

CASO DE ESTUDIO / SOFTWARE REAL

DIGITALIZACIÓN DOCUMENTAL CON IA

De PDFs escaneados y documentos manuscritos a
archivos clasificados y preparados para su consulta.

PYTHON • OPENAI • GEMINI • PYMUPDF

PORTFOLIO PROFESIONAL • DANIEL MATEOS



ESCANEAR • LEER • CLASIFICAR

Ilustración con datos ficticios

Un problema habitual. Una solución práctica.

Digitalizar no consiste solo en escanear: también hay que identificar, separar, nombrar y colocar cada documento.

ANTES

Archivadores y revisión manual

- ✗ Abrir el PDF escaneado
- ✗ Buscar proveedor o referencia
- ✗ Separar y renombrar a mano
- ✗ Buscar la carpeta adecuada

DESPUÉS

Flujo asistido por IA

- ✓ Preparar y agrupar las páginas
- ✓ Interpretar cabeceras y números
- ✓ Generar los PDFs correspondientes
- ✓ Revisar dudas y archivar

Objetivo del proyecto: reducir trabajo repetitivo sin eliminar la revisión humana.

Del escáner al archivo listo para guardar

El sistema recibe un lote de páginas mezcladas y transforma cada documento en un PDF identificable.



Dos rutas de trabajo: facturas de compra (proveedor + nº de factura) y albaranes de venta (nº manuscrito asociado a una factura).

Dos tipos de documentos. Dos flujos especializados.

En lugar de una IA genérica, cada proceso extrae exactamente los datos que necesita el negocio.

IA COMPRAS

Facturas de proveedores

1. Agrupar factura y anexos

2. Leer proveedor y número

3. Crear un PDF por factura

IA VENTAS

Albaranes con escritura a mano

1. Recortar esquina del albarán

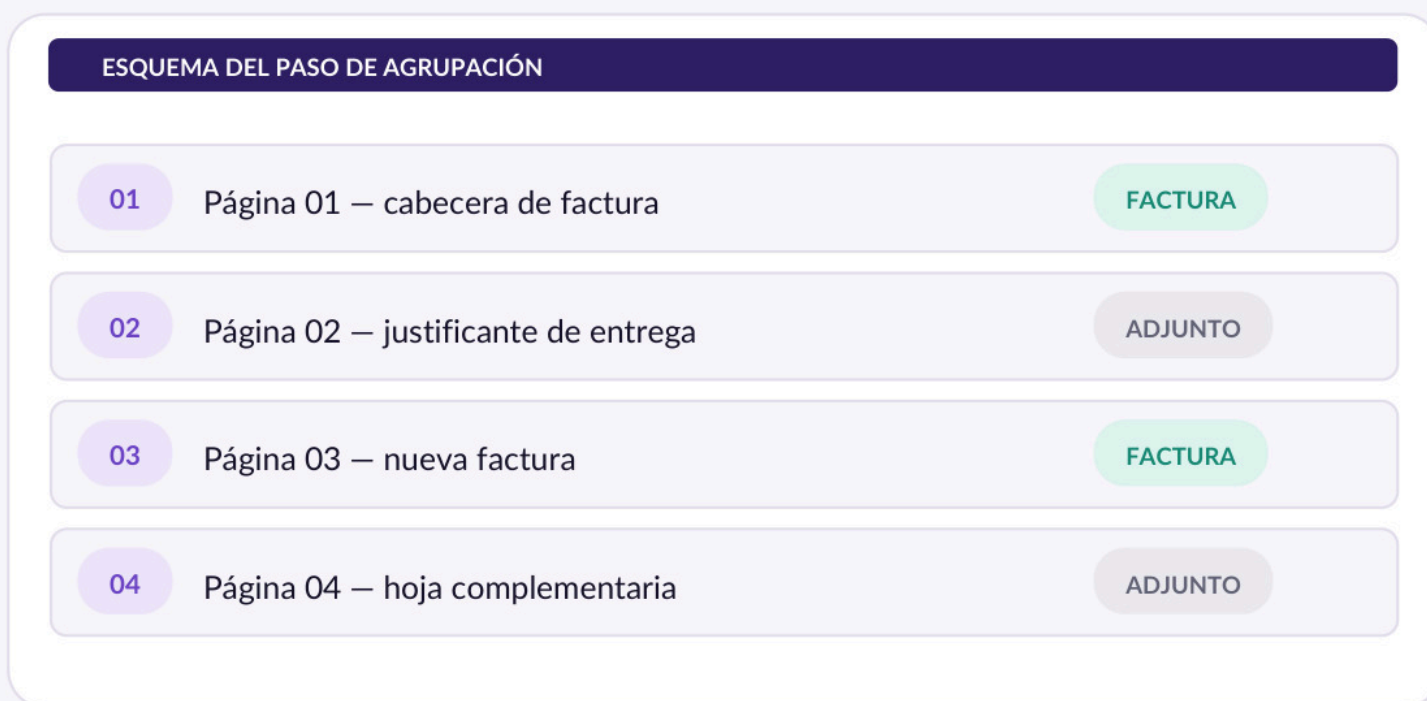
2. Reconocer número manuscrito

3. Vincular con carpeta de factura

Ambos procesos identifican casos dudosos para revisión.

Paso 1 — separar las facturas de sus anexos

Una interfaz Tkinter permite indicar dónde empieza cada factura y qué páginas forman parte del mismo documento.



¿Por qué importa?

Una factura puede ocupar varias páginas e incorporar albaranes o anexos. El sistema **conserva el grupo completo** antes de generar su PDF.

La revisión humana se reserva para una decisión sencilla: indicar si empieza un documento o continúa el anterior.

Representación explicativa basada en las funciones del repositorio; no es una captura de la interfaz.

Paso 2 – leer proveedor y número de factura

Gemini analiza collages de cabeceras para identificar los datos clave sin leer las páginas completas.

EJEMPLO DE COLLAGE PARA IA

FACTURA 1

AGRO DEMO, S.L.

N.º V250092

FACTURA 2

SUMINISTROS NORTE

N.º 14/2025

FACTURA 3

TRANSPORTES EJEMPLO

N.º FA250010

FACTURA 4

CABECERA POCO LEGIBLE

N.º REVISION

Una llamada, varias facturas

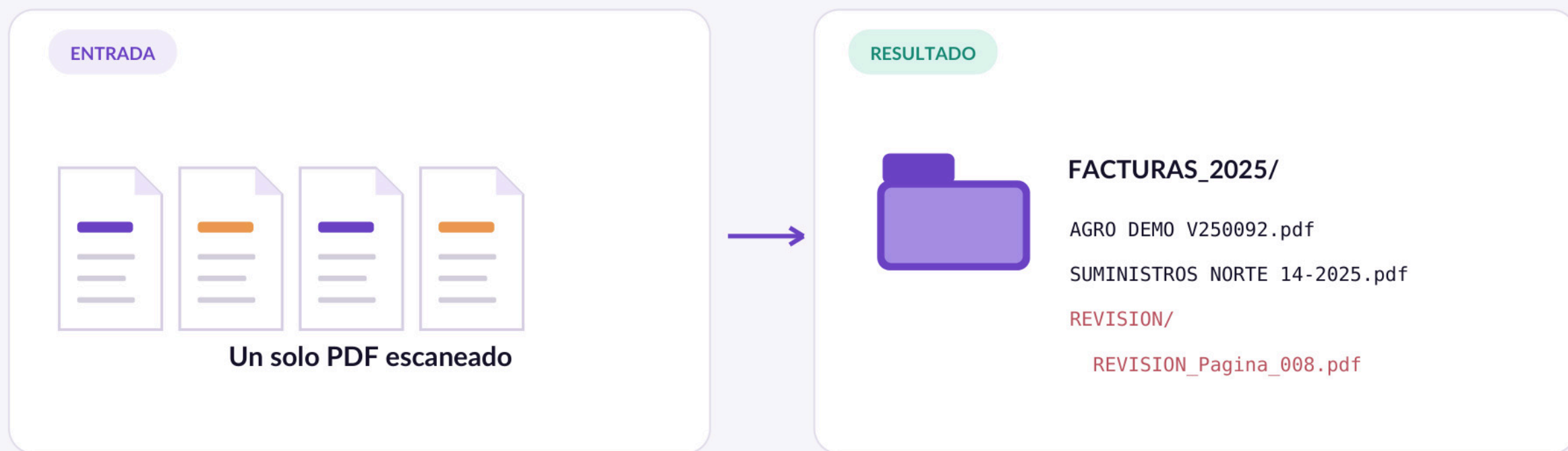
El código prepara **lotes de hasta 10 cabeceras** por collage y solicita proveedor y número para cada una.

Cuando la información no es clara, marca el documento como **REVISION** para un tratamiento específico.

Datos y ejemplo ilustrativos; proceso contrastado con el código publicado.

Paso 3 – generar PDFs individuales y nombrados

PyMuPDF reconstruye cada factura con sus anexos y la guarda con un nombre identificable.



Además, se evitan colisiones de nombres: si existe un PDF igual, se añade un sufijo.

Ejemplos con nombres ficticios; el formato está basado en la lógica real de generación.

Paso 1 – localizar números escritos a mano

Los albaranes escaneados incluyen referencias manuscritas que un OCR convencional puede leer con dificultad.



RECORTE AMPLIADO

2400212

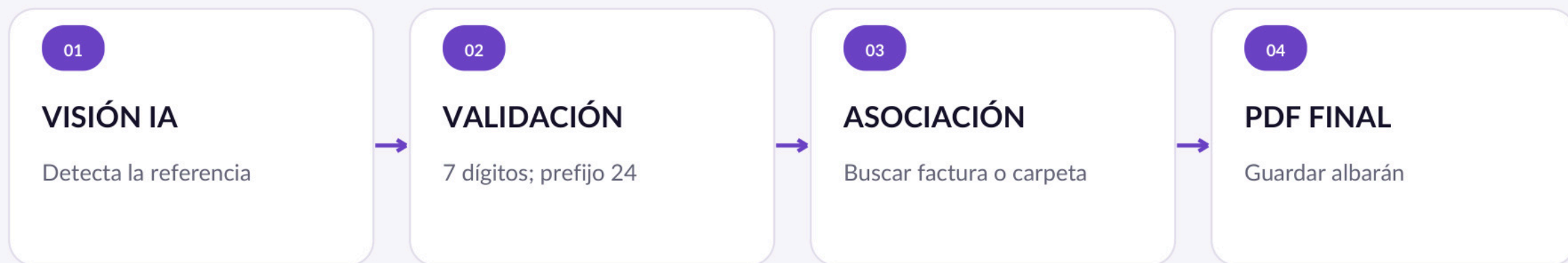
Zona inferior derecha de la página

El script extrae esta región a **300 ppp** y compone collages para su análisis.

Documentación visual ilustrativa con una referencia ficticia de 7 dígitos.

Paso 2 – reconocer y vincular los albaranes

OpenAI analiza los recortes; Gemini puede actuar como respaldo. El número sirve para asociar el albarán a su factura.



SALIDA ORDENADA – EJEMPLO

FRA 2400212 / ALBARAN_01.pdf

FRA 2400212 / ALBARAN_02.pdf

SI NO HAY LECTURA FIABLE

REVISION/

PAGINA_004_INDECISA.pdf

Ejemplos ficticios; los nombres de salida y el criterio de validación proceden del código.

Diseñado para ahorrar llamadas y gestionar dudas

La automatización combina procesamiento por lotes, alternativas de proveedor y puntos de revisión.

01

Procesamiento por lotes

Collages con varias cabeceras o esquinas para reducir consultas repetidas a la IA.

02

Recuperación ante fallos

Gemini puede sustituir a OpenAI en ventas; varias claves permiten gestionar cuotas en compras.

03

Revisión dirigida

Los documentos ilegibles o dudosos se identifican con REVISION para que el usuario los compruebe.

La IA ayuda a clasificar, pero las decisiones dudosas no se presentan como certezas.

Tecnologías al servicio de un proceso real

Una solución de escritorio en Python, orientada a documentos escaneados y al trabajo diario de una empresa.

PYTHON

Orquestación de los procesos y sus reglas.

PYMUPDF

Lectura, recorte y generación de PDFs.

PILLOW

Composición de imágenes y collages.

OPENAI + GEMINI

Modelos de visión para extraer referencias.

TKINTER

Pantallas de selección y revisión.

ENV / CONFIG

Claves fuera del código y configuración local.

Caso real en uso • Código anonimizado • Sin datos de clientes ni credenciales en el repositorio

PROYECTO EN PRODUCCIÓN

UN PROBLEMA REAL. UNA SOLUCIÓN EN USO.

Desarrollé una herramienta interna que transforma lotes de documentos escaneados en archivos PDF identificables, combinando procesamiento de imágenes, visión artificial y revisión humana.

MI APORTACIÓN

Diseño de los flujos de compras y ventas, extracción documental, integración de modelos, control de errores y generación de resultados listos para archivar.

[VER CÓDIGO Y DESCRIPCIÓN EN GITHUB →](#)

github.com/danimatper/digitalizacion-documentos-ia



Python · OpenAI · Gemini

PyMuPDF · Pillow · Tkinter