. de 2019, ISSN: 0022-0418. DOI: 10.1108/JD-07-2018-0114. dirección: https://dx.doi.org/10.1108/JD-07-2018-0114.

[155] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu y C. Zhu, "G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment," *EMNLP 2023 - 2023 Conference on Empirical Methods in Natural Language Processing, Proceedings*, págs. 2511-2522, mar. de 2023. DOI: 10.18653/v1/2023.emnlp-main.153. dirección: https://arxiv.org/pdf/2303.16634.

[156] I. D. Raji, E. M. Bender, A. Paullada, E. Denton y A. Hanna, "AI and the Everything in the Whole Wide World Benchmark," *Advances in Neural Information Processing*

*Systems*, nov. de 2021, ISSN: 10495258. dirección: https://arxiv.org/pdf/2111.15366.

# ANEXOS

Diagrama que muestra la estructura en capas del sistema, incluyendo la capa de presentación (frontend), capa de API (backend), capa de servicios y almacenamiento.

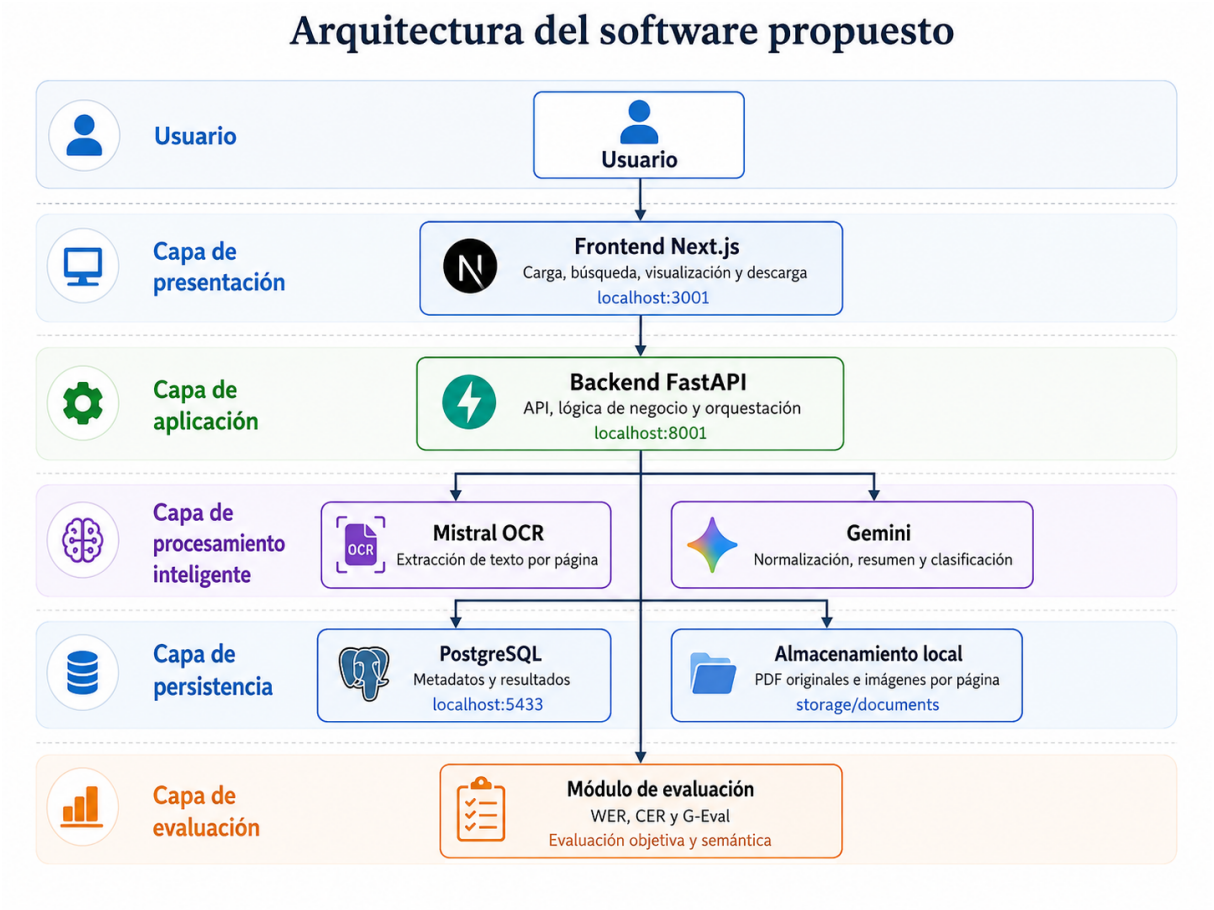


**Figura 28**. Arquitectura del sistema.

Organización jerárquica de carpetas y archivos del sistema, mostrando la distribución de componentes en el directorio principal New_project/.

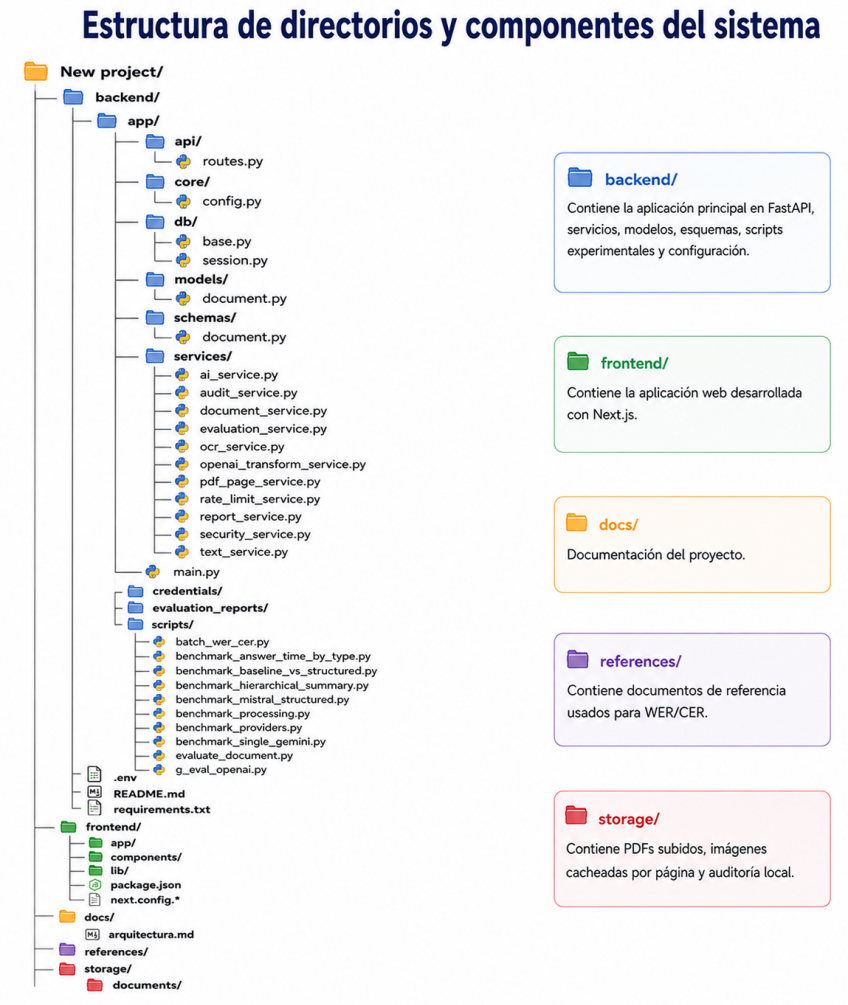


**Figura 29**. Estructura de Directorios del Proyecto.

Captura de pantalla de terminal que muestra la ejecución del frontend del sistema mediante *Next.js*. Se observa la inicialización exitosa del servidor de desarrollo, indicando que la interfaz web se encuentra disponible para el usuario. Desde esta capa se permite cargar documentos PDF, visualizar el texto extraído por páginas, consultar resúmenes generados por IA, buscar documentos, aplicar filtros por fecha y descargar reportes o archivos originales. La aplicación queda disponible en `http://localhost:3001`, desde donde se comunica con la API desarrollada en *FastAPI*.

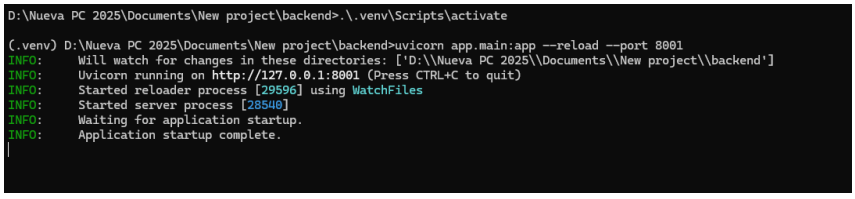


**Figura 30**. Inicialización del Sistema en Entorno Local.

Captura de pantalla de la página principal del sistema, mostrando el área de carga de documentos, y la visualización de resultados.




**Figura 31**. Interfaz de Usuario: Pantalla Principal.

Ventana de diálogo del sistema operativo utilizada para seleccionar el archivo PDF que será cargado al sistema. Esta ventana permite al usuario navegar entre las carpetas locales, elegir el documento correspondiente y enviarlo posteriormente a la aplicación para iniciar el proceso de validación, extracción OCR y análisis mediante inteligencia artificial generativa.

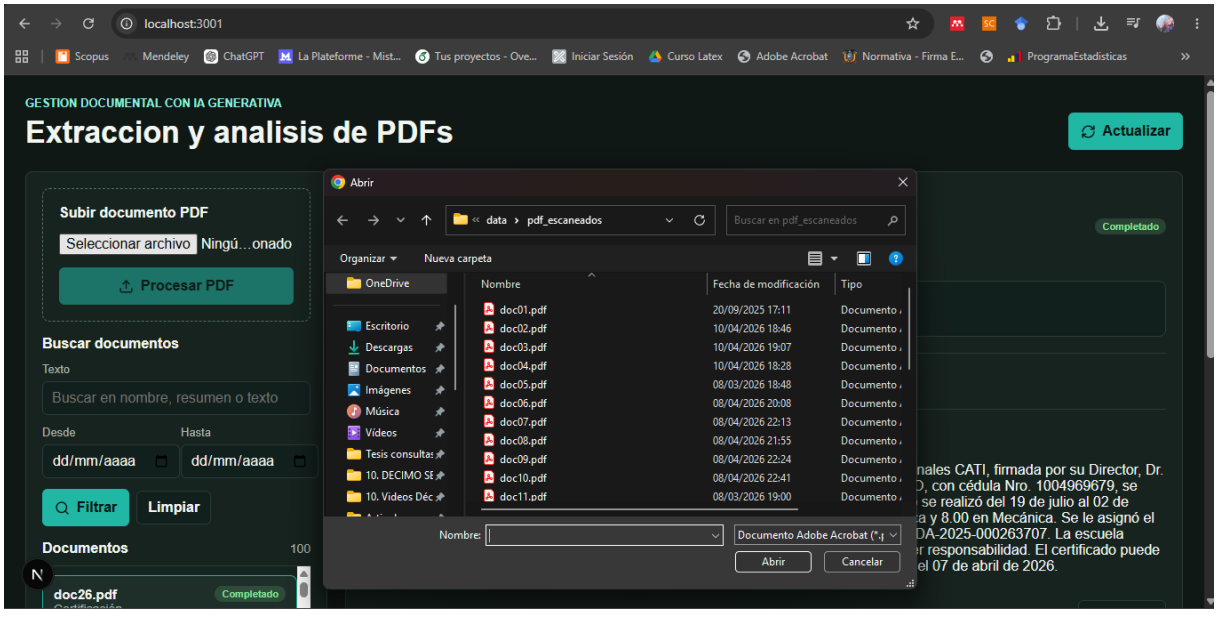


**Figura 32**. Ventana de carga de archivo.

Ventana modal que informa al usuario sobre el proceso de carga y análisis del documento PDF. Esta ventana se muestra mientras el sistema valida el archivo, lo almacena localmente e inicia el procesamiento mediante OCR e inteligencia artificial generativa. Su finalidad es brindar retroalimentación visual al usuario y evitar que realice acciones repetidas mientras el documento se encuentra en procesamiento.

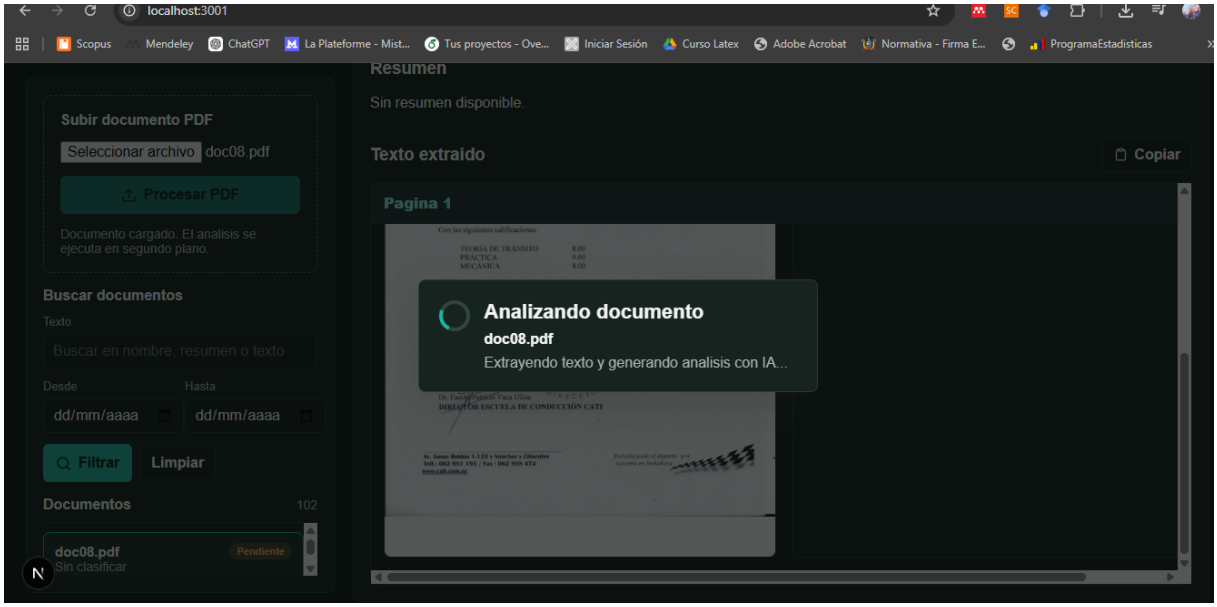


**Figura 33**. Ventana emergente de procesamiento.

Visualización del resumen generado por inteligencia artificial a partir del texto extraído del documento PDF. Esta sección permite al usuario revisar de forma sintetizada el contenido principal del archivo cargado, facilitando la comprensión rápida del documento sin necesidad de leerlo completo. El resumen es generado mediante Gemini a partir del texto procesado por el sistema.

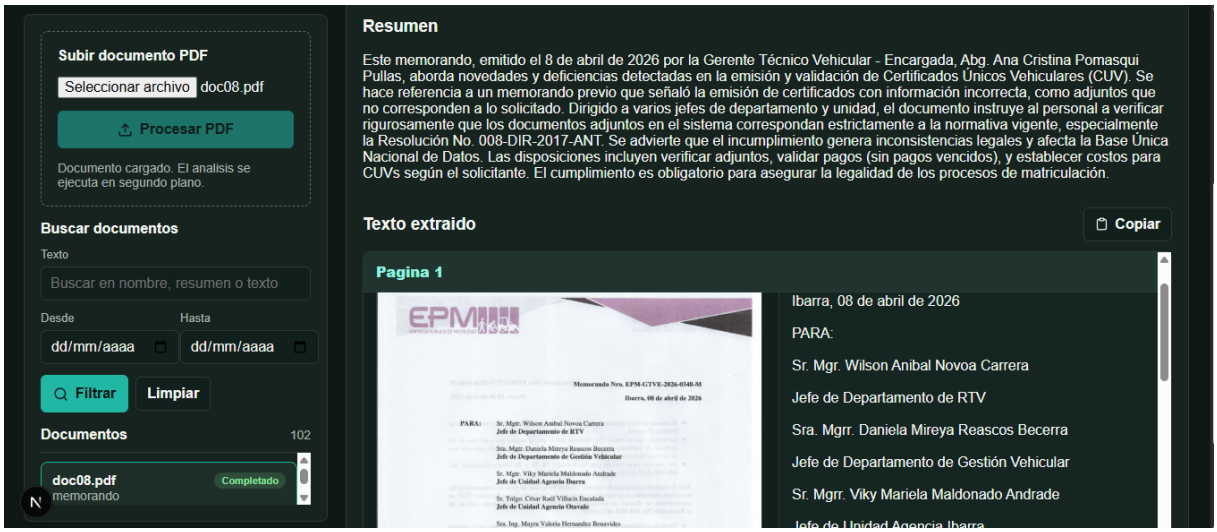


**Figura 34**. Visualización del resumen del documento procesado.

Visualización del texto extraído del documento PDF mediante OCR. La interfaz presenta una vista por páginas, donde en el panel izquierdo se muestra la imagen de la página original del PDF y en el panel derecho se despliega el texto procesado correspondiente a dicha página. Esta organización permite comparar visualmente el documento original con el texto generado por el sistema, facilitando la revisión, validación y análisis del contenido extraído.

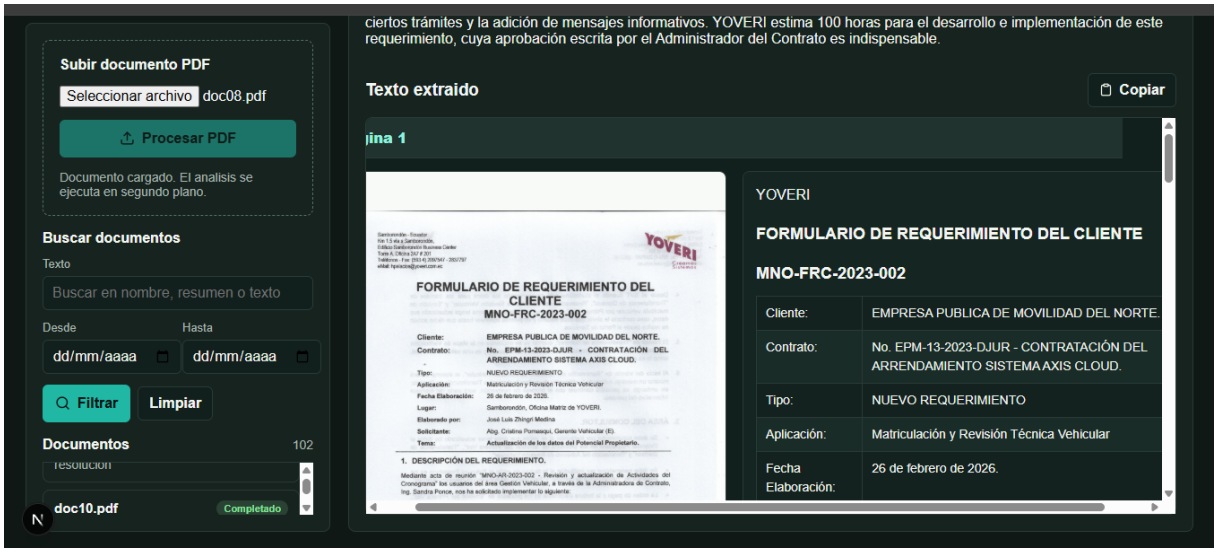


**Figura 35**. Interfaz de visualización del texto extraído.

Esta funcionalidad permite aprovechar el texto extraído mediante OCR para realizar búsquedas dentro de los documentos almacenados. De esta manera, el usuario no necesita abrir cada archivo manualmente, sino que puede localizar documentos relevantes mediante términos específicos.

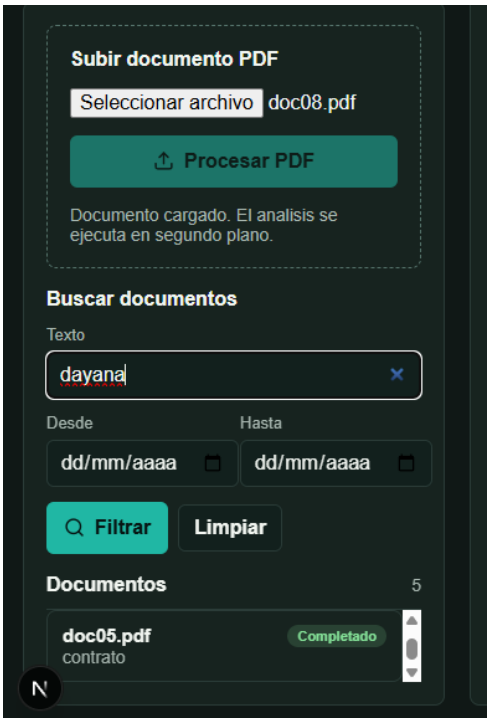


**Figura 36**. Búsqueda de documentos por texto.

Captura de pantalla de la funcionalidad de filtro por fecha del sistema. Esta opción permite al usuario definir un rango de búsqueda mediante los campos “desde” y “hasta”, con el fin de visualizar únicamente los documentos cargados o procesados dentro del período seleccionado. Esta funcionalidad facilita la organización y recuperación de documentos según su fecha de registro en el sistema.

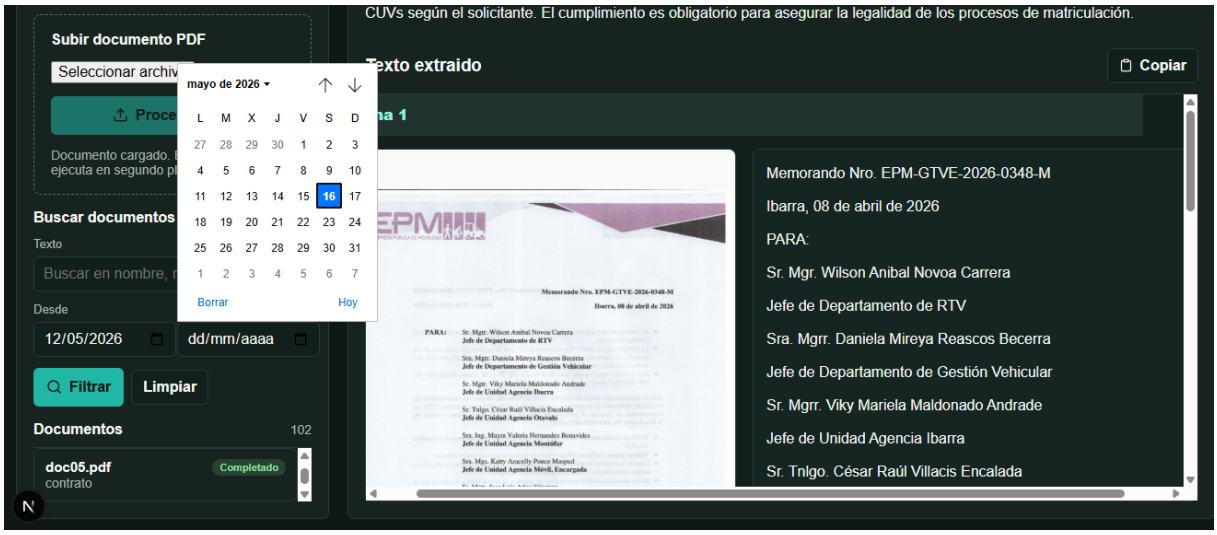


**Figura 37**. Búsqueda filtro por fecha del sistema.

Tiene la funcionalidad de descargar los documentos en PDF, Word o txt, cada uno tiene su botón para que la información se pueda obtener de manera inmediata.




**Figura 38**. Descarga desde la interfaz en formatos PDF, Word o txt del texto extraido.

Se muestra la documentación interactiva generada automáticamente por FastAPI mediante Swagger UI, disponible en la ruta /docs. En esta sección se visualizan los endpoints implementados en el backend del sistema, permitiendo identificar las operaciones disponibles para la carga de documentos PDF, consulta de documentos procesados, búsqueda, descarga de archivos y generación de reportes. Esta interfaz facilita la verificación y prueba de los servicios expuestos por la API.

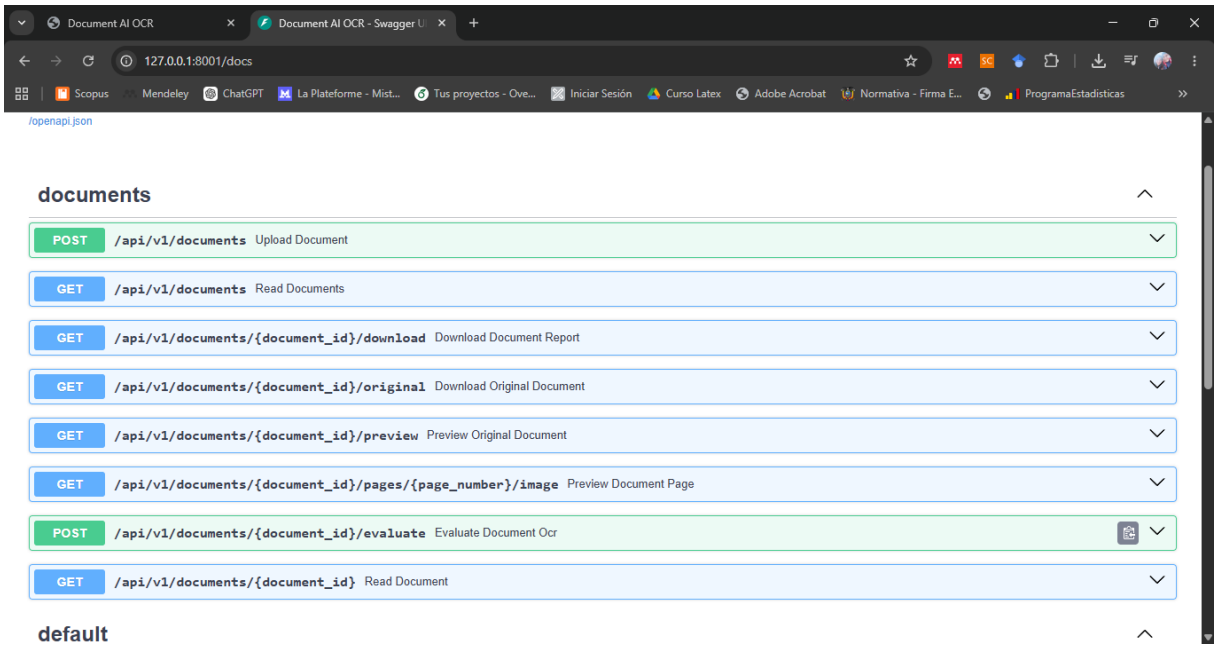


**Figura 39**. Documentación interactiva Swagger/FastAPI.

La ejecución del script experimental de evaluación G-Eval desde la terminal. En esta salida se observa que fueron evaluados 26 documentos utilizando el modelo gpt-4o-mini, con un umbral de aceptación de 0.7. La evaluación consideró cinco criterios principales: fidelidad, completitud, orden de lectura, calidad de tablas y limpieza de ruido, además de un puntaje general.

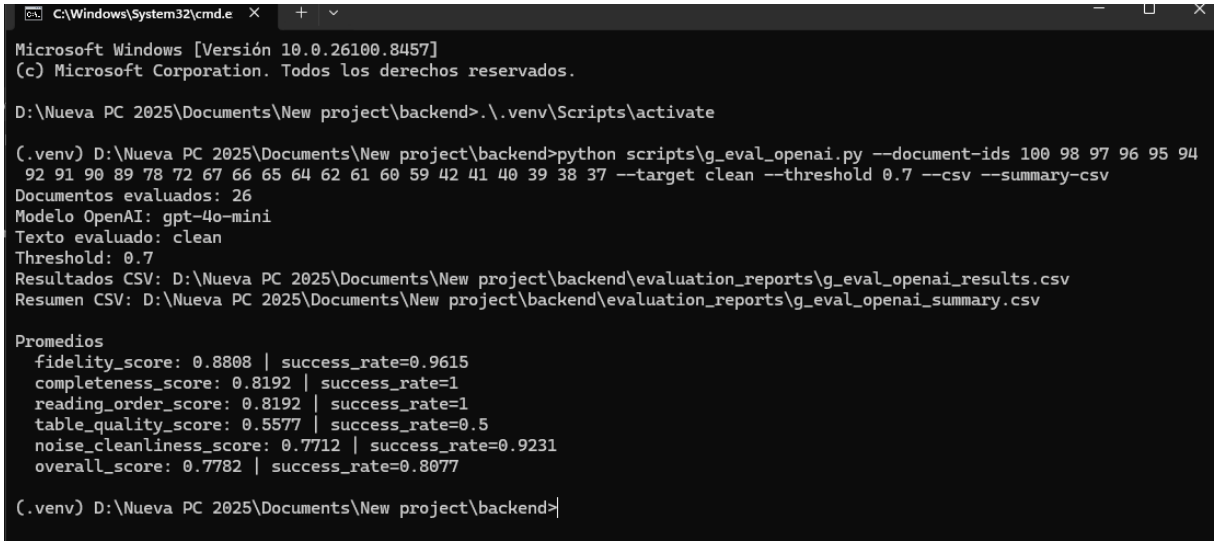


**Figura 40**. Evaluación experimental G-Eval.

Se observa la carpeta `storage\documents`, donde el sistema guarda los archivos PDF originales cargados por el usuario mediante nombres únicos generados automáticamente. Esto permite conservar los documentos sin sobrescribir archivos con nombres repetidos.

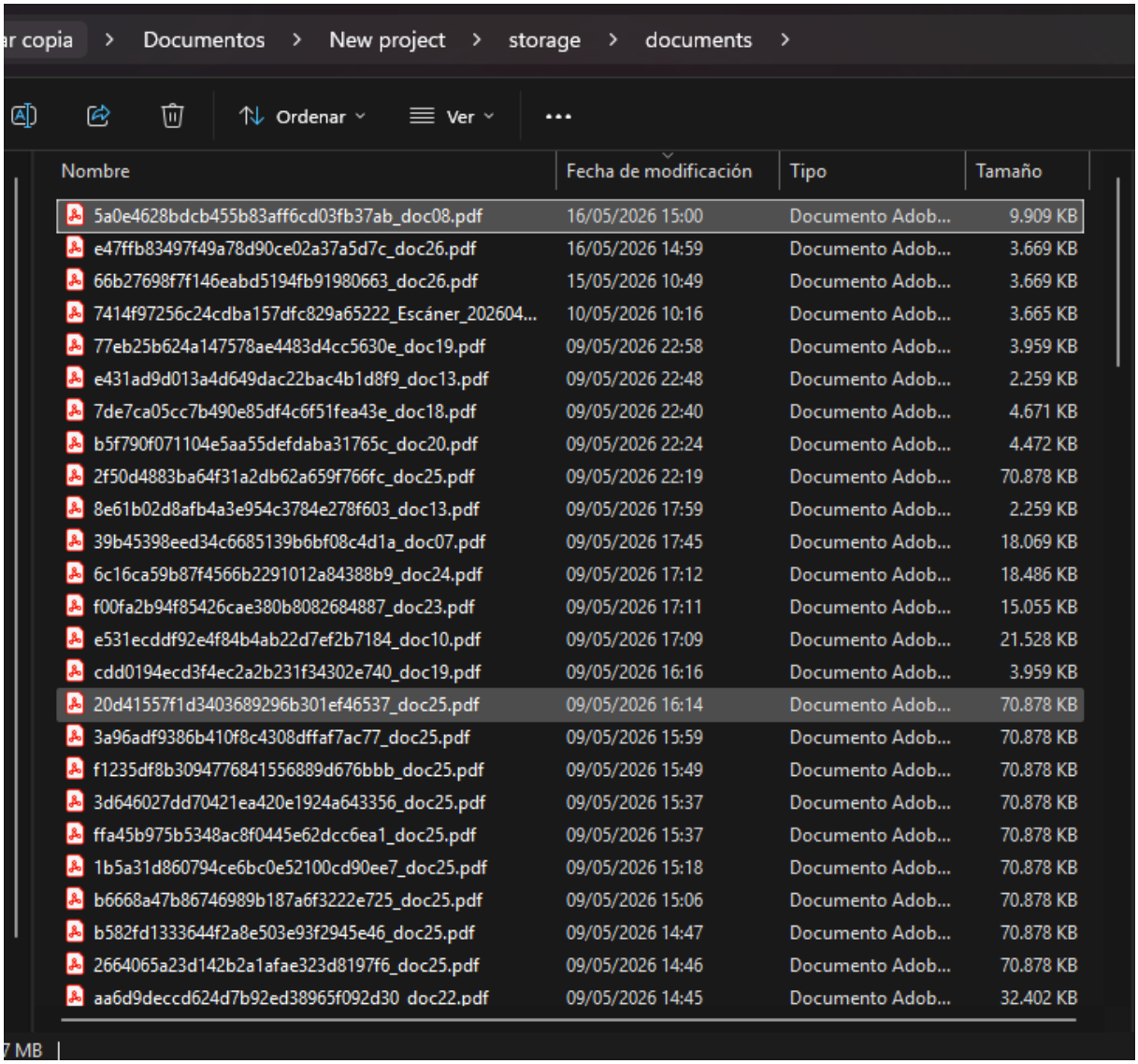


**Figura 41**. Almacenamiento local de documentos PDF.

Se presenta las calificaciones originales asignadas por los tres evaluadores humanos en una escala de 1 a 5. Estos valores fueron utilizados para calcular el promedio por criterio y posteriormente normalizarlos a una escala de 0 a 1, con el fin de compararlos con los resultados obtenidos mediante G-Eval.

Se muestra una vista comparativa entre una página del documento PDF original y el texto extraído por el sistema. En el panel izquierdo se visualiza la imagen de la página procesada,

| Criterio evaluado | Evaluador 1 | Evaluador 2 | Evaluador 3 | Promedio |
|---|---|---|---|---|
| Fidelidad | 3 | 4 | 3 | 3,33 |
| Completitud | 4 | 3 | 3 | 3,33 |
| Orden de lectura | 4,5 | 4 | 3 | 3,83 |
| Calidad de tablas | 2 | 3 | 2 | 2,33 |
| Limpieza de ruido | 3 | 2 | 3 | 2,67 |

**Figura 42**. Resultados de evaluación humana por criterio.

mientras que en el panel derecho se presenta el contenido textual obtenido mediante OCR y procesamiento posterior. Esta comparación permite verificar visualmente la correspondencia entre el documento original y la salida generada por el sistema.

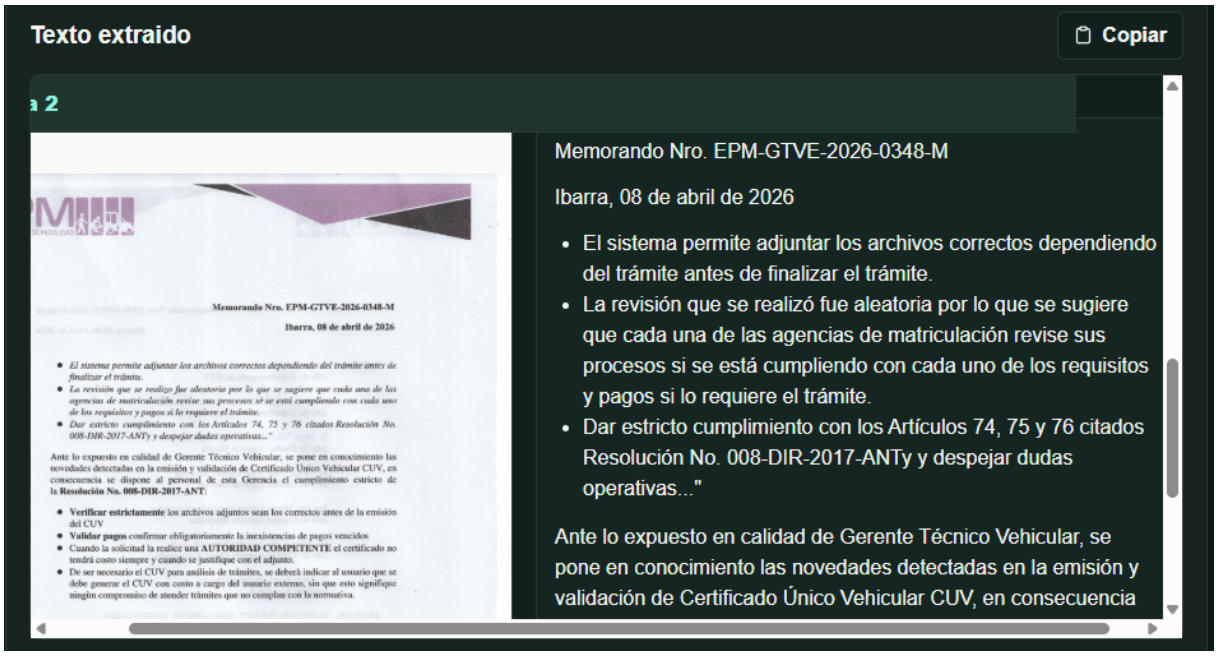


**Figura 43**. Comparación entre PDF original y texto extraído.

Se muestra la configuración de variables de entorno utilizadas por el backend del sistema. Por seguridad, los valores sensibles, como claves de API, credenciales de base de datos y tokens de acceso, fueron ocultados. Estas variables permiten configurar la conexión a PostgreSQL, las credenciales de servicios de IA, la ruta de almacenamiento local y los parámetros de carga de documentos.




**Figura 44**. Configuración de variables de entorno del sistema.

Tabla completa de los 26 documentos con valores individuales de CER y WER.

**Tabla XXVIII**.

Resultados de evaluación CER y WER por documento

| **Documento ID** | **CER** | **WER** |
|---|---|---|
| 66b27698f7f146eabd5194fb91980663 | 0.0056 | 0.0071 |
| 77eb25b624a147578ae4483d4cc5630e | 0.0 | 0.0 |
| e431ad9d013a4d649dac22bac4b1d8f9 | 0.1814 | 0.24 |
| 7de7ca05cc7b490e85df4c6f51fea43e | 0.0008 | 0.0069 |
| b5f790f071104e5aa55defdaba31765c | 0.1483 | 0.6875 |
| 2f50d4883ba64f31a2db62a659f766fc | 0.0156 | 0.0251 |
| 39b45398eed34c6685139b6bf08c4d1a | 0.05 | 0.0535 |
| 6c16ca59b87f4566b2291012a84388b9 | 0.0489 | 0.0585 |
| f00fa2b94f85426cae380b8082684887 | 0.0081 | 0.0119 |
| e531ecddf92e4f84b4ab22d7ef2b7184 | 0.0022 | 0.0086 |
| aa6d9deccd624d7b92ed38965f092d30 | 0.0014 | 0.0078 |
| be38723b060541c79d00056690bd5df6 | 0.0369 | 0.0408 |
| 961dee9157fe4a07a891af89945fabae | 0.0 | 0.0 |
| 7b7923f5896f45a1874bb8e2f6bb1dca | 0.0527 | 0.0536 |
| 2d64dd3fbb314933a4338916dbbc21c7 | 0.0193 | 0.0243 |
| 7548a84ef124460590af3aedbf9f2b33 | 0.053 | 0.1579 |
| 1cfa682f3d714299a28542212507814b | 0.0472 | 0.106 |
| 7d095ac577444483895ba4d387d2f275 | 0.0269 | 0.0226 |

**Tabla XXVIII – Continuación**

| Documento ID | CER | WER |
|---|---|---|
| d87720058d5d40c6937ea8b19468a318 | 0.0335 | 0.0444 |
| ab19a2930c7045fab8370132b1e711fe | 0.0007 | 0.0087 |
| cdcf614a439849aea9eb444808d1332e | 0.0966 | 0.1803 |
| 34dfd2f353d946f6a1b7241b7ead3e40 | 0.0012 | 0.0073 |
| 16140a6dc13142cdaf17b4033c5a2860 | 0.0 | 0.0 |
| 11e6757a6cf843c39cba02374d429f10 | 0.0 | 0.0 |
| 0ed11425d4e04d6d98060cf77cc076bf | 0.2557 | 0.2841 |
| c5474ddcdc4e4402a114560dfbd2456f | 0.0 | 0.0 |