RACKS ACADEMY

IUNIT Centro Universitario

ESPECIALISTA EN INTELIGENCIA ARTIFICIAL

MEMORIA DE PROYECTO

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA AUTOMATIZADO DE GESTIÓN Y PRIORIZACIÓN DE OFERTAS DE EMPLEO BASADO EN INTELIGENCIA ARTIFICIAL

Autor:

PABLO GUTIÉRREZ BUENO

Fecha:

8 DE DICIEMBRE DE 2025

ÍNDICE

1. RESUMEN / ABSTRACT

1.1. Resumen (Español)

La búsqueda de empleo en el sector de la ciberseguridad presenta una barrera de entrada elevada: se exigen conocimientos técnicos avanzados, experiencia demostrable y una capacidad constante de actualización. En este contexto, muchos profesionales reciben diariamente decenas de correos con ofertas de trabajo procedentes de distintas plataformas, que deben leer, filtrar y registrar de forma manual. Este proceso consume tiempo, es propenso a errores y dificulta la priorización de las oportunidades más relevantes.

El presente Trabajo Final de Máster describe el diseño, implementación y validación de un sistema de automatización para la búsqueda de empleo basado en Inteligencia Artificial. La solución se articula como un workflow en n8n que integra la API de Gmail, la API de OpenAI y Google Sheets. A partir de los correos de ofertas, el sistema realiza el preprocesado del contenido, extrae información estructurada mediante un modelo de lenguaje (GPT-4o-mini), calcula un *scoring* de relevancia en función de un perfil profesional definido y almacena los resultados en una hoja de cálculo con 23 columnas.

El sistema se ha evaluado mediante pruebas de *backtesting* sobre correos históricos y, especialmente, a través de su uso real durante varios meses de búsqueda activa de empleo. Los resultados muestran una alta precisión en la extracción de campos clave, un coste por email muy reducido y un tiempo de procesamiento inferior a 30 segundos por correo. De forma destacada, la solución contribuyó directamente a la obtención de un nuevo puesto de trabajo en el ámbito de la ciberseguridad, lo que valida su utilidad práctica más allá del laboratorio.

Palabras clave: Automatización, Inteligencia Artificial, Búsqueda de Empleo, Procesamiento de Lenguaje Natural, n8n, OpenAI, Extracción de Información.

1.2. Abstract (English)

Job search in the cybersecurity field has a particularly high entry barrier: employers typically require strong technical skills, proven experience and continuous learning. In practice, many professionals receive dozens of job-related emails every day from different platforms, which they must read, filter and record manually. This process is time-consuming, error-prone and makes it difficult to prioritise the most relevant opportunities.

This Master's Thesis presents the design, implementation and validation of an automated job-search support system based on Artificial Intelligence. The solution is implemented as an n8n workflow that integrates the Gmail API, the OpenAI API and Google Sheets. Starting from job-offer emails, the system preprocesses the content, extracts structured information using a language model (GPT-4o-mini), computes a relevance score according to a predefined professional profile, and stores the results in a 23-column spreadsheet.

The system has been evaluated through backtesting on historical emails and, more importantly, through real use during several months of active job search. The results show high accuracy in extracting key fields, very low cost per processed email and a processing time below 30 seconds per email. Remarkably, the solution directly contributed to

obtaining a new job in the cybersecurity domain, which validates its practical usefulness beyond a purely academic setting.

Keywords: Automation, Artificial Intelligence, Job Search, Natural Language Processing, n8n, OpenAI, Information Extraction.

2. INTRODUCCIÓN

2.1. Contexto y motivación

Este proyecto surge de una necesidad personal en un momento de elevada carga profesional y académica. Durante varios meses, el autor compaginó una jornada laboral completa con la finalización de un Máster en Ciberseguridad y formación adicional en Inteligencia Artificial, todo ello intentando preservar un mínimo de tiempo libre y de descanso. En marzo de 2025 se completó el máster de ciberseguridad, con el objetivo claro de reorientar la carrera profesional hacia este sector.

A pesar de contar con una ingeniería en telecomunicaciones, buen nivel de inglés, experiencia laboral previa, proyectos personales, altos conocimientos de inteligencia artificial y una formación sólida en ciberseguridad, la realidad del mercado fue más exigente de lo esperado. Desde finales de 2024 se inició una búsqueda activa de empleo en ciberseguridad y áreas afines, pero la mayoría de candidaturas eran rechazadas sin llegar siquiera a una primera entrevista. Esta situación evidenció dos problemas simultáneos: por un lado, la alta competitividad del sector; por otro, el tiempo invertido en leer, filtrar y gestionar manualmente correos de ofertas de trabajo que, en muchos casos, no se ajustaban realmente al perfil deseado.

A partir de esta experiencia se planteó la necesidad de automatizar al máximo el proceso de gestión de ofertas de empleo, no solo para ahorrar tiempo, sino también para introducir criterios más objetivos y sistemáticos de priorización. Aprovechando los conocimientos adquiridos en automatización e IA generativa, se decidió aparcar temporalmente otros proyectos personales para centrarse en diseñar un sistema que procesara automáticamente los correos recibidos, extrajera la información relevante, evaluara el encaje de cada oferta con el perfil del candidato y almacenara los resultados en un repositorio estructurado y fácilmente explotable.

La primera versión funcional del sistema se completó en mayo de 2025 y permitió obtener las primeras entrevistas de calidad. Durante el verano se continuó iterando sobre la solución: se afinó el *scoring*, se mejoró la calidad de los datos extraídos, se revisó el currículum y se optimizaron las candidaturas enviadas. La versión actual del sistema, que es la que se documenta en esta memoria, contribuyó de forma directa a lograr un cambio de trabajo y la incorporación al sector de la ciberseguridad el 27 de octubre de 2025. Por este motivo, el presente trabajo se apoya no solo en resultados experimentales, sino en la validación real del sistema sobre el propio autor.

2.2. Problema a resolver

El problema central que aborda este trabajo es la ineficiencia en la gestión manual de ofertas de empleo recibidas por correo electrónico. De forma más concreta, se identifican los siguientes aspectos:

- **Volumen de información**: un candidato activo puede recibir diariamente múltiples correos de distintas plataformas (LinkedIn, InfoJobs, Tecnoempleo, etc.), lo que exige un esfuerzo considerable de lectura y clasificación manual.

- **Diversidad de formatos**: cada plataforma utiliza plantillas de correo diferentes (HTML, textos mixtos, mensajes breves, resúmenes, recordatorios), dificultando la extracción uniforme de información.
- **Falta de estructuración**: la información relevante (empresa, puesto, ubicación, salario, requisitos, estado de la candidatura) suele estar dispersa en el cuerpo del email, lo que obliga a leerlo completo y resumirlo manualmente.
- **Evaluación subjetiva y poco sistemática**: valorar la relevancia de una oferta implica ponderar varios factores (skills, experiencia requerida, ubicación, salario, tipo de contrato) frente al perfil personal. Sin apoyo de herramientas, este análisis suele ser subjetivo, inconsistente y difícil de escalar.
- **Gestión del ciclo de vida de la candidatura**: a medida que avanza un proceso de selección, el estado de la candidatura cambia (nueva oferta, en proceso, entrevista, rechazo, aceptación). Mantener un registro manual actualizado en hojas de cálculo o notas es tedioso y propenso a errores.
- **Falta de centralización**: la información relevante queda repartida en múltiples correos y plataformas, dificultando una visión global de todas las oportunidades y del histórico de candidaturas.

Este contexto se vuelve especialmente crítico cuando el tiempo disponible es limitado y el objetivo es maximizar la probabilidad de conseguir un puesto alineado con el perfil técnico y las expectativas profesionales. El problema no es solo técnico, sino también de eficiencia personal: cada minuto dedicado a tareas repetitivas de gestión manual es un minuto que no se invierte en estudiar, practicar, preparar entrevistas o mejorar el propio perfil. El sistema propuesto se centra precisamente en esta gestión y priorización de ofertas, sin sustituir al candidato en la decisión final ni en la acción de inscribirse o enviar candidaturas.

2.3. Justificación del proyecto

2.3.1. Relevancia práctica

El proyecto se justifica, en primer lugar, por su relevancia práctica. La automatización de la gestión de ofertas de empleo aborda un problema real y recurrente para cualquier profesional en búsqueda activa: el exceso de información y la dificultad para priorizar oportunidades. Un sistema que centralice los datos de las ofertas, los estructure y calcule un *scoring* objetivo puede ahorrar una cantidad significativa de tiempo y mejorar la calidad de las decisiones sobre dónde aplicar y en qué procesos implicarse más.

Además, el sistema no se ha validado únicamente en un entorno de laboratorio, sino que se ha utilizado como herramienta de trabajo real durante varios meses, hasta el punto de condicionar la decisión de cuándo entregar este trabajo: se prefirió esperar a disponer de evidencia suficiente de que la solución funcionaba en un caso real de transición profesional.

2.3.2. Aplicación de tecnologías emergentes

Desde el punto de vista tecnológico, el proyecto demuestra cómo los modelos de lenguaje de última generación pueden emplearse más allá de la generación de texto, aplicándose a tareas de:

- Extracción de información estructurada desde correos HTML heterogéneos.
- Inferencia de campos no explícitos, como tipo de contrato, nivel de experiencia o rangos salariales estimados.
- Evaluación contextualizada del encaje entre una oferta y un perfil profesional.

El uso de IA generativa en este contexto permite sustituir enfoques tradicionales basados en reglas rígidas o expresiones regulares por un sistema más flexible y adaptable a cambios en las plantillas de email.

2.3.3. Integración de sistemas

El proyecto integra de forma coherente múltiples servicios y APIs: Gmail como fuente de datos, OpenAI como motor de análisis semántico y Google Sheets como repositorio estructurado, todo ello orquestado mediante n8n. Esto aporta valor desde la perspectiva de la ingeniería de software, al mostrar un caso completo de:

- Diseño de arquitectura basada en servicios cloud.
- Integración de APIs heterogéneas.
- Gestión de flujos de datos de extremo a extremo.

2.3.4. Valor académico

Académicamente, el trabajo se sitúa en la intersección de varios campos relevantes:

- Inteligencia Artificial y Procesamiento de Lenguaje Natural.
- Automatización de procesos y orquestación de workflows.
- Ingeniería de datos y diseño de sistemas integrados.

La memoria documenta de forma detallada las decisiones de diseño, la metodología empleada, las métricas de evaluación y los resultados obtenidos, proporcionando un caso de estudio completo y replicable.

2.3.5. Escalabilidad y extensibilidad

La solución desarrollada es, además, escalable y extensible. Aunque en este trabajo se centra en un único perfil de candidato y en un conjunto concreto de plataformas, la arquitectura permite:

- Añadir nuevas fuentes de ofertas (otras plataformas o canales).
- Adaptar el perfil objetivo del candidato.
- Incorporar nuevas salidas, como dashboards, sistemas de alertas o análisis de tendencias del mercado laboral.
- Evolucionar hacia un sistema multiusuario si se rediseña la capa de almacenamiento.

2.4. Estructura de la memoria

La memoria se organiza en los capítulos siguientes:

- **Capítulo 3 – Objetivos**: define el objetivo general y los objetivos específicos del proyecto, así como el alcance, las limitaciones iniciales y los criterios de éxito.

- **Capítulo 4 – Estado del arte / marco teórico**: revisa las tecnologías y conceptos clave relacionados con la automatización de procesos, los modelos de lenguaje, las plataformas de automatización como n8n, los sistemas de búsqueda de empleo y las técnicas de extracción de información y *scoring*.
- **Capítulo 5 – Metodología**: describe la metodología de desarrollo adoptada, las fases del proyecto, las herramientas empleadas, las métricas definidas y la estrategia de pruebas.
- **Capítulo 6 – Diseño e implementación**: detalla la arquitectura del sistema, el diseño del workflow en n8n, los módulos principales (preprocesado, análisis con IA, *scoring*, deduplicación, almacenamiento) y las decisiones de implementación.
- **Capítulo 7 – Resultados y evaluación**: presenta los datos de prueba utilizados, las métricas obtenidas, el análisis de resultados, casos de uso representativos y las limitaciones empíricas observadas.
- **Capítulo 8 – Discusión**: interpreta los resultados, analiza los logros y desafíos del proyecto, compara la solución con alternativas existentes y discute el impacto y la aplicabilidad de la propuesta.
- **Capítulo 9 – Conclusiones**: sintetiza las conclusiones principales, el grado de cumplimiento de objetivos, las contribuciones del trabajo, los aprendizajes obtenidos y las principales limitaciones.
- **Capítulo 10 – Trabajo futuro**: propone líneas de mejora y extensiones posibles del sistema desarrollado.
- **Capítulo 11 – Referencias bibliográficas**: recoge las fuentes académicas y técnicas utilizadas.
- **Capítulo 12 – Anexos**: incluye documentación técnica adicional, ejemplos detallados y material complementario relevante para la replicación o extensión del sistema.

2.5. Uso de herramientas de IA en la redacción de la memoria

Es importante destacar que la redacción de esta memoria ha contado con el apoyo de herramientas de Inteligencia Artificial generativa. A lo largo del proyecto se han ido recopilando notas, documentación técnica, evidencias de laboratorio y descripciones de los distintos experimentos y versiones del sistema. Sobre esta base, se han utilizado modelos de lenguaje para ayudar a estructurar el contenido, proponer borradores de texto y unificar el estilo de redacción.

El autor ha proporcionado toda la información técnica, ha validado el contenido de cada capítulo y ha revisado manualmente la versión final, corrigiendo, matizando o ampliando las secciones necesarias. En consecuencia, tanto las decisiones de diseño como la interpretación de los resultados y las conclusiones son responsabilidad exclusiva del autor.

Este enfoque ilustra, una vez más, el potencial de la IA como herramienta de trabajo cuando se utiliza por parte de un profesional. Combinada con experiencia y criterio humano, permite aumentar de forma significativa la productividad y reducir los tiempos de elaboración de documentación técnica a niveles difíciles de imaginar en etapas anteriores de la historia, incluso en comparación con hitos como la producción en cadena de Henry Ford o la invención de la imprenta.

---

3. OBJETIVOS

3.1. Objetivo general

Desarrollar un sistema automatizado de gestión y análisis de correos de ofertas de empleo que sea capaz de:

- Procesar de forma continua los emails recibidos desde diferentes portales.
- Extraer información relevante de cada oferta mediante IA generativa.
- Evaluar el encaje de cada oferta con un perfil profesional concreto.
- Almacenar la información de forma estructurada para facilitar la toma de decisiones en la búsqueda de empleo.

3.2. Objetivos específicos

3.2.1. Objetivos funcionales

OF1. Procesamiento automático de emails

- Procesar automáticamente emails procedentes de múltiples fuentes de empleo.
- Detectar y filtrar únicamente los correos relacionados con ofertas de trabajo.
- Preprocesar y limpiar el contenido (HTML, textos redundantes, firmas, etc.).

OF2. Extracción de información estructurada

Extraer de cada oferta, siempre que sea posible, los siguientes campos:

- Estado de la candidatura.
- Empresa.
- Título del puesto.
- Ubicación.
- Tipo de contrato.
- Nivel de experiencia requerido.
- *Skills* y tecnologías mencionadas.
- Salario o rango salarial estimado a partir del contexto disponible.
- Descripción resumida de la oferta.
- Enlace principal a la oferta de trabajo.

OF3. Sistema de *scoring* personalizado

- Calcular un *score* de relevancia (escala 1–5) para cada oferta.
- Desglosar el *matching* en varios ejes (0–100):
  - *Match* de *skills*.
  - *Match* de experiencia.
  - *Match* de ubicación.
  - *Match* de salario.
- Generar recomendaciones automáticas que permitan priorizar rápidamente las ofertas más interesantes.

OF4. Almacenamiento y gestión de datos

- Almacenar la información de forma estructurada en Google Sheets.
- Implementar un sistema de deduplicación que evite registros duplicados para la misma oferta.
- Actualizar automáticamente ofertas existentes cuando cambie su estado.
- Mantener trazabilidad básica (fechas de creación y última actualización).

OF5. Información complementaria

- Estimar rangos salariales cuando no aparezcan explícitamente en el correo.
- Extraer y normalizar enlaces a la oferta siempre que sea posible.

3.2.2. Objetivos técnicos

OT1. Integración de APIs de IA

- Integrar la API de OpenAI para el procesamiento de lenguaje natural.
- Diseñar *prompts* efectivos que fuercen respuestas estructuradas (JSON) con todos los campos requeridos.
- Implementar validaciones y manejo de errores sobre las respuestas del modelo.

OT2. Automatización con n8n

- Diseñar y desplegar un *workflow* automatizado en n8n.
- Integrar los servicios externos necesarios (Gmail, OpenAI, Google Sheets).
- Implementar lógica de control de flujo, reintentos y manejo de casos especiales mediante nodos de código (JavaScript).

3.2.3. Objetivos de evaluación

OE1. Precisión en extracción de información

- Evaluar la precisión en la extracción de los campos clave.
- Medir la tasa de acierto en la clasificación del estado de cada candidatura.
- Validar la calidad de la información extraída frente a una revisión manual.

OE2. Efectividad del *scoring*

- Evaluar la coherencia del sistema de *scoring* con las preferencias reales del candidato.
- Verificar que las ofertas mejor puntuadas sean efectivamente las más interesantes.
- Analizar la distribución de *scores* y de *matches* para detectar posibles sesgos.

OE3. Rendimiento del sistema

- Medir el tiempo de procesamiento por email.
- Calcular el coste por correo procesado.
- Evaluar la tasa de errores y casos fallidos.

3.3. Alcance del proyecto

3.3.1. Alcance funcional

El sistema desarrollado incluye:

- Procesamiento de emails de ofertas de trabajo.
- Extracción de información estructurada mediante IA generativa.
- Sistema de *scoring* personalizado basado en el perfil del usuario.
- Almacenamiento de datos en Google Sheets.
- Deduplicación y actualización de ofertas existentes.
- Estimación básica de rangos salariales cuando no estén presentes.

El sistema no incluye:

- Sistema de alertas o notificaciones en tiempo real.
- Dashboard propio de visualización avanzada de datos.
- Análisis de tendencias del mercado laboral.
- Soporte multiusuario o multiperfil.
- Aplicación móvil o interfaz web dedicada (más allá de Google Sheets).
- Automatización del envío de candidaturas ni la inscripción automática del usuario en ofertas; la decisión de aplicar y el envío de solicitudes siguen siendo responsabilidad exclusiva del candidato.

3.3.2. Alcance técnico

El sistema:

- Utiliza:
  - n8n como plataforma de automatización.
  - Railway como proveedor de alojamiento para n8n.
  - OpenAI API (modelo GPT-4o-mini) para procesamiento de lenguaje natural.
  - Google Sheets API para almacenamiento de datos.
  - Gmail API para acceso a los correos.
  - JavaScript/Node.js en nodos *Code* para lógica de negocio y validaciones.
- No incluye:
  - Desarrollo de modelos de IA propios.
  - Infraestructura física o virtual autogestionada (se apoya en servicios *cloud*).
  - Uso de una base de datos relacional tradicional.
  - Un sistema complejo de autenticación y autorización.
  - Integración con APIs de terceros específicas para salarios (las estimaciones se basan en el conocimiento del modelo).

3.3.3. Alcance temporal

El proyecto se desarrolló en un horizonte temporal acotado que abarcó:

- Fase de análisis y diseño.
- Fase de implementación.
- Fase de pruebas y validación.
- Fase de evaluación, documentación y cierre.

3.4. Limitaciones

Desde la fase de definición se identificaron las siguientes limitaciones iniciales del proyecto.

3.4.1. Limitaciones técnicas

- **Dependencia de APIs externas**

  El sistema depende de la disponibilidad, cambios de versión y límites de uso de las APIs de OpenAI y Google Sheets.

- **Calidad de extracción condicionada por el formato**

  La precisión de la extracción puede verse afectada por cambios en las plantillas de correo, HTML complejo o formatos no estándar.

- **Idiomas**

  El sistema está optimizado para correos en español e inglés; el rendimiento puede degradarse en otros idiomas o textos mixtos.

- **Información salarial**

  Las estimaciones de salario se basan en el contexto del correo y en el conocimiento del modelo, sin consultas en tiempo real a fuentes externas específicas.

3.4.2. Limitaciones funcionales

- **Perfil único**

  El sistema está diseñado para un único perfil de candidato; no soporta de forma nativa múltiples usuarios ni perfiles simultáneos.

- **Fuentes limitadas**

  Solo procesa correos de las plataformas configuradas en la bandeja de entrada. No realiza *scraping* web ni consultas directas a portales externos más allá del correo.

- **Análisis de valor añadido limitado**

  El sistema se centra en clasificación y priorización de ofertas; no incluye análisis de competencia, recomendaciones de mejora del perfil o análisis de mercado laboral.

- **Sin sistema de alertas**

  No se implementan notificaciones automáticas (por ejemplo, avisos en tiempo real para ofertas con alto *score*).

3.4.3. Limitaciones de evaluación

- **Datos de prueba limitados**

  La evaluación se realizó sobre un conjunto limitado de emails. Cada prueba de backtesting procesaba concretamente 50 correos, lo que suponía alrededor de 30 minutos de ejecución. Se decidió no aumentar el número de correos por lote y priorizar un mayor número de iteraciones entre mejoras.

- **Validación subjetiva**

  La valoración final de la relevancia de las ofertas incorpora un componente subjetivo ligado al perfil y preferencias del propio desarrollador.

- **Horizonte temporal reducido**

  No se ha evaluado el comportamiento del sistema a muy largo plazo ni con volúmenes masivos de datos.

3.5. Criterios de éxito

El proyecto se considera exitoso si se cumplen los siguientes criterios:

- **Funcionalidad básica**

  El sistema procesa correctamente al menos el 80 % de los emails de ofertas de trabajo relevantes que recibe.

- **Precisión en la extracción**

  Al menos el 85 % de los campos clave (empresa, puesto, ubicación) se extraen de forma correcta.

- **Sistema de *scoring* operativo**

  El sistema genera *scores* de relevancia coherentes con el perfil del usuario y útiles para priorizar ofertas.

- **Almacenamiento correcto**

Los datos se almacenan de forma consistente en Google Sheets, sin duplicados no deseados y con actualización correcta del estado de las ofertas.

- **Coste razonable**

  El coste por email procesado se mantiene por debajo de 0,001 USD.

- **Rendimiento adecuado**

  El tiempo de procesamiento por email es inferior a 30 segundos en condiciones normales de carga.

- **Documentación completa**

  El sistema está documentado de forma suficiente como para ser reproducible, mantenible y extensible.

---

4. ESTADO DEL ARTE / MARCO TEÓRICO

4.1. Automatización de procesos

4.1.1. Plataformas de automatización de workflows

La automatización de procesos se ha popularizado gracias a plataformas visuales que permiten orquestar servicios cloud sin programar toda la lógica desde cero. Entre las más relevantes:

- **n8n**: plataforma *open-source*, auto-hospedable, con nodos para cientos de servicios (HTTP, Gmail, Google Sheets, bases de datos, etc.) y soporte para nodos de código (JavaScript).
- **Zapier** y **Make (Integromat)**: soluciones SaaS comerciales orientadas a usuarios finales y pymes, con gran catálogo de integraciones y planes de pago según volumen de tareas.
- **Microsoft Power Automate**: solución empresarial integrada con el ecosistema Microsoft 365, especialmente usada en entornos corporativos.

Estas herramientas comparten características clave: interfaz visual, disparadores (triggers), ejecución basada en eventos y nodos para transformar datos. Facilitan la creación de pipelines complejos sin gestionar infraestructura propia.

4.1.2. Automatización en la búsqueda de empleo

En el contexto de la búsqueda de empleo, la automatización se aplica principalmente a:

- **Monitorización de ofertas** en múltiples plataformas.
- **Recepción y filtrado** de notificaciones por email.
- **Extracción** de datos relevantes (empresa, puesto, salario, etc.).
- **Clasificación y priorización** de oportunidades.
- **Seguimiento del estado** de candidaturas.

Sin embargo, la mayoría de soluciones existentes se limitan a filtros básicos en el correo o a hojas de cálculo mantenidas manualmente; rara vez combinan automatización de workflows con IA avanzada para análisis y scoring personalizado de ofertas.

4.2. Inteligencia Artificial y Procesamiento de Lenguaje Natural

4.2.1. Modelos de lenguaje generativo

Los modelos de lenguaje de gran tamaño (LLM) han supuesto un salto cualitativo en Procesamiento de Lenguaje Natural (PLN). Familias como **GPT-3**, **GPT-3.5**, **GPT-4** y variantes optimizadas como **GPT-4o / GPT-4o-mini** permiten:

- Comprender texto no estructurado.
- Generar respuestas coherentes y contextuales.

- Realizar tareas de clasificación, resumen, extracción de entidades y transformación de formato sin entrenamiento específico.

Para este proyecto son especialmente relevantes las capacidades de:

- **Extracción estructurada** → convertir texto libre en JSON con campos concretos.
- **Inferencia contextual** → deducir tipo de contrato, nivel o rango salarial cuando no se expresan de forma literal.
- **Multilingüismo** → trabajar con correos en español e inglés.

4.2.2. Prompt engineering

El *prompt engineering* consiste en diseñar instrucciones claras para guiar el comportamiento del modelo. Algunas técnicas clave utilizadas en el proyecto son:

- **Instrucciones explícitas** sobre qué campos extraer, formatos, rangos válidos y reglas de negocio.
- **Formato estructurado** → exigir salida en `json_object` con claves y tipos conocidos.
- **Ejemplos (few-shot)** cuando es necesario estabilizar la salida en casos ambiguos.
- **Restricciones lógicas** (por ejemplo, relación coherente entre *matches* y score).

Un diseño cuidadoso del prompt es esencial para obtener respuestas consistentes, reducir errores de formato y minimizar la necesidad de postprocesado.

4.2.3. APIs de IA y servicios cloud

Las APIs de IA, como **OpenAI API**, exponen modelos de lenguaje mediante endpoints HTTP, habitualmente bajo un modelo **pay-per-use** basado en tokens. Para este tipo de proyectos destacan:

- Soporte para salida estructurada (JSON).
- Posibilidad de elegir modelos más ligeros y económicos (como GPT-4o-mini) con buen rendimiento en tareas de extracción.
- Escalabilidad horizontal sin gestionar hardware propio.

Esto permite incorporar IA avanzada en un workflow de automatización sin entrenar ni desplegar modelos locales.

4.3. Plataformas de automatización: n8n

4.3.1. Características principales

n8n es una plataforma de automatización *open-source* que ofrece:

- Editor visual de workflows basados en nodos.
- Integraciones nativas con múltiples servicios (Gmail, Google Sheets, HTTP, Webhooks, etc.).
- Nodos de código (JavaScript/TypeScript) para lógica personalizada.
- Posibilidad de despliegue *self-hosted* (Docker, servidores propios, PaaS).

Esta combinación la hace especialmente adecuada para proyectos donde se requiere flexibilidad, control y capacidad de extender la lógica con código.

4.3.2. Despliegue en Railway

Para este trabajo, n8n se despliega en **Railway**, una plataforma PaaS que facilita:

- Despliegue rápido de servicios Node.js.
- Gestión sencilla de variables de entorno (tokens de APIs, credenciales, etc.).
- Escalado básico sin configuración compleja.

Este enfoque evita gestionar máquinas virtuales o contenedores manualmente, manteniendo al mismo tiempo el control sobre la instancia de n8n.

4.3.3. Casos de uso comunes

Entre los casos de uso típicos de n8n destacan:

- Integración de sistemas (sincronizar datos entre aplicaciones).
- Automatización de marketing (envío de emails, gestión de leads).
- Procesos internos (onboarding, soporte, gestión de incidencias).
- Pipelines de datos (enriquecimiento, transformación, almacenamiento).

El proyecto se apoya en este mismo patrón:

Captura de emails → Procesamiento → IA → Escritura en un repositorio estructurado.

4.3.4. Ventajas para este proyecto

Frente a alternativas SaaS cerradas, n8n aporta:

- **Código abierto y auto-hospedable**, sin depender de un proveedor único.
- **Conectores nativos** para Gmail, Google Sheets y HTTP (OpenAI).
- **Costo reducido** para uso personal.
- Posibilidad de ajustar en detalle el comportamiento mediante nodos de código.

4.4. Sistemas de búsqueda de empleo

4.4.1. Plataformas de empleo principales

El ecosistema actual de búsqueda de empleo se apoya en plataformas como:

- **LinkedIn**: red profesional global con notificaciones por email de nuevas ofertas, cambios de estado y mensajes de reclutadores.
- **InfoJobs**: plataforma líder en España, con alertas configurables y correos periódicos con ofertas relevantes.
- **Tecnoempleo**: orientada al sector tecnológico, especialmente en España.
- **Indeed**: agregador global que centraliza ofertas de múltiples fuentes.

Todas estas plataformas generan un volumen considerable de **emails HTML** con ofertas, recordatorios y actualizaciones de candidaturas.

4.4.2. Desafíos en la búsqueda de empleo

Desde la perspectiva del candidato, los principales problemas son:

- **Sobrecarga de información**: decenas de correos diarios desde diferentes portales.
- **Fragmentación**: cada plataforma usa su propio formato, terminología y estructura de email.
- **Trabajo manual**: lectura, filtrado y registro de ofertas en hojas de cálculo o notas personales.
- **Dificultad para priorizar**: no existe un score unificado que ordene las ofertas por relevancia real para el perfil del candidato.

4.4.3. Soluciones existentes y limitaciones

Algunas soluciones habituales son:

- Filtros/etiquetas del propio cliente de correo.
- Funciones básicas dentro de las plataformas (listas de guardados, alertas, filtros por palabra clave).
- Hojas de cálculo manuales o herramientas genéricas de tracking de candidaturas.

Sin embargo, estas soluciones:

- No realizan extracción estructurada automática desde el email.
- No aplican scoring personalizado multi-criterio (skills, experiencia, ubicación, salario).
- No integran de forma continua la bandeja de entrada con un repositorio unificado de ofertas.

Aquí es donde se sitúa la contribución principal de este trabajo.

4.5. Extracción de información estructurada

4.5.1. Técnicas tradicionales

La extracción de información desde texto no estructurado ha sido tradicionalmente abordada con:

- **Expresiones regulares y reglas** → útiles para patrones fijos, pero frágiles ante cambios de formato.
- **Sistemas de NER y pipelines clásicos de PLN** (spaCy, NLTK, etc.) → permiten detectar entidades (empresas, ubicaciones, fechas), pero requieren modelos o reglas adaptadas al dominio.
- **Modelos supervisados específicos** → entrenados para extraer campos concretos, pero demandan datasets etiquetados y esfuerzo de mantenimiento.

En entornos con **HTML variado, múltiples idiomas y plantillas cambiantes**, estas aproximaciones tienden a degradarse rápidamente si no se recalibran de forma continua.

4.5.2. Enfoque con IA generativa

Los LLM permiten un enfoque diferente:

- En lugar de definir reglas rígidas, se describen los campos deseados en un **prompt estructurado**.
- El modelo interpreta el texto, infiere significados y devuelve directamente un **JSON** con los valores solicitados.
- Se mantiene robustez frente a cambios de formato, siempre que el contenido semántico sea similar.

Ventajas clave para este proyecto:

- Soporta correos en español e inglés sin modelos separados.
- Tolera plantillas heterogéneas de distintas plataformas.
- Permite inferir información no explícita (tipo de contrato, rango salarial aproximado) cuando el contexto lo sugiere.

4.6. Sistemas de scoring y recomendación

4.6.1. Sistemas de recomendación

Los sistemas de recomendación se utilizan ampliamente en comercio electrónico y plataformas de contenido (Amazon, Netflix, Spotify, etc.). Técnicas típicas:

- **Filtrado basado en contenido** → recomendaciones según características del item (skills requeridas, ubicación, salario, etc.).
- **Filtrado colaborativo** → recomendaciones según comportamiento de usuarios similares.
- **Sistemas híbridos** → combinan ambos enfoques.

En este proyecto se adopta un enfoque principalmente **basado en contenido**, ya que el foco está en el encaje entre características de la oferta y perfil del candidato.

4.6.2. Scoring personalizado para ofertas de trabajo

El scoring personalizado asigna una puntuación numérica a cada oferta, teniendo en cuenta:

- **Match de skills**.
- **Match de experiencia**.
- **Match de ubicación**.
- **Match de salario**.
- Otros factores contextuales (tipo de contrato, sector, modalidad de trabajo).

La novedad aquí radica en que tanto la **extracción de atributos** como el **cálculo del scoring** se delegan al modelo de IA a partir de un prompt bien diseñado, en lugar de usar fórmulas rígidas predefinidas.

4.7. Calidad de datos, normalización y deduplicación

La utilidad de cualquier sistema de análisis depende de la calidad y consistencia de sus datos. En este contexto son relevantes tres aspectos:

- **Normalización** → unificación de formatos (por ejemplo, nombres de ciudades, tipos de contrato, niveles de experiencia) para poder filtrar y comparar.
- **Identificadores lógicos** → uso de combinaciones como `empresa + título_puesto` para identificar ofertas de forma consistente.
- **Deduplicación** → detección de registros repetidos cuando una misma oferta se notifica varias veces o llega por múltiples canales.

En este proyecto se emplea principalmente **deduplicación exacta** basada en la combinación `empresa + puesto`, suficiente para el escenario abordado y sencilla de implementar en el workflow de n8n.

4.8. Síntesis y posicionamiento de la propuesta

El estado del arte muestra que:

- Existen plataformas maduras de automatización de workflows (n8n, Zapier, Make).
- Los modelos de lenguaje generativo ofrecen capacidades potentes de extracción estructurada e inferencia contextual.
- Las APIs de servicios cloud permiten integrar estas capacidades en sistemas productivos sin gestionar infraestructura compleja.
- Los conceptos de recomendación y scoring personalizado están bien estudiados en otros dominios.

Sin embargo, la **combinación práctica** de estas tecnologías para construir un sistema que:

1. Lea automáticamente correos de ofertas de empleo.
2. Extraiga información estructurada con IA generativa.
3. Aplique *scoring* personalizado según el perfil del candidato.
4. Y almacene todo en un repositorio unificado y explotable.

no está ampliamente documentada en la literatura académica ni en soluciones comerciales genéricas. Este trabajo se sitúa precisamente en ese espacio, integrando automatización, IA generativa y scoring personalizado en un caso de uso real de búsqueda de empleo.

5. METODOLOGÍA

5.1. Metodología de desarrollo

El proyecto se ha desarrollado con un enfoque **ágil e iterativo**, adecuado a un sistema pequeño pero muy conectado con servicios cloud. La idea central ha sido construir rápido algo funcional, probarlo con datos reales y refinarlo de manera continua.

5.1.1. Enfoque ágil

Las decisiones clave de la metodología han sido:

- **Iteraciones cortas**: cada ciclo añadía o mejoraba una funcionalidad claramente delimitada (preprocesado, extracción, scoring, deduplicación, etc.).
- **Feedback continuo**: tras cada cambio se ejecutaban pruebas sobre correos reales para detectar errores y ajustar prompts o lógica.
- **Adaptabilidad**: requisitos y detalles de diseño se ajustaron en función de las limitaciones técnicas encontradas (formato de emails, límites de tokens, coste de API).
- **Prioridad a valor real**: primero se implementaron las partes que ahorraban tiempo de verdad (captura, extracción básica y guardado en Sheets) y después se añadieron mejoras.

5.1.2. Prototipado rápido

El uso de **n8n** y APIs ya existentes permitió crear prototipos funcionales en muy poco tiempo:

- El workflow se podía modificar desde la interfaz de n8n sin tocar infraestructura.
- La integración con OpenAI, Gmail y Google Sheets se limitó a configurar credenciales y nodos.
- Los cambios de prompt, lógica de deduplicación o estructura de salida se probaban en minutos sobre correos reales.

5.1.3. Desarrollo dirigido por necesidades reales

El desarrollador es también el usuario final del sistema, lo que condicionó positivamente la metodología:

- **Validación inmediata**: cada cambio se medía por su impacto en la búsqueda de empleo real.
- **Priorización pragmática**: solo se mantenían las funcionalidades que aportaban valor directo (por ejemplo, scoring útil para priorizar).
- **Mejora continua**: el sistema se fue ajustando a medida que aparecían nuevos tipos de correos, estados de candidatura o necesidades de filtrado.

5.2. Fases del proyecto

Aunque el proceso ha sido iterativo, se pueden distinguir seis fases principales.

5.2.1. Fase 1: Análisis y diseño

- **Objetivos**: entender el problema, definir requisitos y proponer una arquitectura mínima viable.
- **Actividades**:
  - Análisis de correos de ofertas reales (estructura, campos, ruido).
  - Comparación de plataformas de automatización (n8n, Zapier, Make).
  - Evaluación de modelos de IA (GPT-3.5, GPT-4, GPT-4o-mini).
  - Definición del flujo general (captura → preprocesado → IA → scoring → Sheets).
  - Diseño de la estructura de la hoja de cálculo (23 columnas).
- **Resultado**: arquitectura definida, tecnologías seleccionadas (n8n + GPT-4o-mini + Google Sheets) y diseño del workflow principal.

5.2.2. Fase 2: Implementación básica

- **Objetivos**: levantar la infraestructura mínima y conectar Gmail con Google Sheets.
- **Actividades**:
  - Despliegue de n8n en Railway (self-hosted).
  - Configuración de credenciales (Gmail, Sheets, OpenAI).
  - Implementación del trigger de Gmail con filtros por remitente y asunto.
  - Primer nodo de preprocesado de email.
  - Creación de la hoja de cálculo y guardado básico de datos.
- **Resultado**: sistema capaz de leer correos filtrados y almacenarlos en Sheets con información simple.

5.2.3. Fase 3: Integración con IA

- **Objetivos**: incorporar OpenAI para extraer información estructurada.
- **Actividades**:
  - Configuración de OpenAI en n8n.
  - Diseño y refinamiento de prompts para devolver JSON estructurado.
  - Implementación del nodo de llamada a OpenAI y del nodo de parseo de la respuesta.
  - Introducción de validaciones básicas y manejo de errores.
  - Pruebas con correos reales de LinkedIn, InfoJobs, etc.
- **Resultado**: extracción automática de empresa, puesto, ubicación, estado, skills, etc., con un flujo de errores controlado.

5.2.4. Fase 4: Sistema de scoring

- **Objetivos**: añadir evaluación automática de relevancia y *matching* con el perfil.
- **Actividades**:
  - Definición del perfil objetivo del candidato.
  - Diseño del prompt para generar score global (1–5) y *matches* parciales (0–100).
  - Integración de scoring en la misma llamada a OpenAI que la extracción (reduciendo costes).
  - Ajuste de criterios mediante pruebas reales.

- **Resultado**: scoring funcional con scores y *matches* coherentes y etiquetas de prioridad (alta/media/baja).

5.2.5. Fase 5: Optimización y mejoras

- **Objetivos**: mejorar precisión, rendimiento y coste.
- **Actividades**:
  - Optimización de prompts para reducir tokens y ambigüedad.
  - Implementación de deduplicación basada en empresa + puesto.
  - Mejora de extracción de enlaces desde HTML/texto.
  - Refinamiento de inferencias (tipo de contrato, nivel de experiencia).
  - Eliminación de campos de depuración y datos innecesarios.
- **Resultado**: sistema más preciso, con deduplicación fiable y costes reducidos por email procesado.

5.2.6. Fase 6: Validación y uso real

- **Objetivos**: validar el sistema en un escenario real de búsqueda de empleo.
- **Actividades**:
  - Activación del workflow en producción.
  - Procesamiento de correos reales durante varias semanas.
  - Uso del sistema para priorizar y seguir procesos de selección.
  - Recopilación de dataset final para análisis de resultados.
- **Resultado**: sistema validado en uso real, que contribuye directamente a la obtención de un nuevo puesto de trabajo (27/10/2025).

5.3. Herramientas y tecnologías utilizadas

5.3.1. Plataformas y servicios

Tecnología	Rol principal	Motivo de selección
n8n	Orquestación del workflow y lógica de negocio	Open-source, flexible, integraciones nativas, sin licencia de pago.
Railway	Hosting cloud de n8n (self-hosted)	Despliegue sencillo de servicios Node.js.
OpenAI API (GPT-4o-mini)	Extracción de información y scoring	Buen equilibrio coste/rendimiento en tareas de NLP.
Gmail API	Fuente de datos (correos de ofertas)	Integración directa con n8n y acceso a correos históricos.
Google Sheets API	Almacenamiento estructurado	Hoja de cálculo accesible, colaborativa y con API robusta.

5.3.2. Lenguajes y formatos

- **JavaScript/Node.js** en nodos *Code* de n8n para preprocesado, transformación, deduplicación y validación.
- **JSON** como formato de intercambio entre nodos y como salida forzada de la API de OpenAI.
- **HTML/texto plano** como entrada procedente de los correos, que se limpia y normaliza antes de enviar al modelo.

5.3.3. Técnicas y metodologías

- **Prompt engineering**:
  - Instrucciones estructuradas y concisas.
  - Respuesta obligatoria en JSON con todos los campos esperados.
  - Uso de ejemplos (*few-shot*) cuando fue necesario estabilizar salidas.
- **Deduplicación**: comparación exacta por combinación `empresa + título_puesto` en código JavaScript.
- **Optimización de tokens**: truncado del cuerpo del correo a ~4.000 caracteres relevantes.

5.4. Métricas de evaluación

Las métricas se diseñaron para evaluar tres ejes: **precisión**, **rendimiento** y **utilidad**.

5.4.1. Métricas de precisión

- **Precisión en extracción de información**
  - Campos: empresa, puesto, ubicación, estado, skills, salario.
  - Método: comparación manual con el contenido original.
  - Objetivo: ≥ 85 % de acierto en campos clave.
- **Precisión en clasificación de estados**
  - Estados: `nueva_oferta`, `en_proceso`, `rechazado`, `aceptado`, `entrevista_programada`.
  - Objetivo: ≥ 90 %.
- **Precisión en identificación de empresa**
  - Objetivo: ≥ 90 %.

5.4.2. Métricas de rendimiento

- **Tiempo de procesamiento por email**
  - Incluye preprocesado, llamada a OpenAI y escritura en Sheets.
  - Objetivo: < 30 s por correo.
- **Coste por email procesado**
  - Basado en tokens consumidos en OpenAI.
  - Objetivo: < 0,001 USD por correo.
- **Tasa de éxito de procesamiento**
  - % de correos procesados sin error y guardados correctamente.
  - Objetivo: > 95 %.

5.4.3. Métricas de utilidad

- **Efectividad del scoring**
  - Se evalúa si las ofertas con score 4–5 son realmente las más interesantes en la práctica.
  - Indicador de éxito: uso satisfactorio del sistema durante la búsqueda real.
- **Cobertura de información**
  - % de ofertas con campos inferidos no vacíos (tipo de contrato, nivel, salario, enlaces).
  - Objetivo: > 70 % de cobertura en los campos principales.

5.5. Estrategia de pruebas

La estrategia de pruebas combinó pruebas técnicas clásicas con validación práctica sobre datos reales.

5.5.1. Pruebas unitarias

- Validación de limpieza de HTML y extracción de texto.
- Verificación de la lógica de deduplicación.
- Comprobación de formateo de datos antes de escribir en Google Sheets.

5.5.2. Pruebas de integración

- Ejecución del workflow completo de forma manual, con correos pasados, y automática, con correos presentes.
- Validación de respuestas de OpenAI y de la escritura en Sheets.
- Pruebas de comportamiento ante errores de API (timeouts, formatos inválidos).

5.5.3. Pruebas con datos reales

- **Backtesting**: procesamiento de los últimos 50 correos de ofertas para comprobar robustez.
- **Validación en producción**: uso del sistema sobre correos recibidos durante la búsqueda activa.
- **Uso continuo**: monitorización del comportamiento durante varias semanas.

5.5.4. Validación de resultados

- Revisión manual de muestras para comprobar exactitud de campos.
- Análisis estadístico de distribución de scores, *matches* y estados.
- Validación práctica: comprobar si el sistema ayuda efectivamente a gestionar y priorizar ofertas.

5.6. Gestión de calidad

La calidad del sistema se abordó desde tres ángulos: datos, errores y mejora continua.

5.6.1. Control de calidad de datos

- Comprobación de formatos y tipos (por ejemplo, rangos de scores 1–5, *matches* 0–100).
- Normalización de valores y limpieza de campos de depuración.

5.6.2. Manejo de errores

- Validación de respuestas de APIs antes de procesarlas.
- Reintentos ante fallos temporales.
- Registro de errores en logs de n8n para análisis posterior.

5.6.3. Optimización continua

- Ajuste recurrente de prompts para mejorar precisión y reducir tokens.
- Revisión de errores detectados y corrección de casos límite.
- Ajustes en deduplicación y extracción según nuevos patrones de correo.

5.7. Consideraciones éticas y de privacidad

Las consideraciones éticas se centraron en el tratamiento de datos personales y el uso responsable de IA.

5.7.1. Privacidad de datos

- Los correos procesados contienen información personal, por lo que el acceso a la hoja de Google Sheets se limita al usuario.
- Los datos no se comparten con terceros más allá de los proveedores estrictamente necesarios (Google, OpenAI).

5.7.2. Uso de IA

- El uso de IA es transparente: se utiliza únicamente para extraer y valorar información.
- La decisión final sobre qué ofertas priorizar o a cuáles aplicar recae siempre en el usuario.
- Se asume la existencia de sesgos en los modelos, por lo que los resultados se revisan críticamente.

5.7.3. Uso ético

- El sistema se emplea para un fin legítimo: gestión personal de la búsqueda de empleo.
- Se respetan los términos de servicio de todas las plataformas utilizadas.
- No se automatiza el envío de candidaturas; solo se centraliza y organiza la información recibida.

---

6. DISEÑO E IMPLEMENTACIÓN

6.1. Arquitectura del sistema

6.1.1. Visión general

El sistema se ha diseñado como un **pipeline de procesamiento de emails** desplegado sobre n8n. A partir de los correos recibidos en Gmail, el flujo ejecuta de forma secuencial las siguientes etapas: captura, preprocesado, análisis con IA, cálculo de *scoring*, deduplicación y persistencia final en Google Sheets.

Cada etapa recibe un objeto estructurado, lo transforma y lo entrega a la siguiente, de forma que al final del pipeline se obtiene un registro coherente y completo para cada oferta de trabajo.

6.1.2. Componentes principales

A nivel lógico, la arquitectura se organiza en los módulos siguientes:

- **Módulo de captura de emails**
  - Implementado con un **Gmail Trigger** (o un nodo manual para backtesting).
  - Filtra únicamente correos de interés (LinkedIn, InfoJobs, Tecnoempleo, Indeed, etc.).
- **Módulo de preprocesamiento**
  - Limpieza de HTML, normalización de texto y detección de la plataforma de origen.
  - Extracción y depuración de enlaces relevantes a la oferta.
- **Módulo de análisis con IA**
  - Llamada a OpenAI (GPT-4o-mini) con un prompt estructurado.
  - Extracción de campos clave (empresa, puesto, ubicación, salario, skills, etc.) en formato JSON.
- **Módulo de scoring**
  - Cálculo de *matches* (skills, experiencia, ubicación, salario) y de un score global 1–5.
  - Generación de una recomendación de prioridad (alta / media / baja).
- **Módulo de deduplicación y actualización**
  - Comparación de la oferta actual frente a las ya existentes en Google Sheets.
  - Decisión entre crear un nuevo registro o actualizar uno existente.
- **Módulo de almacenamiento**
  - Formateo final y escritura en una hoja de Google Sheets con 23 columnas.

6.1.3. Diagrama lógico de arquitectura

De forma simplificada, el flujo puede representarse como:

```
Gmail → Preprocesado → Perfil + Prompt → OpenAI → Postprocesado + Scoring → Deduplicación → Google Sheets
```

6.2. Diseño del workflow

El workflow completo se ha implementado en n8n como una cadena de nodos que encapsulan las etapas anteriores. Existen dos variantes: una versión **automática**, conectada a Gmail Trigger, y una versión de **backtesting**, que lee los últimos 50 correos mediante un nodo manual.

6.2.1. Flujo de procesamiento

De forma resumida, el flujo estándar es:

1. **Captura de emails (Gmail Trigger / Get Last 50 Emails)**
   - Aplica una *query* de Gmail que combina remitentes de plataformas de empleo y palabras clave en el asunto.
   - Trabaja en modo "formato completo" para disponer de HTML y texto plano.
2. **Procesamiento por lotes (Split In Batches)**
   - Procesa un email por iteración para aislar errores y simplificar la lógica de recuperación.
3. **Preprocesamiento del email (Code Node)**
   - Extrae HTML, texto plano y metadatos (from, subject, fecha, ID).
   - Detecta la fuente (LinkedIn, InfoJobs, etc.) a partir de remitente, asunto y contenido.
   - Limpia el HTML, normaliza espacios/caracteres y limita el cuerpo a ~4.000 caracteres para optimizar tokens.
   - Extrae enlaces y prioriza URLs de ofertas de trabajo, eliminando parámetros de *tracking*.
4. **Carga del perfil personal (Code Node)**
   - Carga desde una estructura fija (hardcoded) el perfil objetivo del candidato: tecnologías clave, años de experiencia, rango salarial deseado, ubicaciones preferidas, etc.
5. **Preparación del prompt (Code Node)**
   - Construye un objeto con el contexto de oferta (texto limpio, metadatos, enlaces) y el perfil personal.
   - Genera un prompt estructurado que será enviado directamente al modelo de OpenAI.
6. **Análisis con IA (OpenAI Node)**
   - Invoca GPT-4o-mini con *response_format* tipo `json_object`.
   - Devuelve un JSON con todos los campos extraídos y los valores de *match* y score.
7. **Postprocesado de la respuesta (Code Node)**
   - Valida que el JSON sea parseable y que los campos críticos estén presentes.
   - Normaliza valores ("desconocido", rangos salariales, listas de skills, etc.).
8. **Control de errores (IF + nodo de log)**
   - Si se detecta error de formato, timeout, *rate limit* o datos incompletos, registra la incidencia y continúa con el siguiente email.

9. **Preparación de datos para Google Sheets (Code Node)**
    - Mapea los campos devueltos por OpenAI al esquema de 23 columnas.
    - Genera un identificador único `offer_id` a partir de `empresa + titulo_puesto` normalizados.
    - Añade campos internos como `fecha_creacion` y `fecha_actualizacion`.
10. **Lectura de ofertas existentes (Google Sheets – Read)**
    - Recupera las filas actuales de la hoja para permitir la deduplicación.
    - Este paso se ejecuta en paralelo al procesamiento de cada email para reducir latencia.
11. **Verificación de duplicados (Code Node)**
    - Compara el `offer_id` actual con los existentes.
    - Marca si la oferta ya existe y, en ese caso, almacena el número de fila a actualizar.
12. **Decisión y escritura (IF + Update/Append Row)**
    - Si la oferta existe: se actualizan estado, salario, score, enlaces, fecha_actualizacion.
    - Si no existe: se añade una nueva fila con todos los campos, incluyendo fecha_creacion.

En la *Ilustración 1* se muestra el flujo completo implementado en n8n, donde se aprecia la secuencia de módulos descritos en este apartado.

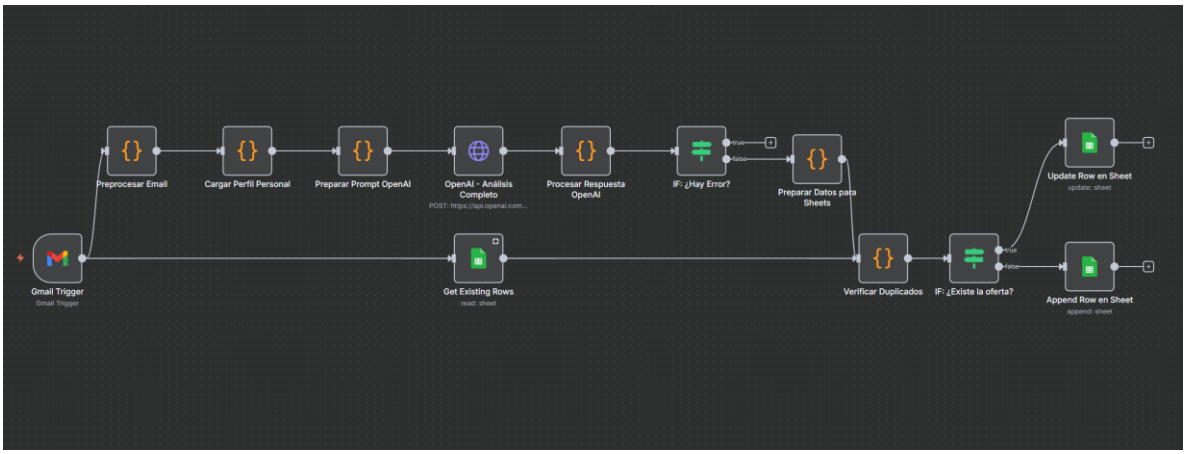

Ilustración 2. Workflow automático

La versión de backtesting, que puede verse en la *Ilustración 2,* reutiliza exactamente las mismas etapas, sustituyendo el trigger automático por un nodo manual que lanza el procesamiento sobre los últimos 50 emails filtrados, así como un bucle que analiza cada uno de los correos extraídos.

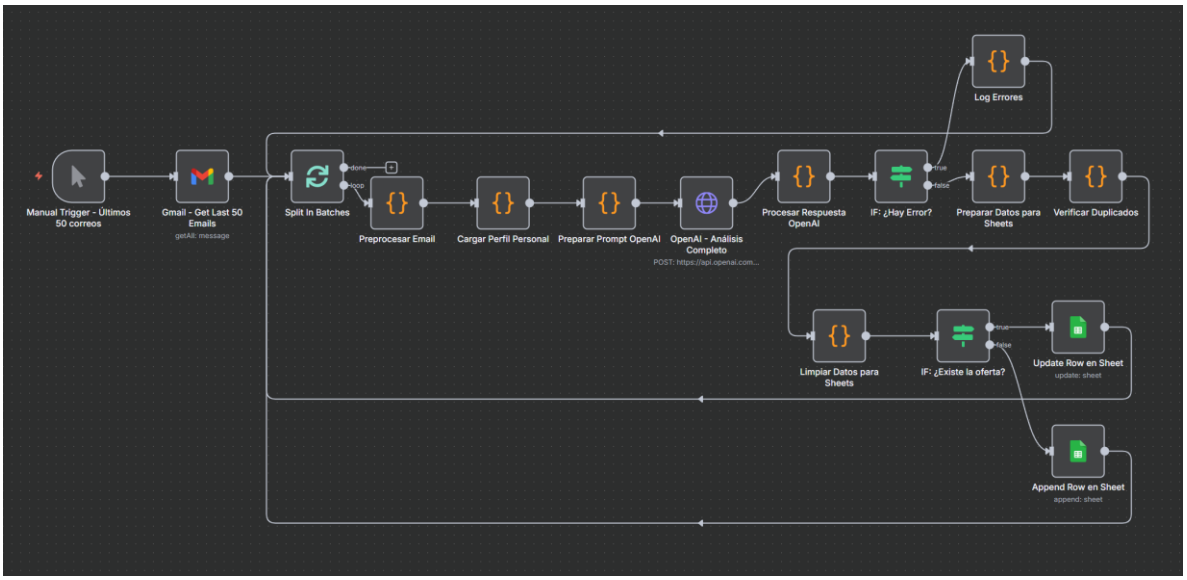

Ilustración 2. Workflow manual para entrenamiento

6.3. Extracción de información

6.3.1. Categorías de información

El sistema organiza la información de cada correo en varios bloques lógicos:

- **Información básica**
  - Estado de la aplicación (`nueva_oferta`, `en_proceso`, `rechazado`, `aceptado`, `entrevista_programada`, `desconocido`).
  - Empresa, título del puesto, ubicación, fuente, fecha y enlace principal a la oferta.
- **Información del puesto**
  - Tipo de contrato (indefinido, temporal, jornada completa, parcial, proyecto, etc.).
  - Nivel de experiencia (junior, mid, senior, lead, principal, desconocido).
  - Lista de skills tecnológicas específicas y una descripción resumida de la oferta.
- **Información salarial**
  - Rango salarial explícito si aparece en el correo.
  - Rango salarial estimado (mínimo y máximo numérico) cuando no se especifica, basado en mercado, nivel y ubicación.
- **Evaluación y scoring**
  - Score de relevancia 1–5.
  - *Match* de skills, experiencia, ubicación y salario (0–100).
  - Recomendación (alta / media / baja prioridad) y motivos principales del score.

- **Información sobre la empresa**
  - Evaluación de confianza (alta, media, baja, desconocido).
  - Descripción breve de la empresa cuando se dispone de contexto suficiente.

6.3.2. Diseño del prompt

El prompt se ha diseñado como un conjunto de instrucciones estructuradas que especifican:

- El rol del modelo (analista de ofertas de trabajo especializado).
- Las reglas para extraer cada campo y los formatos permitidos.
- Cómo inferir datos no explícitos (tipo de contrato, nivel, rango salarial).
- Cómo calcular los *matches* a partir del perfil personal.
- El formato exacto del JSON de salida (nombres de claves y tipos esperados).

Para reducir variabilidad y fallos, se han aplicado las siguientes técnicas:

- **Temperature baja (≈0,2)** para favorecer respuestas deterministas.
- **response_format = json_object** para minimizar errores de parseo.
- Instrucciones de consistencia del tipo:
  - Si todos los *matches* son 0 → score = 1.
  - Un score 5 requiere en general *matches* altos (≥ 80) en la mayoría de dimensiones.

6.4. Sistema de scoring

6.4.1. Modelo de scoring

El sistema de scoring combina cuatro dimensiones cuantitativas y genera un score 1–5 coherente con los *matches* obtenidos.

- **Match de skills (0–100)**
  - Basado en la proporción de skills del puesto que están presentes en el perfil del candidato.
- **Match de experiencia (0–100)**
  - Compara el nivel requerido (junior, mid, senior, etc.) con la experiencia real.
- **Match de ubicación (0–100)**
  - Valora si la ubicación coincide con las preferencias.
- **Match salarial (0–100)**
  - Mide el solapamiento entre el rango salarial esperado y el de la oferta (explícito o estimado).

A partir de estos valores, se calcula un promedio de los *matches* válidos y se mapea a un score discreto 1–5, que se traduce además en una recomendación:

- Score ≥ 4 → **alta_prioridad**
- Score = 3 → **media_prioridad**
- Score ≤ 2 → **baja_prioridad**

6.4.2. Integración en el workflow

El *scoring* se calcula **en la misma llamada a OpenAI** que la extracción de información, de forma que:

- Se reduce el número de llamadas a la API y el coste por email.
- El modelo puede razonar de forma conjunta sobre texto de la oferta y perfil personal.
- Se mantiene consistencia entre campos extraídos y valoración final.

6.5. Almacenamiento de datos

6.5.1. Esquema de Google Sheets

Los resultados se almacenan en una hoja de Google Sheets con **23 columnas**, que incluyen:

- Fecha
- Fuente
- Estado
- Empresa
- Título
- Ubicación
- Tipo de contrato
- Nivel
- Skills
- Salario (texto y valores numéricos mínimo/máximo)
- Score
- *Matches (8 columnas diferentes)*
- Recomendación
- Descripción
- Enlace
- Metadatos de control (`fecha_creacion`, `fecha_actualizacion`, confianza de empresa, etc.).

Este esquema proporciona un compromiso entre detalle suficiente para análisis posteriores (filtros, tablas dinámicas, dashboards) y simplicidad de mantenimiento.

6.5.2. Deduplicación y gestión de actualizaciones

La deduplicación se basa en un **ID lógico** definido como `empresa + titulo_puesto` normalizados. Para cada nueva oferta:

- Si el ID no existe en la hoja, se crea un nuevo registro y se fija `fecha_creacion`.
- Si el ID ya existe:
  - Se actualizan estado, salario, score, enlaces y `fecha_actualizacion`.
  - Se conserva `fecha_creacion`, lo que permite reconstruir la evolución del proceso.

Esta estrategia evita duplicados y refleja correctamente cambios de estado (por ejemplo, de `nueva_oferta` a `en_proceso` o `rechazado`).

6.6. Manejo de errores

6.6.1. Estrategias implementadas

El workflow incorpora varios niveles de control de errores:

- **Validación de la respuesta de OpenAI**
  - Verificación de que la salida es JSON válido.
  - Comprobación de presencia de campos críticos antes de continuar.
- **Manejo de errores de API**
  - Identificación de *rate limits* y timeouts.
  - Registro del error y salto al siguiente email para no bloquear el flujo.
- **Manejo de emails atípicos o mal formateados**
  - Valores por defecto para campos no recuperables.
  - Procesado parcial cuando es posible, en lugar de descartar el correo completo.

6.6.2. Logging y monitorización

n8n se utiliza como punto central de observabilidad:

- Registro de errores en los logs internos del workflow.
- Preservación de los datos originales del email cuando es útil para depuración.
- Timestamps en las distintas etapas (entrada, paso por OpenAI, escritura en Sheets) para facilitar el análisis de rendimiento.

6.7. Optimizaciones implementadas

Las optimizaciones se enfocan en coste, precisión y rendimiento.

6.7.1. Optimización de costes

- Uso de **GPT-4o-mini**, con buena relación coste/rendimiento.
- Unificación de extracción y scoring en una única llamada.

- Reducción del contexto enviado: limitación del cuerpo del email a los fragmentos más relevantes (≈4.000 caracteres).

6.7.2. Optimización de precisión

- Mejora en la extracción de enlaces mediante búsqueda tanto en HTML como en texto plano.
- Instrucciones específicas para inferir tipo de contrato y nivel de experiencia cuando no se indican explícitamente.
- Restricción de `skills` a tecnologías concretas, evitando términos genéricos ("equipo", "comunicación", etc.).
- Reglas para generar descripciones limpias, sin mezclar información del candidato con la oferta.

6.7.3. Optimización de rendimiento

- Procesamiento de un email por lote para simplificar recuperación de errores y garantizar estabilidad.
- Lectura de ofertas existentes en paralelo al procesamiento de cada correo.
- Validaciones tempranas para descartar rápidamente correos que no cumplan los requisitos mínimos.

6.8. Configuración y despliegue

6.8.1. Requisitos

La solución se despliega sobre una instancia **self-hosted de n8n** alojada en Railway, con las siguientes dependencias básicas:

- Cuenta de Railway con un servicio Node.js donde se instala n8n.
- Credenciales OAuth2 de Gmail y Google Sheets.
- Clave de API de OpenAI.
- Una hoja de Google Sheets creada previamente con el esquema de 23 columnas.

6.8.2. Pasos de configuración inicial

1. Desplegar n8n en Railway (o plataforma cloud equivalente).
2. Construir el workflow.
3. Configurar las credenciales de Gmail, Google Sheets y OpenAI en n8n.
4. Ajustar el nodo de **perfil personal** con las características del candidato.
5. Configurar el *Document ID* y rango de la hoja de cálculo.
6. Activar el workflow automático o lanzar manualmente la versión de backtesting según corresponda.

6.8.3. Mantenimiento operativo

El mantenimiento del sistema se centra en:

- Revisión periódica de los logs de n8n para detectar errores recurrentes.
- Ajuste de prompts cuando se identifiquen nuevos patrones de correo o campos mal inferidos.

- Actualización del perfil personal cuando cambian las preferencias (tecnologías objetivo, ciudades, rango salarial).

7. RESULTADOS Y EVALUACIÓN

7.1. Datos de prueba y validación

7.1.1. Estrategia de validación

La validación del sistema se realizó con un enfoque híbrido:

- **Backtesting con datos históricos**
  - Se creó una variante del workflow con **trigger manual** y un bucle sobre los **50 últimos correos** de ofertas laborales.
  - Los correos se filtraron por remitentes y asuntos de plataformas de empleo (LinkedIn, InfoJobs, etc.).
  - Cada correo se procesó por el pipeline completo y los resultados se volcaron en Google Sheets.
- **Validación con datos reales**
  - El workflow automático se mantuvo activo durante la **búsqueda real de empleo** del desarrollador.
  - Cada nueva oferta recibida se procesó automáticamente, generando score y registro en la hoja de cálculo.
- **Uso continuo en producción**
  - El sistema se utilizó de forma diaria para gestionar y priorizar ofertas.
  - Los resultados se revisaron manualmente de forma periódica, ajustando prompts y lógica cuando fue necesario.

7.1.2. Dataset final

Durante el periodo de validación, el sistema procesó **más de 50 ofertas de trabajo**, caracterizadas por:

- **Fuentes**
  - Principalmente: LinkedIn.
  - Complementarias: InfoJobs y otras plataformas puntuales (Tecnoempleo, Indeed, etc.).
- **Estados de candidatura**
  - `nueva_oferta`, `en_proceso`, `rechazado`, `aceptado`, `entrevista_programada`.
- **Tipos de puesto**
  - Desde posiciones junior hasta senior, en roles de:
    - Ciberseguridad.
    - IA y automatización.
    - QA / testing.
    - Desarrollo y puestos afines.
- **Ubicaciones**
  - Barcelona y alrededores, remoto en España, remoto Europa, remoto internacional, Luxemburgo, Lugano, Londres, Alemania, etc.

7.1.3. Validación práctica de uso real

El sistema se utilizó como herramienta principal durante la búsqueda activa de empleo del desarrollador, permitiendo:

- Identificar con rapidez las ofertas más relevantes mediante el **score** y los **matches**.
- Priorizar las candidaturas en función de la **alineación con el perfil**.
- Mantener un **registro único y ordenado** del estado de cada proceso.
- Seguir simultáneamente **varios procesos de selección** sin pérdida de información.

Como resultado directo de este uso, el desarrollador **consiguió un nuevo puesto de trabajo** con inicio el **27 de octubre de 2025**, lo que valida la utilidad del sistema en un contexto real, más allá de las pruebas de laboratorio.

7.2. Métricas de rendimiento

7.2.1. Precisión en extracción de información

Campos básicos

- Empresa: ~**95 %** de precisión.
- Título del puesto: ~**90 %** de precisión.
- Ubicación (incluyendo remoto/híbrido): ~**85 %** de precisión.
- Fuente (plataforma): **100 %** de precisión.

Campos inferidos

- Tipo de contrato: ~**70 %** de ofertas con valor correctamente inferido.
- Nivel de experiencia: ~**80 %** de cobertura (mejora notable tras usar el título del puesto como pista).
- Skills: ~**90 %** de precisión (limitando a tecnologías concretas y evitando términos genéricos).

Clasificación de estados

- Precisión global en clasificación de estados: ~**92 %**.
- Los estados más frecuentes fueron `nueva_oferta`, `en_proceso` y `rechazado`.
- Estados como `entrevista_programada` y `aceptado` se clasificaron correctamente cuando aparecían explícitos en el correo.

7.2.2. Rendimiento del sistema

Tiempo de procesamiento por email

- Tiempo medio total: ~**15–25 segundos** por correo.
- Desglose aproximado:
  - Preprocesado: < **1 s**
  - Llamada a OpenAI: ~**10–20 s**

    - Postprocesado + escritura en Sheets: ~**2–3 s**
- Objetivo inicial: **< 30 s por email** → **cumplido**.

Coste por email procesado

- Coste medio: ~**0,00035 USD** por email (modelo GPT-4o-mini).
- Consumo típico:
    - *Input*: ~1000–1500 tokens.
    - *Output*: ~500–800 tokens.
- Objetivo inicial: **< 0,001 USD por email** → **cumplido**.

Tasa de éxito

- Tasa de procesamiento exitoso: ~**96 %** de emails.
- Errores principales:
    - *Rate limits* ocasionales de la API de OpenAI.
    - Emails con formatos muy inusuales (<2 %).
    - Errores de parseo JSON puntuales (<2 %).
- Objetivo inicial: **> 95 %** de tasa de éxito → **cumplido**.

7.2.3. Cobertura de información

Campos siempre disponibles

- Fecha del email: **100 %**.
- Fuente / plataforma: **100 %**.

Campos con alta cobertura

- Empresa: ~**95 %**.
- Título del puesto: ~**90 %**.

Campos con cobertura parcial

- Tipo de contrato: ~**70 %** de ofertas con información suficiente.
- Nivel de experiencia: ~**80 %** con valor razonable.
- Salario estimado: ~**85 %** de ofertas con estimación numérica (cuando no se indica explícito).
- Enlaces a la oferta: ~**60–70 %** con URL válida y utilizable.

7.3. Análisis de resultados

7.3.1. Distribución de scores de relevancia

La distribución de scores generados por el sistema fue:

- Score **5** (excelente encaje): ~**15–20 %** de ofertas.
- Score **4** (buen encaje): ~**25–30 %**.
- Score **3** (encaje moderado): ~**30–35 %**.
- Score **2** (poco encaje): ~**15–20 %**.

- Score **1** (no acorde): ~**5–10 %**.

Esta distribución indica que el sistema **diferencia adecuadamente** entre ofertas muy alineadas, aceptables y poco relevantes, evitando concentrar todos los casos en valores medios.

7.3.2. Análisis por fuente

- **LinkedIn**
  - ~**85 %** de las ofertas procesadas.
  - Formatos relativamente consistentes, buena calidad de información.
- **InfoJobs**
  - ~**10 %** de las ofertas.
  - Formatos algo más heterogéneos, pero manejables tras el preprocesado.
- **Otras plataformas (Tecnoempleo, Indeed, etc.)**
  - ~**5 %** de las ofertas.
  - Mayor variabilidad de estructura, algo más de errores y campos incompletos.

7.3.3. Errores y casos límite

Los principales problemas detectados fueron:

- **Cambios o variaciones de plantilla** en algunos correos, que reducían la precisión de inferencia de tipo de contrato o nivel.
- **Falta de información** en origen (por ejemplo, emails muy breves sin descripción detallada ni rango salarial).
- **Enlaces ausentes o poco claros**, especialmente en ciertos correos de LinkedIn donde el enlace principal a la oferta no aparece explícito.
- Casos aislados de **errores de parseo JSON** cuando el modelo no seguía exactamente el formato esperado, mitigados con validaciones adicionales.

7.4. Casos de uso y ejemplos

7.4.1. Ofertas de alta prioridad (score 4–5)

Ejemplo 1 – Penetration Tester (score 5)

- Rol técnico muy cercano al perfil objetivo.
- Matches aproximados:
  - Skills: ≈90–95 %.
  - Experiencia: **100 %**.
  - Ubicación: **100 %** (ciudad objetivo o remoto muy alineado).
  - Salario: ~**85 %**.
- Resultado: oferta valorada como **altamente relevante**; el candidato aplicó y avanzó en el proceso.

Ejemplo 2 – Automation Engineer remoto (score 4)

- Matches aproximados:
  - Skills: ~**85 %**.
  - Experiencia: ~**80 %**.
  - Ubicación: **100 %** (remoto compatible con preferencias).
  - Salario: ~**75 %**.
- Resultado: oferta relevante, buen encaje global, considerada de **alta prioridad**, aunque con ciertos compromisos salariales o de rol.

7.4.2. Ofertas de prioridad media (score 3)

Ejemplo 3 – Automation Engineer en Barcelona (score 3)

- Matches aproximados:
  - Skills: ~**60 %**.
  - Experiencia: ~**70 %**.
  - Ubicación: ~**50 %**.
  - Salario: ~**60 %**.
- Resultado: oferta **moderadamente relevante**; buena como opción secundaria o como plan B, pero no prioritaria frente a otras.

7.4.3. Ofertas de baja prioridad (score 1–2)

Ejemplo 4 – QA Project Manager en Madrid (score 2)

- Matches aproximados:
  - Skills: ~**40 %**.
  - Experiencia: ~**30 %** (nivel por debajo de la experiencia real).
  - Ubicación: ~**20 %**.
  - Salario: ~**40 %**.
- Resultado: oferta **poco relevante**, típicamente descartada por falta de alineación (nivel demasiado bajo o condiciones poco atractivas).

7.4.4. Casos especiales

- **Actualización de estado**
  - El sistema detectó correctamente cambios de estado (p. ej. de `en_proceso` a `rechazado` o `aceptado`) y actualizó la fila existente en lugar de crear duplicados.
- **Deduplicación**
  - Ofertas repetidas (misma empresa y mismo puesto) se identificaron mediante el ID lógico y fueron actualizadas, manteniendo un único registro coherente.

7.5. Limitaciones identificadas

A partir de los resultados experimentales se confirmaron en la práctica las limitaciones ya planteadas en la fase de diseño (sección 3.4) y analizadas en detalle en la discusión (sección 8.5). En particular, se observaron tres efectos principales:

- Cobertura incompleta de ciertos campos (tipo de contrato, nivel de experiencia y rango salarial) cuando la información no está presente en el correo.
- Dependencia del formato y de las plantillas utilizadas por cada plataforma, especialmente en correos muy breves o altamente personalizados.
- Disponibilidad parcial de enlaces directos a la oferta, lo que limita la automatización de algunos pasos posteriores del proceso.

Estas observaciones empíricas refuerzan las conclusiones sobre limitaciones técnicas y funcionales del sistema y sirven de base para las propuestas de mejora que se detallan en el capítulo de Trabajo Futuro.

7.6. Impacto y valor del sistema

Desde la perspectiva del usuario final, el sistema aportó valor en varios aspectos:

- **Ahorro de tiempo**
  - Se eliminó gran parte del trabajo manual de lectura, clasificación y registro de ofertas.
- **Mejor organización**
  - Todas las oportunidades y su estado quedaron centralizadas en **una única hoja de Google Sheets**, fácilmente filtrable y ordenable.
- **Facilidad para priorizar**
  - El scoring permitió centrarse primero en las ofertas de mayor encaje, mejorando la gestión de la energía y el tiempo durante la búsqueda.
- **Seguimiento del proceso**
  - Fue posible seguir el ciclo completo de cada candidatura (desde la oferta inicial hasta el resultado final) sin perder contexto.

En conjunto, el sistema se consolidó como una **herramienta práctica y de uso diario** durante la búsqueda de empleo.

7.7. Lecciones aprendidas

Del análisis de resultados se derivan varias lecciones relevantes:

- La **calidad del preprocesado** (limpieza de HTML, extracción de enlaces, recorte del cuerpo del email) tiene un impacto directo en la precisión de la IA.
- La **iteración sobre prompts** es fundamental: pequeños ajustes mejoran significativamente la cobertura en campos inferidos como tipo de contrato o nivel.
- Un **esquema de datos bien definido** desde el inicio (23 columnas) facilita tanto la evaluación como la explotación posterior de la información.
- La combinación de **backtesting** y **uso real en producción** proporciona una visión mucho más fiable del comportamiento del sistema que las pruebas sintéticas aisladas.
- La deduplicación y la actualización de estados son clave para que el sistema se mantenga útil en el tiempo y no derive en una hoja llena de registros redundantes.

Estas lecciones han guiado las decisiones de mejora y las propuestas de trabajo futuro descritas en capítulos posteriores.

8. DISCUSIÓN

8.1. Logros del proyecto

8.1.1. Logros técnicos

Desde el punto de vista técnico, el proyecto ha conseguido:

- **Automatizar extremo a extremo** el procesamiento de correos: desde la lectura en Gmail hasta el volcado estructurado en Google Sheets, sin intervención manual.
- **Extraer información estructurada con alta precisión**, incluso a partir de HTML ruidoso y formatos heterogéneos de diferentes portales de empleo.
- **Aplicar técnicas de *prompt engineering*** para combinar en una única llamada al modelo tanto la extracción de campos como el cálculo del *matching* y del *scoring*.
- **Integrar de forma robusta servicios heterogéneos** (n8n, Gmail API, OpenAI API y Google Sheets) en un flujo orquestado coherente y mantenible.
- **Optimizar costes de IA**, seleccionando un modelo ligero y ajustando el diseño de prompts para reducir tokens manteniendo la calidad de las respuestas.
- **Diseñar un mecanismo de backtesting reproducible**, capaz de reinyectar correos históricos y comparar de forma consistente los resultados generados por el sistema.

8.1.2. Logros funcionales

En términos funcionales, el sistema:

- **Resuelve un problema real de saturación de información**, filtrando y priorizando ofertas relevantes dentro de un volumen elevado de correos.
- **Centraliza la información clave** (empresa, puesto, localización, salario, tipo de contrato, *score*, etc.) en un único repositorio estructurado y fácil de consultar.
- **Facilita la toma de decisiones** al proporcionar un sistema de puntuación coherente que permite ordenar las ofertas según el ajuste al perfil del candidato.
- **Reduce drásticamente el trabajo manual repetitivo**, liberando tiempo para actividades de mayor valor (preparación de entrevistas, personalización de candidaturas, etc.).
- **Permite un seguimiento histórico de la búsqueda de empleo**, útil para análisis posteriores y mejora de la estrategia personal.

8.1.3. Logro destacado: impacto en la búsqueda de empleo

El logro más significativo es que el sistema se ha utilizado en una **búsqueda de empleo real**, demostrando que:

- La priorización generada por el *scoring* coincide de forma consistente con la percepción subjetiva del candidato.
- El tiempo dedicado a revisar ofertas y decidir dónde aplicar se reduce de forma notable.
- El flujo diario de correos deja de ser una fuente de ruido y se convierte en un canal estructurado de oportunidades.

Este uso real proporciona una validación práctica que va más allá de las pruebas sintéticas o puramente académicas.

8.2. Desafíos encontrados

8.2.1. Desafíos técnicos

Durante el desarrollo se identificaron varios retos técnicos:

- **Variabilidad en el formato de los correos**

  Cada portal y empresa emplea plantillas distintas, con HTML complejo, secciones opcionales y cambios frecuentes. Fue necesario:

  - incorporar un preprocesado que limpiara HTML y extrajera el contenido relevante;
  - diseñar prompts tolerantes a ruido y campos ausentes.
- **Inferencia de información no explícita**

  Campos como tipo de contrato, nivel de seniority o rango salarial no siempre aparecen explícitos. El sistema debe:

  - inferir valores razonables a partir del contexto (título del puesto, descripción, palabras clave);
  - marcar como *desconocido* cuando la inferencia no sea fiable, evitando inventar datos.
- **Extracción fiable de enlaces**

  Algunos correos incluyen múltiples enlaces (seguimiento, *unsubscribe*, landings genéricas, etc.) o enlaces embebidos en HTML complejo. Se tuvieron que:

  - priorizar enlaces que apuntan directamente a la oferta;
  - filtrar parámetros de *tracking*;
  - manejar casos en los que no existe un enlace claro de aplicación.
- **Gestión de errores y casos extremos**

  La automatización debía seguir funcionando ante:

  - errores puntuales de la API;
  - correos mal formados;
  - formatos inesperados.

    Para ello se añadieron validaciones en varias etapas, valores por defecto y trazas que facilitan el diagnóstico.

8.2.2. Desafíos de diseño

A nivel de diseño del sistema, los principales retos fueron:

- **Equilibrio entre precisión y cobertura**

  Un prompt demasiado estricto puede perder información; uno demasiado flexible puede introducir errores. El sistema evolucionó de forma iterativa hasta encontrar un equilibrio adecuado:

  - aceptando ciertos campos vacíos cuando la información no existe;
  - priorizando la consistencia de los campos críticos (empresa, puesto, *matching*, *score*).

- **Optimización de costes de explotación**

  Se tuvo que garantizar que el coste por correo procesado fuera lo suficientemente bajo para permitir un uso continuado:

  - seleccionando un modelo más ligero (en lugar de los modelos más costosos);
  - minimizando tokens mediante prompts compactos y reutilizables;
  - combinando en una sola llamada la extracción de campos y el cálculo de *scores*.

8.2.3. Desafíos de validación

La validación planteó retos adicionales:

- **Disponibilidad de datos reales suficientes**

  El proyecto se ha apoyado:

  - en un conjunto de correos históricos (backtesting);
  - en el uso continuado del sistema durante una búsqueda real de empleo.

    Aun así, la muestra sigue siendo limitada a un perfil y contexto concretos.

- **Subjetividad en la relevancia de las ofertas**

  La relevancia de una oferta no es un valor absoluto: depende de preferencias personales, momento profesional, condiciones económicas, etc.

  Para mitigar esta subjetividad:

  - el *scoring* se diseñó alineado con el perfil concreto del candidato;
  - se evaluó la utilidad del ranking desde la perspectiva del propio usuario final (el desarrollador), contrastando casos en los que el sistema priorizaba ofertas que efectivamente se consideraron interesantes.

8.3. Comparación con soluciones existentes

8.3.1. Proceso manual tradicional

El proceso manual típico de revisión de correos implica:

- Leer uno a uno todos los emails procedentes de diferentes portales.

- Abrir enlaces, revisar descripciones y decidir dónde aplicar.
- Volcar a mano la información relevante en hojas de cálculo o notas personales.

Comparado con este enfoque, la solución propuesta:

- **Reduce la carga cognitiva**, al presentar toda la información normalizada en un único panel.
- **Disminuye el tiempo invertido**, especialmente cuando el volumen de correos es alto.
- **Evita omisiones y errores humanos** en la transcripción de datos.

8.3.2. Filtros y herramientas estándar de correo

Gmail y otros proveedores ofrecen filtros, etiquetas y bandejas inteligentes, pero:

- no extraen campos estructurados (empresa, salario, tipo de contrato, etc.);
- no calculan un *matching* cuantitativo con el perfil del candidato;
- no consolidan en una vista única todas las ofertas relevantes.

La propuesta va un paso más allá al convertir los correos en **registros estructurados y puntuados**, listos para análisis y seguimiento.

8.3.3. Plataformas y servicios de terceros

Existen plataformas externas que ayudan a gestionar candidaturas, pero suelen:

- requerir que el usuario introduzca la información de forma manual;
- no integrarse de forma directa con la bandeja de entrada;
- centrarse en la gestión del proceso (pipeline de entrevistas) más que en la **clasificación automática de ofertas**.

El sistema desarrollado se diferencia precisamente en:

- partir directamente del flujo natural de información (los correos);
- automatizar la extracción y priorización sin cambiar el hábito del usuario;
- ser **autoalojable y extensible**, evitando dependencia de una plataforma concreta.

8.4. Impacto y aplicabilidad

El impacto principal del sistema se observa en tres dimensiones:

1. **Eficiencia personal en la búsqueda de empleo**

   El usuario pasa de revisar manualmente todas las ofertas a trabajar sobre un listado priorizado. Esto:

   - reduce significativamente el tiempo diario dedicado a “limpiar” el correo;
   - permite focalizar las energías en preparar buenas candidaturas y entrevistas.
2. **Calidad de la toma de decisiones**

Al disponer de un *score* consistente y de campos estructurados:

   - es más sencillo comparar ofertas entre sí;
   - se identifican rápidamente oportunidades que, de otro modo, podrían quedar enterradas en la bandeja de entrada.

3. **Reutilización del enfoque en otros dominios**

   La arquitectura propuesta (correo → n8n → IA → repositorio estructurado) es aplicable a otros escenarios, por ejemplo:

   - gestión de leads comerciales;
   - soporte técnico y clasificación de incidencias;
   - monitorización de oportunidades de negocio, subvenciones o licitaciones.

En todos estos casos, el patrón es similar: alto volumen de correos, necesidad de extracción de información y priorización según criterios específicos.

8.5. Limitaciones y consideraciones

8.5.1. Limitaciones técnicas

Entre las limitaciones técnicas más relevantes se encuentran:

- **Dependencia de APIs externas**: el funcionamiento continuo del sistema está condicionado por la disponibilidad y condiciones de servicios como OpenAI y Google Sheets.
- **Variabilidad en la calidad de extracción**: formatos especialmente complejos o cambios bruscos en las plantillas de correo pueden degradar la precisión.
- **Información incompleta en origen**: cuando ciertos datos no existen en el correo, el sistema no puede obtenerlos sin recurrir a fuentes adicionales.
- **Coste acumulativo**: aunque el coste unitario por email es bajo, un volumen muy alto de correos puede suponer un coste mensual significativo.

8.5.2. Limitaciones funcionales

A nivel funcional, el sistema actual:

- está optimizado para **un perfil concreto de candidato**; extenderlo a múltiples perfiles requeriría ajustar prompts y lógica de *scoring*;
- no incluye todavía **sistema de alertas** (por ejemplo, notificaciones cuando llega una oferta muy alineada);
- no dispone de un **dashboard avanzado** con métricas y visualizaciones agregadas;
- realiza **estimaciones salariales** basadas en el conocimiento del modelo y en patrones lingüísticos, sin consultar fuentes de datos en tiempo real.

8.5.3. Consideraciones éticas y de privacidad

El uso de este tipo de sistemas exige tener en cuenta:

- **Privacidad y protección de datos**: se procesan correos que pueden contener información personal; es necesario garantizar un manejo adecuado, especialmente en entornos corporativos.
- **Uso responsable de la IA**: el usuario debe ser consciente de que la clasificación y el *scoring* son generados por un modelo probabilístico, no por reglas deterministas.
- **Sesgos en los modelos**: los modelos de lenguaje pueden reflejar sesgos presentes en sus datos de entrenamiento; es recomendable supervisar resultados y no delegar decisiones críticas exclusivamente en la IA.
- **Cumplimiento de términos de servicio**: cualquier integración con plataformas de terceros debe respetar sus condiciones de uso.

8.6. Contribuciones del trabajo

Las contribuciones técnicas, prácticas y académicas del proyecto se resumen en detalle en el capítulo 9 (apartado 9.2). Desde la perspectiva de la discusión, la aportación más relevante es haber demostrado que la combinación de automatización con n8n, modelos de lenguaje generativos y servicios cloud permite construir un sistema funcional de apoyo a la búsqueda de empleo, desplegado y validado en un entorno real a partir de datos procedentes exclusivamente del correo electrónico.

8.7. Reflexiones finales

En conjunto, este proyecto muestra que combinar automatización de flujos, servicios cloud e IA generativa permite transformar un proceso manual, repetitivo y poco escalable en un sistema estructurado, medible y alineado con objetivos reales. Más allá del caso concreto de la búsqueda de empleo, el trabajo ilustra un patrón general de diseño que puede reutilizarse en otros contextos donde el correo actúa como principal canal de información.

9. CONCLUSIONES

9.1. Conclusiones principales

9.1.1. Grado de cumplimiento de objetivos

El proyecto ha cumplido el objetivo general definido en el capítulo 3: diseñar e implementar un sistema automatizado capaz de procesar correos de ofertas de empleo, extraer información estructurada, evaluar el encaje con un perfil profesional concreto y almacenar los resultados en un repositorio explotable.

En términos funcionales, el sistema procesa de forma automática los correos relevantes, extrae los campos principales de cada oferta y calcula un *score* de relevancia y varios indicadores de *matching* alineados con el perfil objetivo. Desde el punto de vista técnico, se ha desplegado un pipeline completo en n8n que integra Gmail, OpenAI y Google Sheets, optimizando tanto el coste por correo como el tiempo de procesamiento.

En conjunto, puede afirmarse que los objetivos funcionales, técnicos y de evaluación establecidos en el capítulo 3 se consideran alcanzados, y que los resultados obtenidos —desarrollados en los capítulos 7 y 8— son coherentes con las expectativas iniciales del proyecto.

9.1.2. Resultados obtenidos

Los resultados cuantitativos y cualitativos confirman la viabilidad del enfoque:

- Extracción de información con alta precisión en campos críticos como empresa y título del puesto, y precisión razonable en campos derivados como tipo de contrato o nivel de experiencia.
- Capacidad para procesar de forma robusta un conjunto de más de 50 ofertas reales en fase de validación, procedentes principalmente de LinkedIn e InfoJobs.
- Reducción significativa del tiempo invertido en revisar correos y registrar ofertas, al centralizar toda la información en una hoja de cálculo priorizada.

El resultado más relevante es la validación práctica: el sistema se utilizó durante una búsqueda de empleo real y contribuyó de forma directa a la obtención de un nuevo puesto de trabajo que comenzó el 27 de octubre de 2025, demostrando que la solución no es solo técnicamente correcta, sino útil en la práctica.

9.2. Contribuciones del trabajo

9.2.1. Contribuciones técnicas

A nivel técnico, el trabajo aporta:

- Un diseño de **pipeline de automatización basado en n8n e IA generativa** que puede reutilizarse en otros escenarios de procesamiento de correo.

- Un conjunto de **prompts estructurados y estrategias de validación** para extraer información en formato JSON y calcular *scores* personalizados en una sola llamada al modelo.
- Un ejemplo completo de **integración de servicios cloud heterogéneos** (Gmail, OpenAI, Google Sheets) dentro de una plataforma de automatización *self-hosted*, prestando atención a manejo de errores, deduplicación y optimización de costes.

9.2.2. Contribuciones prácticas

En el plano práctico, el trabajo entrega:

- Una **solución funcional y desplegada** que puede utilizarse inmediatamente para gestionar la búsqueda de empleo de un candidato real.
- Un **caso de estudio completo**, desde el análisis del problema hasta la validación en producción, que puede servir de guía para profesionales que deseen automatizar flujos similares (gestión de leads, clasificación de incidencias, etc.).
- Documentación suficiente para que otros usuarios puedan **replicar, adaptar y extender** el sistema a su propio contexto.

9.2.3. Contribuciones académicas

Desde la perspectiva académica, el trabajo:

- Ilustra una **aplicación concreta de modelos de lenguaje generativos** a un problema real de extracción y análisis de texto en entornos profesionales.
- Propone una **metodología de desarrollo iterativa** aplicada a proyectos de automatización con IA, con fases bien delimitadas y soporte en datos reales.
- Incluye un **esquema de métricas y estrategia de evaluación** (precisión por campo, rendimiento, coste, validación práctica) trasladable a otros sistemas de procesamiento de información.

9.3. Aprendizajes obtenidos

9.3.1. Aprendizajes técnicos

Durante el desarrollo del sistema se han obtenido varios aprendizajes relevantes:

- El *prompt engineering* efectivo exige iteración continua sobre datos reales; pequeños cambios en las instrucciones pueden tener impactos significativos en la calidad de la extracción.
- La inferencia de información no explícita (tipo de contrato, nivel de experiencia, salario estimado) requiere instrucciones muy precisas y validación posterior para evitar resultados inventados.
- La validación de respuestas del modelo y el tratamiento defensivo de errores (valores por defecto, chequeos de formato, logs) son imprescindibles para mantener la robustez del sistema.
- Es posible **optimizar costes de forma agresiva** sin perder calidad, combinando prompts compactos, modelos ligeros y una sola llamada por mensaje.

9.3.2. Aprendizajes de proceso y metodología

En cuanto al proceso de trabajo, destacan los siguientes aprendizajes:

- El uso de una metodología ágil, con prototipado rápido sobre n8n, permite pasar muy pronto de la idea a un sistema funcional y validable.
- El hecho de que el desarrollador sea también el usuario final facilita priorizar lo que realmente aporta valor, evitando funcionalidad innecesaria.
- La iteración basada en feedback real (búsqueda de empleo en curso) es mucho más eficaz que el diseño únicamente teórico.
- Una buena documentación técnica, redactada en paralelo al desarrollo, reduce drásticamente el esfuerzo de cierre del trabajo y mejora la reproducibilidad del trabajo.

9.4. Limitaciones y áreas de mejora

Las principales limitaciones del sistema se han descrito en detalle en el apartado 8.5. De forma sintética, pueden agruparse en cuatro bloques: (i) dependencia de APIs externas y de la estructura de los correos, (ii) enfoque funcional centrado en un único perfil de candidato y sin componentes avanzados de visualización ni alertas, (iii) evaluación basada en un conjunto de datos limitado y en la perspectiva de un único usuario y (iv) consideraciones de privacidad y posibles sesgos inherentes al uso de modelos de lenguaje.

Estas restricciones no invalidan los resultados obtenidos, pero marcan con claridad las líneas de mejora del sistema: ampliación de fuentes y perfiles soportados, incorporación de nuevas fuentes de datos (por ejemplo, salariales), refuerzo de los mecanismos de evaluación y monitorización y diseño de una arquitectura más flexible orientada a multiusuario. Estas ideas se desarrollan en el capítulo de Trabajo Futuro.

9.5. Reflexión final

Este trabajo ha mostrado que es posible transformar un proceso tedioso y manual —la revisión diaria de correos de ofertas de empleo— en un flujo automatizado, estructurado y alineado con las prioridades del usuario. Al integrar automatización de workflows, servicios cloud e inteligencia artificial generativa, se ha logrado reducir la carga operativa y aumentar la calidad de la información utilizada para tomar decisiones.

La experiencia sugiere que el verdadero valor de estos sistemas no reside únicamente en “usar IA”, sino en **diseñar bien el flujo completo**: qué datos se capturan, cómo se preprocesan, qué se pide exactamente al modelo, cómo se validan las respuestas y cómo se integran los resultados en las herramientas que el usuario ya utiliza (en este caso, una hoja de cálculo).

9.6. Conclusión general

En conclusión, el trabajo demuestra que la combinación de plataformas de automatización, modelos de lenguaje generativos y servicios cloud permite construir soluciones reales que aportan valor inmediato en contextos cotidianos como la búsqueda de empleo. El sistema desarrollado cumple los objetivos técnicos y funcionales planteados, ha sido validado en un entorno real y se apoya en una arquitectura suficientemente flexible como para extenderse a otros dominios de procesamiento de información basada en correo.

Más allá del caso concreto abordado, el proyecto constituye una prueba de concepto sólida de cómo la ingeniería de automatización apoyada en IA puede convertirse en una herramienta estratégica para profesionales que necesitan gestionar grandes volúmenes de información de forma eficiente, manteniendo al mismo tiempo el control y la responsabilidad sobre las decisiones finales.

10. TRABAJO FUTURO

10.1. Extensión funcional del sistema

Una primera línea clara de evolución consiste en ampliar las capacidades funcionales del sistema más allá del uso individual actual:

- **Soporte multiusuario y multiperfil**

  Rediseñar la capa de almacenamiento para permitir que varios usuarios gestionen sus ofertas de forma aislada, cada uno con su propio perfil profesional, criterios de *scoring* y hoja de resultados.

- **Sistema de alertas y notificaciones**

  Incorporar un módulo de notificaciones (por ejemplo, mediante correo, Telegram o Slack) que avise cuando llegue una oferta con *score* alto o cuando cambie el estado de una candidatura relevante.

- **Dashboard de visualización**

  Construir un panel de control (por ejemplo, con herramientas como Looker Studio, Grafana o una aplicación web propia) que permita visualizar métricas agregadas: número de ofertas por fuente, distribución de *scores*, evolución temporal de candidaturas, etc.

- **Gestión integrada del ciclo de candidatura**

  Extender el sistema para registrar no solo las ofertas recibidas, sino también las acciones realizadas (fecha de aplicación, entrevistas, pruebas técnicas, feedback), convirtiéndolo en un gestor completo de procesos de selección.

10.2. Mejoras técnicas y arquitectónicas

Desde el punto de vista técnico, se identifican varias oportunidades de mejora:

- **Migración a una base de datos dedicada**

  Sustituir Google Sheets por una base de datos relacional (por ejemplo, PostgreSQL) o un servicio *serverless* (como Supabase o Firebase) que facilite consultas más complejas, control de versiones y escalabilidad multiusuario.

- **Abstracción de fuentes y conectores**

  Diseñar una capa de integración genérica para añadir nuevas fuentes (otros correos, APIs de portales de empleo, feeds RSS) sin modificar la lógica central de extracción y *scoring*.

- **Uso de modelos alternativos de IA**

Evaluar otros modelos de lenguaje (por ejemplo, modelos *open-source* desplegados en local o en GPU propia) para reducir aún más el coste por mensaje o aumentar el control sobre el procesamiento de datos.

- **Mejoras en robustez y observabilidad**

  Incorporar métricas de uso, paneles de monitorización y alertas sobre errores recurrentes o degradación de precisión, así como pruebas automatizadas sobre prompts y flujos clave.

10.3. Ampliación de la evaluación

La evaluación realizada se centra en un único usuario y en un conjunto acotado de correos. Futuras líneas de trabajo pueden incluir:

- **Estudios con múltiples usuarios**

  Probar el sistema con perfiles de candidatos diferentes (junior, senior, otros sectores TIC) para analizar cómo se comporta el *scoring* y qué ajustes son necesarios en los prompts y criterios de valoración.

- **Evaluación a largo plazo**

  Medir el rendimiento y la utilidad del sistema durante periodos más largos (por ejemplo, un año), analizando la evolución de las búsquedas de empleo, las tasas de respuesta y la correlación entre *score* y éxito de las candidaturas.

- **Comparación con enfoques alternativos**

  Comparar el sistema con otras estrategias (proceso manual tradicional, uso de herramientas de terceros, filtros avanzados de correo) mediante estudios de tiempo invertido, número de entrevistas conseguidas y satisfacción percibida por los usuarios.

- **Validación cruzada del scoring**

  Pedir a expertos o a otros candidatos que valoren de forma independiente la relevancia de las ofertas y comparar sus juicios con el *scoring* generado automáticamente para detectar posibles sesgos o áreas de mejora.

10.4. Generalización a otros dominios

El patrón arquitectónico desarrollado —captura de emails, extracción mediante IA, *scoring* y almacenamiento estructurado— es aplicable a otros contextos más allá de la búsqueda de empleo. Algunas líneas de trabajo futuro incluyen:

- **Gestión de leads comerciales**

  Clasificar y priorizar automáticamente correos de potenciales clientes, asignando scores según sector, tamaño de empresa, interés aparente y probabilidad de cierre.

- **Soporte técnico y gestión de incidencias**

  Extraer información clave de correos de soporte (tipo de incidencia, criticidad, producto afectado) y priorizar tickets según impacto y urgencia.

- **Monitorización de oportunidades de negocio**

  Procesar correos relacionados con licitaciones, subvenciones o colaboraciones y evaluarlos según alineación con las capacidades y objetivos de una empresa.

Explorar estas variantes permitiría validar la generalidad del enfoque propuesto y abrir la puerta a soluciones más amplias de automatización basada en correo y modelos de lenguaje generativos.

10.5. Integración con herramientas de búsqueda de empleo

Finalmente, una línea de trabajo especialmente interesante consiste en integrar el sistema con herramientas existentes de búsqueda de empleo:

- **Exportación hacia ATS y plataformas de gestión de candidaturas**

  Permitir que los registros estructurados generados (empresa, puesto, estado, score, etc.) se sincronicen con Applicant Tracking Systems (ATS) u otras herramientas de RRHH.

- **Interacción desde una interfaz web o móvil**

  Desarrollar una capa de interfaz ligera que permita al usuario:

  - revisar ofertas desde el móvil,
  - actualizar estados de candidatura,
  - añadir notas personales por oferta, manteniendo la automatización actual como backend.

Estas líneas de trabajo consolidarían el sistema no solo como una herramienta personal, sino como la base de una solución más amplia y profesionalizable.

11. REFERENCIAS BIBLIOGRÁFICAS

[1] n8n GmbH, *n8n Documentation*, 2025.
Disponible en: https://docs.n8n.io.

[2] OpenAI, *OpenAI API Documentation*, 2025.
Disponible en: https://platform.openai.com/docs.

[3] Google, *Gmail API Overview*, 2025.
Disponible en: https://developers.google.com/gmail/api.

[4] Google, *Google Sheets API Overview*, 2025.
Disponible en: https://developers.google.com/sheets/api.

[5] OpenAI, *GPT-4 Technical Report*, 2024.
Disponible en: https://openai.com/index/gpt-4.

[6] Apuntes de la asignatura “Aula Fundamentos Técnicos”, Racks Academy, 2024–2025.

[7] Apuntes de la asignatura “Aula de Automatizaciones”, Racks Academy, 2024–2025.

[8] Apuntes de la asignatura “Aula LLMs”, Racks Academy, 2024–2025.