

IA local para PyMEs · Módulo 7

# Documentos y OCR

---

El ruteo por niveles y por qué convertir a texto plano destruye información. Medición campo por campo.

# Qué se construye en este módulo

## El ruteo por niveles

Clasificar un documento en PDF con texto, OCR clásico, parser o VLM

## Por qué texto plano destruye layouts

Comparar tres conversiones del mismo formulario

## Extracción con esquema

Extraer campos con confianza por campo

# Qué se construye en este módulo

 **Set de prueba  
con documentos  
reales y medición  
campo a campo**

Anotar 40 documentos  
del cliente

---

## 7.1 · El ruteo por niveles

# El ruteo por niveles

Una factura electrónica en PDF, una remisión escaneada torcida, un contrato con sellos y firmas, y una foto de celular de una cotización son cuatro problemas distintos. Tratarlos igual —«pasarlos todo por el modelo grande»— es caro y falla. El **ruteo por niveles** manda cada documento al método más barato que lo resuelve.

# En una tabla

| Perfil | Qué pasa                                                                               |
|--------|----------------------------------------------------------------------------------------|
| A      | Niveles 0–1 sin problema; nivel 2 lento; nivel 3 impracticable en vivo (lote nocturno) |
| B      | Niveles 0–2 en vivo; nivel 3 con VLM pequeño                                           |
| C      | Todos en vivo                                                                          |

Cuándo NO usar esto

## El ruteo por niveles

- No mande todo al VLM «porque lee todo»: lee todo y se equivoca en cifras. Módulo 7.4 lo mide.
- No use OCR clásico en tablas: devuelve una sopa de números sin columnas.

# Qué se degrada en cada perfil

## A · sin GPU

Niveles 0–1 sin problema; nivel 2 lento; nivel 3 impracticable en vivo (lote nocturno)

## B · 8–16 GB VRAM

Niveles 0–2 en vivo; nivel 3 con VLM pequeño

## C · 24 GB+

Todos en vivo



---

## 7.2 · Por qué texto plano destruye layouts

# Por qué texto plano destruye layouts

Un PDF de una factura tiene columnas: descripción, cantidad, precio, total. Al extraer «el texto» se obtiene una lista de palabras en orden de lectura, y el orden de lectura de una tabla **no es el de las filas**. El resultado es correcto letra por letra e inútil como dato.

# En una tabla

| Perfil | Qué pasa                                                    |
|--------|-------------------------------------------------------------|
| A      | OCR y parser en CPU (segundos por página); VLM solo en lote |
| B / C  | Los tres en vivo                                            |

# El comando, copiable

```
cd ~/labs/el-tornillo && mkdir -p labs/07-02 && cd labs/07-02
# 1) texto plano por OCR clásico
tesseract remision.png salida-ocr -l spa --psm 6
# 2) parser con estructura (Markdown con tablas) [VOLÁTIL – verificado:
2026-09]
docling remision.pdf --to md --output salida-parser
# 3) VLM: "Transcriba esta remisión como tabla Markdown" con la imagen adjunta
python3 vlm.py remision.png "Transcriba la tabla de esta remisión en Markdown, sin
inventar celdas." > salida-vm.md
```

Si no corre desde cero con un comando, no entra al curso.

Cuándo NO usar esto

## Por qué texto plano destruye layouts

- No convierta a Markdown un documento cuyo valor está en el layout exacto (un plano, un formulario oficial que hay que reproducir): ahí se guarda la imagen y se

---

## 7.3 · Extracción con esquema

# Extracción con esquema

La salida útil de un documento para la PyME no es su transcripción: son sus **campos**. Número, fecha, proveedor, NIT, líneas, totales. La extracción con esquema (módulo 5.2) se aplica aquí sobre la salida del nivel que decidió el router, y añade **confianza por campo** y una regla de **rechazo** clara.

# En una tabla

| Perfil | Qué pasa                                         |
|--------|--------------------------------------------------|
| A      | Lote nocturno: 40 facturas en decenas de minutos |
| B / C  | En vivo: segundos por factura                    |



# El comando, copiable

```
python3 extraer_factura.py facturas/*.pdf --salida resultados.jsonl
python3 - <<'PY'
import json
r=[json.loads(l) for l in open("resultados.jsonl")]
print("aceptadas:", sum(1 for x in r if not x.get("_revision_humana")), "· a revisión
      :", sum(1 for x in r if x.get("_revision_humana")))
PY
```

Si no corre desde cero con un comando, no entra al curso.

Cuándo NO usar esto

## Extracción con esquema

- No baje el umbral de confianza para «aceptar más»: acepta más errores. El umbral se mueve con la medición, no con la prisa.
- No extraiga sin coherencia: un JSON válido con total inventado pasa el esquema.

---

## **7.4 · Set de prueba con documentos reales y medición campo a campo**

# Set de prueba con documentos reales y medición campo a campo

Antes de entregar un extractor se construye un **set de prueba** con documentos **del cliente** —no de internet— anotados a mano, y se mide **campo por campo**. Es la única forma de decir «lee bien el 96 % de los totales» en vez de «funciona».

# En una tabla

| Resultado                                       | Acción                                        |
|-------------------------------------------------|-----------------------------------------------|
| un campo < 90 % en <b>todos</b> los proveedores | el prompt o el esquema de ese campo           |
| un campo < 90 % en <b>un</b> proveedor          | regla específica (o pedirle al proveedor un t |
| fallan los 5 «raros»                            | están en el nivel de router equivocado (7.1)  |
| el desacuerdo humano es alto                    | el campo está mal definido; se redefine con   |

# El comando, copiable

```
# medir.py – exactitud por campo, y qué documentos fallan
import json
esp = json.load(open("set/esperado.json")); res = json.load(open("resultados.json"))
campos = ["numero", "fecha", "proveedor", "nit", "subtotal", "iva", "total"]
for c in campos:
    ok = sum(res[d][c] == esp[d][c] for d in esp)
    print(f"{c:10s} {ok/len(esp)*100:5.1f} %    fallan: {[d for d in esp if res[d][c]
        != esp[d][c]][:5]}")
```

Si no corre desde cero con un comando, no entra al curso.

Cuándo NO usar esto

## Set de prueba con documentos reales y medición campo a campo

- No mida con los mismos documentos con los que ajustó el prompt: sepárelos (20 para ajustar, 20 para medir).
- No reporte el promedio de todos los campos: esconde el campo malo.

# Entregables del módulo

- `router.py` y `labs/07-01/ruteo.csv` con las 30 decisiones y, para 10 de ellas, la verificación manual de que el nivel fue el correcto.
- las tres salidas y la tabla comparativa.
- el extractor y `resultados.jsonl`.
- `set/` (40 documentos, `esperado.json`), `medir.py`, y la tabla por campo en `labs/07-04/medicion.md` con la fecha.



**El ruteo por niveles y por  
qué convertir a texto plano  
destruye información.**