

Informe de Experimentación y Pruebas

Versión del modelo / herramienta

Orquestador: n8n (workflow automatizado)

Modelo de IA: Gemini 2.5 Flash (visión + texto)

Servicio de conversión PDF → imagen: PDF.co API

Servicios de integración:

- Gmail API (lectura de correos con adjuntos PDF)
- Google Drive API (almacenamiento de PDFs)
- Google Sheets API (registro estructurado de resultados)

Prompts utilizados

Prompt principal para Gemini (extracción de datos de factura):

Eres un extractor de FACTURAS o RECIBOS. Analiza esta factura de servicios públicos y extrae la información en formato JSON.

IMPORTANTE: Esta es una factura de servicios públicos (agua, luz, gas, etc.)

Formato JSON requerido:

```
{  
  "numero_contrato": "número de contrato del cliente",  
  "direccion": "dirección completa del inmueble",  
  "codigo_referencia": "código de referencia para pago electrónico o PSE",  
  "total_pagar": "número sin símbolos de moneda (solo el valor numérico)",  
  "empresa": "nombre de la empresa de servicios públicos",  
  "periodo_facturado": "mes y año del periodo facturado",  
  "fecha_vencimiento": "fecha límite de pago"  
}
```

INSTRUCCIONES:

- El número de contrato puede aparecer como "No CONTRATO", "CONTRATO", "No. CONTRATO"
- La dirección suele estar después de "DIRECCIÓN" o "Dirección"
- El código de referencia puede aparecer como "Código de referencia", "Ref. pago electrónico", "PSE"
- El TOTAL A PAGAR es el monto final que debe pagar el cliente
- Si un campo no existe, usa "" para strings y 0 para números
- Para números, NO incluyas símbolos de moneda (\$), puntos o comas

Devuelve SOLO el JSON, sin texto adicional ni explicaciones.

Resultados obtenidos

- Se procesó un PDF de prueba con **10 páginas** (10 facturas).
- El flujo completo se ejecutó en aproximadamente **2 minutos y 10 segundos**.
- Para cada página se obtuvo un JSON con los campos requeridos.

- Los resultados fueron consolidados en un Google Sheets creado automáticamente a partir del nombre del archivo PDF.
- La extracción de campos numéricos (por ejemplo, total_pagar) fue consistente al usar la regla de eliminar símbolos y formatear a número.

Ajustes realizados

Durante la experimentación se realizaron los siguientes ajustes:

- **PDF.co – Conversión de páginas:**
 - Se retiró el parámetro pages restringido (por ejemplo, “2-”), lo que inicialmente hacía que se omitiera la página 1.
 - Se dejó vacío o con 1- para incluir todas las páginas del documento.
- **Limpieza y parseo de JSON:**
 - Se ajustó el nodo *Code* en n8n para:
 - Eliminar delimitadores json y otros caracteres.
 - Intentar parsear el JSON incluso si el modelo añadía texto adicional.
 - Normalizar tipos (strings y números) y campos faltantes.
- **Sincronización de ramas:**
 - Se incorporó un nodo de espera/merge (“Esperar ambas ramas”) para asegurar que la creación del Google Sheet y el parseo de datos se completaran antes de la escritura final.
- **Corrección de binarios y nombres de propiedades:**
 - Se ajustaron los nombres de las propiedades binarias (attachment_0, data, etc.) en n8n para que el nodo de subida a Drive siempre recibiera el archivo correcto.

Observaciones y aprendizajes

La combinación de **n8n + PDF.co + Gemini 2.5 Flash** permite construir flujos complejos sin necesidad de código backend personalizado.

El manejo correcto de **binarios** en n8n (adjuntos de Gmail, PDFs, imágenes) es clave para que la automatización funcione.

El diseño de prompts claros y estrictos (por ejemplo, “devuelve SOLO el JSON”) reduce errores en el parseo posterior.

Es importante contemplar la **sincronización entre nodos** cuando un flujo genera y usa recursos en etapas distintas (por ejemplo, crear un Google Sheet y luego utilizar su ID para registrar filas).

El marcado de correos como “leídos” tras el procesamiento evita reprocesamientos y bucles innecesarios.