Extracción Inteligente de Metadatos del Anexo No. 2 en Reclamación de Garantías: Automatización con Ciencia de Datos Aplicada al FNG

Johan Camilo Duque Carreño
Sergio Peñuela Cardenas

Director: Msc. Mario Julián Mora Cardona

25 de mayo de 2026

Pontificia Universidad Javeriana Cali
Facultad de Ingeniería y Ciencias
Maestría en Ciencia de Datos
Proyecto Aplicado

Resumen

El presente proyecto desarrolla un prototipo funcional en Python para automatizar la extracción de información del Anexo No. 2 del proceso de reclamación de garantías en el Fondo Nacional de Garantías (FNG). Este formulario semiestructurado concentra datos críticos de deudores y codeudores —como nombre o razón social, NIT o cédula, correo electrónico, teléfono, dirección y ciudad— y, en la operación actual, su lectura se realiza de manera manual, lo que genera cuellos de botella operativos y errores de transcripción.

La solución propuesta combina dos componentes complementarios. Para los documentos PDF con texto embebido se implementó un pipeline basado en reglas, extracción por *layout* y validación semántica. Para los formularios manuscritos se construyó y ajustó finamente un modelo TrOCR (`microsoft/trocr-large-handwritten`) sobre un conjunto de recortes de campos del formulario, con el fin de reconocer escritura manuscrita directamente a partir de imágenes. El sistema opera íntegramente en un entorno local, sin depender de servicios externos en la nube.

El corpus de trabajo comprende 551 documentos clasificados en tres tipos: 301 PDFs digitales con texto seleccionable, 240 formularios manuscritos y 10 documentos escaneados. Para el componente manuscrito se construyó un *dataset* de 231 pares imagen–texto, particionados en 80/20 para entrenamiento y validación.

El ajuste fino de TrOCR, una arquitectura `VisionEncoderDecoder` que combina BEiT como codificador de imagen y RoBERTa como decodificador de texto, se realizó en Google Colaboratory con GPU NVIDIA T4. El mejor *checkpoint*, obtenido en la época 8 de 10, alcanzó un *Character Error Rate* (CER) de 0,1176, equivalente a una precisión del 88,2 % a nivel de carácter.

Los resultados demuestran la viabilidad técnica de automatizar la lectura de campos manuscritos en documentos de alta variabilidad caligráfica, sin dependencia de servicios externos en la nube y bajo restricciones operativas de privacidad.

Palabras clave: reconocimiento óptico de caracteres, TrOCR, documentos manuscritos, extracción de información, Transformer, *fine-tuning*, visión por computador, procesamiento de formularios, Fondo Nacional de Garantías.

Dedicatoria

A mi familia, por ser mi pilar fundamental, por sostenerme con amor y por caminar conmigo en cada paso de esta maestría. A mi abuela, mi Tata, que me acompañó desde el inicio de este sueño y que hoy nos sigue acompañando desde el cielo; su presencia continúa siendo la fuerza más hermosa detrás de cada logro. Este proyecto está dedicado a mis padres y, principalmente, a ella, con todo mi amor y gratitud eterna.

Camilo

A mi familia, por su apoyo incondicional, por su confianza y por estar siempre presente en este proceso.

Sergio

Agradecimientos

Agradecemos al Fondo Nacional de Garantías por proporcionar el contexto institucional y operativo que motivó este proyecto, así como por su apertura al permitirnos identificar y abordar necesidades reales de automatización en sus procesos de reclamación de garantías.

A nuestro director, Msc. Mario Julián Mora Cardona, por su guía metodológica, sus orientaciones técnicas y su compromiso con el rigor académico de este trabajo.

A la Pontificia Universidad Javeriana Cali y a los profesores de la Maestría en Ciencia de Datos, por brindarnos las herramientas conceptuales y técnicas que hicieron posible este proyecto.

Índice general

Índice de tablas

Índice de Código

Índice de Acrónimos y Abreviaturas

Acrónimo	**Significado**
API	*Application Programming Interface* (Interfaz de Programación de Aplicaciones)
AMP	*Automatic Mixed Precision* (Precisión Mixta Automática)
BEiT	*Bidirectional Encoder representation from Image Transformers*
CER	*Character Error Rate* (Tasa de Error por Carácter)
CRF	*Conditional Random Fields* (Campos Aleatorios Condicionales)
CRISP-DM	*Cross-Industry Standard Process for Data Mining* (Proceso Estándar entre Industrias para Minería de Datos)
CSV	*Comma-Separated Values* (Valores Separados por Comas)
DIAN	Dirección de Impuestos y Aduanas Nacionales
DPI	*Dots Per Inch* (Puntos Por Pulgada)
IAM	*IAM Handwriting Database* (base de datos de escritura a mano)
FNG	Fondo Nacional de Garantías
FP16	*Floating Point 16-bit* (Punto Flotante de 16 bits)
GPU	*Graphics Processing Unit* (Unidad de Procesamiento Gráfico)
JSON	*JavaScript Object Notation* (Notación de Objetos de JavaScript)
NER	*Named Entity Recognition* (Reconocimiento de Entidades Nombradas)
NIT	Número de Identificación Tributaria
NLP	*Natural Language Processing* (Procesamiento de Lenguaje Natural)
OCR	*Optical Character Recognition* (Reconocimiento Óptico de Caracteres)
PDF	*Portable Document Format* (Formato de Documento Portátil)
PJ	Persona Jurídica
PN	Persona Natural
SIC	Superintendencia de Industria y Comercio
TrOCR	*Transformer-based Optical Character Recognition* (Reconocimiento Óptico de Caracteres basado en Transformers)
VED	*VisionEncoderDecoder*
VRAM	*Video Random Access Memory* (Memoria de Acceso Aleatorio de Video)
WER	*Word Error Rate* (Tasa de Error por Palabra)

Capítulo 1

Introducción

El Fondo Nacional de Garantías (FNG) es una entidad financiera colombiana que respalda créditos otorgados a micro, pequeñas y medianas empresas mediante garantías formalizadas en documentos estandarizados [1]. Uno de estos documentos es el Anexo No. 2, un formulario de tres páginas que recoge la información de identificación de deudores, codeudores, avales y garantes. El procesamiento manual de estos formularios genera cuellos de botella operativos, errores de transcripción y costos administrativos significativos para la entidad.

La digitalización de documentos financieros es un desafío ampliamente reconocido en el campo del procesamiento inteligente de documentos. En el contexto colombiano, la adopción de herramientas de reconocimiento óptico de caracteres en entidades financieras ha avanzado de forma gradual, con restricciones relacionadas con la variabilidad caligráfica de los documentos y los requisitos de protección de datos personales establecidos en la Ley 1581 de 2012 [2]. Trabajos previos en instituciones nacionales han documentado las dificultades de aplicar OCR estándar sobre formularios manuscritos con alta variabilidad de escritura [3].

Los avances recientes en modelos de visión–lenguaje basados en Transformers [4] han abierto una nueva vía para el reconocimiento de escritura manuscrita sin depender de segmentación manual de caracteres. El modelo TrOCR combina un codificador visual de tipo BEiT [5] con un decodificador de lenguaje RoBERTa [6], logrando resultados competitivos en conjuntos de datos de escritura a mano en inglés. Su adaptación a documentos en español con vocabulario específico del dominio financiero representa el problema central de este trabajo.

Para estructurar el proceso de desarrollo se adoptó la metodología CRISP-DM [7], que organiza el trabajo en fases iterativas de comprensión del negocio, comprensión de los datos, preparación de los datos, modelamiento, evaluación e implementación. Esta metodología permitió articular las decisiones técnicas del pipeline con los requerimientos operativos del FNG.

El presente documento se organiza de la siguiente manera. Los capítulos 2 al 7 describen el problema, los objetivos, el alcance y las restricciones legales del proyecto. El Capítulo 8 presenta el marco teórico y el Capítulo 9 los trabajos relacionados. El Capítulo 10 expone el desarrollo del sistema en sus fases principales: marco metodológico, construcción del corpus, definición del esquema de anotación, desarrollo del modelo de extracción y evaluación de resultados. El Capítulo 11 presenta las conclusiones del trabajo y el Capítulo 12 las líneas de desarrollo identificadas para consolidar el prototipo.

Capítulo 2

Definición del Problema

El Fondo Nacional de Garantías (FNG) es una entidad financiera colombiana cuya misión es facilitar el acceso al crédito a personas naturales y jurídicas que no cuentan con las garantías suficientes exigidas por el sistema financiero tradicional [1]. En el proceso de reclamación de garantías, el área de Reclamaciones del FNG recibe y procesa el Anexo No. 2, un formulario institucional que recoge los datos de identificación y contacto de deudores, codeudores, garantes, avales y fiadores vinculados a cada operación de crédito. La versión 4.0 de este formulario, objeto exclusivo del presente proyecto, introdujo por primera vez los campos de correo electrónico, teléfono, dirección y ciudad, ausentes en versiones anteriores.

El Anexo No. 2 se diligencia en dos modalidades principales. En la primera, el formulario se completa digitalmente en el sistema de información del intermediario financiero y se exporta como PDF con texto embebido (documento digital). En la segunda, el formulario se imprime, se diligencia a mano por el deudor y/o codeudor, y se digitaliza mediante escáner o aplicación móvil (documento manuscrito). En ambos casos, el documento llega al área de Reclamaciones como un archivo PDF que debe ser revisado manualmente por un analista para extraer y registrar los datos en los sistemas internos de la entidad.

Este proceso manual genera tres tipos de problemas operativos documentados: errores de transcripción por digitación incorrecta de datos alfanuméricos (NITs, cédulas, correos electrónicos); demoras en el procesamiento por la revisión campo a campo de cada formulario; y dificultad para escalar el proceso ante aumentos en el volumen de reclamaciones [3]. La automatización de la extracción de datos del Anexo No. 2 es, por tanto, un problema con impacto operativo directo y medible en la eficiencia del área de Reclamaciones.

Capítulo 3

Planteamiento del Problema

El problema central del proyecto se puede formular de la siguiente manera: dado un PDF del Anexo No. 2 —digital o manuscrito—, ¿es posible extraer automáticamente y con suficiente precisión los valores de los 13 campos de identificación y contacto de deudores y codeudores, de modo que el proceso manual de transcripción pueda ser reducido o eliminado?

Esta pregunta involucra dos subproblemas técnicos distintos. El primero corresponde a los PDFs digitales con texto embebido, donde el reto no es el reconocimiento de caracteres sino la localización y delimitación precisa de cada campo dentro del *layout* del formulario, que varía según el tipo de persona (jurídica o natural) y la cantidad de titulares registrados por formulario. El segundo corresponde a los formularios manuscritos digitalizados, donde el reto es el reconocimiento de escritura a mano en español colombiano con la diversidad caligráfica real de los analistas y deudores del FNG, para lo cual no existe un modelo preentrenado específico ni un corpus público [4].

La solución a ambos subproblemas debe ser técnicamente viable en el entorno institucional del FNG: sin acceso a datos reales de clientes durante el desarrollo —por las restricciones de la Ley 1581 de 2012 [2]—, operable íntegramente en entornos locales sin dependencia de servicios externos, y produciendo una salida estructurada en formato CSV integrable con los sistemas de información existentes.

Capítulo 4

Objetivos

4.1. Objetivo General

Desarrollar un prototipo funcional en Python que automatice la extracción de los campos de identificación y contacto del Anexo No. 2 versión 4.0 del FNG a partir de documentos PDF digitales y manuscritos, mediante un pipeline híbrido de reglas para documentos con texto embebido y un modelo TrOCR ajustado para formularios manuscritos, operando íntegramente de forma local sin dependencia de servicios externos.

4.1.1. Objetivos Específicos

1. Construir un corpus sintético de 551 documentos del Anexo No. 2 —PDFs digitales, formularios manuscritos y documentos escaneados— que sirva como base para el entrenamiento y evaluación del sistema, respetando las restricciones de la Ley 1581 de 2012 [2].

2. Definir un esquema de anotación de 13 campos clave —7 para Persona Jurídica y 6 para Persona Natural— y construir el conjunto de datos de *ground truth* supervisado de 1.204 cuadros etiquetados para la evaluación cuantitativa del sistema.

3. Implementar y evaluar el pipeline de extracción para PDFs digitales sobre el corpus sintético, documentando la cobertura por campo y la tasa global de cuadros con datos válidos.

4. Realizar el *fine-tuning* del modelo TrOCR sobre el corpus de formularios manuscritos y evaluar su desempeño reportando la tasa de error de caracteres (CER) sobre el conjunto de prueba.

5. Integrar ambos componentes en un flujo unificado de procesamiento que detecte automáticamente el tipo de documento y aplique el componente correspondiente, produciendo una salida estructurada en formato CSV para su integración con los sistemas de información del FNG.

Capítulo 5

Alcance y Limitaciones

5.1. Alcance

El proyecto cubre el diseño, implementación y evaluación de un prototipo funcional en Python que procesa el Anexo No. 2 versión 4.0 del FNG en sus dos escenarios operativos principales: PDFs digitales con texto embebido y formularios manuscritos digitalizados. La salida del prototipo es un archivo CSV con los valores extraídos de los 13 campos del esquema de anotación para cada cuadro del formulario procesado, formato directamente integrable con sistemas SQL, Excel y las plataformas de información del área de Reclamaciones.

El desarrollo se realizó íntegramente sobre un corpus sintético de 551 documentos —301 PDFs digitales, 240 formularios manuscritos y 10 documentos escaneados explorados como trabajo futuro— generados específicamente para el proyecto, sin uso de documentos reales del FNG en ninguna fase del trabajo. El prototipo funcional es el código modular desarrollado en *notebooks* de Python; su salida estructurada en CSV y JSON constituye la forma técnicamente correcta de integración en el entorno institucional, dado que los sistemas de información del sector financiero operan sobre bases de datos relacionales y hojas de cálculo. El desarrollo de una interfaz gráfica de usuario queda fuera del alcance de este trabajo de grado y se propone como trabajo futuro (véase Capítulo 12).

5.2. Limitaciones

- **Corpus sintético:** toda la evaluación se realizó sobre documentos generados artificialmente. El desempeño sobre documentos reales del FNG no ha sido validado en esta fase del proyecto y constituye la principal línea de trabajo futuro, sujeta a la obtención de autorización formal bajo la Ley 1581 de 2012 [2].
- **Versión del formulario:** el prototipo está calibrado para la versión 4.0 del Anexo No. 2. Versiones anteriores con *layout* diferente requieren ajustes en los parámetros del pipeline digital y en los parámetros de segmentación del componente manuscrito.
- **Escenario escaneado:** los documentos escaneados —formularios digitales impresos y luego digitalizados— no fueron incorporados al pipeline de producción en esta fase del proyecto. Su tratamiento se identifica como trabajo futuro (véase Capítulo 12).

- **Idioma y dominio:** el modelo TrOCR fue preentrenado en inglés y ajustado fino sobre escritura manuscrita en español colombiano con vocabulario del dominio financiero. Su desempeño en otros idiomas o formularios de distinto dominio no ha sido evaluado.
- **Sin interfaz gráfica:** el prototipo es código ejecutable en Python, no una aplicación con interfaz de usuario. La salida en CSV y JSON es la forma correcta de integración con los sistemas del FNG en esta etapa del proyecto.

Capítulo 6

Justificación

La justificación del proyecto se articula en tres dimensiones complementarias: operativa, metodológica y académica.

Desde la dimensión **operativa**, la automatización de la extracción del Anexo No. 2 reduce directamente el tiempo de procesamiento por reclamación y elimina los errores de transcripción manual, que tienen consecuencias directas en la trazabilidad de las operaciones y en la calidad de la información registrada en los sistemas del FNG. El volumen de formularios que procesa diariamente el área de Reclamaciones hace inviable la revisión manual exhaustiva sin incurrir en demoras o en la contratación de personal adicional.

Desde la dimensión **metodológica**, el proyecto demuestra que es posible construir un corpus de entrenamiento sintético de alta calidad para un formulario institucional específico sin violar las restricciones de privacidad establecidas en la Ley 1581 de 2012 [2]. La generación programática de documentos digitales y la recolección de formularios manuscritos con datos ficticios —completados por participantes voluntarios— garantiza la ausencia de datos personales reales en todo el ciclo del proyecto. Esta metodología es replicable en otras entidades del sector financiero colombiano que enfrenten restricciones similares.

Desde la dimensión **académica**, el proyecto contribuye a la literatura sobre adaptación de modelos de visión por computador a dominios específicos en español latinoamericano, área subrepresentada en la investigación internacional donde la mayoría de los trabajos se desarrollan sobre corpus en inglés [8]. La combinación de un pipeline de reglas para documentos digitales con un modelo TrOCR ajustado para manuscritos constituye un enfoque híbrido que responde a las condiciones reales de un dominio institucional con recursos limitados y sin corpus público preexistente [4, 3].

Capítulo 7

Restricciones Legales y Evaluación de Alternativas Comerciales

Una decisión técnica central del proyecto fue descartar explícitamente las soluciones comerciales de extracción de documentos disponibles en el mercado. Esta decisión no responde a criterios de desempeño sino a una restricción legal de carácter no negociable: la Ley 1581 de 2012 [2] y el Decreto 1377 de 2013 prohíben la transferencia internacional de datos personales sensibles sin el consentimiento explícito de su titular. Los formularios del Anexo No. 2 contienen NITs, cédulas, correos electrónicos, teléfonos y direcciones de personas naturales y jurídicas, datos que la normativa colombiana clasifica como sensibles y cuyo tratamiento está sujeto a los principios de finalidad, necesidad y circulación restringida.

Todas las soluciones líderes del mercado para extracción de documentos —Google Document AI, Amazon Textract, Azure Form Recognizer y ChatGPT Vision— operan mediante el envío del documento a servidores ubicados fuera del territorio colombiano. Procesar los formularios del FNG a través de cualquiera de estas plataformas constituiría una transferencia internacional de datos sin autorización del titular, infracción sancionable por la Superintendencia de Industria y Comercio (SIC). La Tabla 7.1 resume esta evaluación.

Tabla 7.1: Alternativas de extracción documental frente a la restricción de Habeas Data

Solución	**Proveedor**	**Tipo**	**Habeas Data (Ley 1581)**
ChatGPT Vision (GPT-4o)	OpenAI	Propietario	× Nube EEUU
Google Document AI	Google Cloud	Propietario	× Nube global
AWS Textract	Amazon	Propietario	× Nube EEUU
Azure Form Recognizer	Microsoft	Propietario	× Nube EEUU
Adobe Extract API	Adobe	Propietario	× Nube
TrOCR (HuggingFace)	**Microsoft Res**	**Cód. abierto**	**✓ Local**

Esta restricción refuerza además el argumento central del proyecto: si el propio equipo de de-

sarrollo —con acceso institucional al FNG— no puede utilizar los documentos reales por razones de Habeas Data, ningún servicio de inteligencia artificial externo puede hacerlo bajo la misma justificación legal. La única alternativa técnicamente viable y legalmente admisible es un modelo que opere íntegramente de forma local, sin enviar datos a servidores externos. TrOCR, disponible como modelo de código abierto a través de HuggingFace Transformers, cumple esta condición de forma nativa [4].

Capítulo 8

Marco Teórico

La extracción automática de información a partir de documentos institucionales es un problema activo en la literatura de visión por computador y procesamiento de lenguaje natural [9]. La mayoría de los trabajos existentes se desarrollan sobre corpus genéricos en inglés —facturas comerciales, formularios fiscales, recibos de compra— que difieren significativamente del Anexo No. 2 del FNG en idioma, estructura y tipo de escritura. Esta especificidad del dominio fue determinante en la selección del enfoque técnico del proyecto.

8.1. OCR Clásico y Enfoques Basados en Reglas

Los sistemas de reconocimiento óptico de caracteres de primera generación, como Tesseract [10, 11], operan mediante la segmentación de la imagen en regiones de texto y la clasificación de cada carácter de forma independiente. Combinados con expresiones regulares y reglas heurísticas, constituyen el enfoque más extendido para la extracción de campos en formularios digitales estructurados [3].

Este enfoque es eficaz cuando los documentos tienen texto impreso de alta calidad y *layout* completamente predecible. Sin embargo, Tesseract no fue diseñado para escritura manuscrita en español y sus resultados sobre formularios diligenciados a mano presentan tasas de error que lo hacen inviable para el componente manuscrito del proyecto [12]. Adicionalmente, las reglas basadas en coordenadas absolutas son frágiles frente a variaciones de posición entre versiones del formulario. Estas limitaciones motivaron el desarrollo de un componente especializado basado en aprendizaje profundo para el escenario manuscrito.

8.2. Modelos de Lenguaje con Comprensión de *Layout*

LayoutLM [13] y sus versiones posteriores representan el estado del arte en comprensión de documentos. Estos modelos integran en un único *encoder* Transformer la información textual, las coordenadas espaciales de cada *token* y la representación visual de la página, logrando resultados superiores a enfoques anteriores en tareas de extracción de entidades y clasificación de campos.

No obstante, la aplicación de LayoutLM al Anexo No. 2 presenta un obstáculo central en el contexto de este proyecto: el modelo requiere como entrada las coordenadas exactas (*bounding*

boxes) de cada *token* de texto, lo que implica disponer de un pipeline de detección de texto que opere correctamente sobre documentos manuscritos en español. Construir y anotar ese pipeline adicional excedía el alcance del proyecto, y no existe un corpus público de detección de texto para este formulario específico. Por estas razones, LayoutLM fue descartado como enfoque principal y se identifica como línea de trabajo futuro (véase Capítulo 12).

8.3. Modelos de Lenguaje para Extracción de Entidades

BERT [14] y sus derivados —RoBERTa [6], BETO para español— han demostrado desempeño sobresaliente en tareas de reconocimiento de entidades nombradas (NER) aplicadas a documentos [15]. En el contexto de extracción de formularios, estos modelos pueden identificar y clasificar entidades como nombres, números de identificación y correos electrónicos directamente sobre el texto del documento.

Sin embargo, este enfoque presupone que el texto del documento ya ha sido extraído y está disponible en formato limpio. Para los PDFs digitales del Anexo No. 2 con texto embebido, esto es directo con PyMuPDF [16]; pero para los formularios manuscritos, aplicar NER requeriría primero un OCR de calidad suficiente, introduciendo la misma dependencia que se intenta resolver. Esta limitación hace que los modelos de lenguaje puros no sean aplicables de forma directa al escenario manuscrito del proyecto.

8.4. Modelos *Encoder-Decoder* para Reconocimiento de Texto en Imagen

TrOCR [4] y Donut [17] representan una generación de modelos que integran un codificador de visión con un decodificador de texto, permitiendo reconocer texto directamente a partir de una imagen sin necesidad de segmentación previa ni pipeline de detección de palabras. TrOCR combina un codificador BEiT [5] con un decodificador RoBERTa [6] y fue preentrenado específicamente sobre corpus de escritura manuscrita —base de datos IAM, 115 320 imágenes—, lo que lo posiciona como el candidato más adecuado para el componente manuscrito del proyecto.

Donut adopta un enfoque de extremo a extremo que opera sobre la página completa en lugar de recortes individuales. Si bien sus resultados en *benchmarks* como DocVQA son competitivos, no dispone de un *checkpoint* preentrenado específico para escritura manuscrita en español, lo que habría requerido una fase de preentrenamiento adicional fuera del alcance del proyecto. Por esta razón, Donut fue descartado como alternativa principal.

8.5. Justificación del Enfoque Adoptado

El análisis de los enfoques anteriores, evaluado en el contexto específico del Anexo No. 2 del FNG, condujo a la arquitectura de dos componentes especializados:

- **Pipeline híbrido con PyMuPDF** [16] para PDFs digitales: aprovecha el texto embebido disponible mediante anclas textuales y expresiones regulares con validación semántica por campo. Es el enfoque más eficiente para este escenario dado que el texto ya está disponible sin necesidad de reconocimiento de imagen.
- **TrOCR con ajuste fino** [4] para formularios manuscritos: opera directamente sobre recortes de imagen de cada campo, no requiere *bounding boxes* de *tokens*, dispone de preentrenamiento en escritura a mano y es ejecutable íntegramente de forma local en cumplimiento de la Ley 1581 de 2012 [2].

Esta combinación responde a la naturaleza dual del corpus del FNG y a las restricciones reales del proyecto: ausencia de anotaciones espaciales, operación completamente local y dominio altamente específico sin corpus público preexistente [4, 3].

Capítulo 9

Antecedentes

La extracción automática de información de documentos institucionales ha sido abordada desde múltiples perspectivas en la literatura académica y en la práctica industrial. A continuación se presentan los trabajos y enfoques más relevantes como antecedentes directos del presente proyecto.

Guzmán [3] analizó los desafíos de automatización en procesos documentales de entidades financieras colombianas, identificando la variabilidad de formato y la coexistencia de documentos digitales y manuscritos como los principales obstáculos para la implementación de sistemas de extracción automática. Su trabajo documentó que los enfoques basados exclusivamente en coordenadas fijas presentan tasas de fallo elevadas cuando el *layout* varía entre versiones del formulario, lo que motivó el enfoque híbrido adoptado en el presente proyecto.

Zanibbi y Doermann [12] presentaron una revisión comprensiva del estado del arte en análisis y reconocimiento de imágenes de documentos, identificando como desafíos abiertos la variabilidad caligráfica, la degradación de imagen por escaneo y la falta de corpus etiquetados en idiomas distintos al inglés. Estas limitaciones son directamente relevantes para el Anexo No. 2 del FNG, cuyos formularios manuscritos exhiben exactamente estas características.

Xu et al. [13] propusieron LayoutLM, un modelo que integra representaciones textuales y espaciales para la comprensión de documentos, alcanzando resultados de estado del arte en *benchmarks* de extracción de información. Su análisis de las limitaciones de los enfoques basados exclusivamente en texto —sin considerar la posición espacial de los elementos— fundamenta la decisión de utilizar coordenadas de *layout* en el componente digital del presente proyecto.

Li et al. [4] presentaron TrOCR, modelo *encoder-decoder* basado en Transformers que combina un codificador visual BEiT [5] con un decodificador de lenguaje RoBERTa [6] para el reconocimiento de texto en imágenes. Los autores reportaron resultados de estado del arte en los *benchmarks* IAM y SROIE, con una tasa de error de caracteres (CER) de 2,89 % en el conjunto de prueba de IAM. Este trabajo constituye la base técnica del componente de reconocimiento manuscrito del presente proyecto.

PyMuPDF [16] es la biblioteca de referencia para la manipulación programática de PDFs en Python, utilizada en este proyecto tanto para la extracción de texto embebido y sus coordenadas de posición como para la conversión de páginas del formulario a imágenes rasterizadas. Su capacidad para extraer bloques de texto con *bounding boxes* precisas y acceder a la estructura interna del PDF constituye la base técnica del componente de extracción digital.

Capítulo 10

Desarrollo del Proyecto

10.1. Marco metodológico

El proyecto adoptó el marco CRISP-DM (Cross-Industry Standard Process for Data Mining) como guía metodológica para el desarrollo de la herramienta de extracción automática. CRISP-DM es un modelo de proceso iterativo ampliamente utilizado en proyectos de ciencia de datos aplicados a contextos empresariales, que estructura el trabajo en seis fases: comprensión del negocio, comprensión de los datos, preparación de los datos, modelado, evaluación y despliegue [7].

La característica distintiva de CRISP-DM es su naturaleza cíclica: los resultados de cada fase pueden retroalimentar fases anteriores, permitiendo ajustes iterativos basados en el conocimiento adquirido durante el proceso. Este carácter iterativo resultó fundamental para este proyecto, donde decisiones iniciales sobre arquitecturas de modelos fueron revisadas y ajustadas a medida que se comprendía mejor la naturaleza de los datos y los resultados preliminares de evaluación [7].

10.2. Aplicación de CRISP-DM al proyecto

10.2.1. Comprensión del negocio

La fase de comprensión del negocio se enfocó en entender la necesidad operativa del FNG, las restricciones del dominio y los objetivos del prototipo. El proceso manual de extracción de información del Anexo No. 2 genera tres problemas críticos identificados con el área de Reclamaciones del FNG: cuellos de botella operativos en períodos de alta carga, errores de transcripción en campos alfanuméricos largos y falta de trazabilidad del proceso de extracción [3].

Se identificaron dos restricciones críticas que moldearon el diseño completo del proyecto: la confidencialidad de los datos regulada por la Ley 1581 de 2012 [2], que impide el envío de documentos reales a servicios de inteligencia artificial externos, y el requisito de operación local como consecuencia directa de esa restricción. Se definió que el resultado del proyecto sería un prototipo funcional en formato de notebooks, con objetivos alineados a la reducción del trabajo manual de transcripción y con salida estructurada en CSV integrable con los sistemas de información del FNG.

10.2.2. Comprensión de los datos

La fase de comprensión de los datos reveló la heterogeneidad del Anexo No. 2 y fundamentó la decisión de construir un corpus sintético. El análisis preliminar identificó tres escenarios operativos reales: PDFs digitales generados mediante herramientas de ofimática, manuscritos completados a mano y posteriormente digitalizados, y formularios impresos y escaneados.

El análisis de los 13 campos del formulario reveló una distinción técnica fundamental: campos con patrón fijo, como NIT, correo electrónico, teléfono y cédula, susceptibles de validación mediante expresiones regulares, y campos contexto-dependientes, como nombre, razón social, dirección y ciudad, que requieren comprensión contextual. Esta distinción guió el diseño del pipeline: los campos de patrón fijo se manejan mediante reglas de validación estricta, mientras que los campos contexto-dependientes requieren anclas textuales [3].

Las restricciones de protección de datos personales establecidas en la Ley 1581 de 2012 [2] impidieron el uso de documentos reales del FNG como corpus. Esta restricción, identificada desde la fase de comprensión del negocio, determinó que la estrategia de datos fuera la generación sintética controlada: una solución técnicamente sólida que además garantiza la ausencia total de datos personales reales y refuerza la necesidad de una solución completamente local [12].

10.2.3. Preparación de los datos

La fase de preparación de los datos fue la más extensa del proyecto, abarcando la generación sintética de documentos, el preprocesamiento diferenciado por escenario y la construcción del conjunto de datos anotado. Se construyó un corpus de 551 documentos sintéticos distribuidos en tres escenarios: 301 PDFs digitales generados programáticamente, 240 formularios manuscritos obtenidos mediante la participación de aproximadamente 20 voluntarios con caligrafías diversas, y 10 documentos escaneados empleados en pruebas exploratorias del tercer escenario.

Cada escenario requirió técnicas de preprocesamiento específicas. Para los PDFs digitales, el preprocesamiento se enfocó en la extracción estructurada de texto mediante PyMuPDF [16]. Para los manuscritos, el preprocesamiento constituyó el componente técnicamente más complejo: incluyó detección dinámica del inicio de la tabla mediante análisis del perfil de brillo vertical, detección de cuadros mediante derivada del perfil suavizado, recorte de campos por coordenadas relativas, binarización adaptativa, corrección de orientación mediante transformada de Hough y normalización de contraste [18].

Se definió un esquema de 13 campos clave, 7 para Persona Jurídica y 6 para Persona Natural, con un coeficiente Kappa interanotador de 0,94 en campos de patrón fijo y 0,87 en campos contexto-dependientes, construyendo un conjunto de datos supervisado de 1.204 cuadros etiquetados. El entrenamiento del modelo TrOCR se realizó exclusivamente sobre los formularios manuscritos, construyendo un conjunto de 231 pares imagen-texto [4].

Durante esta fase surgió una decisión técnica relevante: el anteproyecto planteaba el uso de

modelos CRF y BERT para el reconocimiento de entidades sobre el texto extraído. Sin embargo, el análisis de los documentos digitales reveló que la estructura del Anexo No. 2 es suficientemente predecible para ser tratada con reglas de posición y expresiones regulares, sin necesidad de entrenamiento supervisado [3]. Esta constatación condujo a adoptar un pipeline híbrido por reglas para los PDFs digitales, reservando el aprendizaje supervisado con TrOCR para el reconocimiento de escritura manuscrita.

10.2.4. Modelado

La fase de modelado se dividió según el escenario operativo. Para los PDFs digitales, se implementó un pipeline híbrido de reglas en tres capas: extracción por widgets, extracción por anclas textuales y validación semántica por expresiones regulares. Este enfoque se justifica porque los PDFs digitales tienen texto ya disponible y una estructura predecible; un modelo de aprendizaje automático requeriría miles de ejemplos etiquetados sin garantía de superar el rendimiento de reglas bien diseñadas para este dominio estructurado [3].

Para formularios manuscritos, se realizó el ajuste fino del modelo `microsoft/trocr-large-handwritten` sobre el conjunto de 231 pares imagen-texto [4]. La arquitectura TrOCR combina un codificador visual que procesa la imagen y un decodificador de lenguaje que genera texto carácter por carácter. Los hiperparámetros empleados incluyen 10 épocas, lote efectivo de 16, tasa de aprendizaje de 5×10^{-5}, optimizador AdamW y entrenamiento en precisión mixta FP16. Para el escenario de documentos escaneados, el pipeline requerirá reconocimiento óptico sobre texto tipográfico degradado, integrable con las mismas anclas textuales del componente digital; su desarrollo completo se describe en el capítulo 10.

10.2.5. Evaluación

La evaluación del pipeline digital se realizó sobre los 301 PDFs del corpus sintético, alcanzando una cobertura global del 80,3 % de cuadros con datos válidos. Los campos estructurales NIT, nombre y documento de identidad alcanzaron las coberturas más altas, mientras que los campos de Persona Jurídica sin widgets y los campos de ciudad presentaron los valores más bajos. Para el componente manuscrito, el modelo TrOCR alcanzó un CER de 0,1176 sobre el conjunto de prueba de 47 pares, equivalente al 88,2 % de precisión a nivel de carácter. Los resultados se presentan en detalle en el capítulo 10.7.3.

10.2.6. Despliegue

La fase de despliegue se concretó en un conjunto de notebooks Python modulares y reproducibles que procesan documentos del Anexo No. 2 y generan salidas estructuradas en formato CSV y JSON. Estos formatos son los adecuados para la integración con los sistemas de información del sector

financiero colombiano, que operan con bases de datos relacionales y hojas de cálculo. El prototipo funcional valida la viabilidad técnica de la automatización y constituye la base sobre la cual se proyectan las líneas de evolución descritas en el capítulo 10.

10.2.7. Carácter iterativo del proceso

El desarrollo del proyecto ejemplifica el carácter cíclico de CRISP-DM. Los resultados de evaluación del pipeline digital retroalimentaron la fase de preparación de los datos, conduciendo a la adición de mecanismos de validación semántica más estrictos. El análisis de la estructura predecible de los PDFs durante la fase de modelado confirmó que un pipeline de reglas era suficiente, descartando los modelos CRF y BERT planteados en el anteproyecto. La restricción de acceso a datos reales, conocida desde la fase de comprensión del negocio, fue determinante para diseñar una estrategia de corpus sintético que fortaleció tanto la calidad del conjunto de datos como la justificación del enfoque local del proyecto [7, 12].

10.3. Recolección y organización de la muestra de documentos

La construcción de un corpus representativo es el primer paso del ciclo CRISP-DM [7] y el cimiento sobre el que se apoya la evaluación del sistema. Las restricciones de protección de datos personales establecidas en la Ley 1581 de 2012 [2] y la ausencia de un convenio de tratamiento de datos firmado con el FNG durante esta etapa del proyecto determinaron, desde la fase de comprensión del negocio, que la estrategia de obtención de datos fuera la generación sintética controlada. Esta decisión permitió construir un corpus con verdad de referencia perfecta por construcción, sin exponer información personal real y sin depender de acuerdos institucionales externos al alcance del proyecto.

10.3.1. Justificación de la estrategia de corpus sintético

El anteproyecto contemplaba el uso de documentos reales del proceso de reclamaciones del FNG como corpus de entrenamiento y evaluación. El análisis de viabilidad realizado durante la fase de comprensión del negocio [7] identificó tres condiciones que hacían técnica y legalmente inviable ese enfoque:

- **Restricción normativa:** los documentos del Anexo No. 2 contienen datos personales de deudores y codeudores, cuyo tratamiento para fines distintos al proceso de reclamación requiere autorización explícita bajo la Ley 1581 de 2012 [2].
- **Restricción institucional:** el acceso a documentos reales del FNG requería un convenio formal de tratamiento de datos que no pudo gestionarse dentro del calendario del proyecto.

- **Ausencia de referencia validada:** los documentos reales disponibles no contaban con un conjunto de referencia anotado, lo que habría exigido una campaña de etiquetado manual extensa antes de poder utilizarlos para entrenamiento o evaluación.

La generación sintética controlada resuelve estas tres condiciones simultáneamente: los documentos se crean a partir de datos completamente ficticios, la verdad de referencia es perfecta por construcción y no se requiere acceso a los sistemas del FNG. Este enfoque está documentado en la literatura de aprendizaje automático para documentos institucionales [3] y es coherente con la restricción de operación local que rige el proyecto.

Adicionalmente, esta estrategia reforzó la hipótesis central del proyecto: si el equipo de desarrollo, aun con acceso institucional al contexto del problema, no podía utilizar documentos reales, ningún servicio de inteligencia artificial externo podría hacerlo bajo la misma justificación legal. Esto respalda la pertinencia de una solución completamente local [12].

10.3.2. Diseño del corpus: tres escenarios de documento

El diseño del corpus se estructuró en torno a los tres tipos de documento que el sistema de reclamaciones del FNG recibe en la práctica:

1. **PDFs digitales (Escenario 1):** formularios completados directamente en el sistema del intermediario financiero y exportados como PDF con texto embebido. Son documentos con estructura predecible y texto extraíble directamente sin necesidad de reconocimiento de imagen.

2. **Formularios manuscritos (Escenario 2):** formularios impresos en blanco, diligenciados a mano por el deudor o codeudor y digitalizados mediante escáner o aplicación móvil. Presentan alta variabilidad caligráfica y de calidad de imagen, y constituyen el escenario técnicamente más exigente.

3. **Documentos escaneados (Escenario 3):** formularios generados digitalmente, impresos y posteriormente escaneados. Combinan texto tipográfico con la degradación visual propia del proceso de impresión y escaneo.

El corpus final quedó conformado por 551 documentos distribuidos como se presenta en la Tabla 10.1.

Tabla 10.1: Distribución final del corpus sintético del Anexo No. 2

Tipo de documento	Documentos	% del total
PDFs digitales	301	54,6 %
Formularios manuscritos digitalizados	240	43,6 %
Documentos escaneados	10	1,8 %
Total	**551**	**100 %**

Los 240 formularios manuscritos constituyen el conjunto de datos de entrenamiento y evaluación del modelo TrOCR, por ser el escenario con mayor variabilidad entre documentos. Los 301 PDFs digitales constituyen la base de evaluación del pipeline de reglas. Los 10 documentos escaneados se utilizaron para pruebas exploratorias del Escenario 3, cuyo desarrollo completo queda como trabajo futuro.

10.3.3. Generación de PDFs digitales (Escenario 1)

Los 301 PDFs digitales se generaron programáticamente a partir de tres plantillas del formulario Anexo No. 2 en formato `.docx`, sobre las cuales se inyectaron datos ficticios mediante la biblioteca `docxtpl`. El uso de tres plantillas distintas —con bordes visibles, sin bordes con mayor espaciado y con diseño libre sin tabla estructurada— permite simular la variabilidad real de los PDFs generados por distintas herramientas de ofimática en los intermediarios financieros.

Los datos ficticios respetan las restricciones de formato del formulario: nombres propios colombianos, NITs con dígito verificador válido, correos electrónicos, teléfonos de entre 7 y 10 dígitos, direcciones con nomenclatura urbana colombiana y ciudades del país. La distribución entre tipos de persona respetó las proporciones operativas del FNG: 60 % Persona Natural y 40 % Persona Jurídica.

10.3.3.1. Carga de la plantilla y los datos

```
from docxtpl import DocxTemplate
import pandas as pd
from google.colab import drive

drive.mount("/content/drive")

ruta_plantilla = "/content/drive/MyDrive/objetivo/Anexo No 02.docx"
ruta_excel     = "/content/drive/MyDrive/objetivo/datos_ficticios.xlsx"

df = pd.read_excel(ruta_excel)
```

```
print(f"Datos cargados: {len(df)} documentos a generar.")
```

Código 10.1: Carga de la plantilla y el archivo de datos en Google Colab

10.3.3.2. Limpieza de valores y renderizado

Antes de inyectar los datos, cada valor se limpia para evitar que celdas vacías del Excel generen errores de tipo o texto `nan` visible en el documento:

```
def limpiar_valor(valor):
if pd.isna(valor):
return ""
if isinstance(valor, float) and valor.is_integer():
return str(int(valor))
return str(valor)

def limpiar_fila(fila):
return {k: limpiar_valor(v) for k, v in fila.items()}

doc = DocxTemplate(ruta_plantilla)
doc.render(limpiar_fila(df.iloc[0]))
doc.save("Documento_001.docx")
```

Código 10.2: Limpieza de valores y generación del documento Word renderizado

La función `is_integer()` resuelve un problema frecuente al leer NITs y cédulas desde Excel: Python los interpreta como `float` y los almacena con decimal `.0`. La conversión a entero antes de formatear como cadena garantiza que el campo quede como `900123456` en lugar de `900123456.0`.

10.3.3.3. Conversión a PDF con LibreOffice

```
import os

def convertir_a_pdf(ruta_docx, ruta_pdf_destino):
carpeta = os.path.dirname(ruta_docx)
os.system(
f'libreoffice --headless --convert-to pdf '
f'"{ruta_docx}" --outdir "{carpeta}"'
)
generado = ruta_docx.replace(".docx", ".pdf")
if os.path.exists(generado):
os.rename(generado, ruta_pdf_destino)
```

Código 10.3: Conversión de DOCX a PDF mediante LibreOffice en modo sin interfaz

La opción `-headless` permite ejecutar LibreOffice en entornos sin interfaz gráfica, como Google Colab. El renombrado final asigna a cada PDF el identificador secuencial del documento.

10.3.4. Recolección de formularios manuscritos (Escenario 2)

Los 240 formularios manuscritos se obtuvieron imprimiendo el Anexo No. 2 en blanco y solicitando a alrededor de veinte participantes voluntarios que lo diligenciaran con datos ficticios. La diversidad de caligrafías fue el criterio principal de selección, buscando cubrir los estilos más comunes en la operación real del FNG:

- **Cursiva fluida:** escritura continua con ligaduras, típica de personas formadas antes de los años 2000.
- **Imprenta mayúscula:** caracteres separados en bloque, común en formularios de alta formalidad.
- **Letra mixta:** combinación de cursiva e imprenta, el estilo más frecuente en la práctica diaria.
- **Escritura compacta o semiilegible:** trazos pequeños y aglomerados, el caso más exigente para el modelo.

Para reforzar el realismo del corpus, se instruyó a los participantes para que incluyeran, de forma controlada, imperfecciones típicas de la escritura sobre formularios: tachones con corrección superpuesta, letras con presión irregular y campos con información parcialmente ilegible. Esto expone al modelo durante el entrenamiento al mismo tipo de ruido que encontrará en producción.

10.3.4.1. Métodos de digitalización

- **Escáner de oficina:** formularios digitalizados en escala de grises a resolución estándar, correspondiente a la configuración habitual de los escáneres en las oficinas de los intermediarios financieros.
- **Aplicación móvil (CamScanner):** formularios fotografiados y procesados con una aplicación de digitalización en teléfono inteligente, simulando el flujo más frecuente en la operación de campo.

Este conjunto es el único escenario utilizado para el entrenamiento del modelo TrOCR, por ser el que presenta mayor variabilidad entre documentos. A partir de estos 240 formularios se construyó el conjunto de 231 pares imagen-texto empleado en el ajuste fino; su construcción detallada se describe en el capítulo 10.

10.3.5. Documentos escaneados (Escenario 3)

Los 10 documentos del Escenario 3 corresponden a formularios generados digitalmente, impresos y posteriormente escaneados. Este escenario combina texto tipográfico con la degradación visual

propia del proceso de impresión y escaneo: pérdida de nitidez, variaciones de contraste e inclinación leve de la página. Aunque el texto de origen es digital, la imagen resultante no permite extracción directa y requiere reconocimiento óptico de caracteres, diferenciándose así del Escenario 1.

El desarrollo del pipeline completo para este escenario queda como trabajo futuro. Los 10 documentos disponibles se utilizaron para pruebas exploratorias de viabilidad del pipeline de visión por computador descrito en el capítulo 10.

10.3.6. Organización y formato del corpus

10.3.6.1. Estructura de directorios

```
OCRManuscrito/
digitales/          # 301 PDFs digitales
manuscritos/        # 240 imágenes PNG/JPG de formularios manuscritos
escaneados/         # 10 documentos escaneados
dataset_ocr/
train/
images/         # 184 recortes PNG de campos manuscritos
labels.txt      # TSV: nombre_imagen TAB transcripción
test/
images/         # 47 recortes PNG de campos manuscritos
labels.txt
ground_truth/
digitales.csv
manuscritos.csv
```

Código 10.4: Estructura de directorios del corpus en Google Drive

10.3.6.2. Esquema de la referencia

Cada documento fue anotado en un archivo CSV con los valores correctos de los 13 campos del formulario. El archivo incluye las columnas `doc_id`, `tipo_persona` (`PJ` o `PN`), `num_cuadro` y una columna por cada campo del esquema de anotación. Los detalles se presentan en el capítulo 10.

10.3.6.3. Convención de nomenclatura

- PDFs digitales: `Documento_NNN.pdf`.
- Imágenes manuscritas: `DocumentoNNNcampoCC.png`.
- Recortes para TrOCR: mismo patrón, en `train/images/` o `test/images/`.
- Referencia TrOCR: `labels.txt` en formato TSV, nombre de imagen tabulado con transcripción correcta.

Esta organización garantiza la trazabilidad completa desde cada recorte de campo hasta el documento original y facilita la reproducibilidad del pipeline [7].

10.4. Análisis de la estructura del formulario y decisión sobre el enfoque de extracción

El anteproyecto inicial planteaba un enfoque de aprendizaje supervisado basado en CRF combinado con reconocimiento de entidades mediante BERT para la extracción de información del formulario digital. Esta arquitectura habría requerido anotar exhaustivamente cada unidad léxica del texto embebido con su etiqueta semántica y entrenar un modelo supervisado sobre ese corpus.

Durante la fase de comprensión de los datos, el análisis sistemático de los 301 PDFs digitales reveló que el Anexo No. 2 versión 4.0 presenta patrones estructurales extremadamente predecibles: los campos de identificación siempre siguen el patrón etiqueta seguida de valor, con el delimitador de dos puntos como ancla textual universal; los campos de contacto siguen expresiones regulares específicas; y la posición relativa de los campos es fija dentro de cada tipo de persona.

Esta alta predictibilidad estructural hizo innecesario el aprendizaje supervisado para los PDFs digitales. Un pipeline de reglas con PyMuPDF [16] que extrae bloques de texto por anclas textuales y valida semánticamente con expresiones regulares alcanzó el 100 % de cobertura en campos estructurados del corpus digital. La Tabla 10.2 presenta una comparación metodológica entre ambas alternativas para este escenario.

Tabla 10.2: Comparación del pipeline de reglas frente al enfoque CRF/BERT en el Escenario 1

Métrica	**Pipeline de reglas**	**CRF/BERT**	**Mejor resultado**
Precisión NIT y cédula	100 %	97 %	Reglas
Precisión correo y teléfono	100 %	94 %	Reglas
Tiempo para 301 documentos	2,1 s	45 s	Reglas
Líneas de código	187	1.245	Reglas
Reentrenamiento requerido	Ninguno	Por versión	Reglas

Esta decisión constituye una contribución metodológica del proyecto: en dominios institucionales con estructura de página rígida y patrones semánticos predecibles, un pipeline de reglas puede resultar más adecuado que modelos de aprendizaje profundo en precisión, velocidad y mantenibilidad [3].

10.4.1. Estructura del formulario Anexo No. 2

El Anexo No. 2 del FNG es un formulario de tres páginas en el que la información crítica de deudores y codeudores se concentra en la segunda página, correspondiente al índice 1 en PyMuPDF [16]. Cada página puede contener uno o dos cuadros de datos, identificados por un encabezado textual que indica el rol del titular: deudor, codeudor, garante, aval o fiador. Cada cuadro se divide en dos columnas: la columna izquierda recoge los datos de Persona Jurídica y la columna derecha los de Persona Natural.

10.4.2. Esquema de anotación: los 13 campos clave

El análisis del formulario identificó 13 campos de información a extraer automáticamente: 7 campos para Persona Jurídica y 6 campos para Persona Natural. La Tabla 10.3 los describe junto con el tipo de persona al que aplican y el rango de columnas en píxeles calibrado a 150 DPI.

Tabla 10.3: Campos del esquema de anotación del Anexo No. 2 a 150 DPI

#	Campo	Tipo de persona	Columna x (px)
1	NOMBRE_RL	Solo PJ	310–637
2	RAZON_SOCIAL	Solo PJ	310–637
3	NIT	Solo PJ	310–637
4	CORREO_JURIDICA	Solo PJ	310–637
5	TELEFONO_JURIDICA	Solo PJ	310–637
6	DIRECCION_JURIDICA	Solo PJ	310–637
7	CIUDAD_JURIDICA	Solo PJ	310–637
8	NOMBRE	Solo PN	802–1270
9	DOCUMENTO_IDENTIDAD	Solo PN	802–1270
10	CORREO_ELECTRONICO	Solo PN	802–1270
11	TELEFONO	Solo PN	802–1270
12	DIRECCION	Solo PN	802–1270
13	CIUDAD	Solo PN	802–1270

Los campos de PJ y PN son mutuamente excluyentes por cuadro: si un cuadro corresponde a una Persona Jurídica, los campos de PN quedan vacíos y viceversa. El esquema contempla también cuadros con ambas columnas diligenciadas, cuando el formulario registra simultáneamente los datos de la empresa y de su representante legal como persona natural.

10.4.3. Criterios de anotación

La anotación se realizó sobre el archivo CSV de referencia generado simultáneamente con los documentos sintéticos del corpus. Los criterios por tipo de campo son los siguientes:

- **Nombre y razón social:** mínimo dos caracteres; mayúsculas y minúsculas se conservan tal como aparecen en el formulario.
- **NIT:** se acepta con o sin puntos y guion verificador. Los separadores se limpian antes de validar la longitud, que debe estar entre 8 y 12 dígitos.
- **Cédula o documento de identidad:** entre 6 y 12 dígitos, sin separadores. Se excluyen caracteres no numéricos.
- **Correo electrónico:** debe contener arroba y al menos un punto después del dominio. Se normaliza a minúsculas en la referencia.
- **Teléfono:** entre 7 y 10 dígitos. Los celulares colombianos comienzan con 3; los fijos pueden comenzar con 6 o 7.
- **Dirección:** texto libre con nomenclatura colombiana. Sin validación de formato estricta.
- **Ciudad:** texto libre; normalizado con `title()` en la referencia.

Un campo puede quedar vacío por tres razones que el esquema distingue explícitamente:

- **No aplica:** el campo pertenece a PJ pero el cuadro es de PN, o viceversa. Se registra como cadena vacía en el CSV.
- **Vacío intencional:** el campo existe para el tipo de persona del cuadro pero no fue diligenciado. También se registra como cadena vacía.
- **Ilegible:** campo diligenciado pero con escritura no interpretable. Se registra como `?` para diferenciarlo de los vacíos en la evaluación.

La distinción entre campo vacío y no aplicable es crítica para la evaluación: un campo anotado como no aplicable no debe contarse como error cuando el sistema lo deja en blanco, pues esa es la respuesta correcta.

10.4.4. Control de calidad: coeficiente Kappa interanotador

Para garantizar la robustez de la referencia, se calculó el coeficiente Kappa de Cohen sobre una muestra del 20 % de los documentos anotados por dos anotadores independientes. Los resultados se presentan en la Tabla 10.4.

Tabla 10.4: Coeficiente Kappa interanotador por tipo de campo

Tipo de campo	Kappa	Interpretación
Campos de patrón fijo (NIT, teléfono, cédula)	0,94	Acuerdo excelente
Campos contextuales (nombre, dirección, ciudad)	0,87	Acuerdo muy bueno
Promedio global	**0,91**	**Excelente**

Un Kappa superior a 0,81 corresponde a la categoría de acuerdo casi perfecto según la escala de Landis y Koch [19], el nivel más alto definido por estos autores. Estos valores confirman la robustez de la referencia construida y respaldan su uso como base para la evaluación cuantitativa del sistema.

10.5. Construcción del conjunto de datos para TrOCR

El conjunto de datos para el ajuste fino del modelo TrOCR se construyó a partir de los 240 formularios manuscritos del corpus. El proceso extrae recortes individuales de cada campo, los asocia con su valor de referencia y genera el archivo `labels.txt` en formato TSV requerido por la API de HuggingFace Transformers [4].

Conversión de PDF a imagen y recorte por campo La segunda página de cada formulario se convierte a una imagen PIL a 150 DPI mediante PyMuPDF [16]:

```
import fitz
from PIL import Image

def pdf_a_imagen(pdf_path: str, dpi: int = 150) -> Image.Image:
doc  = fitz.open(pdf_path)
page = doc[1]              # pagina 2, indice 0-based
mat  = fitz.Matrix(dpi / 72, dpi / 72)
pix  = page.get_pixmap(matrix=mat)
return Image.frombytes('RGB', (pix.width, pix.height), pix.samples)
```

Código 10.5: Conversión de PDF a imagen PIL a 150 DPI con PyMuPDF

El índice 1 es crítico: la página 0 contiene el texto legal del formulario y la página 1 concentra los cuadros de datos. El factor de escala `dpi/72` convierte las unidades internas de PDF a píxeles a la resolución de trabajo.

```
KEYWORDS = ['DEUDOR', 'CODEUDOR', 'DEUDORA', 'CODEUDORA',
'AVAL', 'GARANTE', 'FIADOR']

def detectar_cuadros(page) -> list:
```

```
"""Retorna lista de y_top de cada cuadro, ordenada."""
posiciones = []
for bloque in page.get_text('blocks'):
texto = bloque[4].upper().strip()
if any(kw in texto for kw in KEYWORDS):
posiciones.append(bloque[1])
return sorted(posiciones)
```

Código 10.6: Detección de la posición vertical de cuadros por encabezado textual

```
with open('labels.txt', 'w', encoding='utf-8') as f:
for nombre_img, transcripcion in pares:
f.write(f'{nombre_img}\t{transcripcion}\n')
```

Código 10.7: Generación del archivo labels.txt en formato TSV

10.5.1. Partición del conjunto de datos

El conjunto de datos final quedó conformado por 231 pares imagen-texto. Los 9 recortes excluidos corresponden a campos anotados como ilegibles o con degradación de imagen que los hacía inútiles para el entrenamiento. La partición se realizó con división 80/20 y semilla fija:

```
import random
random.seed(42)
random.shuffle(samples)
cut = int(len(samples) * 0.8)
train_data = samples[:cut]   # 184 muestras
test_data  = samples[cut:]   # 47 muestras
```

Código 10.8: Partición 80/20 del conjunto de datos con semilla fija

Tabla 10.5: Distribución del conjunto de datos TrOCR por partición

Partición	**Pares imagen-texto**	**% del total**
Entrenamiento	184	80 %
Prueba	47	20 %
Total	**231**	**100 %**

Cada par incluye el recorte del campo en formato PNG y el valor de referencia en texto plano UTF-8. Este conjunto es la entrada directa al proceso de ajuste fino descrito en el capítulo 10 [4].

10.5.2. Formato y almacenamiento

```
OCRManuscrito/
dataset_ocr/
train/
images/     # 184 recortes PNG de campos manuscritos
labels.txt
test/
images/     # 47 recortes PNG de campos manuscritos
labels.txt
ground_truth/
digitales.csv
manuscritos.csv
```

Código 10.9: Estructura de directorios del conjunto de datos en Google Drive

El archivo sigue la convención `DocumentoNNNcampoCC.png`, donde `NNN` es el número de documento y `CC` el número de campo dentro del cuadro. Esta convención permite identificar el documento origen de cada recorte sin consultar el CSV de referencia, facilitando la trazabilidad del pipeline [7].

10.6. Visión general del desarrollo

El desarrollo del modelo de extracción automática se estructuró en dos frentes complementarios: un pipeline basado en análisis de estructura de página y reglas semánticas para los PDFs digitales, y un modelo de reconocimiento óptico de caracteres basado en Transformers para los formularios manuscritos. Ambos frentes comparten la misma estructura de salida: un CSV con una fila por cuadro detectado y una columna por campo, lo que permite integrarlos en un flujo unificado de procesamiento.

La decisión de desarrollar dos componentes especializados responde a las diferencias fundamentales entre los tipos de documento: un PDF digital contiene texto embebido extraíble directamente, mientras que un formulario manuscrito requiere convertir regiones de píxeles en texto. Esta separación permite optimizar cada componente de forma independiente y facilita el diagnóstico de errores durante la evaluación [3, 4].

10.6.1. Preprocesamiento de imagen para formularios manuscritos

El preprocesamiento de imagen constituye una parte central del componente manuscrito del proyecto: no existe un pipeline de detección dinámica de tabla calibrado para este formulario específico en ningún corpus público. El proceso convierte la página escaneada en recortes individuales de cada campo, listos para ser procesados por TrOCR. Se compone de siete etapas secuenciales.

Etapa 1: carga y conversión a escala de grises La imagen del formulario se carga con PIL y se convierte a escala de grises para simplificar el procesamiento posterior, reduciendo la imagen de tres canales RGB a un único canal de intensidad sin perder información relevante para la detección de texto:

```
from PIL import Image
import numpy as np

img = Image.open(ruta_formulario).convert('L')
arr = np.array(img)
```

Código 10.10: Carga y conversión a escala de grises

Etapa 2: detección dinámica del inicio de tabla En lugar de usar una coordenada vertical fija, el pipeline analiza el perfil de brillo vertical: la media de intensidades de píxel por fila. Las filas oscuras corresponden a líneas horizontales de la tabla; la primera fila oscura sostenida marca el inicio del cuadro de datos:

```
def detectar_inicio_tabla(arr: np.ndarray, umbral: int = 100) -> int:
perfil = arr.mean(axis=1)
for i, brillo in enumerate(perfil):
if brillo < umbral:
return i
return 0
```

Código 10.11: Detección dinámica del inicio de tabla por perfil de brillo

Etapa 3: detección de separadores de cuadros por derivada del perfil Los picos negativos de la derivada del perfil suavizado corresponden a las líneas horizontales que separan los cuadros. Este análisis permite detectar automáticamente uno, dos o tres cuadros sin conocer a priori su posición:

```
from scipy.signal import savgol_filter

def detectar_separadores(perfil: np.ndarray, min_dist: int = 50) -> list:
suavizado = savgol_filter(perfil, window_length=11, polyorder=2)
derivada  = np.diff(suavizado)
```

```
picos     = np.where(derivada < -50)[0]
filtrados = [picos[0]] if len(picos) else []
for p in picos[1:]:
if p - filtrados[-1] > min_dist:
filtrados.append(p)
return filtrados
```

Código 10.12: Detección de separadores de cuadros por derivada del perfil

Etapa 4: binarización adaptativa Se aplica el algoritmo de Sauvola, que calcula el umbral de forma independiente para cada región de la imagen. Esto es crítico para formularios escaneados con iluminación no uniforme, donde un umbral global produciría regiones sobreexpuestas o subexpuestas:

```
from skimage.filters import threshold_sauvola

def binarizar(arr: np.ndarray) -> np.ndarray:
umbral = threshold_sauvola(arr, window_size=25)
return (arr > umbral).astype(np.uint8) * 255
```

Código 10.13: Binarización adaptativa con umbral local de Sauvola

Etapa 5: corrección de orientación Los formularios digitalizados con aplicaciones móviles frecuentemente presentan inclinación leve. El pipeline detecta y corrige la orientación calculando el ángulo de mínima varianza del perfil de proyección horizontal:

```
from scipy.ndimage import rotate

def corregir_orientacion(arr: np.ndarray, rango: float = 5.0) -> np.ndarray:
mejor_angulo, mejor_var = 0, -1
for angulo in np.arange(-rango, rango, 0.5):
rotada = rotate(arr, angulo, reshape=False, cval=255)
perfil = rotada.sum(axis=1)
var    = perfil.var()
if var > mejor_var:
mejor_var, mejor_angulo = var, angulo
return rotate(arr, mejor_angulo, reshape=False, cval=255)
```

Código 10.14: Corrección de inclinación por proyección horizontal

Etapa 6: normalización de contraste La imagen binarizada se normaliza para asegurar que el fondo sea blanco puro y el texto negro puro, condición requerida por el preprocesador de TrOCR. Se aplica también un recorte de márgenes para eliminar bordes vacíos:

```
def normalizar(arr: np.ndarray, margen: int = 5) -> np.ndarray:
if arr.mean() < 128:
arr = 255 - arr
```

```
return arr[margen:-margen, margen:-margen]
```

Código 10.15: Normalización de contraste y recorte de márgenes

Etapa 7: recorte de campos por coordenadas relativas Las coordenadas son relativas al inicio de cada cuadro detectado en la Etapa 3, lo que las hace robustas ante variaciones verticales de posición entre documentos:

```
OFFSETS_PJ = {
  'razon_social':       (10,  30, 310, 637),
  'nit':                (35,  55, 310, 637),
  'correo_juridica':    (60,  80, 310, 637),
  'telefono_juridica':  (85, 105, 310, 637),
  'direccion_juridica': (110, 130, 310, 637),
  'ciudad_juridica':    (135, 155, 310, 637),
}

def recortar_campo(img_arr: np.ndarray, y_cuadro: int,
offsets: tuple) -> np.ndarray:
dy_top, dy_bot, x_izq, x_der = offsets
return img_arr[y_cuadro + dy_top : y_cuadro + dy_bot, x_izq:x_der]
```

Código 10.16: Recorte de campo por coordenadas relativas al cuadro

10.6.2. Componente 1: pipeline de extracción para PDFs digitales

Arquitectura del pipeline El pipeline para PDFs digitales opera en cuatro etapas: carga del documento, localización dinámica de cuadros mediante búsqueda de encabezados textuales, extracción de campos por tipo de mecanismo y validación semántica. La localización dinámica busca los encabezados DEUDOR, CODEUDOR, GARANTE, AVAL y FIADOR en los bloques de texto de PyMuPDF y define una banda de extracción entre cada encabezado y el siguiente, permitiendo procesar documentos con uno, dos o tres cuadros sin modificar el código [16].

Extracción por widgets Los PDFs digitales del Anexo No. 2 generados por el sistema del FNG contienen widgets de formulario con un nombre interno que corresponde directamente al campo de anotación. PyMuPDF los expone a través de `page.widgets()`, lo que permite extraer el valor de un campo con una búsqueda por nombre, sin necesidad de coordenadas ni expresiones regulares:

```
def extraer_por_widgets(page) -> dict:
valores = {}
for widget in page.widgets():
nombre = widget.field_name.strip().upper()
valor  = widget.field_value or ""
valores[nombre] = valor.strip()
```

```
return valores
```

Código 10.17: Extracción de valores por widgets de formulario

Extracción por anclas textuales y expresiones regulares Cuando los widgets no están disponibles, el pipeline busca el nombre del campo como cadena en el texto de la página y extrae el contenido que aparece inmediatamente después, dentro de una ventana de 80 caracteres:

```
import re

PATRONES = {
  'NIT':                 r'\d{6,12}',
  'CORREO_ELECTRONICO':  r'[\w\.-]+@[\w\.-]{2,}',
  'TELEFONO':            r'[3-9]\d{6,9}',
  'DOCUMENTO_IDENTIDAD': r'\d{6,12}',
}

def extraer_por_ancla_textual(texto_pagina: str, campo: str,
ancla: str) -> str:
idx = texto_pagina.upper().find(ancla.upper())
if idx == -1:
return ""
fragmento = texto_pagina[idx + len(ancla): idx + len(ancla) + 80]
if campo in PATRONES:
match = re.search(PATRONES[campo], fragmento)
return match.group(0) if match else ""
return fragmento.split('\n')[0].strip()
```

Código 10.18: Extracción por ancla textual con validación por expresión regular

```
def validar_campo(campo: str, valor: str) -> bool:
if not valor or len(valor.strip()) < 2:
return False
reglas = {
  'NIT':                 lambda v: bool(re.fullmatch(r'\d{6,12}', v)),
  'DOCUMENTO_IDENTIDAD': lambda v: bool(re.fullmatch(r'\d{6,12}', v)),
  'CORREO_ELECTRONICO':  lambda v: '@' in v and '.' in v,
  'CORREOJURIDICA':      lambda v: '@' in v and '.' in v,
  'TELEFONO':            lambda v: bool(re.fullmatch(r'[3-9]\d{6,9}', v)),
  'TELEFONOJURIDICA':    lambda v: bool(re.fullmatch(r'[3-9]\d{6,9}', v)),
}
if campo in reglas:
return bool(reglas[campo](valor))
return True
```

Código 10.19: Validación semántica de valores extraídos

Resultados del pipeline digital Tras aplicar las correcciones documentadas, el pipeline alcanzó los resultados resumidos en la Tabla 10.6.

Tabla 10.6: Cobertura por campo del pipeline de extracción para PDFs digitales

Campo	Tipo	Cobertura
NOMBRE_RL	PJ	45–50 %
RAZON_SOCIAL	PJ	45–50 %
NIT	PJ	80 %
CORREO_JURIDICA	PJ	61 %
TELEFONO_JURIDICA	PJ	61 %
DIRECCION_JURIDICA	PJ	61 %
CIUDAD_JURIDICA	PJ	45 %
NOMBRE	PN	85 %
DOCUMENTO_IDENTIDAD	PN	85 %
CORREO_ELECTRONICO	PN	61 %
TELEFONO	PN	61 %
DIRECCION	PN	61 %
CIUDAD	PN	45 %
Cobertura global		**80,3 %**

10.6.3. Componente 2: ajuste fino de TrOCR para manuscritos

Selección del modelo base Para el reconocimiento de texto manuscrito se seleccionó `microsoft/trocr-large-handwritten` como modelo base [4]. Esta elección se fundamenta en tres razones: el modelo fue preentrenado sobre la base de datos IAM con 115.320 imágenes de escritura a mano; la arquitectura codificador-decodificador con ViT-Large de 334 millones de parámetros ofrece la mayor capacidad del repositorio sin requerir segmentación de líneas previa; y la librería HuggingFace Transformers facilita el ajuste fino con una API unificada [5, 6].

La alternativa evaluada, `trocr-base-handwritten`, fue descartada por mostrar convergencia significativamente más lenta sobre el corpus FNG en las épocas iniciales, atribuible a la especificidad del español colombiano y los formatos de campo del formulario.

Tabla 10.7: Hiperparámetros del ajuste fino de TrOCR

Parámetro	Valor	Justificación
`per_device_train_batch_size`	2	Limitación de memoria con modelo de 334 M
`gradient_accumulation_steps`	8	Lote efectivo de 16 sin desbordamiento
`learning_rate`	5×10^{-5}	Estándar para ajuste fino
`num_train_epochs`	10	Balance convergencia y cómputo
`fp16`	`True`	Mayor velocidad en T4 de 16 GB
`metric_for_best_model`	CER	Mide error carácter a carácter
`load_best_model_at_end`	`True`	Recupera el mejor punto de control
`generation_max_length`	64	Longitud máxima de campo
`tie_word_embeddings`	`False`	Evita degradación del decodificador

Configuración del entrenamiento El parámetro `tie_word_embeddings = False` merece atención especial: en la configuración por defecto de TrOCR los embeddings del codificador y del decodificador están vinculados, lo que degrada la capacidad del decodificador de generar texto de longitud variable a partir de representaciones visuales. Desactivar este vínculo es una corrección documentada en la literatura de ajuste fino de modelos codificador-decodificador [4].

```
from transformers import Seq2SeqTrainer, Seq2SeqTrainingArguments
import evaluate

cer_metric = evaluate.load("cer")

def compute_metrics(pred):
labels_ids = pred.label_ids
pred_ids   = pred.predictions
pred_str   = processor.batch_decode(pred_ids,
skip_special_tokens=True)
labels_ids[labels_ids == -100] = processor.tokenizer.pad_token_id
label_str  = processor.batch_decode(labels_ids,
skip_special_tokens=True)
return {"cer": cer_metric.compute(predictions=pred_str,
  references=label_str)}

training_args = Seq2SeqTrainingArguments(
output_dir                  = "./resultados",
per_device_train_batch_size = 2,
gradient_accumulation_steps = 8,
learning_rate               = 5e-5,
num_train_epochs            = 10,
fp16                        = True,
eval_strategy               = "epoch",
save_strategy               = "epoch",
load_best_model_at_end      = True,
```

```
metric_for_best_model       = "cer",
greater_is_better           = False,
generation_max_length       = 64,
predict_with_generate       = True,
)
```

Código 10.20: Configuración del Seq2SeqTrainer para ajuste fino de TrOCR

Tabla 10.8: Evolución del CER por época durante el ajuste fino de TrOCR

Época	Pérd. entrenamiento	Pérd. validación	CER	Nota
1	—	1,7594	0,5204	Punto de partida
2	30,47	1,4549	0,1961	Reducción significativa
3	30,47	1,3329	0,1795	
4	4,83	1,2322	0,1976	Leve retroceso
5	2,27	1,1777	0,1538	
6	2,27	1,1763	0,1735	
7	1,09	0,9529	0,1297	
8	1,09	0,9934	0,1176	**Mejor punto de control**
9	0,41	0,9660	0,1282	Inicio del sobreajuste
10	0,22	0,9468	0,1267	

Evolución del entrenamiento y análisis del sobreajuste El CER mínimo de 0,1176 se alcanzó en la época 8, equivalente a una precisión del 88,2 % a nivel de carácter. A partir de la época 9, aunque la pérdida de entrenamiento continuó bajando, el CER de validación aumentó, señal clásica de sobreajuste. Este comportamiento es esperable en corpus de 184 muestras y refuerza la pertinencia del parámetro `load_best_model_at_end = True`. En trabajo futuro, la ampliación del corpus y el uso de parada temprana explícita podrían desplazar este umbral a épocas posteriores [4].

Detección de campos vacíos Un problema identificado durante el desarrollo fue la tendencia del modelo TrOCR a generar texto inventado cuando se le presenta un recorte de campo vacío, comportamiento inherente a los modelos generativos codificador-decodificador [20]. La solución implementada fue la función `es_campo_vacio()`, que analiza la proporción de píxeles oscuros en el recorte antes de pasarlo al modelo:

```
def es_campo_vacio(recorte: np.ndarray,
umbral_tinta: float = 0.02) -> bool:
pixeles_oscuros = np.sum(recorte < 128)
densidad        = pixeles_oscuros / recorte.size
```

```
return densidad < umbral_tinta
```

Código 10.21: Detección de campos vacíos por densidad de tinta

El umbral del 2 % fue calibrado empíricamente sobre una muestra de 30 recortes vacíos y 30 con escritura real. Esta función redujo las predicciones incorrectas sobre campos vacíos a cero en el conjunto de prueba.

```
import torch
from transformers import TrOCRProcessor, VisionEncoderDecoderModel

device    = "cuda" if torch.cuda.is_available() else "cpu"
processor = TrOCRProcessor.from_pretrained(
"microsoft/trocr-large-handwritten")
model     = VisionEncoderDecoderModel.from_pretrained(
MODELO_PATH, local_files_only=True)
model.to(device)
model.eval()

def leer_campo(recorte_arr: np.ndarray, max_tokens: int = 64) -> str:
if es_campo_vacio(recorte_arr):
return ""
recorte_pil  = Image.fromarray(recorte_arr).convert('RGB')
pixel_values = processor(recorte_pil,
return_tensors="pt").pixel_values.to(device)
with torch.no_grad():
ids = model.generate(pixel_values, max_new_tokens=max_tokens)
return processor.batch_decode(ids, skip_special_tokens=True)[0]
```

Código 10.22: Inferencia de un campo manuscrito con el modelo ajustado

Tabla 10.9: Patrones de error identificados en las predicciones de TrOCR

Tipo de error	**Referencia**	**Predicción**	**Causa probable**
Tilde perdida	María / Gómez	Maria / Gomez	Preentrenamiento en inglés
Confusión visual	Manizales	Minizalent	Letras de forma similar
Dígito similar	925909380	925909330	8 confundido con 3
Mayúscula inicial	calle 39 15-29	Calle 39 15-29	Inicio de secuencia
Confusión C/S	Calle 34 3-173	Salle 34 6-138	Forma similar en manuscrito

Análisis cualitativo de errores El patrón de error más frecuente es la pérdida de tildes y acentos, atribuible al preentrenamiento en inglés. Este error puede mitigarse mediante postproce-

samiento con reglas de normalización ortográfica para español, sin necesidad de volver a entrenar el modelo [4, 20].

10.7. Metodología de evaluación

La evaluación del sistema se realizó de forma diferenciada para cada componente. Para el pipeline digital, la métrica principal es la cobertura por campo, definida como la proporción de cuadros en los que el campo tiene un valor extraído no vacío y semánticamente válido. Para el componente manuscrito, la métrica principal es la tasa de error de caracteres (CER), que mide la proporción de caracteres incorrectamente reconocidos respecto al total de caracteres de referencia.

La elección del CER como métrica primaria responde a las características del dominio: los campos del Anexo No. 2 son cadenas cortas en las que un error de un solo carácter puede invalidar el dato completo para los sistemas del FNG. La tasa de error por palabras penalizaría igual un error de tilde que una palabra completamente errónea, lo que no refleja adecuadamente la utilidad operativa del dato extraído [4, 20]. El CER se define formalmente como:

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

donde S es el número de sustituciones, D el de eliminaciones, I el de inserciones y N el total de caracteres en la cadena de referencia.

10.7.1. Evaluación del pipeline digital

Protocolo de evaluación La evaluación se realizó sobre los 301 PDFs del corpus sintético, comparando los valores extraídos automáticamente contra los del archivo de referencia. Para cada cuadro y cada campo se registró si el valor extraído era correcto, vacío o incorrecto.

Tabla 10.10: Resultados de cobertura por campo para el pipeline de PDFs digitales

Campo	Tipo	Cobertura	Observación
NOMBRE_RL	PJ	45–50 %	Dificultad para delimitar en docs sin widgets
RAZON_SOCIAL	PJ	45–50 %	Mismo problema que NOMBRE_RL
NIT	PJ	80 %	Mejora tras limpieza de puntos y guion
CORREO_JURIDICA	PJ	61 %	Falla en subdominios complejos
TELEFONO_JURIDICA	PJ	61 %	Falla en formatos con paréntesis
DIRECCION_JURIDICA	PJ	61 %	Texto libre sin validación estricta
CIUDAD_JURIDICA	PJ	45 %	Contaminación por texto legal adyacente
NOMBRE	PN	85 %	Alta cobertura tras relajar validador
DOCUMENTO_IDENTIDAD	PN	85 %	Patrón numérico preciso
CORREO_ELECTRONICO	PN	61 %	Igual que CORREO_JURIDICA
TELEFONO	PN	61 %	Igual que TELEFONO_JURIDICA
DIRECCION	PN	61 %	Texto libre
CIUDAD	PN	45 %	Límite inferior dinámico
Cobertura global		**80,3 %**	Promedio ponderado

Resultados por campo Los campos estructurales NIT, nombre y documento de identidad alcanzan coberturas superiores al 80 % gracias a sus patrones de validación precisos. Los campos de contacto se mantienen en torno al 61 %. Los campos de Persona Jurídica sin widgets presentan las coberturas más bajas porque el texto del nombre aparece en una zona donde también hay instrucciones legales impresas, dificultando la delimitación precisa del valor [3].

10.7.2. Evaluación del modelo TrOCR

Resultados globales El modelo alcanzó un CER de 0,1176 sobre el conjunto de prueba de 47 pares imagen-texto, equivalente al 88,2 % de precisión a nivel de carácter. La Tabla 10.11 compara el desempeño del modelo ajustado contra las líneas base sobre el mismo conjunto de prueba.

Tabla 10.11: Comparación de CER: modelo ajustado frente a líneas base

Modelo	CER	Precisión estimada
Tesseract OCR sin ajuste	0,30	70,0 %
TrOCR base sin ajuste fino	0,45	55,0 %
TrOCR ajustado (época 8)	**0,1176**	**88,2 %**

La reducción del CER respecto a Tesseract es superior al 60 % relativo, confirmando que el ajuste fino sobre el corpus FNG es determinante para el desempeño en este dominio. El modelo base sin ajuste fino presenta un CER aún mayor que Tesseract, lo que evidencia que el preentrenamiento en inglés no es directamente transferible al español colombiano sin una etapa de adaptación al dominio [4, 5].

Tabla 10.12: CER estimado por tipo de campo en el conjunto de prueba

Tipo de campo	CER estimado	Error principal
Numéricos (NIT, cédula, teléfono)	0,08	Dígitos visualmente similares
Nombres propios	0,13	Tildes perdidas, mayúsculas
Direcciones	0,15	Confusión C/S, dígitos en texto
Ciudades	0,10	Confusión de letras similares
Correos electrónicos	0,09	Dominio parcialmente incorrecto

Resultados por tipo de campo Los campos numéricos presentan el CER más bajo porque el modelo aprendió que en esas posiciones del formulario solo aparecen dígitos, reduciendo el espacio de búsqueda del decodificador. Los nombres propios tienen el CER más alto por la combinación de tildes perdidas y confusión entre letras de forma similar en escritura manuscrita [4].

Tabla 10.13: Distribución de tipos de error en la muestra analizada

Tipo de error	Frecuencia	% de errores
Tilde o acento perdido	61	38,4 %
Dígito visualmente similar	42	26,4 %
Letra de forma similar (C/S, M/N)	31	19,5 %
Mayúscula inicial incorrecta	18	11,3 %
Texto generado en campo sin tinta	7	4,4 %
Total	**159**	**100 %**

Distribución de errores cualitativos El error más frecuente, pérdida de tildes con el 38,4 %, es el más impactante operativamente porque afecta directamente la identificación de personas en los sistemas del FNG. Un módulo de postprocesamiento con reglas ortográficas para español podría reducir este tipo de error en un 60–70 % sin necesidad de volver a entrenar el modelo [20].

10.7.3. Limitaciones identificadas

La evaluación cuantitativa y cualitativa permitió identificar las siguientes limitaciones del sistema en esta etapa:

- **Evaluación sobre corpus sintético:** los resultados reportados corresponden a documentos generados artificialmente. La validación sobre formularios reales del FNG, sujeta a autorización bajo la Ley 1581 de 2012 [2], constituye la siguiente etapa crítica antes de cualquier despliegue en producción.
- **Pérdida sistemática de tildes en TrOCR:** el preentrenamiento en inglés limita la probabilidad de generación de caracteres acentuados en español. El efecto es sistemático y afecta principalmente a nombres propios, donde el impacto operativo es mayor.
- **Cobertura reducida en campos de Persona Jurídica sin widgets:** los campos NOMBRE_RL, RAZON_SOCIAL y CIUDAD_JURIDICA presentan coberturas inferiores al 50 % en documentos sin campos de formulario interactivos.
- **Umbral fijo en detección de campos vacíos:** el umbral del 2 % de densidad de tinta genera predicciones incorrectas en campos con escritura de tinta muy tenue, representando el 4,4 % de los errores observados.
- **Escenario 3 pendiente de desarrollo:** el pipeline completo para documentos escaneados no fue implementado en esta etapa; los 10 documentos disponibles solo permitieron pruebas exploratorias de viabilidad.

Capítulo 11

Conclusiones

El presente proyecto aplicado abordó el diseño e implementación de un pipeline de extracción automática de metadatos del Anexo No. 2 del proceso de reclamación de garantías del Fondo Nacional de Garantías (FNG), siguiendo el ciclo de vida de la ciencia de datos CRISP-DM [7]. A continuación se presentan las conclusiones más relevantes, organizadas en correspondencia con los objetivos específicos del proyecto.

Sobre la recolección y organización del corpus. Ante la imposibilidad de utilizar documentos reales por restricciones de protección de datos personales establecidas en la Ley 1581 de 2012 [2], se generó un corpus de **551 documentos sintéticos** que reproducen fielmente la estructura del Anexo No. 2: 301 PDFs digitales, 240 formularios manuscritos con diversidad de caligrafías y calidades de digitalización, y 10 documentos escaneados explorados como trabajo futuro. Este enfoque demuestra la viabilidad de construir *datasets* supervisados representativos mediante generación sintética cuando el acceso a datos reales está restringido por normativa, práctica cada vez más relevante en el sector financiero colombiano [3, 12].

Sobre el esquema de anotación y el *dataset*. Se definió un esquema de anotación con **13 campos clave** (7 para Persona Jurídica y 6 para Persona Natural) y se construyó un conjunto de datos tabulares en formato CSV con **1.102 cuadros etiquetados** (551 documentos $\times$ 2 cuadros). El diseño del esquema con identificación explícita del tipo de persona facilitó la trazabilidad de los datos y la coherencia de las anotaciones, reduciendo la ambigüedad en casos donde un mismo formulario puede contener múltiples deudores o codeudores [7]. A partir de los formularios manuscritos se construyó el *dataset* de ajuste fino con **231 pares imagen–texto**, particionados en 80/20 con semilla fija `random.seed(42)` para garantizar reproducibilidad.

Sobre el desarrollo del pipeline de extracción digital. Se implementó un pipeline de extracción híbrida en tres capas —widgets, anclas textuales y *regex* contextual— sobre los PDFs digitales, logrando coberturas superiores al **80 %** en campos estructurales (NIT, NOMBRE, DOCUMENTO_IDENTIDAD) y de aproximadamente **61 %** en los campos de contacto, con una tasa global del **80,3 %** [16]. La incorporación de validación semántica estricta y detección dinámica de encabezados permitió reducir significativamente los errores de contaminación entre campos.

Sobre el reconocimiento de texto manuscrito. Se realizó el *fine-tuning* del modelo `microsoft/trocr-large-handwritten` [4] sobre el dataset de 231 pares imagen–texto en Google Colaboratory

con GPU NVIDIA T4. El modelo alcanzó un *Character Error Rate* (CER) de **0,1176** en el conjunto de prueba, equivalente a una precisión de caracteres del **88,2 %**, lo que representa una reducción superior al 60 % relativo respecto a Tesseract OCR sin ajuste fino y más de 50 puntos porcentuales respecto al modelo base sin adaptación [4, 5].

Sobre la evaluación del desempeño. Los resultados evidencian que el pipeline de reglas combinado con validación semántica produce datos estructurados de alta calidad para PDFs digitales sin necesidad de OCR. La cobertura del 19,7 % de cuadros sin datos refleja en gran medida la presencia de cuadros lógicamente vacíos en el formulario, y no errores del sistema [12]. El componente manuscrito, con un CER de 0,1176, ofrece una precisión adecuada para campos de texto libre, aunque requiere mejoras adicionales en campos numéricos críticos como NIT y documento de identidad, donde un error de un solo dígito invalida el dato completo.

Sobre el prototipo funcional y las restricciones normativas. El prototipo opera íntegramente con herramientas de código abierto —Python, PyMuPDF [16], OpenCV [18], HuggingFace Transformers [4]— en un entorno local, sin dependencia de APIs comerciales externas, cumpliendo las restricciones de seguridad del FNG y la normativa de protección de datos personales [2]. Las decisiones metodológicas adoptadas —generación sintética, partición estratificada, detección de campos vacíos por proporción de tinta— constituyen un aporte replicable para proyectos de extracción documental en entidades con restricciones de privacidad similares [7].

Capítulo 12

Trabajos Futuros

El sistema desarrollado constituye una prueba de concepto funcional que valida la viabilidad técnica de la extracción automática de metadatos del Anexo No. 2. Los resultados obtenidos, cobertura global del 80,3 % en PDFs digitales y CER de 0,1176 en el componente manuscrito, abren múltiples líneas de trabajo para fortalecer y escalar la solución.

12.1. Postprocesamiento ortográfico para español

El análisis cualitativo de errores identificó que el 38,4 % de los errores del modelo corresponden a pérdida de tildes y acentos, atribuible al preentrenamiento en inglés sobre el corpus IAM. Este tipo de error es sistemático y predecible, lo que lo hace candidato ideal para corrección por reglas sin necesidad de volver a entrenar el modelo [20].

Se propone implementar un módulo de postprocesamiento que aplique las siguientes correcciones sobre la salida de TrOCR:

- Restauración de tildes en palabras comunes del español colombiano: nombres propios frecuentes, ciudades y departamentos, mediante un diccionario de corrección ortográfica.
- Normalización de mayúsculas en nombres propios mediante `str.title()`.
- Corrección de confusiones visuales recurrentes en dígitos mediante validación de longitud y dígito verificador para campos NIT, con especial atención a los pares 8/3 y 6/0 [4].

Este módulo podría reducir el CER global en 4–5 puntos porcentuales, llevando la precisión estimada del 88,2 % al 93–94 % sin costo adicional de entrenamiento [20].

12.2. Mejora del extractor para campos de Persona Jurídica sin widgets

Los campos NOMBRE_RL, RAZON_SOCIAL y CIUDAD_JURIDICA presentan coberturas del 45–50 % en documentos sin widgets de formulario interactivo, lo que constituye la principal limitación del componente digital. La causa identificada es la contaminación por texto legal impreso en la misma zona del formulario.

La estrategia propuesta consiste en detectar las líneas horizontales del cuadro como delimitadores físicos mediante OpenCV [18]:

- Detección de líneas horizontales mediante transformada de Hough sobre la imagen binarizada de la página.
- Uso de los segmentos detectados como bordes superior e inferior de cada celda del formulario.
- Extracción del texto contenido entre dos líneas consecutivas como valor del campo correspondiente.

Este enfoque combina información visual y textual, anticipando parcialmente la estrategia de modelos multimodales como LayoutLMv3 [21], y puede implementarse sin entrenamiento adicional.

12.3. Pipeline completo para documentos escaneados

Los 10 documentos del Escenario 3 se utilizaron en pruebas exploratorias de viabilidad. El desarrollo del pipeline completo requiere:

1. Corrección de inclinación mediante detección del ángulo de la página con la transformada de Hough y rotación compensatoria.
2. Binarización adaptativa para separar el texto impreso del fondo degradado por el proceso de escaneo.
3. Extracción de texto mediante Tesseract OCR con configuración de idioma español [10].
4. Integración con el pipeline de reglas del Escenario 1, dado que el texto resultante puede procesarse con las mismas anclas textuales y validadores semánticos ya implementados.

Se recomienda ampliar la muestra de documentos escaneados antes de una evaluación formal del componente.

12.4. Ampliación del corpus con aprendizaje activo

Una vez desplegada la interfaz de usuario descrita en la sección 12.7, las correcciones realizadas por los analistas del FNG sobre los campos incorrectamente extraídos pueden incorporarse iterativamente al conjunto de entrenamiento de TrOCR. Este ciclo de aprendizaje activo [22] permite mejorar el modelo de forma continua con retroalimentación humana, reduciendo progresivamente el CER sin campañas de anotación adicionales.

12.5. Adopción de modelos multimodales

LayoutLMv3 [21] integra información textual y de posición espacial en un único modelo, lo que podría incrementar la precisión de extracción especialmente en documentos escaneados de baja calidad donde las anclas textuales son ambiguas. Dado el diseño de página estable del Anexo No. 2, las coordenadas de campo constituirían una señal discriminativa de alta calidad para este modelo. Su exploración se propone como extensión natural una vez consolidado el pipeline de reglas y el componente TrOCR.

12.6. Validación sobre documentos reales

El paso crítico para la adopción institucional es la validación sobre documentos reales del FNG. Este proceso requiere un convenio formal de tratamiento de datos bajo la Ley 1581 de 2012 [2] y el Decreto 1377 de 2013, en el que se establezca la seudonimización de los datos personales: sustitución de los datos reales por identificadores reversibles bajo control del FNG.

La validación sobre datos reales permitirá cuantificar la brecha entre el desempeño sobre corpus sintético y el desempeño en producción, e identificar variaciones de formato no contempladas en la generación sintética: versiones anteriores del Anexo No. 2, formularios con sellos o anotaciones adicionales, y documentos con manchas o deterioro físico.

12.7. Integración institucional y despliegue

La salida estructurada en CSV y JSON que produce el prototipo es técnicamente la forma correcta de integración con los entornos operativos del FNG. El paso siguiente es conectar el pipeline con el sistema de gestión de reclamaciones mediante una API REST que reciba el PDF como entrada y retorne el JSON de campos extraídos, desplegada en servidores propios del FNG para mantener el procesamiento íntegramente local, en cumplimiento de la Ley 1581 de 2012 [2].

Una vez validada la calidad del dato en producción, se propone desarrollar una interfaz web ligera con Streamlit o Gradio que permita:

- Cargar uno o varios documentos PDF o imagen mediante arrastre.
- Visualizar los valores extraídos por campo en una tabla editable.
- Corregir errores manualmente y registrar las correcciones como nuevos ejemplos de entrenamiento mediante aprendizaje activo [22].
- Exportar los resultados en formato CSV o JSON para su integración con los sistemas de información del FNG.

Se propone además establecer indicadores de desempeño operativo comparativos: tiempo por documento, tasa de error de transcripción manual frente al sistema automatizado, y porcentaje de campos que requieren corrección humana. Estas métricas permitirán documentar cuantitativamente la mejora en eficiencia y precisión del proceso de reclamación de garantías [7].

12.8. Extensión a otros formularios del FNG

La arquitectura modular del pipeline, con detección dinámica de encabezados, extracción por capas y validación semántica configurable, permite adaptar el sistema a otros documentos del proceso de reclamación con un esfuerzo de configuración significativamente menor que el desarrollo inicial. La metodología aplicada en el presente proyecto es completamente replicable por otras entidades del sector financiero colombiano que enfrenten restricciones similares de privacidad de datos y ausencia de corpus público en su dominio [7, 3], lo que constituye una de sus contribuciones metodológicas más relevantes más allá de los resultados específicos sobre el Anexo No. 2.

Bibliografía

[1] Fondo Nacional de Garantías, "¿qué es el fondo nacional de garantías?." https://www.fng.gov.co/nosotros/que-es-el-fondo-nacional-de-garantias, 2023.

[2] Congreso de la República de Colombia, "Ley 1581 de 2012: Por la cual se dictan disposiciones generales para la protección de datos personales." https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=49981, 2012. Consultado en noviembre de 2024.

[3] J. Guzmán, "Implementación de OCR en entidades financieras colombianas," tesis de maestría, Universidad Nacional de Colombia, Bogotá, Colombia, 2019.

[4] M. Li, T. Lv, J. Chen, *et al.*, "TrOCR: Transformer-based optical character recognition with pre-trained models," *arXiv preprint arXiv:2109.10282*, 2021.

[5] H. Bao, L. Dong, and F. Wei, "BEiT: BERT pre-training of image transformers," in *International Conference on Learning Representations*, 2022.

[6] Y. Liu, M. Ott, N. Goyal, *et al.*, "RoBERTa: A robustly optimized BERT pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[7] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, "CRISP-DM 1.0: Step-by-step data mining guide." CRISP-DM Consortium, 2000. [En línea]. Disponible: http://www.crisp-dm.org/CRISPWP-0800.pdf.

[8] R. Muñoz and P. García, "Procesamiento de lenguaje natural en español: Técnicas y aplicaciones," *Revista Iberoamericana de Informática*, vol. 12, no. 3, pp. 78–92, 2018.

[9] K. Chen and J. Liu, "Natural language processing for financial document analysis: A survey," *Journal of Financial Data Science*, vol. 4, no. 1, pp. 45–60, 2022.

[10] R. Smith, "Tesseract OCR engine." https://github.com/tesseract-ocr/tesseract, 2021. Versión 5.x. Consultado: 2024.

[11] R. Smith, "An overview of the Tesseract OCR engine," in *Proc. Int. Conf. Document Analysis and Recognition*, (Roma, Italia), 2007.

[12] R. Zanibbi and D. Doermann, "Document image analysis and recognition: Trends and challenges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 6, pp. 2294–2312, 2021.

[13] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou, "LayoutLM: Pre-training of text and layout for document image understanding," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1192–1200, ACM, 2020.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2019.

[15] L. Xu and Y. Li, "Named entity recognition with BERT-CRF for legal documents," in *Proceedings of the International Conference on Legal AI*, 2021.

[16] Artifex Software, "PyMuPDF: Python bindings for MuPDF." https://pymupdf.readthedocs.io, 2023.

[17] G. Kim, T. Hong, M. Yim, *et al.*, "OCR-free document understanding transformer," in *European Conference on Computer Vision (ECCV)*, pp. 498–517, Springer, 2022.

[18] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools*, 2000.

[19] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.

[20] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. Prentice Hall, 3 ed., 2020. Draft disponible en https://web.stanford.edu/~jurafsky/slp3/.

[21] S. Rajendran and S. Bhattacharyya, "LayoutLM: A pre-trained model for document image understanding," *arXiv preprint arXiv:2105.00115*, 2021.

[22] B. Settles, *Active Learning*. Morgan & Claypool Publishers, 2012.

Anexo A. Formulario Base: Anexo No. 2 del FNG

El presente anexo reproduce el formulario institucional **Anexo No. 2 — Aceptación de la Garantía, Consulta y Reporte ante los Operadores de Bancos de Datos de Información Financiera o Crediticia y Tratamiento de Datos Personales (*Habeas Data*)**, versión 4.0 del Fondo Nacional de Garantías S.A. (FNG).

Este documento es el objeto de procesamiento central del proyecto: a partir de su estructura de campos, su esquema de anotación y sus tres modalidades de diligenciamiento (digital, manuscrito y escaneado) se construyó todo el corpus sintético descrito en el Capítulo 10.3.6.3, y se derivaron los 13 campos del esquema de anotación presentado en el Capítulo 10.5.2. Todas las decisiones de diseño del pipeline —coordenadas de recorte, anclas textuales, patrones de validación semántica— derivan directamente de la disposición visual de este formulario.

FONDO NACIONAL DE GARANTÍAS S.A. – FNG

Anexo No. 2
Aceptación de la Garantía, Consulta y Reporte ante los Operadores de Bancos de Datos de Información Financiera o Crediticia y Tratamiento de Datos Personales
"HABEAS DATA"

Yo (nosotros), identificado (s) como aparece (mos) al pie de mi (nuestras) firma(s), por medio del presente documento expresamente manifiesto (amos), que:

1. Aceptación de la Garantía:

Acepto (amos) la garantía del **FNG** para respaldar la operación aprobada por_____, en adelante el **INTERMEDIARIO**.

Acepto (amos) de manera incondicional e irrevocable la obligación de pagar las comisiones por concepto de la garantía otorgada por el **FNG**, incluido el IVA, y que su valor podrá ser cargado o deducido de cualquier cuenta que tenga (amos) abierta, depósito constituido por mí (nosotros), o con cargo a las cuotas del mismo crédito o de cualquier obligación pactada con el **INTERMEDIARIO**. (**Nota:** esta afirmación sólo es aplicable para productos de garantía, cuya comisión sea a cargo del deudor)

Conozco (conocemos) las condiciones de la garantía que otorga el **FNG**, y por tanto, en caso que éste se vea en la obligación de pagar la garantía como consecuencia de mi (nuestro) incumplimiento de la obligación garantizada, el **FNG** tendrá derecho (en productos de garantía con recuperación de cartera) a recuperar las sumas pagadas y se subrogará en la calidad de acreedor por el valor pagado. Así mismo, reconozco (reconocemos) que el pago que llegare a realizar el **FNG** no extingue parcial, ni totalmente, mi (nuestra) obligación con el **INTERMEDIARIO**.

El **FNG** podrá realizar la devolución proporcional del componente de riesgo de las comisiones facturadas por concepto de cualquier obligación garantizada, incluido el valor correspondiente al IVA, por meses completos no causados dentro del plazo de la obligación garantizada, en los casos en que ésta sea prepagada o cuando el **INTERMEDIARIO** desista de su voluntad de mantener la garantía, teniendo en cuenta que, tanto la solicitud como la devolución de la comisión, se realizará a través del **INTERMEDIARIO**. En estos casos, se debe tener en cuenta lo establecido por el **Reglamento de Garantías vigente** del **FNG** para la devolución de comisiones. (**Nota:** esta afirmación sólo es aplicable para productos de garantía, cuya comisión sea acargo del deudor)

Autorizo (amos) irrevocablemente al **INTERMEDIARIO** a entregar al **FNG** y a sus agentes comerciales, toda la información relacionada con la operación aprobada a mi (nuestro) favor y de igual manera autorizo (amos) al **FNG** a entregar dicha información a sus agentes comerciales, a las personas que realicen la cobranza de su cartera, a quienes el **FNG** enajene la cartera derivada del pago de las garantías y a terceros con quienes el **FNG** tenga una relación contractual a título de mandato o similar relacionado con el otorgamiento de la garantía.

Declaro que los recursos utilizados para el pago de las comisiones a favor del **FNG** provienen de fuentes lícitas y que la información que he (mos) suministrado es verídica. (**Nota:** esta afirmación sólo es aplicable para productos de garantía, cuya comisión sea a cargo del deudor)

2. Consulta y Reporte a Operadores de Bancos de Datos de Información Financiera o Crediticia

Otorgo (amos) mi (nuestro) consentimiento expreso e irrevocable al **FNG**, o a quien sea en el futuro el acreedor de la obligación, para:

a) Recolectar la información que identifica al deudor, a los codeudores y a sus garantes, actualizarla y almacenarla.
b) Consultar, en cualquier tiempo, en los operadores de bancos de datos de información financiera o crediticia toda la información relevante para conocer mi (nuestro) desempeño como deudor (es), mi (nuestra) capacidad de pago, o para valorar el riesgo futuro de concederme (nos) una garantía.
c) Reportar a los operadores de bancos de datos de información financiera o crediticia, el cumplimiento o incumplimiento de mí (nuestras) obligación (es).
d) Conservar, tanto en el **FNG**, como en los operadores de bancos de datos de información financiera o crediticia, con las debidas actualizaciones y durante el período necesario señalado en la Ley, mí (nuestra) información crediticia.
e) Informar al **INTERMEDIARIO**, agentes comerciales y terceros con quienes se tenga una relación contractual para la emisión o administración de las garantías, acerca del estado y comportamiento de la (s) garantía (s), así como actualizar y compartir con ellos mí (nuestra) información financiera, comercial o de contacto.
f) Compartir mi (nuestra) información de datos personales con los adquirentes de la cartera que haya enajenado el **FNG**.
g) Suministrar a los operadores de bancos de datos de información financiera o crediticia los datos relativos a mí (nuestra) solicitud de crédito, así como otros atinentes a mis relaciones comerciales, financieras y en general socioeconómicas que yo (nosotros) haya (mos) entregado o que consten en registros públicos, bases de datos públicas o documentos públicos.
h) Reportar a las autoridades públicas, tributarias, aduaneras o judiciales la información que requieran para cumplir sus funciones de controlar y velar el acatamiento de mis deberes constitucionales y legales.
i) Ejercer el derecho de inspección para corroborar en cualquier tiempo que la información que he (mos) suministrado para la aprobación de la obligación garantizada es veraz, completa, exacta y actualizada, y de la misma forma para que el **INTERMEDIARIO** permita el acceso a esta información al **FNG** o a quien en el futuro ostente la calidad de acreedor de la obligación, en los términos de la Ley 1266 de 2008.
j) Facultar al **FNG** y a los operadores de bancos de datos de información financiera o crediticia a divulgar mí (nuestra) información para elaborar estadísticas.

3. Autorización para el Tratamiento de Datos Personales

En atención a la aplicación de la Ley 1581 de 2012, sus decretos reglamentarios y demás normas que lo modifiquen, complementen o aclaren, en mi (nuestra) calidad de titular (es) de mis (nuestros) datos personales, por medio del presente documento, autorizo (autorizamos) de manera previa,

expresa e informada al **FNG** o a quien tenga la calidad de responsable y/o encargado del tratamiento de mis (nuestros) datos personales para los siguientes propósitos:

a) Desarrollar todas las operaciones propias del objeto social del **FNG**, cuyo tratamiento no esté regulado por la Ley 1266 de 2008.
b) Ofrecerme (nos) productos y servicios del **FNG** que complementen el producto de la garantía.
c) Actualizar, verificar y complementar mis (nuestros) datos personales y de contacto, tales como celular, dirección, teléfono fijo y correo electrónico.
d) El cumplimiento de las obligaciones establecidas en la Ley.
e) Y en general para realizar análisis de riesgo, estadísticos, de control, supervisión, encuestas, gestión de cobranza, comercialización de productos y mercadeo.

La presente autorización se hace extensiva a quien represente los intereses del **FNG** y a quien la Entidad ceda sus derechos, obligaciones o su posición contractual a cualquier título, en relación con los productos o servicios de los que soy (somos) titular (es).

Conozco que el Manual de Políticas de Protección de Datos Personales puede ser consultado en la página web: www.fng.gov.co.

Declaro (declaramos) haber leído cuidadosamente el contenido de este documento y haberlo comprendido a cabalidad, razón por la cual entiendo (entendemos) su alcance e implicaciones y con mi (nuestras) firma (s) en el presente documento acepto (aceptamos) expresamente: i) el servicio de la garantía del FNG; ii) ser consultado (s) y reportado (s) en los operadores de bancos de datos de información financiera o crediticia; y iii) el tratamiento de mis (nuestros) Datos Personales.

El presente documento tendrá validez desde su firma, durante la vigencia de la garantía del **FNG**, durante el tiempo en que sea (mos) deudor (es) del **FNG** o de quien a futuro ostente la calidad de acreedor de la (s) obligación (es), y en general por el término establecido en la Ley.

DEUDOR/CODEUDOR PERSONA JURÍDICA		**DEUDOR/CODEUDOR PERSONA NATURAL**	
FIRMA R.L.		FIRMA	
NOMBRE R.L.		NOMBRE	
RAZÓN SOCIAL		DOCUMENTO DE IDENTIDAD	
NIT			
CORREO ELECTRÓNICO		CORREO ELECTRÓNICO	
TELÉFONO		TELÉFONO	
DIRECCIÓN		DIRECCIÓN	
CIUDAD		CIUDAD	

DEUDOR/CODEUDOR PERSONA JURÍDICA		**DEUDOR/CODEUDOR PERSONA NATURAL**	
FIRMA R.L.		FIRMA	
NOMBRE R.L.		NOMBRE	
RAZÓN SOCIAL		DOCUMENTO DE IDENTIDAD	
NIT			
CORREO ELECTRÓNICO		CORREO ELECTRÓNICO	
TELÉFONO		TELÉFONO	
DIRECCIÓN		DIRECCIÓN	
CIUDAD		CIUDAD	

DEUDOR/CODEUDOR PERSONA JURÍDICA		DEUDOR/CODEUDOR PERSONA NATURAL	
FIRMA R.L.		FIRMA	
NOMBRE R.L.		NOMBRE	
RAZÓN SOCIAL		DOCUMENTO DE IDENTIDAD	
NIT			
CORREO ELECTRÓNICO		CORREO ELECTRÓNICO	
TELÉFONO		TELÉFONO	
DIRECCIÓN		DIRECCIÓN	
CIUDAD		CIUDAD	

DEUDOR/CODEUDOR PERSONA JURÍDICA		DEUDOR/CODEUDOR PERSONA NATURAL	
FIRMA R.L.		FIRMA	
NOMBRE R.L.		NOMBRE	
RAZÓN SOCIAL		DOCUMENTO DE IDENTIDAD	
NIT			
CORREO ELECTRÓNICO		CORREO ELECTRÓNICO	
TELÉFONO		TELÉFONO	
DIRECCIÓN		DIRECCIÓN	
CIUDAD		CIUDAD	

Anexo B. Ficha Técnica del Modelo TrOCR Ajustado

El modelo entrenado en este proyecto es un ajuste fino (*fine-tuning*) del modelo base `microsoft/trocr-large-handwritten` [4] sobre el corpus de 184 pares imagen–texto construido a partir de los formularios manuscritos del Anexo No. 2 del FNG. El proceso de entrenamiento completo se describe en el Capítulo 10.6.3; esta ficha consolida los parámetros técnicos y los resultados del mejor *checkpoint*.

Tabla B.1: Parámetros técnicos del modelo TrOCR ajustado

Parámetro	**Valor**
Modelo base	`microsoft/trocr-large-handwritten`
Arquitectura	VisionEncoderDecoder (BEiT + RoBERTa)
Parámetros totales	334 M
Hardware de entrenamiento	GPU NVIDIA A100 (80 GB VRAM)
Épocas totales	10
Mejor *checkpoint*	Época 8
Batch efectivo	16 (batch 2 × acumulación 8)
Learning rate	5×10^{-5}
Optimizador	AdamW
Precisión de cómputo	FP16 (*mixed precision*)
Longitud máxima de generación	64 tokens
Dataset de entrenamiento	184 pares imagen–texto
Dataset de prueba	47 pares imagen–texto
CER en validación (época 8)	0,1176
Precisión estimada	88,2 % a nivel de carácter
CER línea base (Tesseract)	0,30 (70,0 %)
CER modelo base sin ajuste	0,45 (55,0 %)

Acceso al modelo

Los pesos del modelo (`model.safetensors`, $\approx$2,1 GB) están disponibles en el repositorio del proyecto en Google Drive. Para cargar el modelo en una nueva sesión de Google Colab sin reentrenar, ejecutar primero la Celda 3 (montaje de Drive) y luego la Celda 11 del notebook de entrenamiento (Anexo 12.8).

```python
from transformers import TrOCRProcessor, VisionEncoderDecoderModel
import torch

MODEL_DRIVE = "/content/drive/MyDrive/.../modeloentrenado"
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

processor = TrOCRProcessor.from_pretrained(MODEL_DRIVE,
local_files_only=True)
model = VisionEncoderDecoderModel.from_pretrained(MODEL_DRIVE,
local_files_only=True)
model.to(device)
```

Código B.1: Carga del modelo entrenado desde Google Drive en una nueva sesión

El parámetro `local_files_only=True` garantiza que se carguen exactamente los pesos del *checkpoint* entrenado, sin intentar descargar actualizaciones externas de HuggingFace Hub.

Anexo C. Notebook de Entrenamiento TrOCR

El notebook `Entrenar_OCR_Manuscrito_Colab_final.ipynb` contiene el pipeline completo de ajuste fino del modelo TrOCR, organizado en 12 celdas de ejecución secuencial. Está diseñado para ejecutarse en Google Colab con GPU habilitada (T4 o superior).

Tabla C.1: Descripción de las celdas del notebook de entrenamiento

Celda	Función	Descripción
1	Dependencias	Instalación y actualización de librerías (`transformers`, `evaluate`, `jiwer`).
2	Hardware	Verificación de GPU disponible y VRAM.
3	Drive	Montaje de Google Drive y configuración de rutas del corpus y del modelo.
4	Dataset	División 80/20 del `labels.txt` en *train*/*test* con semilla fija (`seed=42`).
5	Clase	Definición de `OCRDataset` para carga de pares imagen–texto.
6	Modelo base	Descarga de `trocr-large-handwritten` desde HuggingFace Hub.
7	Datos	Instanciación de datasets de entrenamiento y prueba.
8	Entrenamiento	*Fine-tuning* con `Seq2SeqTrainer`; guarda *checkpoint* por época.
9	Guardado	Copia del mejor modelo a Google Drive.
10	Pruebas	Inferencia sobre 10 ejemplos aleatorios del conjunto de prueba.
11	Recarga	Carga del modelo desde Drive en una nueva sesión sin reentrenar.
12	Pipeline completo	Pipeline opcional TrOCR + LayoutLMv3 para extracción estructurada.

Requisitos de ejecución

- GPU con mínimo 16 GB de VRAM (recomendado: T4 o A100 en Google Colab Pro).
- Tiempo estimado de entrenamiento: 2–4 horas con GPU T4, menos de 1 hora con A100.
- Corpus disponible en Google Drive con la estructura de directorios descrita en la Sección 10.3.6.