UNIVERSIDAD EUROPEA DE MADRID

MÁSTER UNIVERSITARIO EN ANÁLISIS DE DATOS MASIVOS

TRABAJO FIN DE MÁSTER

EVALUACIÓN COMPARATIVA DE MODELOS OCR Y LLMS PARA EL PROCESAMIENTO INTELIGENTE DE FACTURAS ESCANEADAS

JULIO VALDERRAMA MARTÍNEZ

Dirigido por

DAVID DIAZ VICO

CURSO 2024 - 2025

TÍTULO: EVALUACIÓN COMPARATIVA DE MODELOS OCR Y LLMS PARA EL PROCESAMIENTO INTELIGENTE DE FACTURAS ESCANEADAS

AUTOR: JULIO VALDERRAMA MARTÍNEZ

TITULACIÓN: MÁSTER UNIVERSITARIO EN ANÁLISIS DE DATOS MASIVOS

DIRECTOR DEL PROYECTO: DAVID DIAZ VICO

FECHA: [SEPTIEMBRE] de 2025

RESUMEN

La digitalización de documentos ha incrementado la necesidad de automatizar el procesamiento de facturas de suministros (electricidad, agua y gas), tradicionalmente gestionadas de forma manual con riesgo de errores, altos costes y tiempos prolongados. Este Trabajo Fin de Máster aborda este reto mediante el diseño, implementación y evaluación de un sistema automatizado que combina técnicas de Reconocimiento Óptico de Caracteres (OCR) con modelos de lenguaje multimodales de última generación (LLMs), integrados en una arquitectura basada en microservicios y orquestada con n8n y PostgreSQL.

El proyecto siguió un enfoque experimental y comparativo, evaluando modelos líderes como GPT-4o, Gemini 2.5 Flash y Claude Sonnet 4, junto con soluciones comerciales como Azure AI Document Intelligence. El sistema incorpora un flujo completo: ingesta automática de facturas desde Google Drive, preprocesamiento condicional (PDF a imagen), extracción estructurada mediante *prompts* diseñados con precisión y almacenamiento en base de datos relacional.

Un aporte clave es el microservicio `ocr-evaluator`, que permite calcular métricas de desempeño (precisión, *recall* y F1-score) e indicadores de coste y eficiencia temporal, contrastando automáticamente los resultados con un *ground truth* curado manualmente. Los experimentos muestran que la combinación de OCR y LLMs supera ampliamente a los métodos tradicionales, logrando más del 95 % de precisión en campos críticos y reduciendo significativamente la intervención manual. Asimismo, se demostró que el diseño del *prompt* influye directamente en la obediencia del modelo y en la calidad de la extracción.

En conclusión, la solución propuesta no solo mejora la eficiencia y reduce errores humanos, sino que también establece un marco reproducible para la evaluación objetiva de tecnologías OCR/LLM, ofreciendo criterios sólidos para la toma de decisiones en entornos empresariales.

Palabras clave: OCR, LLMs, ingeniería de prompts, extracción de información, automatización documental, facturas de suministros, evaluación de modelos

ABSTRACT

The digitalization of documents has increased the need to automate the processing of utility invoices (electricity, water, and gas), traditionally managed manually with high risk of errors, costs, and long execution times. This Master's Thesis addresses this challenge by designing, implementing, and evaluating an automated system that combines Optical Character Recognition (OCR) with state-of-the-art multimodal Large Language Models (LLMs), integrated into a microservice-based architecture orchestrated with n8n and PostgreSQL.

The project followed an experimental and comparative approach, benchmarking leading models such as GPT-4o, Gemini 2.5 Flash, and Claude Sonnet 4, alongside commercial solutions like Azure AI Document Intelligence. The system includes a full pipeline: automated ingestion of invoices from Google Drive, conditional preprocessing (PDF to image), structured extraction guided by carefully engineered prompts, and storage in a relational database.

A key contribution is the `ocr-evaluator` microservice, which computes performance metrics (precision, recall, and F1-score) and indicators of cost and efficiency, automatically comparing results with a manually curated ground truth. Experiments show that combining OCR and LLMs significantly outperforms traditional methods, achieving over 95 % accuracy on critical fields while reducing manual intervention. Furthermore, results highlight that prompt design directly impacts model compliance and extraction quality.

In conclusion, the proposed solution not only improves efficiency and reduces human errors but also establishes a reproducible framework for the objective evaluation of OCR/LLM technologies, providing solid criteria for decision-making in business environments.

Keywords: OCR, LLMs, prompt engineering, information extraction, document automation, utility invoices, model evaluation

Índice general

Índice de Figuras

Índice de Tablas

Capítulo 1. INTRODUCCIÓN

1.1 Contexto y justificación

Con el avance de la digitalización, las empresas e instituciones se enfrentan al reto de gestionar cantidades cada vez mayores de documentos no estructurados, siendo las facturas uno de los más problemáticos. En sectores como el financiero, contable o de gestión de suministros, gran parte del trabajo operativo se centra en revisar manualmente documentos escaneados para extraer los datos clave. Este proceso es costoso, lento y, además, altamente propenso a errores humanos.

El caso de las facturas de suministros (electricidad, agua, gas) resulta particularmente complejo: no existe un formato estándar y cada proveedor emite documentos con estructuras, tipografías y layouts distintos. Además, muchos de estos documentos se reciben en condiciones visuales deficientes (escaneos de baja resolución, fotografías con sombras o arrugas, documentos multipágina), lo que incrementa la dificultad para los sistemas de procesamiento automático.

Ante este escenario, tecnologías como el Reconocimiento Óptico de Caracteres (OCR) y los Modelos de Lenguaje de Última Generación (LLMs) representan una oportunidad única para transformar un proceso tradicionalmente manual en un flujo totalmente automatizado y escalable. Los avances recientes en inteligencia artificial, combinando visión por computador y procesamiento del lenguaje natural (NLP), permiten no solo detectar texto en imágenes, sino también comprender el contexto, razonar sobre instrucciones específicas y generar información estructurada lista para ser almacenada en bases de datos.

De esta manera, este trabajo se sitúa en la intersección entre la investigación académica y la aplicación práctica, con el objetivo de diseñar un sistema robusto que pueda aplicarse en entornos reales y que sirva también como caso de estudio para evaluar la madurez de las tecnologías disponibles en el mercado.

1.2 Planteamiento del problema

A pesar del desarrollo de múltiples herramientas OCR comerciales y de código abierto, la automatización del procesamiento de facturas sigue siendo una tarea no resuelta en su totalidad. La diversidad de formatos, los errores derivados de la digitalización deficiente y la necesidad de normalizar los datos extraídos hacen que las soluciones actuales dependan todavía, en muchos casos, de la validación humana.

Incluso un pequeño error, como la confusión en un dígito del número de cuenta o en la fecha de vencimiento, puede ocasionar consecuencias relevantes: desde discrepancias contables hasta problemas legales o interrupciones en el pago de suministros. Esta realidad obliga a buscar un enfoque más integral: no basta con detectar caracteres correctamente, sino que es

necesario aplicar **capas de validación, normalización y comprensión semántica**, tareas en las que los LLMs multimodales destacan frente a los OCR tradicionales.

El problema principal que aborda este TFM es, por tanto, cómo diseñar un sistema capaz de **extraer, estructurar y validar automáticamente información relevante** a partir de facturas escaneadas heterogéneas, combinando OCR con LLMs y garantizando resultados fiables, comparables y utilizables en entornos reales.

1.3 Objetivos del proyecto

El proyecto tiene como objetivo general diseñar, implementar y evaluar un sistema automatizado de procesamiento de facturas escaneadas, integrando OCR y LLMs en una arquitectura modular.

De manera específica, se persiguen los siguientes objetivos:

- Desarrollar un flujo completo que abarque desde la recepción automática de facturas hasta su almacenamiento estructurado en una base de datos relacional.
- Implementar microservicios dedicados a la conversión de documentos (PDF a imagen), a la extracción mediante modelos de IA y a la evaluación comparativa de resultados.
- Establecer métricas claras (precisión, *recall*, F1-score, tiempo de ejecución y coste por inferencia) que permitan comparar de forma objetiva diferentes tecnologías.
- Analizar y contrastar el rendimiento de los modelos líderes en el mercado —como Gemini, GPT-4o y Claude Sonnet— junto con soluciones OCR tradicionales (Tesseract, Azure AI Document Intelligence). Este análisis no solo busca identificar al “mejor” modelo, sino también comprender sus puntos fuertes y débiles en escenarios reales.
- Reducir la dependencia de procesos manuales, minimizando errores humanos y optimizando tiempos y costes en la gestión documental.

La finalidad última es doble: por un lado, demostrar la viabilidad de un sistema de procesamiento automatizado de facturas en un contexto real; y por otro, aportar evidencia empírica que permita comparar y elegir de manera fundamentada entre las principales tecnologías disponibles.

1.4 Resultados obtenidos

Al finalizar el proyecto, se ha construido un sistema funcional que procesa facturas escaneadas de diversa calidad visual y complejidad estructural. Se experimentó con un conjunto de 110 facturas reales, anonimizadas y provenientes de diferentes proveedores, lo que permitió evaluar de manera rigurosa la capacidad de generalización del sistema.

Los resultados muestran que la integración de OCR con LLMs multimodales es factible y ventajosa: se logró extraer de forma correcta gran parte de la información clave (fechas, direcciones, importes, consumos, etc.), y los modelos comparados exhibieron diferencias significativas en precisión, velocidad y coste. La metodología implementada demostró que es

posible no solo automatizar este flujo, sino también establecer un marco comparativo sólido para futuras investigaciones o aplicaciones industriales.

1.5 Estructura de la memoria

La memoria se organiza de forma que guíe al lector desde los objetivos iniciales hasta las conclusiones finales, manteniendo una línea narrativa clara:

- En el capítulo 2 se presentan los objetivos generales y específicos del proyecto, estableciendo las metas a alcanzar.
- El capítulo 3 expone el marco teórico, los antecedentes y el estado del arte en OCR y LLMs, situando el trabajo en el contexto académico y tecnológico actual.
- En el capítulo 4 se describe la metodología adoptada y las herramientas empleadas, detallando cómo se planificaron, desarrollaron y evaluaron las distintas fases.
- El capítulo 5 explica en profundidad la implementación técnica, la arquitectura de microservicios y los flujos orquestados con n8n.
- En el capítulo 6 se presentan los resultados experimentales y las métricas obtenidas.
- El capítulo 7 discute dichos resultados, comparándolos con la literatura y los objetivos iniciales, destacando los aprendizajes y limitaciones.
- Finalmente, el capítulo 8 recoge las conclusiones del proyecto y sugiere líneas de trabajo futuras.
- La memoria se completa con las **referencias bibliográficas** y los **anexos**, que incluyen material técnico complementario.

Capítulo 2. OBJETIVOS

2.1 Objetivo general

El objetivo principal de este Trabajo Fin de Máster es **diseñar, implementar y evaluar un sistema automatizado para la extracción, estructuración, validación y almacenamiento de información contenida en facturas escaneadas de suministros básicos** (electricidad, agua y gas).

El sistema integra tecnologías de **Reconocimiento Óptico de Caracteres (OCR)** con **modelos de lenguaje multimodales (LLMs)**, dentro de una arquitectura modular basada en microservicios y contenedores Docker. Se busca demostrar la viabilidad de un enfoque híbrido que combine detección visual, comprensión semántica y obediencia a reglas de normalización, asegurando resultados fiables y aplicables en entornos productivos.

El proyecto tiene un doble propósito:

- Proponer una **solución práctica y escalable** que reduzca la dependencia de la transcripción manual de facturas, optimizando tiempos, costes y reduciendo errores.
- Construir un **marco de evaluación comparativa** que permita contrastar objetivamente las prestaciones de distintas tecnologías líderes en el mercado (Gemini, GPT-4o, Claude Sonnet, Azure Document Intelligence) bajo criterios de precisión, recall, F1, obediencia al prompt, tiempo de ejecución y coste económico.

2.2 Objetivos específicos

Para alcanzar este objetivo general, se han definido los siguientes objetivos específicos:

- **Integración de tecnologías Big Data y de IA:** aplicar los conocimientos adquiridos durante el máster para combinar NLP, OCR avanzado, aprendizaje profundo y bases de datos relacionales en una solución coherente y profesional.
- **Diseño de una arquitectura de microservicios contenedorizados:** implementar una infraestructura desacoplada basada en Docker y Traefik, que facilite el mantenimiento, la portabilidad y la escalabilidad del sistema.
- **Orquestación del flujo de procesamiento:** desarrollar un *pipeline* extremo a extremo con n8n, que abarque desde la ingesta de facturas (Google Drive) hasta su almacenamiento en PostgreSQL, incluyendo conversión de documentos, llamadas a modelos externos y validación de resultados.
- **Construcción de microservicios especializados:**
  - `pdf-processor`: conversión de PDFs en imágenes PNG optimizadas.
  - `ocr-evaluator`: evaluación automática frente a un *ground truth* anotado manualmente.
- **Desarrollo de un dataset de evaluación:** recopilar y anonimizar un conjunto de fac-

turas reales de gas, agua y electricidad, anotadas manualmente en campos clave (direcciones, fechas, importes, proveedores), garantizando privacidad y reproducibilidad experimental.

- **Comparación de modelos líderes en el mercado:** realizar pruebas con LLMs multimodales de OpenAI, Google y Anthropic, así como con soluciones OCR tradicionales, valorando:
  - Precisión, recall y F1-score por campo y global.
  - Capacidad de obediencia a instrucciones del prompt (por ejemplo, formatos estrictos en fechas o normalización de direcciones).
  - Adaptabilidad a diferentes layouts y calidad de documentos.
  - Eficiencia medida en tiempo de respuesta por factura.
  - Coste económico por inferencia.
- **Formulación de métricas de evaluación avanzadas:** definir un marco que combine métricas tradicionales con indicadores prácticos de negocio (coste/tiempo), generando un análisis coste-beneficio reproducible.
- **Visualización y análisis de resultados:** desarrollar notebooks en Python para generar gráficos comparativos (barras, radar, dispersión coste-tiempo), facilitando la interpretación visual de los resultados y su comunicación en un entorno académico y profesional.
- **Documentación y validación de la solución:** garantizar la trazabilidad de cada módulo, la integridad de los datos y la reproducibilidad de los experimentos, de forma que el sistema pueda servir de base para futuras líneas de investigación y mejora.

2.3 Beneficios del proyecto

Este trabajo ofrece una doble aportación: por un lado, la aplicación práctica de los conocimientos adquiridos en el Máster en Big Data; y por otro, el desarrollo de una solución real y profesional a un problema de gran relevancia en la gestión documental.

Entre los principales beneficios destacan:

- **Optimización de procesos:** reducción del tiempo y coste en la gestión de facturas, sustituyendo tareas manuales por flujos automatizados.
- **Reducción de errores:** incremento de la fiabilidad en el tratamiento de datos financieros mediante validaciones automáticas y normalización estricta.
- **Marco de evaluación robusto:** aportación de un sistema metodológico para comparar tecnologías OCR/LLM con criterios objetivos y cuantificables.
- **Flexibilidad y escalabilidad:** despliegue modular en contenedores Docker, adaptable a distintos entornos empresariales o académicos.
- **Interoperabilidad:** integración con aplicaciones externas (Google Drive, APIs de modelos) y bases de datos relacionales.
- **Contribución al campo del IDP:** avance en el ámbito del *Intelligent Document Processing*, demostrando cómo la IA generativa puede resolver casos de uso críticos en

sectores energéticos, financieros y administrativos.

Capítulo 3. MARCO TEÓRICO

En este capítulo del TFM, se aborda el estado actual de la investigación y las soluciones ya disponibles para el procesamiento automático de documentos, con especial atención en la extracción de información de facturas. Para contextualizar la contribución del presente proyecto y justificar su enfoque, se lleva a cabo un análisis de las tecnologías clave y los trabajos previos más relevantes en este campo.

A pesar de los avances en IA, visión por computador y PLN, la automatización de la extracción de datos de facturas escaneadas presenta limitaciones críticas. Los sistemas OCR son sensibles al ruido y calidad de imagen, carecen de comprensión estructural y no abarcan la diversidad de formatos [1], [2]. Los LLMs han demostrado mejoras significativas en precisión y adaptabilidad, pero su desempeño depende en gran medida de la complejidad de los documentos y de la calidad del *prompt* que guía la extracción. Como han señalado estudios recientes, pequeños cambios en el diseño del prompt (ej. formato JSON frente a texto libre, instrucciones concisas frente a detalladas) pueden alterar drásticamente la precisión y la consistencia de las respuestas [3], [4]. A ello se suma que las soluciones IDP comerciales suelen implicar costes elevados, mientras que las alternativas open-source requieren un esfuerzo significativo de adaptación [1]. La falta de *benchmarks* y datasets de calidad dificulta aún más la evaluación objetiva.

Este proyecto busca desarrollar y validar un método automatizado, preciso, eficiente y viable económicamente para la extracción de datos de facturas escaneadas, creando una herramienta de evaluación reproducible que permita comparar LLMs con visión y servicios IDP comerciales en escenarios reales, midiendo precisión, coste, tiempo de procesamiento y grado de obediencia al prompt.

3.1 Estado del Arte: De la extracción manual a la IA

Históricamente, la gestión manual de facturas ha implicado costes elevados, procesos lentos y una propensión a errores humanos. La implementación de la automatización a través de tecnología OCR (Reconocimiento Óptico de Caracteres) supuso un avance importante al permitir la transformación de imágenes o PDFs en texto estructurado. Sin embargo, los sistemas OCR tradicionales presentan limitaciones importantes como su sensibilidad al ruido, a la calidad deficiente de imagen, a la orientación incorrecta del texto o a la rotura de líneas por escaneado mal alineado [2].

Para superar estas dificultades, la investigación ha evolucionado hacia la integración del OCR con técnicas más sofisticadas, incluyendo el análisis de layout, el uso de modelos de lenguaje entrenados y arquitecturas profundas. Un ejemplo de esta evolución es OCRMiner, que combina información textual y posicional, logrando tasas de recuperación superiores al 88-90 % según el idioma y el conjunto de datos [5].

3.1.1. Evolución de los modelos OCR

Los motores OCR han evolucionado de enfoques heurísticos (como Tesseract v3) a modelos basados en deep learning. Por ejemplo:

- Tesseract v4+ ya incorpora redes neuronales LSTM.
- TrOCR (Transformer-based OCR) de Microsoft plantea un pipeline completo basado en encoder-decoders tipo BERT + GPT2.
- Donut prescinde del paso OCR y genera directamente respuestas estructuradas desde la imagen [6].

Estos avances reflejan la transición del OCR como herramienta aislada a sistemas integrados que combinan visión y lenguaje.

3.1.2. La revolución de BERT y el surgimiento de los LLMs

Un hito fundamental en el procesamiento de texto fue el desarrollo de BERT por Google en 2018, que introdujo el aprendizaje bidireccional contextualizado. Este modelo marcó el inicio de la era moderna del NLP, dando paso a variantes como RoBERTa, T5 o GPT-3, culminando en LLMs multimodales como GPT-4o Vision o Gemini 2.5. En este TFM, se aprovecha su capacidad para seguir instrucciones en lenguaje natural y extraer campos clave de facturas con ruido o formatos variados.

La investigación reciente, sin embargo, subraya que la efectividad de los LLMs no depende solo del modelo seleccionado, sino también del diseño del prompt. Estudios de Atreja et al. muestran cómo la inclusión de definiciones, la estructura del output y la longitud del prompt influyen de manera decisiva en la precisión y consistencia de las anotaciones generadas por LLMs [3]. De igual modo, la investigación de Pawlik evidencia que formatos estructurados como JSON mejoran la exactitud de las respuestas y reducen los costes de inferencia [4]. Estas conclusiones refuerzan la idea de que la ingeniería de prompts constituye un componente central de la metodología de extracción.

3.2 Avances recientes y nuevas tendencias

En los últimos años, los avances en inteligencia artificial y visión artificial han impulsado el desarrollo de nuevos enfoques para la comprensión automatizada de documentos, aplicados específicamente a la extracción de información en facturas. Existen tres grandes paradigmas actuales:

- **Enfoques basados en grafos**: Capturan relaciones espaciales entre elementos de texto, pero resultan costosos en documentos complejos.
- **Deep Learning**: Modelos entrenados con grandes datasets anotados para detectar patrones de extracción.

- **LLMs multimodales**: Procesan texto extraído o directamente imágenes para generar salidas estructuradas [1].

3.2.1. Fusión de OCR y LLMs

Uno de los enfoques más prometedores es la integración de OCR y LLMs en pipelines híbridos. Primero se extrae texto bruto con OCR, y luego se interpreta estructuralmente con LLMs como Gemini o Mistral. Estos modelos han mostrado mejoras significativas en precisión en datasets como SROIE o FATURA [5].

Además, estudios comparativos como el de [7] han mostrado que soluciones especializadas como Azure AI Document Intelligence superan en precisión (94.33 %) a modelos LLM generales como GPT-4o (85 %), lo cual indica que el diseño del sistema y la configuración del prompt siguen siendo factores decisivos.

3.3 Retos persistentes y oportunidades

3.3.1. Retos principales identificados en la literatura

La revisión de la literatura pone de manifiesto una serie de desafíos clave que siguen limitando la eficacia de los sistemas automatizados de extracción de información en facturas escaneadas:

- **Variedad de formatos y calidad:** La diversidad de layouts, idiomas y calidades de las facturas supone un reto para la estandarización de las soluciones, exigiendo modelos adaptativos y robustos [2].
- **Sensibilidad al input y errores del OCR:** El OCR sigue siendo vulnerable a imágenes borrosas, ruidos, escaneos incompletos o textos desalineados, que afectan negativamente a las etapas posteriores de extracción semántica [7].
- **Falta de datasets de calidad y validados:** Muchas investigaciones dependen de datasets propios, poco representativos o sin anotación manual rigurosa, lo que dificulta la generalización y comparación entre soluciones [1], [8].
- **Dependencia del prompt en LLMs:** Incluso los modelos más avanzados muestran variabilidad significativa en su desempeño según la formulación del prompt. Esto implica que la reproducibilidad y la calidad de los resultados dependen tanto del modelo como del diseño de la instrucción [3], [4].

3.3.2. Oportunidades emergentes

A pesar de los retos, también existen oportunidades significativas que pueden ser aprovechadas por nuevas arquitecturas basadas en LLMs:

- **Fusión de OCR y LLMs:** Las arquitecturas híbridas (OCR + LLM) permiten aprovechar lo mejor de ambos mundos: el OCR para la detección rápida y robusta de texto, y el LLM para la comprensión contextual y la estructuración avanzada [3].
- **Optimización mediante ingeniería de prompts:** Diseñar prompts estructurados, concisos y ajustados a las capacidades del modelo puede reducir costes y mejorar la precisión, como demuestran los estudios recientes en otros dominios [3], [4].
- **Generalización a múltiples layouts y lenguajes:** El uso de modelos LLMs y pipelines adaptativos incrementa la capacidad de generalizar a nuevos formatos de factura sin necesidad de re-entrenamiento intensivo o plantillas fijas [1].
- **Automatización completa y reducción del error humano:** La integración de estas tecnologías permite minimizar la intervención humana, reducir el error y acelerar drásticamente los tiempos de procesamiento [8].

3.4 Planteamiento del problema

La revisión del estado del arte evidencia que, a pesar de los importantes avances en inteligencia artificial, visión por computador y procesamiento del lenguaje natural, persisten limitaciones críticas para la automatización de la extracción y estructuración de datos en facturas de suministros escaneadas.

Los sistemas OCR tradicionales presentan una alta sensibilidad al ruido, a la calidad de imagen y a la desalineación, lo que genera frecuentes errores de extracción. Además, carecen de comprensión estructural y semántica del documento, siendo insuficientes por sí solos para abordar la diversidad de formatos y lenguajes presente en las facturas reales [2], [5].

Por su parte, los métodos más recientes basados en modelos de lenguaje de última generación (LLMs), especialmente aquellos con capacidades visuales, han mostrado un potencial considerable para enriquecer la extracción de información y mejorar la precisión. Sin embargo, su desempeño puede verse afectado por la complejidad y longitud de los documentos, y por la dependencia de prompts o configuraciones específicas. Investigaciones recientes confirman que incluso variaciones menores en la formulación de un prompt pueden afectar tanto a la precisión como a la coherencia de las salidas generadas [3], [4].

Además, las soluciones comerciales de Procesamiento Inteligente de Documentos (IDP) suelen implicar costes elevados o dependencia de servicios en la nube, mientras que las alternativas open-source requieren un esfuerzo de adaptación y entrenamiento significativo para cada caso de uso [1].

A estas limitaciones se suma la falta de *benchmarks* estandarizados y datasets de calidad que permitan una evaluación objetiva y reproducible entre tecnologías. Esto dificulta la identificación de la mejor estrategia tecnológica y arquitectónica para resolver el problema en entornos reales, donde la variedad de formatos, idiomas y calidades de imagen es la norma, y donde los recursos económicos o de infraestructura pueden estar limitados.

En consecuencia, el problema que aborda este proyecto es el desarrollo y validación de un método automatizado, preciso, eficiente y económicamente viable para la extracción y estructuración de datos clave de facturas de suministros escaneadas, superando los desafíos identificados en el estado del arte. Para ello, se propone crear una herramienta de evaluación reproducible que permita comparar, de manera rigurosa y práctica, el rendimiento de los modelos más avanzados (LLMs con capacidades de visión y servicios IDP comerciales) en escenarios reales, utilizando datasets anotados manualmente, y midiendo métricas como precisión campo a campo, coste, tiempo de procesamiento y obediencia al prompt.

Capítulo 4. METODOLOGÍA

4.1 Metodología

La metodología de este Trabajo Fin de Máster se fundamenta en un enfoque ágil e iterativo, orientado a la experimentación tecnológica y a la construcción de una arquitectura funcional y escalable. Dado que el objetivo principal era diseñar un sistema real para el procesamiento automatizado de facturas de suministros, se adoptaron prácticas inspiradas en el desarrollo ágil: modularidad, entregas incrementales, validación continua con datos reales y adaptación flexible a los cambios de requisitos.

Este enfoque permitió no solo implementar la arquitectura técnica, sino también **evaluar y comparar tecnologías emergentes de OCR y LLMs en condiciones realistas**, garantizando que las decisiones estuvieran basadas en evidencia empírica y no únicamente en documentación teórica.

4.1.1. Enfoque ágil e iterativo

El desarrollo se estructuró en cinco iteraciones principales:

- **Primera fase: investigación y diseño.** Se analizaron tecnologías de vanguardia en OCR y NLP, incluyendo modelos comerciales (Azure AI Document Intelligence, Tesseract OCR) y modelos de lenguaje multimodales (Gemini 1.5/2.5, GPT-4o Vision, Claude Sonnet). Paralelamente, se definió la arquitectura de microservicios basada en Docker [9], con componentes como n8n [10], PostgreSQL [11] y Traefik [12], diseñando desde el inicio un entorno contenedorizado, modular y reproducible.
- **Segunda fase: prototipado técnico.** Se implementó un flujo inicial en n8n que recibía facturas vía Google Drive, las almacenaba en un volumen compartido y las procesaba con un microservicio Python para conversión PDF $\rightarrow$ imagen mediante `pdf2image` [13]. Este prototipo permitió validar la comunicación entre contenedores y sentar las bases para flujos más complejos.
- **Tercera fase: desarrollo del flujo completo.** Se construyó la integración con modelos externos vía HTTP, adaptando la entrada según cada caso (PDF en Base64 para Gemini, imágenes PNG para GPT-4o o Claude). Los resultados fueron almacenados en PostgreSQL con un esquema tabular relacional, y se añadieron validaciones mediante agentes LLM, que actuaban como capa adicional de control semántico.
- **Cuarta fase: evaluación y contenerización del comparador.** Se diseñó un microservicio independiente en Python (`comparar_facturas.py`), basado en FastAPI, para contrastar resultados de OCR contra un *ground truth*. Este servicio fue desplegado como contenedor propio y se integró en el flujo de n8n, habilitando la evaluación continua y automática de modelos.

- **Quinta fase: análisis de resultados y visualización.** Tras la ejecución del flujo completo y la evaluación automatizada, se incorporó una fase adicional centrada en el **procesamiento analítico de los datos generados**. Los resultados obtenidos de cada modelo (incluyendo métricas campo a campo, métricas globales, costes y tiempos de procesamiento) fueron exportados y tratados en `notebooks` de Python, utilizando bibliotecas como `Pandas` y `Matplotlib`. En esta fase se generaron **gráficos comparativos y dashboards**, tales como diagramas de radar para observar el balance entre *precision*, *recall*, F1 y eficiencia, así como gráficos de barras y dispersión para analizar las relaciones entre coste y tiempo. Este proceso permitió no solo cuantificar las diferencias entre los modelos evaluados, sino también **interpretar visualmente su comportamiento**, identificando patrones, fortalezas y limitaciones de cada tecnología. Además, se establecieron criterios de síntesis para destacar qué modelos resultaban más adecuados en entornos productivos, conectando los hallazgos empíricos con las **conclusiones estratégicas** que se desarrollan en los capítulos finales del proyecto.

4.1.2. Modularidad y desarrollo incremental

Cada componente se concibió como un módulo autónomo, facilitando su testeo independiente y su despliegue progresivo. Esta modularidad garantizó que mejoras en un servicio (por ejemplo, el microservicio de conversión PDF $\rightarrow$ PNG) no afectaran al resto de la arquitectura.

Entre los principales módulos destacan:

- **Conversión PDF $\rightarrow$ Imagen:** microservicio `pdf-processor` con librerías como `pdf2image` y `Pillow`.
- **Procesamiento OCR y enrutamiento:** lógica en n8n que selecciona el modelo más adecuado según el tipo de archivo.
- **Validación y razonamiento:** agentes LLM que comprueban coherencia y obediencia al prompt.
- **Persistencia:** almacenamiento en PostgreSQL, con tablas diferenciadas para resultados OCR, métricas y *ground truth*.
- **Comparación de resultados:** microservicio independiente que utiliza `RapidFuzz` [14] para aplicar coincidencia difusa campo a campo.

4.1.3. Gestión y priorización

La gestión del proyecto combinó un tablero Kanban físico con listas digitales, dividiendo las tareas en cuatro áreas principales: 1) arquitectura Docker, 2) desarrollo de flujos en n8n, 3) creación de scripts Python, 4) validación de resultados.

La priorización se orientó a asegurar entregables funcionales en cada iteración: primero un

flujo OCR completo, después el comparador de resultados, luego la optimización de prompts y métricas, y finalmente la integración segura mediante Traefik y dominio propio.

4.1.4. Adaptación de prácticas ágiles

Aunque el desarrollo fue individual, se aplicaron principios de agilidad: iteraciones cortas, pruebas con datos reales, documentación incremental y capacidad de respuesta rápida ante errores. Esto permitió que el sistema evolucionara de un prototipo simple a una plataforma robusta de evaluación en menos tiempo y con mayor estabilidad.

4.2 Herramientas utilizadas

Las herramientas tecnológicas fueron seleccionadas no solo por su potencia técnica, sino también por su **adecuación al problema** y su potencial de integración en un entorno modular y escalable. A continuación, se describen las más relevantes y el papel que desempeñaron en el proyecto:

Visual Studio Code. Entorno principal de desarrollo y edición, utilizado para la escritura de scripts Python, configuración de `docker-compose`, archivos `.env`, y documentación en Markdown. Permitió centralizar el trabajo en un único IDE, facilitando la depuración y el control de versiones.

Python 3.9 y bibliotecas clave.

- `pdf2image` + `poppler-utils`: conversión de PDF a imágenes de alta resolución.
- `Pillow` y `OpenCV`: manipulación de imágenes, concatenación multipágina y optimización de legibilidad para OCR.
- `RapidFuzz`: coincidencia difusa entre textos, fundamental para evaluar variaciones semánticas en direcciones o nombres de proveedor.
- `Pandas`: análisis tabular de resultados y generación de métricas comparativas.
- `Subprocess`: coordinación segura entre scripts, utilizado en la interacción entre `process_pdfs.py` y `convertir_pdfs.py`.

n8n. Orquestador central de flujos de trabajo. Sus nodos permitieron integrar fuentes externas (Google Drive), gestionar lógica condicional de enrutamiento de archivos, invocar modelos externos mediante HTTP, validar datos y persistir resultados en PostgreSQL. La flexibilidad de n8n fue clave para prototipar y ajustar el flujo de procesamiento de forma ágil.

PostgreSQL. Base de datos relacional encargada de almacenar los resultados estructurados de OCR, las métricas de evaluación y los costes asociados a cada inferencia. Se diseñó

un esquema tabular relacional con tablas específicas para `ground_truth`, `ocr_results` y `ocr_pricing`, garantizando integridad y consistencia en la evaluación.

Docker y Traefik. La contenerización completa del entorno aseguró portabilidad y reproducibilidad. Docker permitió aislar cada servicio en un contenedor, mientras que Traefik gestionó el enrutamiento y la seguridad, ofreciendo certificados TLS automáticos para la exposición de servicios vía HTTPS.

Modelos de OCR e IA evaluados. El núcleo experimental del proyecto incluyó la comparación de Gemini 2.5 Flash, GPT-4o y Claude Sonnet 4. Cada modelo fue probado en escenarios heterogéneos de facturas, evaluando su capacidad de obediencia al prompt, su precisión campo a campo y su relación coste-tiempo.

GitHub. Plataforma utilizada para control de versiones, documentación técnica, organización de código fuente y respaldo de configuraciones de entornos. Garantizó trazabilidad y colaboración futura.

Namecheap y dominio propio. Se adquirió el dominio `tfm-master-big-data.xyz` a través de Namecheap, necesario para la recepción de webhooks externos (Google Drive), configuración OAuth y exposición segura de microservicios mediante HTTPS con certificados TLS de Let's Encrypt.

En conjunto, la combinación de estas herramientas no solo permitió construir el sistema técnico, sino también establecer un **marco experimental sólido**, donde cada componente cumplió un papel concreto dentro del flujo de procesamiento, integración y evaluación de modelos de Inteligencia Artificial.

Capítulo 5. DESARROLLO

Para la realización de este Trabajo Fin de Máster se ha seguido una metodología de desarrollo incremental, basada en la implementación práctica de una arquitectura distribuida orientada a la automatización del procesamiento de facturas mediante modelos de inteligencia artificial y herramientas de orquestación. La metodología se ha dividido en distintas fases, cada una centrada en resolver un problema concreto del flujo completo: desde la recepción de facturas escaneadas hasta la comparación y validación de los resultados extraídos por distintos modelos OCR.

5.1 Arquitectura general

El sistema ha sido diseñado como una **arquitectura de microservicios** desplegada en un entorno de contenedores Docker. Este enfoque ofrece varias ventajas clave: la separación clara de responsabilidades entre servicios, la portabilidad entre entornos (de desarrollo a producción), la escalabilidad granular y un mantenimiento simplificado. El despliegue se realiza sobre un servidor virtual en la nube (DigitalOcean) y está orquestado mediante un archivo `docker-compose.yml` que define todos los servicios interconectados. Gracias a esta aproximación, el entorno puede reconstruirse fácilmente en cualquier otra máquina con un único comando, favoreciendo la reproducibilidad de los experimentos y garantizando la trazabilidad de cada componente.

Traefik como proxy inverso. El primer componente esencial es **Traefik**, que actúa como proxy inverso y balanceador de carga. Su papel es dirigir el tráfico entrante hacia el servicio correspondiente y gestionar la exposición segura mediante HTTPS. La integración con `Let's Encrypt` permite obtener certificados TLS de forma automática, asegurando que todos los servicios desplegados sean accesibles bajo el dominio único `tfm-master-big-data.xyz`. La elección de Traefik frente a alternativas más tradicionales como Nginx se justifica por su descubrimiento dinámico de contenedores y su compatibilidad nativa con Docker Compose, lo que reduce la complejidad de configuración.

Orquestación con n8n en modo queue. El núcleo del sistema lo constituyen **n8n-main** y **n8n-worker**, que implementan el patrón *queue-worker*. El contenedor `n8n-main` gestiona los flujos de trabajo y recibe eventos externos mediante webhooks, mientras que `n8n-worker` se encarga de ejecutar las tareas pesadas de forma asíncrona. Ambos utilizan **Redis** como sistema de colas en memoria, lo que permite distribuir la carga de forma eficiente y absorber picos de entrada. Este esquema desacopla la recepción de eventos de la ejecución intensiva, aumentando la resiliencia del sistema y facilitando la escalabilidad: basta con añadir más workers para incrementar la capacidad de procesamiento.

Redis como gestor de colas. **Redis** desempeña un papel crítico como backend de colas. Al operar en memoria, ofrece baja latencia y garantiza que las tareas se distribuyan de manera ordenada entre los workers disponibles. En la práctica, esto significa que múltiples facturas pueden procesarse en paralelo sin saturar el servicio frontal, lo que resulta especialmente relevante en escenarios de alta concurrencia.

Bases de datos PostgreSQL. Se utilizan dos instancias de **PostgreSQL** con roles diferenciados:

- **postgres-n8n:** almacena la configuración interna de n8n, incluyendo flujos, credenciales y logs de ejecución.
- **postgres-test:** se reserva exclusivamente para la evaluación de modelos y contiene tres tablas fundamentales:
  - `ground_truth_invoices`, con los datos de referencia anotados manualmente.
  - `ocr_results`, con los resultados obtenidos por cada modelo OCR/LLM.
  - `ocr_pricing`, con los registros de coste y duración de cada inferencia.

Esta separación lógica no es arbitraria: permite aislar el entorno productivo de n8n del banco de pruebas de evaluación, evitando interferencias y preservando la consistencia de los datos experimentales. Además, garantiza que los análisis comparativos puedan reproducirse sin afectar al funcionamiento ordinario del sistema de orquestación.

Microservicios de procesamiento y evaluación. El flujo incluye dos microservicios adicionales, desarrollados en Python y desplegados en contenedores independientes:

- **pdf-processor:** encargado de convertir documentos PDF en imágenes PNG de alta calidad. Este paso es imprescindible para modelos que no aceptan PDFs directamente y requieren entrada visual. Para documentos multipágina, genera además una imagen vertical unificada que preserva el layout original.
- **ocr-evaluator:** compara automáticamente los resultados generados por los modelos con los valores del *ground truth*. Calcula métricas campo a campo (accuracy, precision, recall, F1), métricas globales y estadísticas de coste y rendimiento. Expone esta información mediante una API REST, lo que permite integrarla en n8n o en notebooks de análisis.

Redes y volúmenes compartidos. Todos los contenedores están conectados a través de dos redes Docker diferenciadas: `traefik`, para la exposición externa mediante HTTPS, e `internal`, para la comunicación interna segura entre microservicios. Además, comparten un volumen persistente denominado `shared_data`, que almacena los PDFs originales, las imágenes procesadas y los resultados intermedios. Esta solución evita transferencias redun-

dantes de archivos, reduce la latencia y asegura que todos los servicios trabajen sobre una misma fuente de verdad.

Gestión de configuración y seguridad. Las variables sensibles y las credenciales de acceso se centralizan en un archivo `.env`, referenciado por cada servicio mediante la opción `env_file`. Esto proporciona flexibilidad para adaptar el despliegue a distintos entornos y refuerza la seguridad al evitar valores *hard-coded*. Asimismo, la arquitectura limita la exposición: únicamente Traefik queda accesible desde Internet, mientras que el resto de servicios opera en la red interna, minimizando la superficie de ataque.

Ventajas de la arquitectura. Esta infraestructura modular no solo facilita el mantenimiento y la escalabilidad, sino que también permite aislar entornos de evaluación para mantener la consistencia de los datos y garantizar la reproducibilidad de resultados. Frente a arquitecturas monolíticas o a despliegues en entornos serverless, la elección de microservicios en Docker Compose equilibra simplicidad y control: ofrece la potencia necesaria para un TFM de carácter experimental sin incurrir en la complejidad operativa de Kubernetes ni en la dependencia de costes variables de plataformas cloud totalmente gestionadas.

5.2 Dominio y configuración del entorno

Se adquirió un dominio personalizado (`tfm-master-big-data.xyz`) a través de Namecheap, necesario para integraciones que requieren HTTPS como webhooks de Google Drive, OAuth o exposición segura de servicios. Se configuró con Traefik como proxy reverso con certificados TLS vía Let's Encrypt.

5.3 Flujo de entrada: Recepción y almacenamiento de PDFs

Uno de los flujos clave del sistema es la **ingesta automática de documentos** desde Google Drive, que permite una recepción completamente desatendida y en tiempo real. Para ello, se ha configurado un **webhook en n8n** que actúa como punto de entrada al sistema, asegurando que cada nuevo archivo PDF sea detectado, descargado y puesto a disposición del resto de microservicios sin necesidad de intervención manual. Este mecanismo constituye la base de la automatización, garantizando eficiencia, trazabilidad y consistencia en la fase inicial de procesamiento.

Activación del webhook. Cada vez que se añade un nuevo archivo PDF a la carpeta definida en Google Drive, la API emite un evento que activa automáticamente el flujo en n8n. Este paso evita que un usuario deba subir manualmente los documentos al servidor o ejecutar procesos de importación, reduciendo el riesgo de errores humanos y acelerando la ingesta. Además, al tratarse de un evento *push*, se elimina la necesidad de consultas periódicas a la nube, optimizando el consumo de recursos y reduciendo la latencia.

Descarga del archivo. Una vez recibido el evento, el flujo descarga directamente el archivo desde Google Drive mediante la credencial OAuth configurada en n8n. En este proceso se asigna un identificador único (`pdf_id`) que permite mantener la trazabilidad del documento durante todo el pipeline. De esta forma, el sistema puede relacionar cada PDF con los resultados posteriores de extracción, las métricas asociadas y los costes de procesamiento.

Almacenamiento local compartido. El documento descargado se guarda en la ruta */data/pdfs_sin_procesar*, que forma parte del volumen compartido `shared_data`. Este volumen está montado en múltiples contenedores del sistema (`n8n`, `pdf-processor`, `ocr-evaluator`), lo que significa que el archivo es accesible de inmediato para todos ellos, sin necesidad de duplicaciones ni transferencias adicionales por red. La configuración se basa en un mecanismo conocido como *bind mount*, que vincula una carpeta del host (`./shared_data`) con la ruta interna `/data` de cada contenedor.

```
volumes:
  shared_data:
    driver: local

services:
  n8n-main:
    volumes:
      - ./shared_data:/data
  pdf-processor:
    volumes:
      - ./shared_data:/data
  ocr-evaluator:
    volumes:
      - ./shared_data:/data
```

Gracias a este mecanismo, si n8n guarda un archivo en `/data/pdfs_sin_procesar`, dicho archivo será visible y accesible instantáneamente para el resto de servicios. Esto no solo mejora la eficiencia y reduce la latencia, sino que también simplifica la arquitectura, ya que elimina la necesidad de soluciones más complejas como servidores de archivos externos o transferencias vía API interna.

Ventajas del enfoque. El diseño del flujo de entrada ofrece múltiples beneficios:

- **Automatización total:** no se requiere intervención humana en la carga de documentos.
- **Eficiencia operativa:** los archivos se ponen a disposición de todos los servicios en cuestión de segundos.

- **Escalabilidad:** múltiples facturas pueden procesarse en paralelo, manteniendo la misma carpeta como fuente única de verdad.
- **Consistencia y trazabilidad:** todos los documentos siguen la misma ruta de entrada, lo que facilita su seguimiento y auditoría.

Limitaciones y retos. A pesar de sus ventajas, este flujo presenta ciertos desafíos:

- **Duplicados:** un mismo archivo puede subirse más de una vez, lo que requiere estrategias de deduplicación basadas en *hashes* (SHA256).
- **Control de versiones:** si se reemplaza un PDF existente con una versión corregida, puede sobrescribir la anterior y afectar la trazabilidad.
- **Formatos incorrectos:** aunque el flujo está pensado para PDFs, podrían cargarse accidentalmente otros tipos de archivo que deben filtrarse.

Perspectivas de mejora. Como posibles líneas futuras de ampliación se contemplan:

- **Ingesta multicanal:** incorporar fuentes adicionales como correo electrónico, SFTP seguro o integraciones API directas con proveedores.
- **Metadatos enriquecidos:** registrar información adicional como el usuario que subió el archivo, la fecha exacta y la carpeta de origen.
- **Gestión avanzada de colas:** introducir un sistema de priorización de documentos críticos o urgentes.

En definitiva, este flujo convierte a Google Drive en un **punto único de entrada confiable** al sistema, asegurando que cada documento recibido sea procesado de forma eficiente, segura y reproducible desde el primer instante.

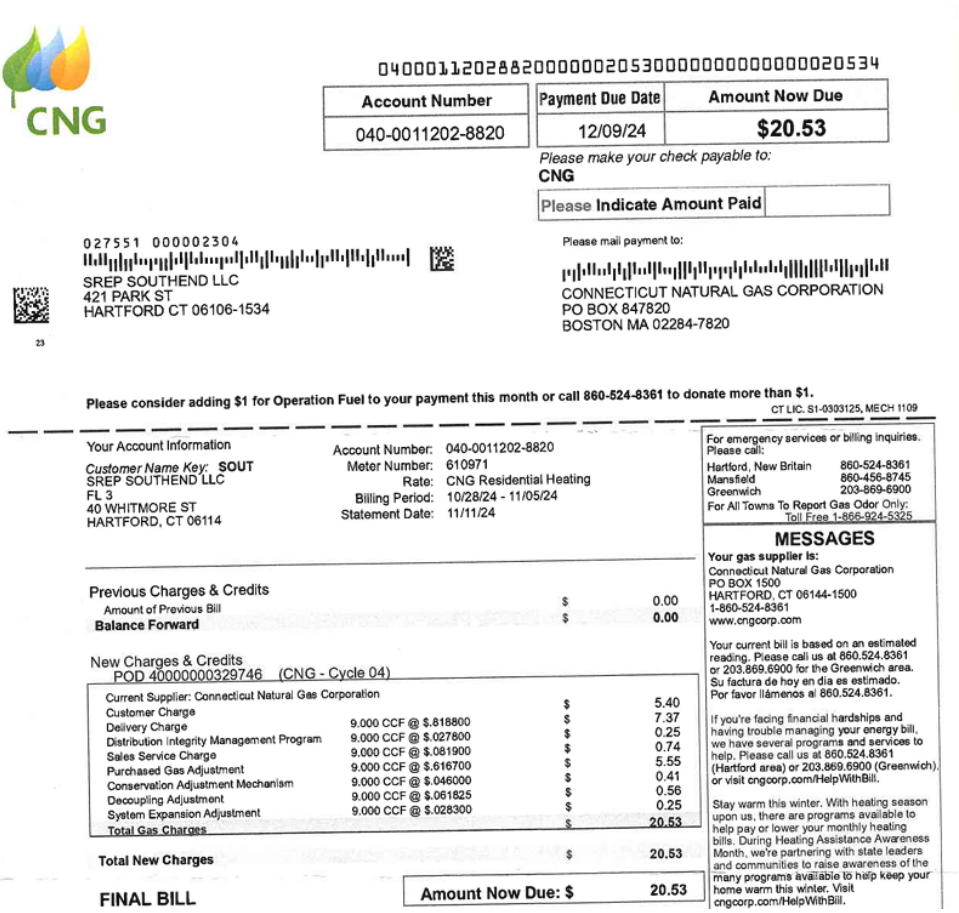
5.4 Casos de Estudio: Facturas Seleccionadas para Evaluación

Se seleccionaron tres facturas reales para validar el sistema. Se han anonimizado previamente para cumplir con los principios éticos y evitar el tratamiento de datos personales. No contienen información sensible ni identificadores personales.

Las facturas provienen de tres proveedores distintos (gas, electricidad y agua), cada una con layouts complejos y estructuras variables. Su estudio permite cubrir un rango amplio de dificultades presentes en escenarios reales, incluyendo diferencias visuales, semánticas, de formato y normalización.

Importante: Cada factura contiene entre 1 y 4 páginas, lo que introduce una dificultad adicional significativa. El sistema debe ser capaz de detectar el inicio y fin de cada factura, incluso dentro de un mismo archivo multipágina, razonando sobre la continuidad semántica y clasificando cada unidad de forma independiente.

Factura 1 (CNG – Gas) Factura de *Connecticut Natural Gas Corporation*. Incluye múltiples direcciones, con la de servicio dividida en varias líneas. Requiere normalización y reordenamiento. El proveedor aparece como *Connecticut Natural Gas* y debe mapearse a *CNG* según el prompt.



CNG

040001120288200000020530000000000000020534

Account Number	Payment Due Date	Amount Now Due
040-0011202-8820	12/09/24	$20.53

Please make your check payable to:
CNG

Please Indicate Amount Paid

027551 000002304
SREP SOUTHEND LLC
421 PARK ST
HARTFORD CT 06106-1534

Please mail payment to:
CONNECTICUT NATURAL GAS CORPORATION
PO BOX 847820
BOSTON MA 02284-7820

Please consider adding $1 for Operation Fuel to your payment this month or call 860-524-8361 to donate more than $1.

CT LIC. S1-0303125, MECH 1109

Your Account Information
Customer Name Key: **SOUT**
SREP SOUTHEND LLC
FL 3
40 WHITMORE ST
HARTFORD, CT 06114

Account Number: 040-0011202-8820
Meter Number: 610971
Rate: CNG Residential Heating
Billing Period: 10/28/24 - 11/05/24
Statement Date: 11/11/24

For emergency services or billing inquiries. Please call:
Hartford, New Britain 860-524-8361
Mansfield 860-456-8745
Greenwich 203-869-6900
For All Towns To Report Gas Odor Only:
Toll Free 1-866-924-5325

MESSAGES

Your gas supplier is:
Connecticut Natural Gas Corporation
PO BOX 1500
HARTFORD, CT 06144-1500
1-860-524-8361
www.cngcorp.com

Your current bill is based on an estimated reading. Please call us at 860.524.8361 or 203.869.6900 for the Greenwich area. Su factura de hoy en dia es estimado. Por favor llámenos al 860.524.8361.

If you're facing financial hardships and having trouble managing your energy bill, we have several programs and services to help. Please call us at 860.524.8361 (Hartford area) or 203.869.6900 (Greenwich), or visit cngcorp.com/HelpWithBill.

Stay warm this winter. With heating season upon us, there are programs available to help pay or lower your monthly heating bills. During Heating Assistance Awareness Month, we're partnering with state leaders and communities to raise awareness of the many programs available to help keep your home warm this winter. Visit cngcorp.com/HelpWithBill.

Previous Charges & Credits

Amount of Previous Bill		$ 0.00
Balance Forward		$ **0.00**

New Charges & Credits
POD 40000000329746 (CNG - Cycle 04)

Current Supplier: Connecticut Natural Gas Corporation		$ 5.40
Customer Charge	9.000 CCF @ $.818800	$ 7.37
Delivery Charge	9.000 CCF @ $.027800	$ 0.25
Distribution Integrity Management Program	9.000 CCF @ $.081900	$ 0.74
Sales Service Charge	9.000 CCF @ $.616700	$ 5.55
Purchased Gas Adjustment	9.000 CCF @ $.046000	$ 0.41
Conservation Adjustment Mechanism	9.000 CCF @ $.061825	$ 0.56
Decoupling Adjustment	9.000 CCF @ $.028300	$ 0.25
System Expansion Adjustment		$ 20.53
Total Gas Charges		
Total New Charges		$ 20.53

FINAL BILL **Amount Now Due: $ 20.53**

Figura 5.1. Ejemplo de sección de factura de gas (CNG)

Factura 2 (Eversource – Electricidad) Factura de *Eversource*, más densa visualmente. Contiene fechas con años abreviados (ej. `08/24/23`) y múltiples columnas. Representa un reto en la normalización y segmentación semántica.

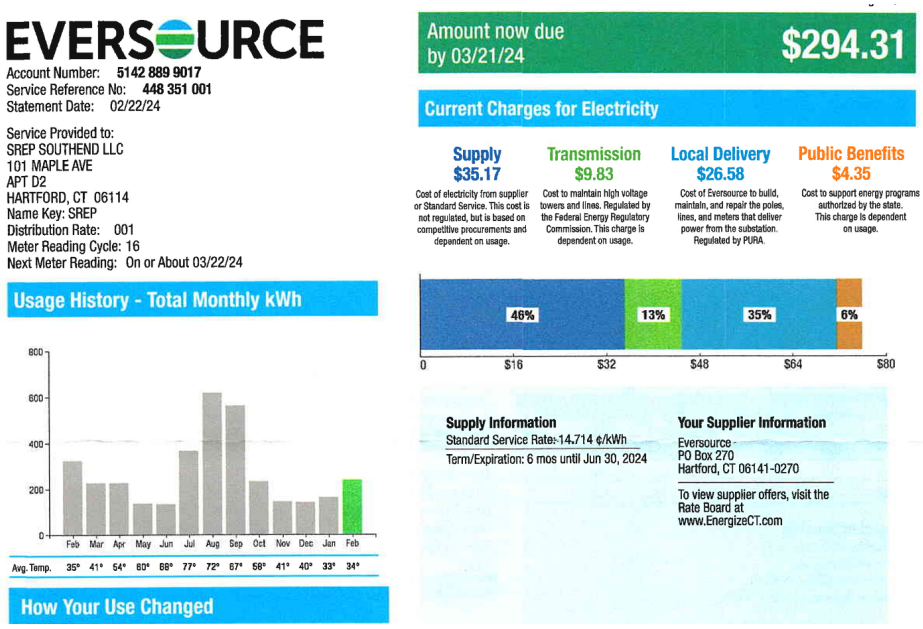
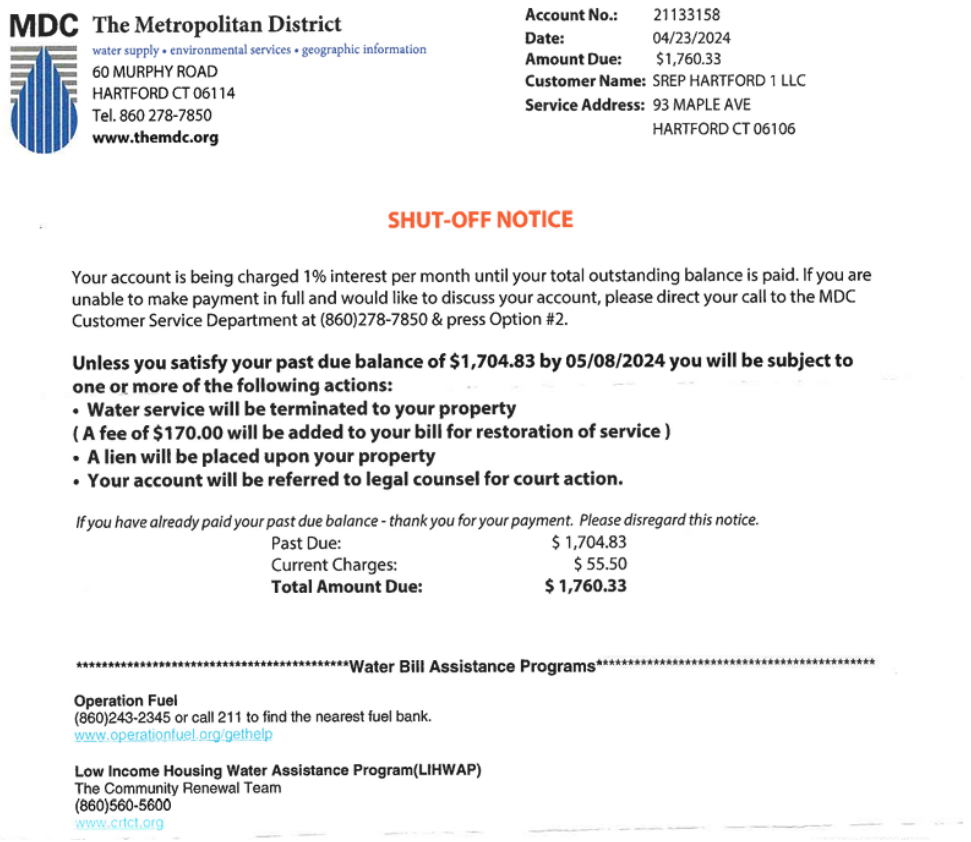


Figura 5.2. Ejemplo de sección de factura eléctrica (Eversource)

Factura 3 (Aquarion – Agua) Factura de *Aquarion*. Texto inclinado, estructuras no convencionales, y cantidades negativas entre paréntesis. Los importes están embebidos en tablas con layouts irregulares.



MDC The Metropolitan District
water supply • environmental services • geographic information
60 MURPHY ROAD
HARTFORD CT 06114
Tel. 860 278-7850
www.themdc.org

Account No.: 21133158
Date: 04/23/2024
Amount Due: $1,760.33
Customer Name: SREP HARTFORD 1 LLC
Service Address: 93 MAPLE AVE
HARTFORD CT 06106

SHUT-OFF NOTICE

Your account is being charged 1% interest per month until your total outstanding balance is paid. If you are unable to make payment in full and would like to discuss your account, please direct your call to the MDC Customer Service Department at (860)278-7850 & press Option #2.

Unless you satisfy your past due balance of $1,704.83 by 05/08/2024 you will be subject to one or more of the following actions:
- **Water service will be terminated to your property**
(A fee of $170.00 will be added to your bill for restoration of service)
- **A lien will be placed upon your property**
- **Your account will be referred to legal counsel for court action.**

If you have already paid your past due balance - thank you for your payment. Please disregard this notice.

Past Due:	$ 1,704.83
Current Charges:	$ 55.50
Total Amount Due:	**$ 1,760.33**

**********Water Bill Assistance Programs**********

Operation Fuel
(860)243-2345 or call 211 to find the nearest fuel bank.
www.operationfuel.org/gethelp

Low Income Housing Water Assistance Program(LIHWAP)
The Community Renewal Team
(860)560-5600
www.crtct.org

Figura 5.3. Ejemplo de sección de factura de agua (Aquarion)

Este conjunto de facturas representa un banco de pruebas riguroso para evaluar la fidelidad semántica, la obediencia al prompt y la capacidad de razonamiento de los modelos LLM. Se profundizará más adelante en otras peculiaridades del layout, lenguaje, segmentación visual y relaciones condicionales presentes en los documentos reales.

5.5 Fase de pre-procesamiento de documentos

La diversidad de requisitos de entrada de los diferentes modelos de Inteligencia Artificial (IA) obliga a diseñar un flujo previo de adaptación de documentos que garantice que cada modelo reciba la entrada en el formato más adecuado. Esta fase constituye un eslabón crítico en la cadena de procesamiento, ya que de su eficacia depende la calidad de los resultados posteriores. Mientras algunos modelos pueden trabajar directamente con PDFs en bruto, otros exigen representaciones visuales optimizadas, lo que hace necesaria la existencia de una capa de pre-procesamiento robusta y flexible.

El sistema desarrollado gestiona esta diversidad de forma dinámica mediante un conjunto de reglas de enrutamiento y servicios especializados. El flujo de procesamiento se articula en torno a tres etapas fundamentales:

1. Pre-procesamiento condicional de documentos, donde se adaptan los PDFs a los formatos esperados por cada modelo.
2. Extracción de datos mediante modelos LLM multimodales, encargados de interpretar y estructurar la información.
3. Persistencia estructurada de resultados en base de datos, garantizando trazabilidad y normalización.

Lógica de enrutamiento en n8n. Una vez que los archivos PDF han sido recepcionados y almacenados localmente en la carpeta `/data/pdfs_sin_procesar` (véase sección 5.3), el sistema orquestado en `n8n` ejecuta una lógica de pre-procesamiento condicional. Esta lógica evalúa automáticamente las características del documento (por ejemplo, número de páginas, peso, formato) y selecciona la ruta más eficiente según la compatibilidad del modelo de destino.

- **Gemini 2.5 Flash:** único modelo capaz de recibir directamente un PDF codificado en Base64, eliminando la necesidad de conversión previa a imágenes. Esto supone una ventaja en términos de latencia y coste computacional, al reducir un paso intermedio en el pipeline.
- **GPT-4o Vision y Claude Sonnet 4:** requieren que el PDF se convierta previamente a imágenes PNG unificadas. En estos casos, el documento se redirige hacia un microservicio especializado que realiza la conversión.

Este enrutamiento inteligente aporta flexibilidad y asegura que cada modelo trabaje en condiciones óptimas.

Microservicio `pdf-processor`. La conversión de documentos está centralizada en el microservicio `pdf-processor`, desarrollado en Python y desplegado como contenedor Docker. Este servicio se activa mediante una petición HTTP POST enviada desde n8n, aunque la comunicación se realiza únicamente dentro de la red interna de Docker. De este modo, se garantiza tanto la eficiencia como la seguridad, al evitar tráfico por la red externa y simplificar la configuración de firewalls.

El microservicio encapsula dos scripts principales que reparten las tareas:

- `process_pdfs.py`: actúa como servidor `Flask`, exponiendo un endpoint en la ruta `/api/pdf-processor/process`. Su función principal (`process_all_pdfs`) orquesta la conversión:
  - Escanea periódicamente la carpeta `/data/pdfs_sin_procesar` en busca de nuevos documentos.
  - Por cada fichero detectado, invoca a `convertir_pdfs.py` como subproceso mediante `subprocess.run`, pasándole las rutas de entrada y salida.

  - Una vez completada la conversión, mueve el PDF original a `/data/pdfs_procesados`, manteniendo un sistema de archivos limpio y garantizando la trazabilidad de los documentos ya procesados.
- `convertir_pdfs.py`: constituye el núcleo del proceso de conversión. Recibe la ruta del PDF y los directorios de salida para las imágenes individuales y para la imagen unificada. Sus principales funciones son:
  - Usar la librería `pdf2image` para transformar cada página en un PNG de alta calidad (300 DPI), optimizando la legibilidad para OCR.
  - Para documentos multipágina, emplear la librería `Pillow (PIL)` con el fin de unir todas las páginas en una única imagen vertical (`<nombre_pdf>_joined.png`).
  - Este formato consolidado es el preferido por modelos como GPT-4o Vision y Claude Sonnet 4, que requieren analizar el layout completo como una única entidad visual, preservando la disposición espacial y el contexto semántico.

Uso del volumen compartido. Un aspecto esencial para la eficiencia es que tanto `n8n` como `pdf-processor` trabajan sobre el mismo volumen compartido `shared_data`. Esto elimina transferencias redundantes entre contenedores y asegura que todos los servicios tengan acceso inmediato a los archivos más recientes. El volumen actúa como repositorio común y confiable, garantizando coherencia, reduciendo latencias y simplificando el flujo de datos.

Valor añadido de esta fase. El pre-procesamiento condicional de documentos ofrece varias ventajas prácticas:

- Permite maximizar la compatibilidad con distintos modelos sin necesidad de duplicar lógicas en cada uno de ellos.
- Reduce la probabilidad de errores de formato, que podrían afectar negativamente la calidad de la extracción de datos.
- Centraliza las operaciones de transformación, lo que facilita el mantenimiento, la extensibilidad y la reproducibilidad de los experimentos.
- Contribuye a la seguridad, al mantener todo el flujo de archivos dentro de la red interna de Docker, sin exponer servicios sensibles al exterior.

En definitiva, esta fase previa de procesamiento no solo es un requisito técnico para adaptar los documentos a los modelos, sino que constituye una pieza estratégica de la arquitectura: asegura eficiencia, coherencia y seguridad en el flujo, y sienta las bases para la fase siguiente de **extracción de datos mediante modelos de lenguaje multimodales**.

5.6 Extracción de Datos con Modelos de Lenguaje (LLMs)

En esta fase se concentra la inteligencia del sistema. Una vez adaptados los formatos de entrada (PDF en Base64 para Gemini, PNG concatenado para GPT-4o y Claude), los documentos se envían a las APIs externas de cada modelo para su procesamiento.

5.6.1. OCR y NLP combinados

A diferencia de los sistemas OCR tradicionales, cuyo objetivo principal es la conversión de imágenes a texto plano y la segmentación de caracteres, los Modelos de Lenguaje de Última Generación (LLMs) multimodales representan un salto cualitativo en el procesamiento documental.

Estos modelos no se limitan a la detección visual del texto, sino que integran:

- **Reconocimiento Óptico de Caracteres (OCR):** la capacidad de extraer texto desde imágenes o documentos escaneados, garantizando una detección robusta incluso en layouts complejos o con variaciones tipográficas.
- **Procesamiento y Comprensión del Lenguaje Natural (NLP):** la facultad de interpretar el contexto semántico, aplicar instrucciones expresadas en lenguaje natural y estructurar la información en un formato coherente y normalizado.

La combinación de ambas capacidades habilita un nivel de procesamiento muy superior, ya que el sistema no solo "lee" el documento, sino que además:

- **Interpreta** instrucciones específicas del prompt (por ejemplo, seleccionar exclusivamente la dirección de servicio frente a la de facturación).
- **Desambigua** entre múltiples candidatos (como fechas o direcciones duplicadas en la misma factura).
- **Normaliza** formatos heterogéneos, transformando valores a estructuras estándar (e.g., fechas al formato `MM/DD/YYYY` o importes numéricos sin símbolos adicionales).
- **Integra el layout visual** en su razonamiento, preservando la relación espacial entre elementos y el contexto de la factura.

De esta manera, los LLMs multimodales ofrecen una **comprensión profunda y contextual** de los documentos, superando ampliamente las limitaciones de un OCR tradicional. Este enfoque híbrido resulta especialmente crítico en la extracción de datos de facturas de suministros, donde existen múltiples layouts, campos de alta variabilidad (como las direcciones) y la necesidad de reglas estrictas de normalización.

5.6.2. El rol crítico de la ingeniería del prompt

La interacción con los modelos LLM multimodales se articula a través de un *prompt* cuidadosamente diseñado, que funciona como una auténtica **interfaz de programación en lenguaje natural**. Este prompt dicta al modelo qué información extraer, cómo estructurar la salida y qué reglas de normalización aplicar. La ingeniería de prompts constituye, por tanto, un punto neurálgico del sistema: de su claridad, especificidad y robustez dependen tanto la calidad final de la extracción como la capacidad del sistema para manejar errores y asegurar la coherencia de los datos almacenados.

Es importante remarcar que, aunque un modelo pueda disponer de una excelente capacidad OCR subyacente (es decir, una "lectura" visual precisa del texto), su rendimiento en la extracción estructurada depende en gran medida de su obediencia al prompt. En los experimentos realizados se observa que algunos modelos, a pesar de leer correctamente los caracteres, son incapaces de seguir instrucciones complejas sobre normalización o exclusión de datos irrelevantes. Otros, en cambio, muestran un alto grado de disciplina y consistencia en la aplicación de reglas, lo que los convierte en candidatos más adecuados para tareas críticas de extracción de facturas.

Este hallazgo refuerza la idea de que la **obediencia al prompt** es un criterio diferenciador clave, tanto o más importante que la precisión pura del OCR. Modelos que pueden aplicar instrucciones negativas (ej. "nunca unir números separados en direcciones") o que saben desambiguar contextos complejos (ej. distinguir entre dirección de facturación y dirección de servicio) demuestran un nivel de comprensión semántica imposible de alcanzar con OCR tradicionales.

5.6.3. Estructura del prompt

El prompt completo se incluye en el Anexo Técnico A. A continuación se desglosan sus componentes principales, reforzados con reglas, ejemplos y advertencias específicas tomadas de facturas reales (véanse Anexos B–D):

- **Definición de rol:** Se comienza con una instrucción explícita que establece el contexto: ‘‘`You are an expert utility invoice extractor.`’’ Este rol guía el comportamiento del modelo, alineando su conocimiento interno con la tarea específica y evitando desviaciones hacia explicaciones narrativas.
- **Instrucciones generales:** Se impone una regla estricta: ‘‘`Extract ONLY the fields specified`’’. El objetivo es que el modelo se limite a los campos solicitados, sin generar inferencias adicionales ni añadir información irrelevante. Esta instrucción es clave para evitar la llamada "alucinación" de LLMs, donde pueden inventar datos inexistentes.
- **Dirección de servicio (`address_line`):** Uno de los campos más complejos y críticos. El prompt instruye de manera explícita:

  ```
  Extract ONLY the full service address. If multiple addresses are
  present, always select the one associated with usage info. NEVER
  join separate numbers. Return as a single line.
  ```

  Las reglas incluyen:
  - Buscar encabezados o pistas semánticas como "Service Address" para localizar la dirección correcta.
  - Concatenar en una sola línea direcciones partidas en varias.
  - Mover códigos de unidad después de la calle principal (ejemplo: `A C4 101 MAPLE AVE` $\rightarrow$ `101 MAPLE AVE A C4`).

  - Nunca unir números separados, evitando errores típicos de OCR ("31 33 PARK ST" → "3133 PARK ST").
  - Conservar siempre letras, números o códigos asociados a la dirección.
  - Ignorar explícitamente apartados postales (PO BOX), direcciones de facturación o de envío.

  Además, se incluyen ejemplos de entradas problemáticas y de la salida esperada, reforzando la comprensión del modelo.
- **Número de cuenta (`account_number`):** La instrucción es devolver solo caracteres numéricos, eliminando guiones, espacios o símbolos. Así, `040-0011184-7196` debe convertirse en `04000111847196`. Esto garantiza un identificador único y válido para integrarlo en la base de datos.
- **Tipo de factura (`invoice_type`):** El prompt pide clasificar la factura en función de las unidades de consumo o encabezados contextuales (por ejemplo, diferenciar entre gas, agua o electricidad). Esto implica una tarea de **clasificación contextual**, que va más allá de la mera extracción literal y pone a prueba la capacidad semántica del modelo.
- **Fecha de vencimiento (`payment_due_date`):** El prompt impone un formato rígido: `MM/DD/YYYY`.

  ```
  Always return as a string in MM/DD/YYYY format, even if the year
  is 2 digits, convert it to 4 digits. Do not use ISO or any alternative
  formats.
  ```

  Por ejemplo, `11/22/24` debe convertirse en `11/22/2024`. Esta normalización es crucial para cálculos financieros y comparaciones entre facturas.
- **Importes (`amount_due_now` y `total_current_charges`):** El prompt exige devolver siempre valores numéricos planos, eliminando símbolos, comas o paréntesis. Ejemplo: `Total Amount Due: $1,234.56` → `1234.56`. También se contemplan importes negativos, representados a menudo entre paréntesis, que deben conservarse como valores numéricos válidos.
- **Nombre del proveedor (`vendor_name`):** Se definen reglas de estandarización para nombres de proveedores conocidos. Por ejemplo: `Connecticut Natural Gas Corporation` → `CNG`. Esta tarea requiere capacidades de **mapping semántico** que un OCR clásico no podría realizar, pues implica reconocer entidades y asignar alias predefinidos.
- **Multi-página y detección de facturas:** El prompt instruye al modelo a razonar sobre documentos multipágina:
  - Determinar si varias páginas corresponden a la misma factura o a facturas diferentes.
  - Identificar el inicio de una nueva factura cuando cambian campos clave (`account_number`, `payment_due_date`, `vendor_name`).
  - Mantener la coherencia entre páginas de un mismo documento, unificando los datos cuando sea necesario.
- **Normalización de valores:** Para reforzar la consistencia, se incluyen reglas de norma-

lización:

- **Direcciones:** siempre en una sola línea, sin unir números separados, con códigos de unidad detrás de la calle.
- **Fechas:** estrictamente en formato `MM/DD/YYYY`, con años expandidos a cuatro dígitos.
- **Importes:** valores numéricos sin símbolos, comas o paréntesis, incluyendo negativos cuando aplique.

- **Formato de salida:** El prompt obliga a devolver siempre un `JSON array` válido, con un objeto por factura. Esta salida estrictamente tipada asegura:
  - Integración directa en la base de datos PostgreSQL.
  - Homogeneidad de resultados entre modelos.
  - Reducción de errores de integración o postprocesamiento.

En conjunto, este diseño convierte al prompt en un auténtico **motor de reglas expresadas en lenguaje natural**, que combina capacidades de extracción, clasificación y normalización. La capacidad del modelo para obedecer fielmente estas instrucciones —más allá de reconocer caracteres— constituye uno de los factores diferenciales más relevantes en la evaluación comparativa presentada en este trabajo.

5.6.4. Ejemplo de Instrucción en el Prompt

Un ejemplo textual del prompt aplicado es el siguiente:

```
Normaliza todos los valores monetarios eliminando símbolos de moneda,
comas o espacios, y devuelve siempre el resultado como un valor numérico
plano (por ejemplo, "$1,234.56"→ "1234.56").
```

Este nivel de detalle convierte al prompt en un auténtico “motor de reglas”, capaz de realizar inferencia semántica, aplicar transformaciones textuales y garantizar salidas consistentes.

5.6.5. Importancia de campos críticos en la extracción

El diseño del prompt no distribuye la misma intensidad de reglas a todos los campos. Algunos, como `vendor_name` o `account_number`, presentan una relativa estabilidad en su detección y normalización. Sin embargo, otros campos requieren una atención mucho mayor debido a su complejidad intrínseca y a las implicaciones prácticas de sus errores. En este trabajo, dos campos destacan como **indicadores clave de la robustez de un LLM**: `address_line` y `payment_due_date`.

La dirección de servicio (`address_line`). La extracción correcta de la dirección de servicio es un desafío histórico en los sistemas de OCR y procesamiento documental. Su dificultad radica en la enorme variabilidad de formatos que puede adoptar:

- Inclusión de múltiples direcciones en un mismo documento (facturación, envío, servicio).
- División en varias líneas que deben ser concatenadas en una sola.
- Presencia de códigos de unidad, apartamento o edificio sin un patrón fijo (`A C4 101 MAPLE AVE`).
- Errores típicos de OCR que tienden a unir números separados ("31 33 PARK ST" → "3133 PARK ST").

El prompt responde a estas dificultades con reglas muy estrictas: reordenar el código de unidad detrás de la calle principal, concatenar líneas en un solo bloque y prohibir expresamente unir números separados. Estas instrucciones ponen a prueba la **capacidad de inferencia, el seguimiento de reglas negativas y la manipulación textual precisa** del modelo. A diferencia de un OCR clásico, que solo transcribe lo que "ve", un LLM debe comprender la lógica semántica del campo y aplicar transformaciones normativas. Un error en este campo puede tener consecuencias graves para la lógica de negocio, como el direccionamiento erróneo de notificaciones, pagos o comunicaciones. En la sección de Resultados se analizará cómo distintos modelos enfrentaron este reto y qué impacto tuvo en su rendimiento global.

La fecha de vencimiento (`payment_due_date`). El segundo campo crítico es la fecha de vencimiento de la factura. Su extracción no se limita a la identificación visual, sino que exige un proceso de normalización con múltiples pasos:

- Identificar la fecha correcta dentro de un documento donde pueden aparecer múltiples fechas (emisión, lectura, vencimiento).
- Convertirla a un formato `MM/DD/YYYY` estandarizado, independientemente de cómo aparezca representada.
- Inferir el siglo correcto cuando la factura utiliza años abreviados con dos dígitos ("11/22/24" → "11/22/2024").

Estas instrucciones convierten la fecha en una auténtica **prueba de obediencia al prompt**. Un modelo que falla en este punto no solo incurre en errores formales, sino que compromete la puntualidad de los pagos, la coherencia contable y, en última instancia, la salud financiera de la organización.

Relevancia en la evaluación. Tanto `address_line` como `payment_due_date` actúan como **campos "termómetro"** para medir la capacidad de los LLM de seguir instrucciones complejas de normalización y aplicar razonamiento contextual. En la evaluación de resultados, estos dos campos constituyen una referencia fundamental para discriminar entre modelos

que simplemente transcriben texto y aquellos que son capaces de ejecutar transformaciones semánticas complejas. La variación observada entre modelos en el tratamiento de estos campos será analizada en detalle, ofreciendo una visión clara sobre cuáles son más aptos para escenarios productivos de procesamiento documental.

5.6.6. Formato de Salida (Output) en el Prompt

Además de las reglas de extracción y normalización, el prompt define de forma estricta el **formato de salida** que debe generar el modelo. Este aspecto es crítico, ya que garantiza que los resultados puedan ser consumidos automáticamente por el sistema sin necesidad de postprocesamiento adicional.

Se exige que el modelo devuelva siempre un **JSON válido**, estructurado como un arreglo donde cada objeto corresponde a una factura procesada. Dentro de cada objeto se incluyen únicamente los campos especificados (`account_number`, `invoice_type`, `payment_due_date`, `amount_due_now`, `total_current_charges`, `vendor_name`, `address_line`, `city`, `post_code`), aplicando las reglas de normalización previamente establecidas.

Este control estricto del formato asegura:

- Compatibilidad inmediata con la base de datos PostgreSQL del sistema.
- Homogeneidad en los resultados entre diferentes modelos, facilitando la comparación objetiva.
- Reducción de errores de integración debidos a respuestas inconsistentes o con información adicional no solicitada.

En síntesis, el diseño del output en el prompt no solo define *qué* información se extrae, sino también *cómo* debe representarse, reforzando la fiabilidad del sistema en un entorno de producción.

5.7 Procesamiento de la Respuesta y Persistencia

Una vez recibidas las respuestas de los modelos de IA, el sistema implementa un flujo de procesamiento automatizado en `n8n` que garantiza la validez, consistencia y trazabilidad de los datos antes de su almacenamiento definitivo en la base de datos. Este flujo contempla las siguientes fases:

Parseo y Validación Las respuestas de los modelos llegan en formato `JSON`. `n8n` realiza un parseo automático y un proceso de validación estructural. En esta etapa se asegura que:

- El `JSON` devuelto cumpla con el esquema predefinido.
- Todos los campos obligatorios estén presentes.
- No existan valores nulos o tipos de datos inconsistentes (ejemplo: fechas en texto libre o importes con caracteres no numéricos).

Este control temprano evita que datos defectuosos contaminen las tablas de resultados.

Normalización de Datos Una vez validados, los datos se someten a un riguroso proceso de normalización, necesario debido a la variabilidad en los formatos devueltos por los distintos modelos. En esta fase se aplican reglas como:

- Conversión de fechas a un formato uniforme `MM/DD/YYYY`.
- Estandarización de importes mediante la eliminación de símbolos monetarios, comas o espacios, devolviendo siempre un valor numérico plano.
- Normalización de direcciones y nombres de proveedores, respetando las reglas específicas definidas en el *prompt*.

Esta estandarización asegura la coherencia del dataset y facilita tanto la persistencia como los análisis comparativos posteriores.

Almacenamiento en Base de Datos Los resultados normalizados se insertan en la tabla `ocr_results` de la base de datos `postgres-test` (`PostgreSQL`). La comunicación se realiza de forma interna en la red Docker, accediendo al contenedor mediante la URL `http://postgres-test:5432/`. La tabla `ocr_results` almacena:

- Los datos estructurados de cada factura.
- Metadatos como el `pdf_id` (clave foránea al `ground_truth`), el modelo utilizado y la fecha de procesamiento.

Adicionalmente, los registros de coste (`input_price`, `output_price`, `total_price`) y tiempo de ejecución (`duration_s`) de cada invocación se almacenan en la tabla `ocr_pricing`. Esta información permite evaluar la eficiencia económica de cada modelo y realizar análisis de coste-tiempo con gran nivel de detalle.

Fiabilidad y Trazabilidad La elección de `PostgreSQL` garantiza integridad referencial y soporte para consultas complejas. Gracias a la relación entre las tablas `ground_truth`, `ocr_results` y `ocr_pricing`, el sistema ofrece un entorno robusto y auditable, cumpliendo con los principios de un sistema de *Intelligent Document Processing* (IDP) preparado para un uso productivo y escalable.

5.8 Evaluación Automatizada y Comparación Flexible

Uno de los pilares fundamentales de este TFM es la **evaluación objetiva, continua y automatizada** de los diferentes modelos de inteligencia artificial aplicados a la extracción de datos de facturas. Con este propósito, se ha desarrollado un microservicio denominado `ocr-evaluator`, implementado en `Python` con el framework `FastAPI` y desplegado como contenedor `Docker`. Esta aproximación garantiza modularidad, escalabilidad y seguridad dentro de la arquitectura global.

El servicio `ocr-evaluator` compara los datos extraídos por los modelos LLM (Gemini 2.5, GPT-4o y Claude Sonnet 4) con un **conjunto de referencia validado manualmente** (`ground truth`). La comunicación con la base de datos `postgres-test` se realiza mediante una red interna de `Docker`, lo que proporciona rendimiento óptimo y un aislamiento seguro.

5.8.1. Arquitectura del Microservicio `ocr-evaluator`

El proceso de evaluación se articula en tres fases principales:

1. **Carga y preparación de datos**:
   - Se extraen las tablas `ground_truth_invoices`, `ocr_results` y `ocr_pricing`.
   - Se realiza un `merge` entre los resultados OCR y el `ground truth` mediante la clave `pdf_id`, garantizando la correcta correspondencia entre facturas y modelos.
2. **Comparación campo a campo inteligente y flexible**: El sistema adapta el tipo de comparación en función de la naturaleza del campo, reconociendo que los LLM pueden introducir variaciones sintácticas, semánticas o tipográficas. Se aplican tres estrategias:
   - **Coincidencia exacta (`exact_match`)**: aplicada en campos donde se requiere precisión literal, como `invoice_type` o `account_number`.
   - **Coincidencia numérica (`numeric_match`)**: utilizada en importes monetarios (`amount_due_now`, `total_current_charges`), tolerando ligeras diferencias de redondeo o formato (p. ej., “1,234.50$” frente a “1234.50”).
   - **Coincidencia difusa (`fuzzy_match`)**: el núcleo metodológico del sistema, empleado en campos donde se esperan variaciones válidas como `address_line`, `vendor_name` o `payment_due_date`.
3. **Registro de métricas y eficiencia**: Los resultados se enriquecen con métricas de evaluación clásicas (precisión, recall, F1, accuracy) junto con información de coste y tiempo (`ocr_pricing`), permitiendo un análisis comparativo integral.

5.8.2. Importancia del Fuzzy Matching en el Contexto de los LLMs

En determinados campos, la coincidencia exacta resulta insuficiente. Los modelos de lenguaje pueden devolver resultados que, aunque no coincidan de forma literal, son equivalentes desde una perspectiva semántica o contextual. Ejemplos ilustrativos incluyen:

- “4A Avenue Street” vs. “A4 Avenue Street”
- “Connecticut Natural Gas” vs. “CNG”
- “08/23/2024” vs. “8/23/2024”

Para gestionar esta casuística, se emplea la librería `rapidfuzz`, que ofrece funciones como

`fuzz.token_sort_ratio()` o `fuzz.ratio()`, capaces de tolerar cambios de orden, abreviaturas o errores menores de formato.

Se definen umbrales específicos por campo, con el objetivo de equilibrar flexibilidad y rigor:

```
FUZZY_THRESHOLDS = { 'address_line': 90, 'vendor_name': 85, 'payment_due_date': 90 }
```

Este enfoque asegura que variaciones mínimas (como "Str." en lugar de "Street") o reordenamientos de tokens no penalicen injustamente al modelo, siempre que se mantenga la coherencia estructural esperada.

Es especialmente relevante en el campo `vendor_name`, donde los modelos pueden devolver acrónimos (p. ej., "EVS" en lugar de "Eversource"). Si dicha simplificación no ha sido contemplada explícitamente en el *prompt*, debe considerarse un error.

En conclusión, la introducción del **fuzzy matching** constituye un aporte metodológico clave de este TFM, al permitir una evaluación más justa y adaptada a la naturaleza flexible de los LLMs, sin perder el control sobre la precisión necesaria para un entorno productivo.

5.9 Evaluación de la Comprensión del Prompt

Un objetivo central de este TFM es evaluar hasta qué punto los modelos LLM **comprenden y obedecen las instrucciones explícitas del prompt**. Todos los modelos analizados reciben exactamente el mismo conjunto de instrucciones, cuidadosamente diseñadas para cubrir los campos críticos de la factura (como `address_line`, `payment_due_date`, o `amount_due_now`).

En consecuencia, las diferencias en sus resultados no dependen del input, sino de la **capacidad del modelo para seguir las directrices del prompt**. Ejemplos de instrucciones críticas incluyen:

- **No fusionar números separados:** "31 33 PARK ST" debe conservarse así, y nunca transformarse en "3133 PARK ST".
- **Formato estricto de fechas:** siempre en `MM/DD/YYYY`, incluso cuando el documento presente el año en dos dígitos.
- **No alterar la estructura semántica de direcciones:** respetar el orden de calle, número y código de unidad, aplicando normalización pero sin introducir errores de concatenación.

Permitir un umbral demasiado laxo en la comparación podría enmascarar **errores reales de interpretación**, dando una falsa impresión de rendimiento superior. Por ello, el sistema implementa un **equilibrio metodológico**:

- **Suficientemente estricto:** para penalizar desviaciones graves que comprometen la utilidad práctica del dato.

- **Moderadamente tolerante:** para no castigar variaciones irrelevantes desde la perspectiva humana (p. ej., “08/23/2024” frente a “8/23/2024”).

Este enfoque convierte la evaluación de la obediencia al prompt en un **indicador diferencial de calidad**, capaz de discriminar entre modelos que simplemente realizan OCR y aquellos que realmente comprenden y aplican las instrucciones de negocio.

5.10 Métricas y Exposición

El microservicio `ocr-evaluator` calcula y expone métricas de rendimiento mediante endpoints REST (`/metrics`, `/metrics/{modelo}`), los cuales son invocados desde flujos de `n8n`. Estas métricas permiten un análisis granular y se agrupan en tres categorías principales:

- **Por campo:** incluyen `accuracy`, `precision`, `recall`, `f1`, el número total de muestras (`total_samples`) y los aciertos absolutos (`correctos`). Estas métricas son clave para identificar qué campos específicos presentan mayor dificultad para cada modelo.
- **Globales:** agregan el rendimiento total del modelo considerando todas las predicciones, con métricas como `accuracy`, `precision`, `recall` y `f1-score` a nivel global. Este enfoque permite comparar modelos en términos de **desempeño general**.
- **De coste y eficiencia:** se incluyen parámetros como la duración media de procesamiento por factura (`avg_duration_s`), el coste medio por predicción (`avg_total_price`) y las sumas acumuladas de coste y tiempo (`total_cost`, `total_duration_s`). Estos indicadores son fundamentales para un análisis coste-beneficio de cada modelo.

Un aspecto central en la evaluación de modelos de extracción de información es el **equilibrio entre *precision* y *recall***. La *precision* mide la proporción de predicciones positivas correctas respecto al total de predicciones positivas realizadas, es decir, qué tan fiable es el modelo cuando afirma haber identificado un campo. Por otro lado, el *recall* refleja la capacidad del modelo para identificar todos los casos relevantes, indicando qué porcentaje de los elementos correctos fueron realmente detectados. En el contexto de facturas, un modelo con alta *precision* pero bajo *recall* cometería pocos errores al predecir campos, pero dejaría muchos sin reconocer; mientras que uno con alto *recall* pero baja *precision* capturaría la mayoría de los campos, aunque con un alto número de falsos positivos.

Este dilema hace necesario un indicador que combine ambas métricas. El **F1-score**, definido como la media armónica entre *precision* y *recall*, proporciona una visión más equilibrada del rendimiento. A diferencia de la media aritmética, la media armónica penaliza los valores extremos, de modo que un modelo solo obtiene un F1 elevado cuando ambas métricas son simultáneamente altas. Por esta razón, el F1 se usa ampliamente en la literatura sobre procesamiento de documentos y clasificación automática, ya que permite valorar la calidad global de un sistema más allá de la exactitud pura, especialmente en escenarios con distribuciones desbalanceadas de clases o cuando los falsos positivos y negativos tienen costes distintos en la práctica.

Todos los resultados se devuelven en formato `JSON`, lo que permite:

- Su **procesamiento automático en n8n**, donde pueden validarse, transformarse y persistirse en base de datos.
- Su integración en notebooks de análisis (`visualizacion.ipynb`), facilitando la construcción de **gráficos comparativos y dashboards**.
- El disparo de **alertas automáticas**, por ejemplo cuando un modelo cae por debajo de ciertos umbrales de precisión o coste definidos por el sistema.

Este diseño convierte al microservicio no solo en un **evaluador técnico**, sino también en una pieza clave de la arquitectura de monitorización y gobernanza del sistema IDP, habilitando un ciclo de mejora continua.

Capítulo 6. RESULTADOS

Tras la ejecución automatizada del microservicio `ocr-evaluator` y el análisis posterior en el notebook de visualización (`visualizacion.ipynb`), se han obtenido los resultados comparativos de los modelos de Procesamiento Inteligente de Documentos (IDP) evaluados. La comparativa se ha realizado sobre un conjunto de 990 predicciones por modelo, utilizando métricas precisas por campo, métricas globales y datos de coste y duración, todos ellos obtenidos de un sistema real de evaluación de OCR sobre facturas de energía con su *ground truth* asociado.

6.1 Comparativa Global de Rendimiento

Las métricas de **Accuracy Global** y **F1 Global** proporcionan una visión general del rendimiento de cada modelo, mientras que la cantidad de **Correctos Globales** indica el número total de campos extraídos correctamente. Estos indicadores son cruciales para entender la fiabilidad general de cada solución de IDP y reflejan la capacidad de los modelos para interpretar y adherirse a las instrucciones del prompt.

Modelo	Precisión Global	F1 Global	Correctos Globales	Total Predicciones
claude-sonnet-4	0.74	0.85	741	990
gemini-2.5-flash	0.98	0.99	980	990
gpt-4o	0.93	0.96	929	990

Tabla 6.1. Métricas Globales de Precisión y Aciertos Totales

Del análisis de la Tabla 1 se desprende que:

- **Gemini 2.5 Flash** ha demostrado ser el modelo más preciso, con una **Accuracy Global** del 98.99 % y una puntuación **F1 Global** de 0.9949. Esto lo posiciona como el modelo más fiable del estudio, logrando 980 extracciones correctas de un total de 990 campos predichos.
- **GPT-4o** se sitúa en segundo lugar, con una **Accuracy Global** del 93.84 % y una **F1 Global** de 0.9682, con 929 aciertos globales. Aunque muy sólido, queda ligeramente por debajo de Gemini.
- **Claude Sonnet 4** presenta un rendimiento significativamente inferior, con una **Accuracy Global** del 74.85 % y una **F1 Global** de 0.8562. Sus 741 aciertos globales lo convierten en la opción menos precisa.

Es importante destacar una limitación técnica específica que afectó al modelo Claude: en varias ocasiones, al intentar enviar imágenes de facturas a la API de Anthropic, se produjo un error porque el tamaño de la imagen superaba el límite de 5MB. En estos casos, el sistema registró valores nulos (`null`), lo cual penalizó directamente sus métricas y redujo su **Accuracy Global**. Este hecho refleja una limitación real en un escenario de producción.

Para una visualización más intuitiva de estas métricas, se generaron los siguientes gráficos:

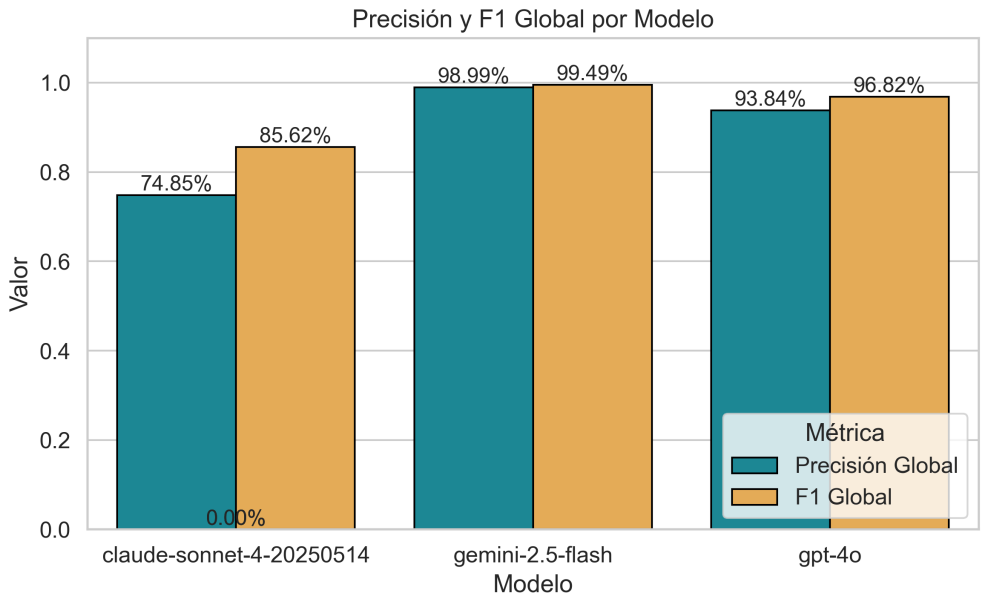


Figura 6.1. Comparativa Global entre Modelos IDP: Precisión y F1

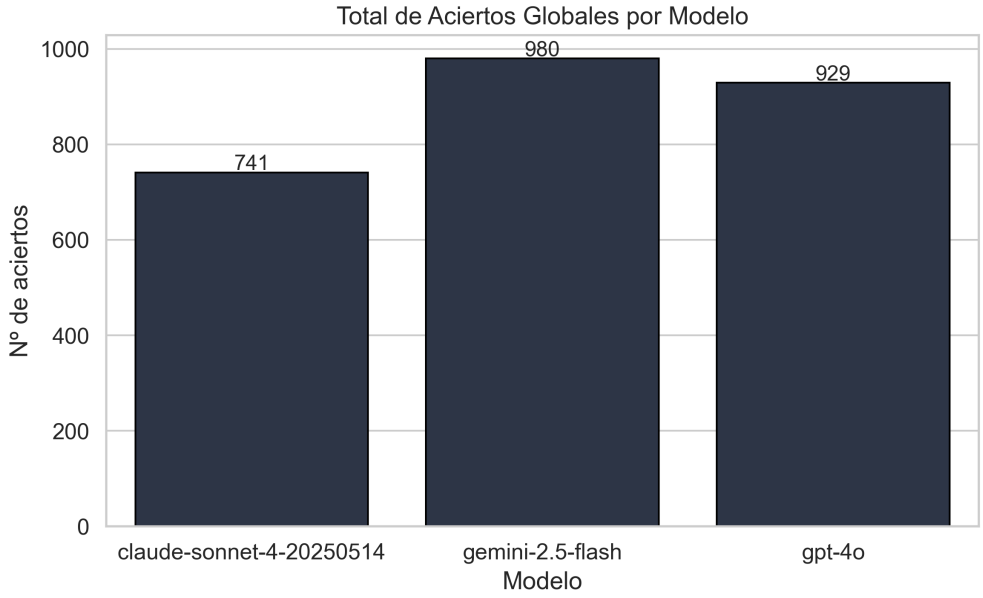


Figura 6.2. Comparativa Global entre Modelos: Nº Total de Aciertos

Como puede observarse en las Figuras 6.1 y 6.2, se muestra claramente la brecha de rendimiento: **Gemini 2.5 Flash** se acerca a la perfección, seguido de **GPT-4o** y, en última posición, **Claude Sonnet 4**. La Figura 6.1 refleja el desempeño porcentual, mientras que la Figura 6.2 corrobora esta tendencia mostrando el número absoluto de campos extraídos correctamente.

6.2 Precisión por Campo

Más allá de la precisión global, es crucial comprender el rendimiento de cada modelo en campos específicos. Esto es debido a que ciertos datos de las facturas (como la dirección de servicio o la fecha de vencimiento del pago) presentan una mayor criticidad para las operaciones del negocio y, a menudo, mayor complejidad en su extracción debido a formatos variables o a la necesidad de contextualización. Es en este punto donde la capacidad del LLM para interpretar y aplicar el *prompt* se pone a prueba de manera más exigente.

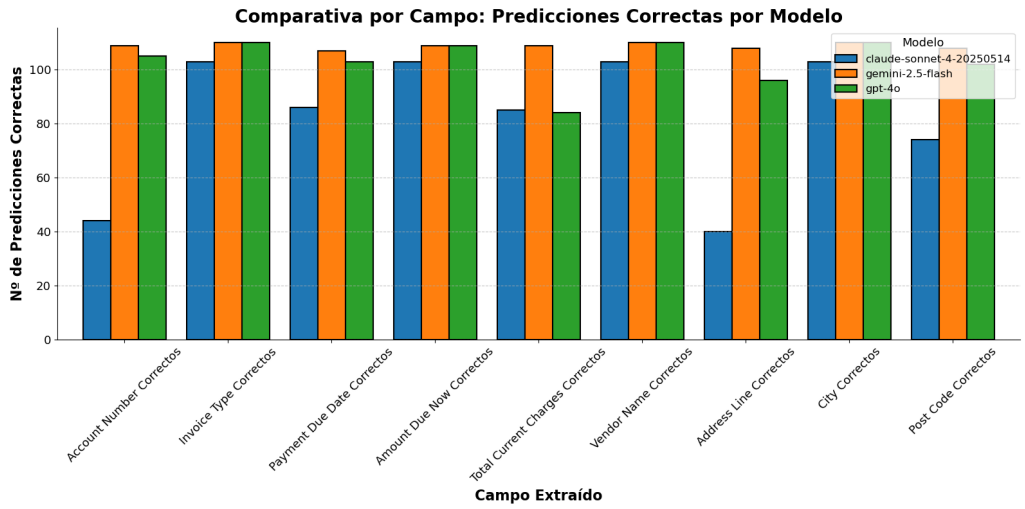


Figura 6.3. Comparativa de Rendimiento por Campo y Modelo

Del análisis del Gráfico 6.3 se observan patrones de rendimiento diferenciados por campo, que revelan directamente la eficacia de cada LLM en la interpretación de las instrucciones del *prompt*:

- **Gemini 2.5 Flash:** mantiene un liderazgo claro, con tasas de acierto prácticamente perfectas en todos los campos clave. Destaca su consistencia excepcional en la precisión a nivel de campo, alcanzando el 100 % o 99 % de exactitud en `account_number` (109 aciertos), `invoice_type` (110), `city` (110), `amount_due_now` (109) y *total_current_charges* (109). Incluso en `address_line`, uno de los campos más complejos por su variabilidad estructural y las reglas de normalización del *prompt*, logra un 98.18 % (108 aciertos).
- **GPT-4o:** presenta resultados también muy sólidos (por encima del 96 % en la mayoría de los campos), reflejando una buena comprensión del *prompt*. Sin embargo, muestra una caída significativa en `total_current_charges` (84 aciertos, 76.36 %), así como un rendimiento inferior a Gemini en `address_line` (96 aciertos).
- **Claude Sonnet 4:** evidencia una gran variabilidad en los resultados por campo, reflejando una menor adherencia a las instrucciones del *prompt*. Aunque mantiene una buena precisión en `vendor_name` (103 aciertos) e `invoice_type` (103), falla gravemente en `address_line` (40 aciertos, 36.36 %) y `account_number` (44 aciertos, 40 %). Estas

deficiencias comprometen seriamente su fiabilidad en un entorno productivo.

6.2.1. Relevancia de la interpretación del prompt.

Uno de los objetivos centrales de este TFM es evaluar la capacidad de los modelos para comprender y aplicar de manera correcta las instrucciones del *prompt*. Al aplicar el mismo *prompt* a todos los modelos, se observa que algunos muestran una mayor disciplina en seguir reglas de normalización y advertencias explícitas, mientras que otros fallan en su cumplimiento. Este aspecto resulta crítico, ya que un modelo puede tener un buen OCR en la lectura de texto, pero ser ineficaz en la aplicación de las reglas semánticas que garantizan la validez de la extracción.

Para ello, los umbrales de *fuzzy_match* definidos en la evaluación no son excesivamente flexibles: fueron diseñados para penalizar desviaciones significativas en la interpretación del *prompt* y asegurar que la comparación refleje no solo la calidad de la lectura (OCR), sino también la capacidad de entendimiento e interpretación semántica (NLP).

6.3 Costes y Tiempos de Ejecución

Además de la precisión, un sistema de Procesamiento Inteligente de Documentos (IDP) debe ser evaluado desde una perspectiva económica y de eficiencia temporal. Para ello, se analizaron tres métricas principales: el **coste total**, el **coste medio por predicción** y la **duración media por factura**. Estos indicadores permiten valorar la viabilidad de cada modelo en escenarios de despliegue real a gran escala, donde el coste acumulado y la latencia de procesamiento son factores críticos.

Tabla 6.2. Métricas de Coste y Duración por Modelo

Modelo	Coste Total (€)	Coste Medio (€)	Duración Media (s)
Claude Sonnet 4	0.7724	0.0074	5.97
Gemini 2.5 Flash	0.3692	0.0033	7.99
GPT-4o	0.7792	0.0070	7.19

Del análisis de la Tabla 6.2 se desprenden varias conclusiones relevantes:

- **Gemini 2.5 Flash** ofrece la mejor relación *coste-beneficio*. Su coste medio por predicción (0.00336 €) es el más bajo del estudio, lo que se traduce en un coste total de apenas 0.369 € para las 990 predicciones realizadas. Aunque su duración media se acerca a los 8 segundos, esta latencia es perfectamente aceptable en entornos asíncronos. En consecuencia, Gemini no solo destaca por su precisión (sección 6.1), sino también por ser la opción más rentable en términos operativos.
- **Claude Sonnet 4** y **GPT-4o** presentan costes totales muy similares, en torno a los 0.77–0.78 €, lo que supone prácticamente duplicar el coste de Gemini. Claude Sonnet

4, sin embargo, tiene la ventaja de ser el modelo más rápido (5.97 segundos por factura), aunque su baja precisión lo convierte en la opción menos eficiente en términos globales. GPT-4o, por su parte, alcanza un tiempo de procesamiento comparable al de Gemini (7.19 segundos), pero con un coste significativamente más alto, lo que reduce su competitividad en escenarios masivos.

Visualización de Coste y Duración

Para complementar el análisis numérico, se desarrollaron representaciones gráficas que ilustran las relaciones entre precisión, coste y velocidad.

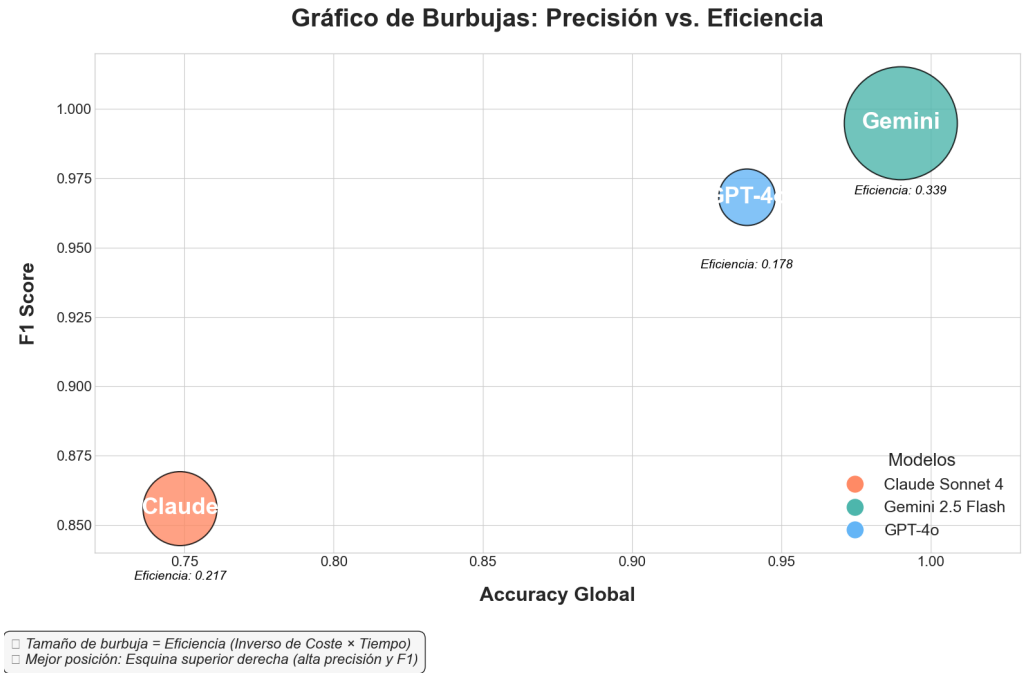


Figura 6.4. Precisión, Coste y Velocidad de Procesamiento — Modelos IDP

El Gráfico de Burbujas 6.4 proporciona una visión tridimensional del rendimiento de los tres modelos evaluados:

- **Eje X:** Precisión Global.
- **Eje Y:** F1 Score.
- **Tamaño de la burbuja:** Eficiencia combinada (inversa de *Coste Total × Duración Media*).

Los resultados muestran de forma clara que:

- **Gemini 2.5 Flash** domina en todas las dimensiones: máxima precisión, alto F1 y mayor eficiencia global, reflejada en el gran tamaño de su burbuja.

- **Claude Sonnet 4** se sitúa en la parte inferior izquierda, sacrificando precisión a cambio de rapidez. Su eficiencia intermedia indica que, aunque es rápido, los fallos en la extracción reducen drásticamente su valor práctico.
- **GPT-4o** ocupa una posición intermedia: sólido en precisión, pero penalizado por un coste más elevado, lo que lo convierte en el menos eficiente del grupo.

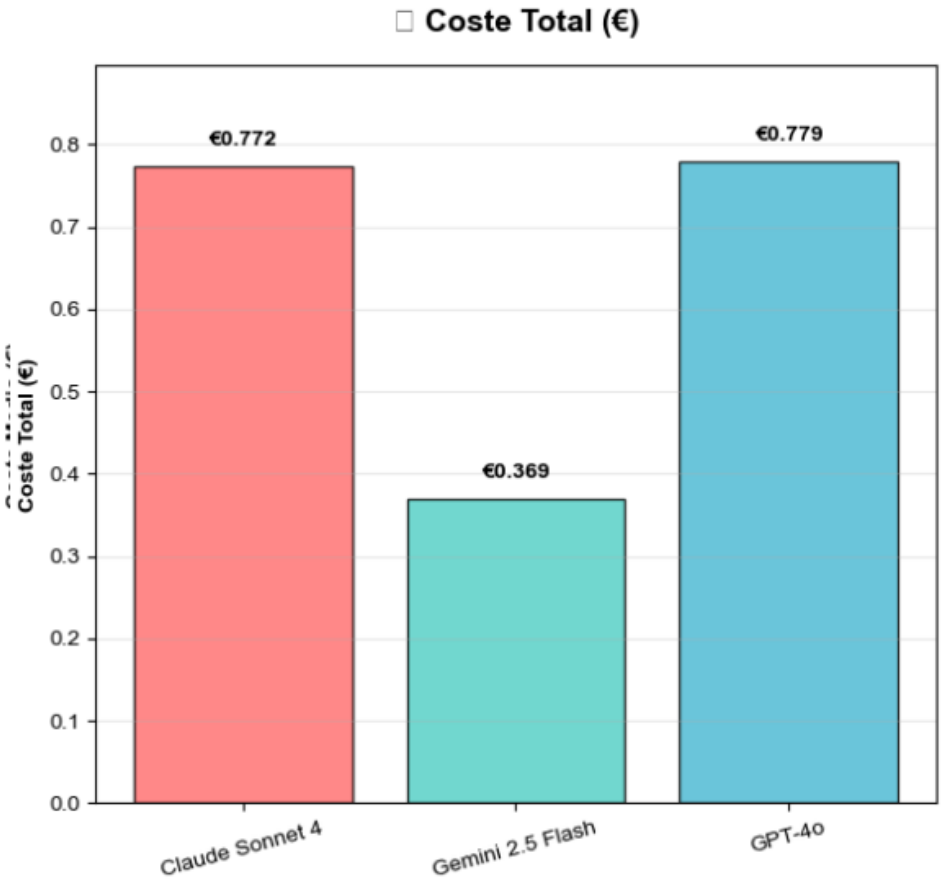


Figura 6.5. Comparativa Coste Total por Modelo

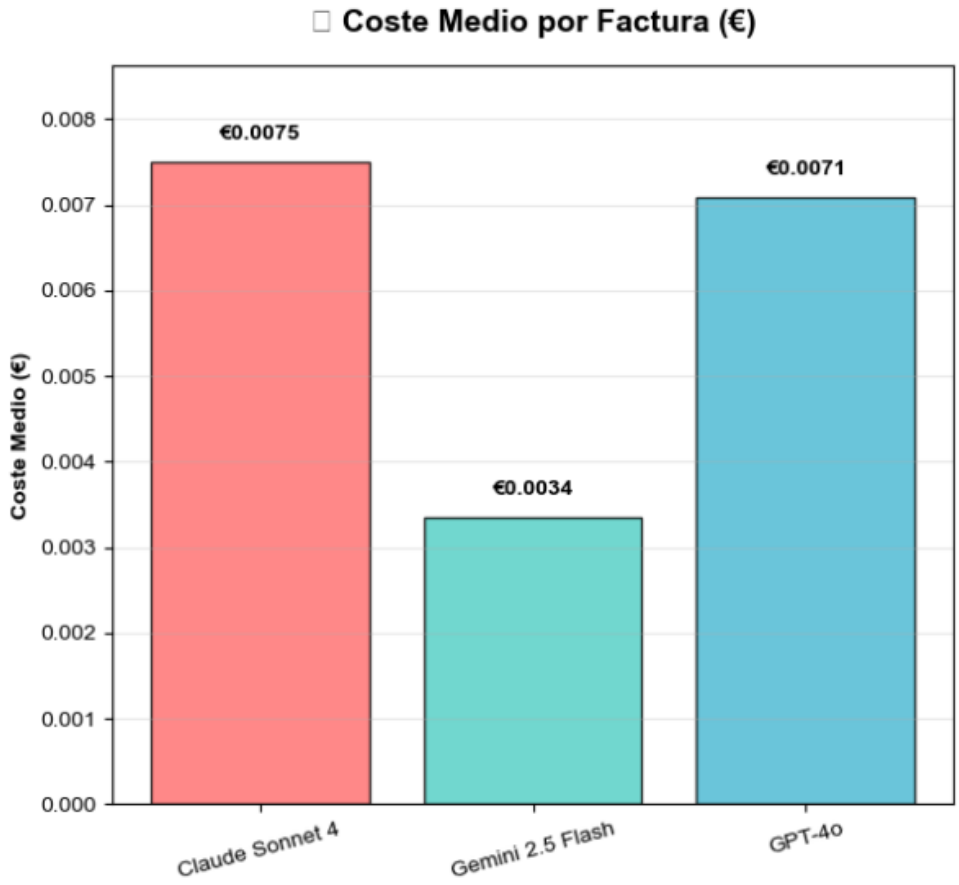


Figura 6.6. Comparativa Coste Medio por Factura por Modelo

Los gráficos 6.5 y 6.6 refuerzan las conclusiones anteriores:

- **Coste Total (€):** Gemini 2.5 Flash es el modelo más económico con 0.369 €, mientras que Claude Sonnet 4 y GPT-4o duplican aproximadamente este valor (0.77–0.78 €). Esta diferencia implica que, en proyectos de alto volumen, Gemini permite escalar el procesamiento con un impacto financiero mucho menor.
- **Coste Medio por Factura (€):** La tendencia se mantiene al nivel unitario: Gemini presenta un coste medio de 0.0034 €, frente a los 0.007 € de Claude Sonnet 4 y GPT-4o. Esto supone que, por cada factura procesada, Gemini es más del doble de eficiente en términos de coste.

En conjunto, los resultados confirman que **Gemini 2.5 Flash no solo es el modelo más preciso, sino también el más rentable**, lo que lo convierte en la opción preferente para aplicaciones reales de IDP a gran escala. GPT-4o, aunque competitivo en precisión, se ve limitado por su coste; y Claude Sonnet 4, a pesar de su rapidez, queda descartado por sus problemas de fiabilidad.

Capítulo 7. DISCUSIÓN

Este Trabajo Fin de Máster ha permitido comprobar en un escenario real las fortalezas y debilidades de los modelos de lenguaje multimodales aplicados al Procesamiento Inteligente de Documentos (IDP). A diferencia de un análisis meramente técnico, este capítulo busca interpretar los resultados obtenidos, contrastarlos con la literatura existente, identificar limitaciones metodológicas y valorar el alcance de las aportaciones para entornos productivos.

7.1 Obediencia al Prompt como Factor Diferenciador

Más allá de la precisión numérica, uno de los hallazgos centrales es que la **obediencia al prompt** constituye un factor decisivo para la fiabilidad de los modelos. Mientras que Gemini 2.5 Flash mostró una capacidad consistente para aplicar reglas complejas (como no fusionar números de direcciones o devolver fechas estrictamente en formato `MM/DD/YYYY`), Claude Sonnet 4 tendió a incumplir estas instrucciones, evidenciando una menor robustez semántica. GPT-4o se situó en una posición intermedia: sólido en términos de exactitud, pero con fallos puntuales en normalización.

Estos resultados coinciden con estudios recientes que demuestran que la calidad y el diseño del *prompt* impactan directamente en la precisión y consistencia de los LLMs [3], [4]. La capacidad de seguir reglas negativas específicas (ej. "`NEVER join numbers`") permite diferenciar entre modelos con verdadero razonamiento contextual y aquellos que solo reproducen patrones estadísticos.

7.2 Comparativa Crítica entre Modelos

El análisis comparativo revela un **trade-off** entre precisión, coste y velocidad, también identificado en estudios previos sobre Document AI [1], [5]:

- **Gemini 2.5 Flash:** mejor balance global, con un F1 superior al 0.99 y el menor coste medio por factura, aunque con tiempos de respuesta algo más altos (sobre los 8s).
- **GPT-4o:** gran precisión (F1=0.97), pero con un coste medio más elevado, lo que lo hace menos competitivo en escenarios de gran escala.
- **Claude Sonnet 4:** tiempos muy bajos (<6s) pero con F1=0.85 y frecuentes errores de normalización, además de limitaciones técnicas con imágenes >5MB.

Estos resultados (ver Figuras 6.4 y 6.5) sugieren que la elección del modelo depende del **contexto de aplicación**: Gemini es idóneo para proyectos de alto volumen donde la precisión es crítica; GPT-4o para tareas con menor volumen pero gran necesidad de exactitud; Claude únicamente en usos exploratorios o de baja exigencia.

7.3 Limitaciones y Desafíos

El proyecto ha puesto de manifiesto varias limitaciones relevantes:

- **Dependencia del prompt:** pequeñas variaciones en la formulación pueden alterar significativamente los resultados, reforzando la importancia de la ingeniería de prompts como disciplina.
- **Restricciones técnicas:** algunos modelos imponen límites de tamaño de archivo o de tokens que reducen su aplicabilidad en entornos empresariales.
- **Dataset reducido:** aunque se usaron 330 facturas (110 por modelo), sería deseable ampliar el corpus a miles de documentos de distintos países e idiomas para mejorar la validez externa.

7.4 Aportes al Estado del Arte

La principal contribución frente a la literatura revisada es la **integración de evaluación semántica mediante fuzzy matching** como criterio complementario de exactitud. Este enfoque permite valorar la verdadera comprensión de los modelos más allá de la extracción literal, respondiendo a la falta de métricas estandarizadas señalada en la literatura [1], [2].

Además, el marco metodológico reproducible basado en microservicios constituye un aporte práctico y extensible a otros dominios documentales (contratos, informes médicos, albaranes).

7.5 Implicaciones para IDP en Producción

Los hallazgos sugieren que los sistemas IDP deben incorporar los siguientes principios:

- **Arquitectura modular:** que permita combinar modelos en función de la tarea y reemplazarlos sin alterar el flujo completo.
- **Estrategias híbridas:** donde un modelo rápido filtre o clasifique documentos y otro más preciso realice la extracción final.
- **Evaluación continua:** con métricas automáticas integradas en pipelines de monitorización, como mecanismo de control de calidad y ventaja competitiva.

Esto conecta con las tendencias actuales en IDP descritas en revisiones sistemáticas recientes [1], [8], que apuntan hacia sistemas adaptativos, gobernados por métricas y diseñados para entornos reales. La replicabilidad del presente sistema constituye, por tanto, un aporte directo a la adopción práctica del IDP en organizaciones.

Capítulo 8. CONCLUSIONES

Este Trabajo Fin de Máster ha logrado diseñar, implementar y validar un sistema completo de Procesamiento Inteligente de Documentos (IDP) orientado a la extracción de datos de facturas de suministros. A continuación, se presentan las conclusiones principales, organizadas en torno a los objetivos del proyecto.

8.1 Logros Principales

- **Arquitectura modular y escalable:** se implementó una infraestructura basada en microservicios (n8n, FastAPI y PostgreSQL en contenedores Docker), capaz de gestionar el ciclo de vida completo de las facturas y facilitar la integración de distintos modelos LLM.
- **Automatización de extremo a extremo:** el flujo abarca desde la recepción de documentos hasta la normalización y almacenamiento estructurado, reduciendo significativamente la intervención manual.
- **Evaluación innovadora:** el microservicio `ocr-evaluator` introdujo una metodología novedosa que combina métricas tradicionales con fuzzy matching, permitiendo evaluar no solo la precisión, sino la obediencia semántica a los prompts.
- **Comparación objetiva de modelos:** se analizaron tres LLMs (Claude Sonnet 4, GPT-4o y Gemini 2.5 Flash) con un dataset real, obteniendo métricas comparables de precisión, coste y tiempo.

8.2 Conclusiones sobre los Modelos

- **Gemini 2.5 Flash:** modelo más preciso (98.99 % de exactitud), más obediente al prompt y más económico, lo que lo convierte en la mejor opción para proyectos de gran escala.
- **GPT-4o:** ofrece precisión sólida (93.84 %), pero con un coste más alto que limita su escalabilidad.
- **Claude Sonnet 4:** destaca por su rapidez, pero sus limitaciones de precisión y robustez lo hacen poco adecuado para entornos exigentes.

8.3 Beneficios y Aportaciones

- Reducción significativa de costes y tiempos frente a procesos manuales.
- Mejora de la calidad y fiabilidad de los datos extraídos.
- Contribución metodológica mediante un marco reproducible de evaluación que puede aplicarse a futuros proyectos de IDP.

8.4 Líneas Futuras de Investigación

- Ampliar el dataset de facturas a otros sectores e idiomas para mejorar la generalización.
- Incorporar modelos adicionales, incluyendo soluciones open-source como Donut o LayoutLMv3.

- Explorar técnicas de optimización de prompts dinámicos y agentes que ajusten automáticamente las instrucciones según el documento.

En definitiva, este trabajo demuestra que la integración de LLMs multimodales con una arquitectura robusta y una evaluación rigurosa no solo es técnicamente viable, sino que constituye una solución práctica y escalable para la automatización de la gestión documental.

Bibliografía

[1] L. Le Saout, G. Carbonnel, O. Boissier y F. Gandon, «A Systematic Review of Document Intelligence: Trends, Challenges and Opportunities,» *Information Processing & Management*, vol. 61, n.º 2, pág. 103 402, 2024.

[2] K. Vijaya y K. Rajalakshmi, *Invoice Data Extraction Using Deep Learning OCR*, Available online, 2024. dirección: https://www.ijert.org/invoice-data-extraction-using-deep-learning-ocr.

[3] S. Atreja, J. Ashkinaze, L. Li, J. Mendelsohn y L. Hemphill, «What's in a Prompt?: A Large-Scale Experiment to Assess the Impact of Prompt Design on the Compliance and Accuracy of LLM-Generated Text Annotations,» *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 19, n.º 1, págs. 122-145, jun. de 2025. DOI: 10.1609/icwsm.v19i1.35807. dirección: https://ojs.aaai.org/index.php/ICWSM/article/view/35807.

[4] L. Pawlik, «How the Choice of LLM and Prompt Engineering Affects Chatbot Effectiveness,» *Electronics*, vol. 14, n.º 5, 2025, ISSN: 2079-9292. DOI: 10.3390/electronics14050888. dirección: https://www.mdpi.com/2079-9292/14/5/888.

[5] M. J. Ha, S. Lee, J. Jang et al., «OCRMineR: Extracting Text and Layout Features for Invoice Information Extraction,» *Journal of Information Science*, 2022.

[6] G. Kim, T. Hong, M. Yim et al., *OCR-free Document Understanding Transformer*, 2022. arXiv: 2111.15664 [cs.LG]. dirección: https://arxiv.org/abs/2111.15664.

[7] L. Li, *Handwriting Recognition in Historical Documents with Multimodal LLM*, 2024. arXiv: 2410.24034 [cs.CV]. dirección: https://arxiv.org/abs/2410.24034.

[8] R. Tafesse, *Empowering Innovation with Workday AI: Comparative Study on Document AI Tools*, Master's Thesis, KTH Royal Institute of Technology, 2024. dirección: https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1834560.

[9] D. Inc., *Docker - Empowering App Development for Developers*, Último acceso: septiembre 2025, 2024. dirección: https://www.docker.com.

[10] n8n.io, *n8n - Workflow automation tool*, Último acceso: septiembre 2025, 2024. dirección: https://n8n.io.

[11] T. P. G. D. Group, *PostgreSQL - The world's most advanced open source database*, Versión 15, Último acceso: septiembre 2025, 2024. dirección: https://www.postgresql.org.

[12] T. Labs, *Traefik Proxy - Cloud Native Edge Router*, Último acceso: septiembre 2025, 2024. dirección: https://traefik.io.

[13] Belval, *pdf2image - Python library for PDF to image conversion*, Último acceso: septiembre 2025, 2024. dirección: https://github.com/Belval/pdf2image.

[14] M. Bachmann, *RapidFuzz: Fuzzy String Matching in Python*, Versión 3.6.1, Último acceso: septiembre 2025, 2024. dirección: `https://maxbachmann.github.io/RapidFuzz`.

Capítulo 9. ANEXOS

9.1 Prompt utilizado para la extracción

Prompt completo

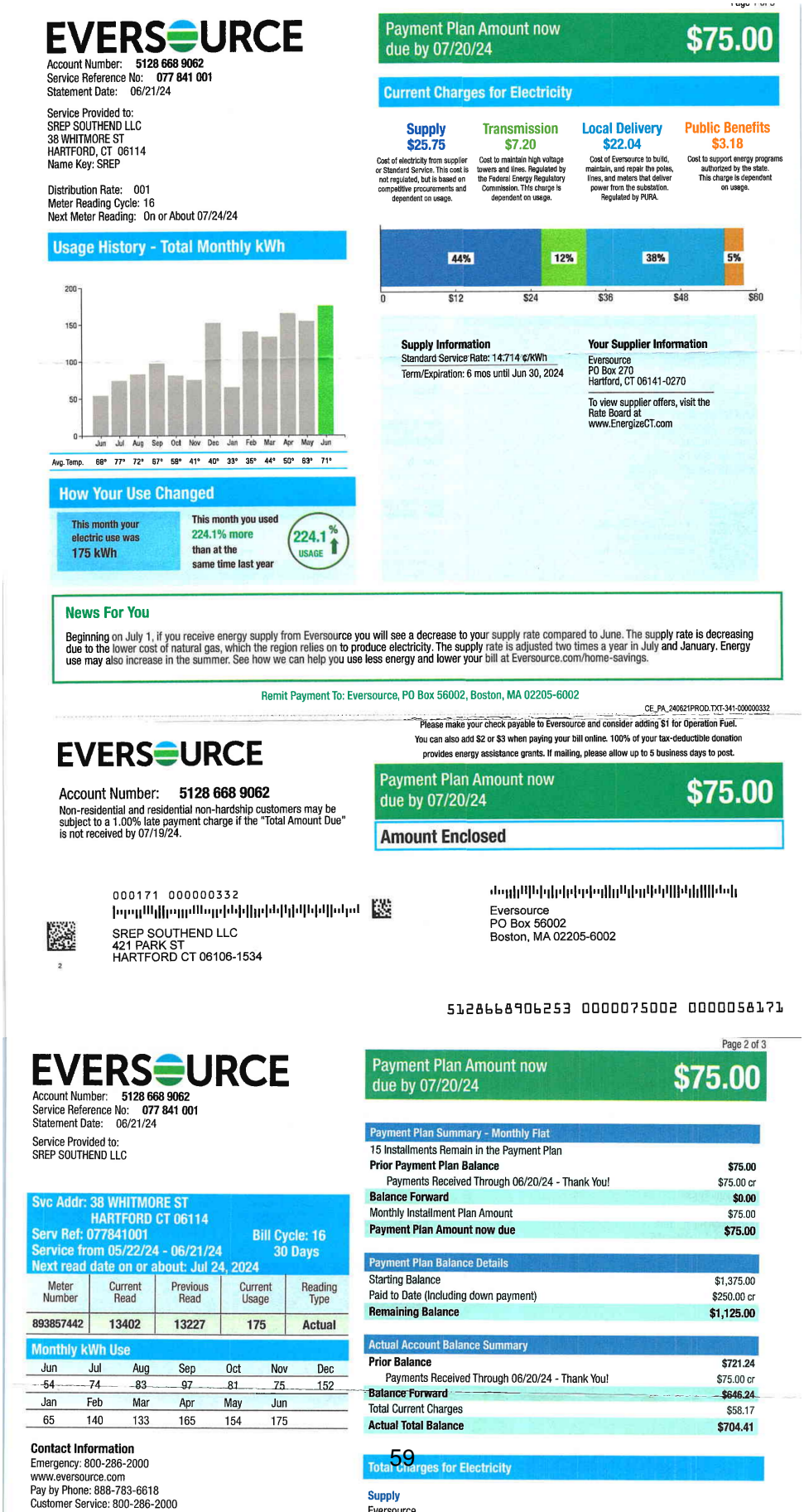
```
Eres un extractor experto de facturas de servicios públicos.

Tu tarea es extraer únicamente los siguientes campos de la factura:

- vendor_name
- account_number
- address_line
- payment_due_date
- amount_due_now
- total_current_charges

INSTRUCCIONES GENERALES:
- Devuelve solo los campos solicitados, sin explicaciones ni inferencias.
- El resultado debe estar en formato JSON array válido.
- Cada factura debe ser un objeto independiente en el JSON.

REGLAS ESPECÍFICAS:
1. address_line:
   - Extrae ÚNICAMENTE la dirección de servicio.
   - Si hay varias direcciones, selecciona la asociada al uso/servicio.
   - NUNCA unir números separados. Ejemplo: "31 33 PARK ST" != "3133 PARK ST".
   - Concatenar en una sola línea direcciones divididas.
   - Mover códigos de unidad al final. Ejemplo: "A C4 101 MAPLE AVE" → "101 MAPLE AVE A C4"
   - Ignorar direcciones irrelevantes (PO BOX, facturación, envío).

2. account_number:
   - Devuelve solo caracteres numéricos, sin guiones ni espacios.
   - Ejemplo: "040-0011184-7196" → "04000111847196".

3. payment_due_date:
   - Devuelve en formato MM/DD/YYYY.
   - Si el año aparece en 2 dígitos, conviértelo a 4 dígitos.
   - Ejemplo: "11/22/24" → "11/22/2024".
```

```
4. amount_due_now y total_current_charges:
   - Devuelve como número decimal.
   - Elimina símbolos de moneda, comas o espacios.
   - Ejemplo: "Total Amount Due: $160.70" → 160.70.

5. vendor_name:
   - Normaliza el nombre del proveedor a su abreviación conocida.
   - Ejemplo: "Connecticut Natural Gas Corporation" → "CNG".

6. Multi-página:
   - Si un PDF contiene varias facturas, devuelve cada una como objeto independiente.
   - Detecta nuevas facturas cuando cambie vendor_name, account_number o payment_due_date.

SALIDA FINAL:
Devuelve SIEMPRE un JSON array válido.

Ejemplo de salida:
[
  {
    "vendor_name": "CNG",
    "account_number": "04000111847196",
    "address_line": "101 MAPLE AVE A C4",
    "payment_due_date": "11/22/2024",
    "amount_due_now": 160.70,
    "total_current_charges": 245.15
  }
]
```

9.2 Ejemplos de facturas utilizadas

A continuación, se muestran ejemplos representativos de las facturas reales utilizadas en el dataset. Todas las imágenes han sido anonimizadas previamente para garantizar la privacidad de los datos.

Factura de Luz



EVERSOURCE

Account Number: **5128 668 9062**
Service Reference No: **077 841 001**
Statement Date: 06/21/24

Service Provided to:
SREP SOUTHEND LLC
38 WHITMORE ST
HARTFORD, CT 06114
Name Key: SREP

Distribution Rate: 001
Meter Reading Cycle: 16
Next Meter Reading: On or About 07/24/24

Payment Plan Amount now due by 07/20/24 — $75.00

Current Charges for Electricity

Supply	Transmission	Local Delivery	Public Benefits
$25.75	$7.20	$22.04	$3.18
Cost of electricity from supplier or Standard Service. This cost is not regulated, but is based on competitive procurements and dependent on usage.	Cost to maintain high voltage towers and lines. Regulated by the Federal Energy Regulatory Commission. This charge is dependent on usage.	Cost of Eversource to build, maintain, and repair the poles, lines, and meters that deliver power from the substation. Regulated by PURA.	Cost to support energy programs authorized by the state. This charge is dependent on usage.

Usage History - Total Monthly kWh

How Your Use Changed

This month your electric use was **175 kWh**

This month you used **224.1% more** than at the same time last year

224.1 % USAGE

Supply Information
Standard Service Rate: 14.714 ¢/kWh
Term/Expiration: 6 mos until Jun 30, 2024

Your Supplier Information
Eversource
PO Box 270
Hartford, CT 06141-0270

To view supplier offers, visit the Rate Board at www.EnergizeCT.com

News For You

Beginning on July 1, if you receive energy supply from Eversource you will see a decrease to your supply rate compared to June. The supply rate is decreasing due to the lower cost of natural gas, which the region relies on to produce electricity. The supply rate is adjusted two times a year in July and January. Energy use may also increase in the summer. See how we can help you use less energy and lower your bill at Eversource.com/home-savings.

Remit Payment To: Eversource, PO Box 56002, Boston, MA 02205-6002

CE_PA_240621PROD.TXT-341-000000332

Please make your check payable to Eversource and consider adding $1 for Operation Fuel.
You can also add $2 or $3 when paying your bill online. 100% of your tax-deductible donation provides energy assistance grants. If mailing, please allow up to 5 business days to post.

EVERSOURCE

Account Number: **5128 668 9062**
Non-residential and residential non-hardship customers may be subject to a 1.00% late payment charge if the "Total Amount Due" is not received by 07/19/24.

Payment Plan Amount now due by 07/20/24 — $75.00

Amount Enclosed

000171 000000332

SREP SOUTHEND LLC
421 PARK ST
HARTFORD CT 06106-1534

Eversource
PO Box 56002
Boston, MA 02205-6002

5128668906253 0000075002 0000058171

EVERSOURCE

Account Number: **5128 668 9062**
Service Reference No: **077 841 001**
Statement Date: 06/21/24

Service Provided to:
SREP SOUTHEND LLC

Payment Plan Amount now due by 07/20/24 — $75.00

Svc Addr: 38 WHITMORE ST HARTFORD CT 06114
Serv Ref: 077841001 — **Bill Cycle: 16**
Service from 05/22/24 - 06/21/24 — **30 Days**
Next read date on or about: Jul 24, 2024

Meter Number	Current Read	Previous Read	Current Usage	Reading Type
893857442	13402	13227	175	Actual

Monthly kWh Use

Jun	Jul	Aug	Sep	Oct	Nov	Dec
54	74	83	97	81	75	152
Jan	Feb	Mar	Apr	May	Jun	
65	140	133	165	154	175	

Payment Plan Summary - Monthly Flat

15 Installments Remain in the Payment Plan	
Prior Payment Plan Balance	**$75.00**
Payments Received Through 06/20/24 - Thank You!	$75.00 cr
Balance Forward	**$0.00**
Monthly Installment Plan Amount	$75.00
Payment Plan Amount now due	**$75.00**

Payment Plan Balance Details

Starting Balance	$1,375.00
Paid to Date (Including down payment)	$250.00 cr
Remaining Balance	**$1,125.00**

Actual Account Balance Summary

Prior Balance	**$721.24**
Payments Received Through 06/20/24 - Thank You!	$75.00 cr
Balance Forward	**$646.24**
Total Current Charges	$58.17
Actual Total Balance	**$704.41**

Total Charges for Electricity

Supply
Eversource

Contact Information
Emergency: 800-286-2000
www.eversource.com
Pay by Phone: 888-783-6618
Customer Service: 800-286-2000

Factura de Agua

MDC The Metropolitan District
water supply • environmental services • geographic information
The Metropolitan District
555 Main Street
Hartford, CT 06103
www.themdc.org / 860.278.7850

Account No.: 21142931 **Page:** 1 of 3
Invoice No.: 630001183428 **Invoice Date:** 03/25/2025
Customer Name: SREP Southend 2, LLC
Service Address: 47 49 FRANKLIN AVE
HARTFORD CT 06106

Billing Summary

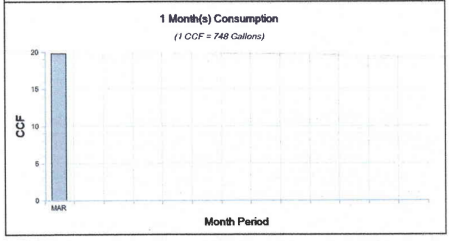
Billing Period
02/26/2025 - 03/16/2025 (19 Days)

Previous Balance	$0.00
Payments Received	$0.00
Remaining Balance	**$0.00**
Service Charges	$182.84
Other Charges & Adjustments	$0.00
Total Current Charges	**$182.84**
TOTAL AMOUNT DUE	**$182.84**

Due Date: 04/21/2025 — **$182.84** Total Amount Due

1% INTEREST ADDED IF NOT **PAID** BY DUE DATE.

Usage Summary (CCF)



CUSTOMER INFORMATION
- 25-day payment terms: The District allows five (5) additional days after the payment due date for customer payments to post to their account before the 1% interest is applied to any remaining unpaid balance.
- If you have already paid your previous balance, thank you for your payment.
- Go Paperless! Go to www.themdc.org and click "Pay Your Bill" to set up your online profile.

Email change of mailing address to customerservice@themdc.com or call (860) 278-7850
Keep this portion for your records. Go Paperless. Pay online at www.themdc.org.
MP250326.csv-5983-000001138
Please return this portion with your payment.

MDC The Metropolitan District
water supply • environmental services • geographic information
Mail Payment / Make Payable to:
The Metropolitan District
P.O. Box 5535
Binghamton, NY 13902-5535

Invoice No.: 630001183428 **Invoice Date:** 03/25/2025
Service Address: 47 49 FRANKLIN AVE **Account No.:** 21142931
HARTFORD CT 06106

Amount Due By 04/21/2025	$ 182.84
DONATION add a dollar or other amount to Operation Fuel	
Total Amount Enclosed	$

002992 000001138
SREP SOUTHEND 2. LLC
421 PARK ST
HARTFORD CT 06106-1534

5535000000000182840421202500111335204000021142931163000118342881

MDC The Metropolitan District
water supply • environmental services • geographic information
The Metropolitan District
555 Main Street
Hartford, CT 06103
www.themdc.org / 860.278.7850

Account No.: 21142931 **Page:** 3 of 3
Invoice No.: 630001183428 **Invoice Date:** 03/25/2025
Customer Name: SREP Southend 2, LLC
Service Address: 47 49 FRANKLIN AVE
HARTFORD CT 06106

***The Clean Water Project (CWP) Charge:**
The Federal Environmental Protection Agency (EPA) and CT State Department of Energy and Environmental Protection (DEEP) mandated Clean Water Project (CWP) Charge is applied for the repayment of the CWP costs and is calculated based on water consumption for customers who receive both water and sewer services. The CWP is a sewer infrastructure improvement project. Learn more, visit www.themdc.org/the-clean-water-project .

Due Date: 04/21/2025 — **$182.84** Total Amount Due

For an explanation of charges or to pay your bill online, please go to **www.themdc.org/customers/billing-services**

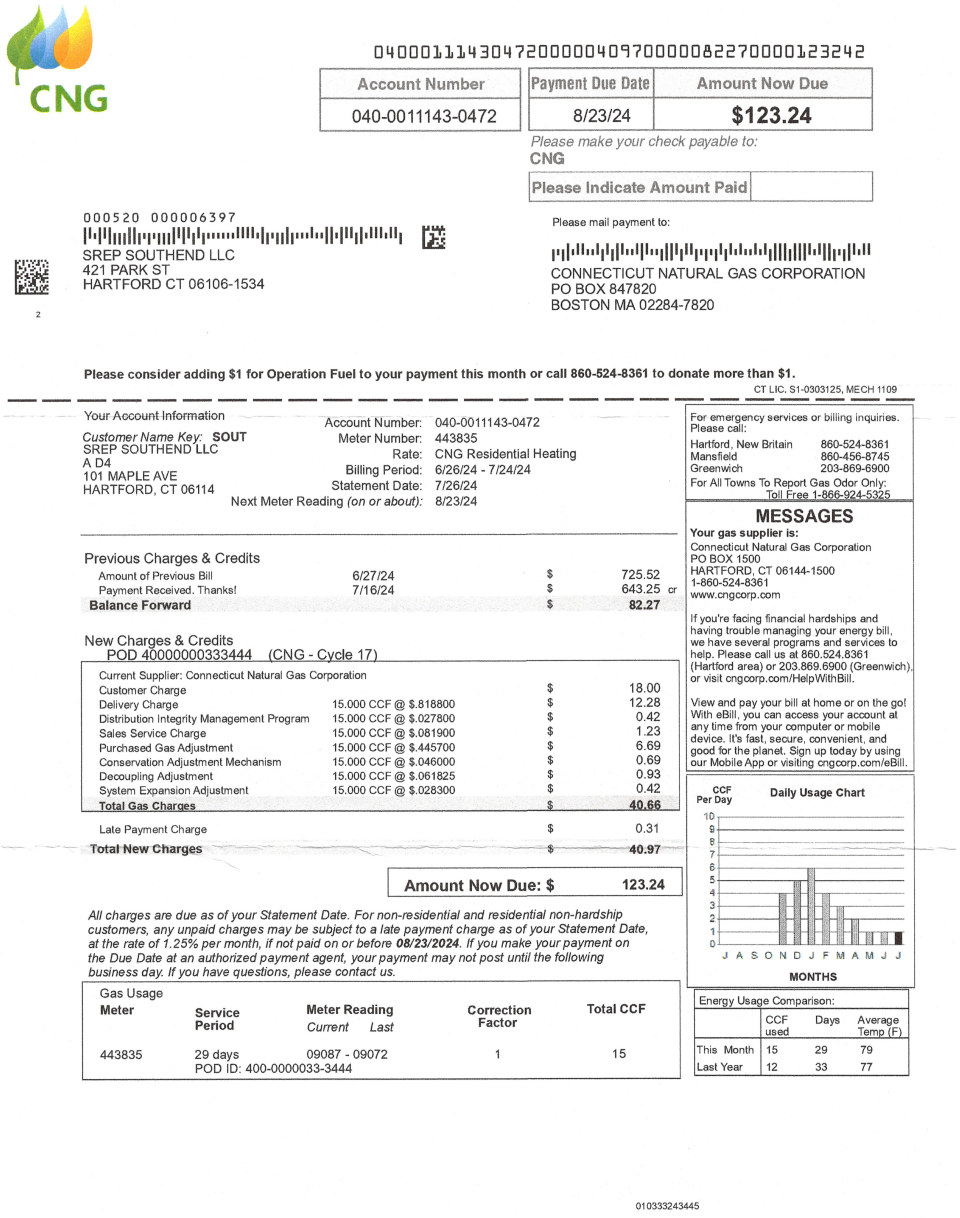
Current Charges

Service Charges	**$182.84**
Water Service	**$86.79**
2025 Water Used Charge @ $3.91 x 19.77 CCF	$77.30
(MDC Water rate for 2025 is $0.0052 per US-Gallon)	
2025 Water Customer Service Charge - 1"	$9.49
Sewer Service	**$5.70**
2025 Sewer Customer Service Charge	$5.70
Federal / State Regulatory Compliance Fees	**$90.35**
2025 DEEP/EPA CWP Charge* @ $4.57 x 19.77 CCF	$90.35

Meter Readings

Reading Date (03/16/2025)	
Meter Number	50010126
Meter Size	1"
Current Meter Reading	3,784.45
Previous Meter Reading	3,764.68 E
Water Usage (CCF)	19.77
Type of Meter Reading	Actual Reading

Factura de Gas



CNG

040001111430472000000409700000822700001232 42

Account Number	Payment Due Date	Amount Now Due
040-0011143-0472	8/23/24	$123.24

Please make your check payable to:
CNG

Please Indicate Amount Paid

000520 000006397
SREP SOUTHEND LLC
421 PARK ST
HARTFORD CT 06106-1534

Please mail payment to:
CONNECTICUT NATURAL GAS CORPORATION
PO BOX 847820
BOSTON MA 02284-7820

Please consider adding $1 for Operation Fuel to your payment this month or call 860-524-8361 to donate more than $1.

CT LIC. S1-0303125, MECH 1109

Your Account Information

Customer Name Key: **SOUT**
SREP SOUTHEND LLC
A D4
101 MAPLE AVE
HARTFORD, CT 06114

Account Number: 040-0011143-0472
Meter Number: 443835
Rate: CNG Residential Heating
Billing Period: 6/26/24 - 7/24/24
Statement Date: 7/26/24
Next Meter Reading *(on or about)*: 8/23/24

For emergency services or billing inquiries. Please call:
Hartford, New Britain 860-524-8361
Mansfield 860-456-8745
Greenwich 203-869-6900
For All Towns To Report Gas Odor Only:
Toll Free 1-866-924-5325

MESSAGES

Your gas supplier is:
Connecticut Natural Gas Corporation
PO BOX 1500
HARTFORD, CT 06144-1500
1-860-524-8361
www.cngcorp.com

If you're facing financial hardships and having trouble managing your energy bill, we have several programs and services to help. Please call us at 860.524.8361 (Hartford area) or 203.869.6900 (Greenwich), or visit cngcorp.com/HelpWithBill.

View and pay your bill at home or on the go! With eBill, you can access your account at any time from your computer or mobile device. It's fast, secure, convenient, and good for the planet. Sign up today by using our Mobile App or visiting cngcorp.com/eBill.

Previous Charges & Credits

Amount of Previous Bill	6/27/24	$	725.52
Payment Received. Thanks!	7/16/24	$	643.25 cr
Balance Forward		$	82.27

New Charges & Credits
POD 40000000333444 (CNG - Cycle 17)

Current Supplier: Connecticut Natural Gas Corporation			
Customer Charge		$	18.00
Delivery Charge	15.000 CCF @ $.818800	$	12.28
Distribution Integrity Management Program	15.000 CCF @ $.027800	$	0.42
Sales Service Charge	15.000 CCF @ $.081900	$	1.23
Purchased Gas Adjustment	15.000 CCF @ $.445700	$	6.69
Conservation Adjustment Mechanism	15.000 CCF @ $.046000	$	0.69
Decoupling Adjustment	15.000 CCF @ $.061825	$	0.93
System Expansion Adjustment	15.000 CCF @ $.028300	$	0.42
Total Gas Charges		$	40.66
Late Payment Charge		$	0.31
Total New Charges		$	40.97

Amount Now Due: $ 123.24

*All charges are due as of your Statement Date. For non-residential and residential non-hardship customers, any unpaid charges may be subject to a late payment charge as of your Statement Date, at the rate of 1.25% per month, if not paid on or before **08/23/2024**. If you make your payment on the Due Date at an authorized payment agent, your payment may not post until the following business day. If you have questions, please contact us.*

CCF Per Day — Daily Usage Chart

Gas Usage

Meter	Service Period	Meter Reading Current	Meter Reading Last	Correction Factor	Total CCF
443835	29 days	09087	09072	1	15

POD ID: 400-0000033-3444

Energy Usage Comparison:

	CCF used	Days	Average Temp (F)
This Month	15	29	79
Last Year	12	33	77

010333243445

Figura 9.3. Ejemplo de factura completa de suministro de gas utilizada en las pruebas.