

# HireGen: Tool to automate the review of resumes and generate employment

Juan Camilo Valencia  
Maria Angelica Portocarrero Quintero  
Xilena Atenea Rojas Salazar

June 7, 2025

## 1 Introducción

En el mundo actual, encontrar a la persona adecuada para un puesto de trabajo puede ser un desafío. Las empresas suelen recibir muchísimos currículums (CVs) para cada oferta, y revisar cada uno a mano es una tarea que consume muchísimo tiempo y esfuerzo. Esto es especialmente complicado para una startup como la nuestra, HireGen.

A diferencia de las grandes empresas de contratación que tienen presupuestos muy grandes, nos propusimos abordar este problema de una forma inteligente y eficiente con un presupuesto muchísimo menor.

La idea es automatizar el proceso inicial de filtrado de CVs. Para lograrlo, nos apoyamos en herramientas muy accesibles pero potentes, como n8n, que nos permite conectar diferentes servicios sin necesidad de escribir mucho código. Además, usamos las APIs de Google (para manejar correos, documentos, etc.) y la API de Telegram y Google para recibir notificaciones con relación al proceso o vacante. Con esta combinación, podemos recibir un CV, procesarlo, extraer la información clave, clasificarla y notificar de forma automática.

El objetivo de este informe mostrar, paso a paso y de forma muy clara, cómo hemos diseñado y construido este sistema de automatización. A lo largo de las siguientes secciones, profundizaremos en el detalle del funcionamiento interno de HireGen para filtrar los CVs de manera eficiente, explicando cada componente y su propósito.

## 2 Marco Teórico

### 2.1 Plataformas Low-Code/No-Code

Las plataformas Low-Code/No-Code son entornos de desarrollo de software que permiten a los usuarios crear aplicaciones y automatizaciones con una mínima (Low-Code) o ninguna (No-Code) escritura de código. Funcionan mediante interfaces visuales, componentes preconstruidos y lógica de arrastrar y soltar, democratizando la creación de soluciones digitales.[1]

## 2.2 APIs e integración de servicios

Una Interfaz de Programación de Aplicaciones (API) es un conjunto de reglas, protocolos y herramientas que define cómo diferentes aplicaciones de software deben comunicarse entre sí. Permite que una aplicación solicite servicios o acceda a datos de otra aplicación de manera estandarizada.

La integración de servicios, es el proceso de conectar y coordinar múltiples APIs y sistemas para crear un ecosistema cohesivo que comparte datos y automatiza flujos de trabajo complejos a través de diferentes plataformas.[2]

## 2.3 Modelo de Lenguaje Grande (LLM)

Por sus siglas en inglés 'Large Language Model' es un tipo específico de modelo de Inteligencia Artificial que ha sido entrenado con vastas cantidades de datos textuales. Esto le permite comprender, generar y procesar lenguaje humano con una coherencia y complejidad avanzadas, siendo capaz de realizar funciones como el análisis de texto, la extracción de información, la generación de contenido y la clasificación semántica.[3]

## 2.4 Procesamiento de Lenguaje Natural (NLP)

El Procesamiento de Lenguaje Natural (PLN o NLP por sus siglas en inglés) es una rama de la Inteligencia Artificial que se enfoca en permitir que las computadoras comprendan, interpreten, generen y manipulen el lenguaje humano. Implica el uso de técnicas para analizar texto y voz, identificar significados, extraer información clave y responder de manera coherente.[6]

## 2.5 Workflow / Flujo de Trabajo

En el contexto de la automatización, un workflow o flujo de trabajo es una secuencia estructurada y predefinida de tareas o pasos que se ejecutan de manera consecutiva para completar un proceso. Cada paso puede ser una acción, una decisión o una integración con otro sistema, y el flujo define la lógica y el orden en que se realizan.[4]

## 2.6 API Key

Una API Key (o clave de API) es un código único que se utiliza para identificar a un usuario, una aplicación o un proyecto cuando este realiza solicitudes a una Interfaz de Programación de Aplicaciones (API). Actúa como una credencial de autenticación que permite a los servicios verificar quién está haciendo la solicitud y controlar el acceso a sus funcionalidades.[5]

# 3 Descripción del problema

Para entender bien por qué desarrollamos HireGen, conversamos con una experta en

temas de contratación. Ella cuenta con más de 10 años de experiencia en grandes empresas, donde ha trabajado con equipos dedicados a la búsqueda y selección de personal. Nos explicó que, incluso en estas compañías con todos sus recursos y personal especializado, el proceso de selección inicial –es decir, ese primer filtro donde se revisan los requisitos obligatorios de cada currículum (CV)– consume una cantidad considerable de tiempo. La gran cantidad de vacantes y la diversidad de perfiles profesionales que manejan hacen que esta fase sea muy compleja y lenta.

Pero la experta nos hizo ver un punto clave. Si para las empresas grandes, con toda su infraestructura, esto ya es un reto, la situación es aún más complicada para las pequeñas o microempresas. Esto se debe a que ellas casi nunca tienen un equipo dedicado a la contratación, ni el tiempo suficiente, ni el presupuesto para invertir en software costoso. Cuando necesitan contratar a alguien, la tarea de revisar CVs, llamar candidatos y hacer el seguimiento se convierte en una carga enorme que les desvía de su negocio principal. Esto a menudo lleva a que los procesos de contratación sean menos eficientes o tomen mucho más tiempo para encontrar a la persona adecuada.

Es aquí donde HireGen entra en juego. Pensamos en esto como una oportunidad para que nuestra iniciativa de startup ofrezca una solución práctica. HireGen permite que el filtrado inicial de CVs se realice de forma automatizada y sencilla. Esta solución se adapta al bajo costo y está pensada especialmente para las microempresas que buscan optimizar sus procesos de selección.

## 4 Planteamiento de la Solución

Para la automatización del proceso de selección HireGen se organiza en dos bloques principales dentro de n8n, utilizando también varias APIs (Google y Telegram). Estos bloques son: Procesamiento de datos del CV y el de Notificaciones.

### 4.1 Procesamiento de datos del CV

- **Recepción de la CV:** Se recibe la CV vía E-mail como archivo adjunto
- **Extracción y limpieza de la información:** Una vez que tenemos el CV, el sistema se encarga de sacar los datos más importantes de él, como el título académico, el tiempo de experiencia en años o meses. Además, limpia el texto del CV para que esté en el formato ideal para el siguiente paso.
- **Clasificación:** Usamos un modelo LLM, le damos a esta LLM unas instrucciones claras ("Prompts") para que lea la información extraída de la CV y lo clasifique. La decisión es sencilla: el CV es marcado como "relacionado" si cumple con todos los requisitos obligatorios del puesto, o "no relacionado" si le falta alguno de ellos. También evalúa si cumple con los deseables añadiéndolos a la salida que genera el modelo.

## 4.2 Notificaciones

Una vez que el CV ha sido clasificado por el modelo LLM se hace el envío de notificación cada vez se clasifica una CV como relacionada.

- Gmail: Se envía un correo electrónico alertando sobre un CV clasificado como relacionado a la vacante
- Telegram: Se envía un mensaje instantáneo alertando sobre un CV clasificado como relacionado a la vacante

## 5 Entendimiento de los datos

Para este flujo, los datos de entrada son archivos en formato PDF que contienen hojas de vida (CVs) enviadas por candidatos a través del correo electrónico.

El proceso inicia con la recepción de estos correos, los cuales deben incluir como adjunto un archivo PDF correspondiente al CV del remitente. Esta condición es necesaria para activar el flujo automatizado.

A nivel estructural, el sistema presupone dos elementos preexistentes: (1) una cuenta de Google vinculada al flujo de trabajo de **n8n**, la cual debe contener una carpeta designada en Google Drive para almacenar los CVs recibidos, y (2) un archivo de Google Sheets previamente creado, que cuente con dos hojas o pestañas destinadas a almacenar los resúmenes generados. Estas hojas corresponden a las categorías “**relacionado**” y “**no relacionado**”, las cuales reflejan la evaluación automática de cada candidato con respecto a una vacante determinada.

## 6 Preparación de los datos

Durante el desarrollo del flujo, los datos recibidos son hojas de vida en formato PDF. Estas atraviesan una serie de transformaciones diseñadas para asegurar su correcta interpretación por parte del modelo de lenguaje, así como su posterior estructuración y almacenamiento.

El primer paso en este proceso ocurre tras la descarga del archivo PDF desde Google Drive. Este archivo es enviado a un módulo de extracción de texto, el cual se encarga de leer el contenido del documento y convertirlo en una cadena de texto sin formato. Esta transformación se hizo debido a que los modelos de lenguaje como LLaMA 3.2 requieren datos en texto plano para su análisis semántico.

Una vez obtenido el texto, este se transmite al modelo de lenguaje mediante un componente de tipo *Basic LLM Chain*, el cual está configurado con un prompt especializado. Este prompt instruye al modelo para que evalúe el contenido del CV, lo compare con una descripción de vacante predefinida, y produzca una salida estructurada en formato JSON. Dicha salida incluye campos como el nombre del candidato, vacante objetivo, resumen profesional, principales habilidades, información de contacto y una categoría que indica si el perfil es “relacionado” o “no relacionado” con la vacante. En la sección 7.3 se describe

en detalle las estrategias llevadas a cabo para optimizar el output del modelo de lenguaje.

Debido a la naturaleza de la respuesta generada por el modelo, es necesario realizar una transformación adicional, ya que esta es presentada como texto plano. Esta transformación es llevada a cabo por un nodo de código personalizado, el cual toma la respuesta como cadena de texto y aplica un proceso de limpieza para remover caracteres innecesarios y asegurar que el formato JSON sea válido.

A continuación, el texto es interpretado como un objeto JSON, y se extraen los campos de interés. El resultado es una estructura uniforme que facilita su inserción en una hoja de cálculo en el siguiente nodo.

Finalmente, este objeto estructurado se transmite a un nodo de Google Sheets, donde se añade una nueva fila en la hoja correspondiente según la clasificación del candidato (“relacionado” o “no relacionado”). A partir de esta actualización, se activan otros flujos destinados al envío de notificaciones automatizadas por correo electrónico o Telegram, lo cual completa el ciclo de transformación y uso de los datos.

## 7 Modelado - Visualización y métricas

### 7.1 Generación de credenciales

Para que HireGen pueda funcionar y comunicarse con las diferentes herramientas que usamos, como Google y Telegram, necesita sus credenciales para acceder a esos servicios.

Generar y configurar estas credenciales es un paso fundamental. Con ellas, n8n puede, leer correos y enviar mensajes por Telegram de forma segura y controlada. Sin credenciales bien configuradas, el sistema no podría conectarse ni realizar ninguna tarea de automatización.

#### 7.1.1 Configuración Google Cloud

Para iniciar a usar los servicios de Google, es necesario tener una cuenta, ingresamos a <https://console.cloud.google.com>, si es la primera vez que entras, es posible que te pida que aceptes los términos y condiciones.

Una vez tenemos la cuenta y aceptamos términos y condiciones es hora de crear nuestro proyecto

- **Creación de un nuevo proyecto** Para crear el proyecto nuevo, daremos clic al lado del icono de Google Cloud como lo muestra la siguiente figura.  
Ahora crearemos un nuevo proyecto dando clic en “proyecto nuevo”.

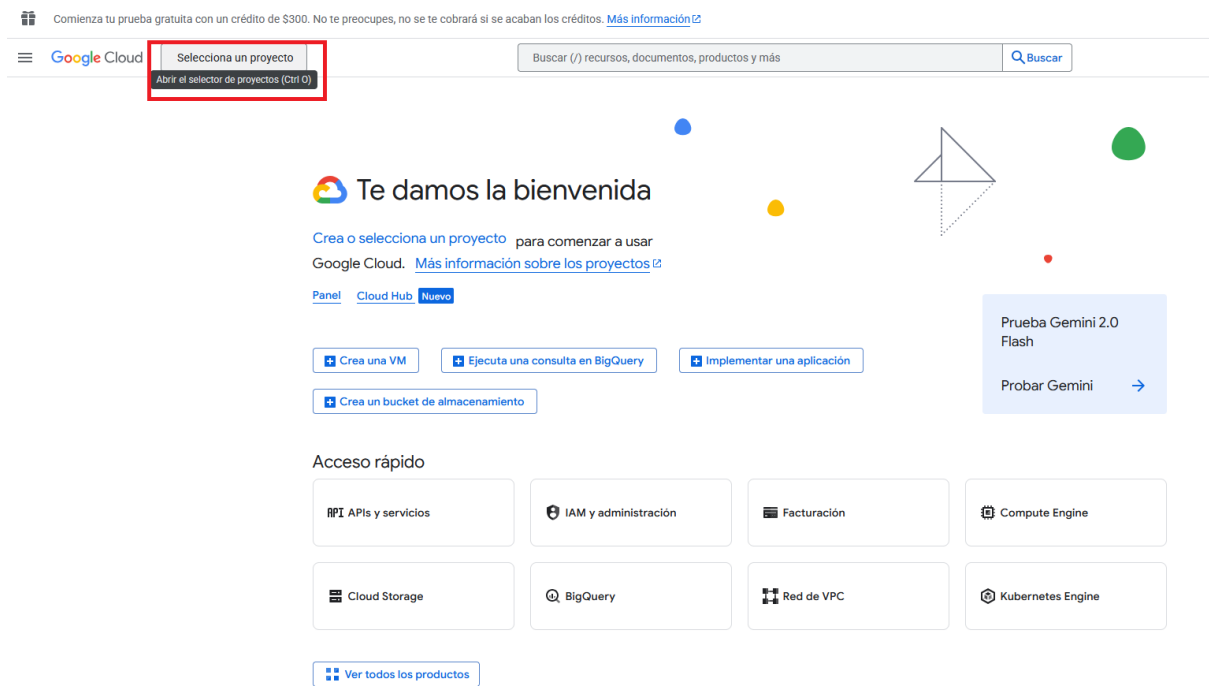


Figure 1: Creación de nuevo proyecto en Google Cloud

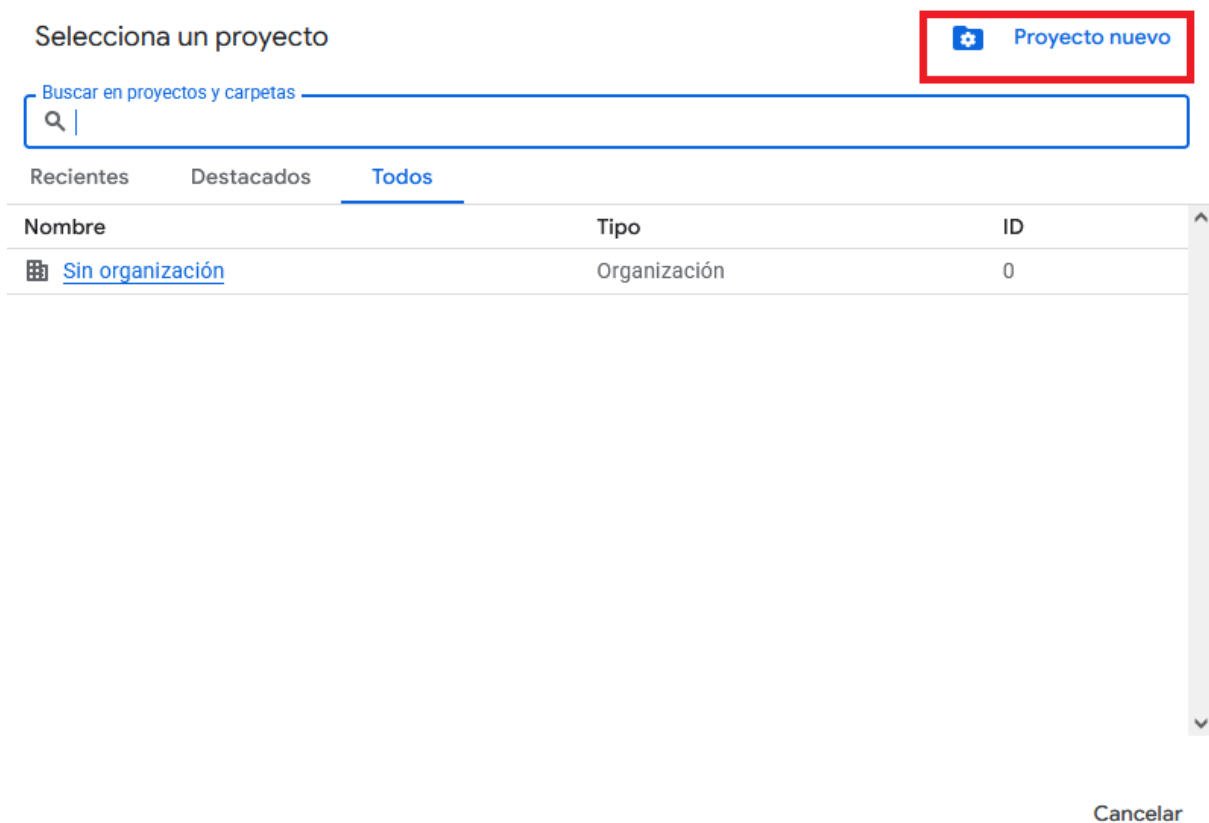


Figure 2: Proyecto nuevo

- **Creacion de credenciales de Google** Ya con un proyecto creado crearemos las credenciales OAuth que usara n8n para la comunicación con las API de Google, iremos al menú de navegación, seleccionaremos la opción "API y servicios" junto con "Credenciales" como se muestra en la siguiente figura.

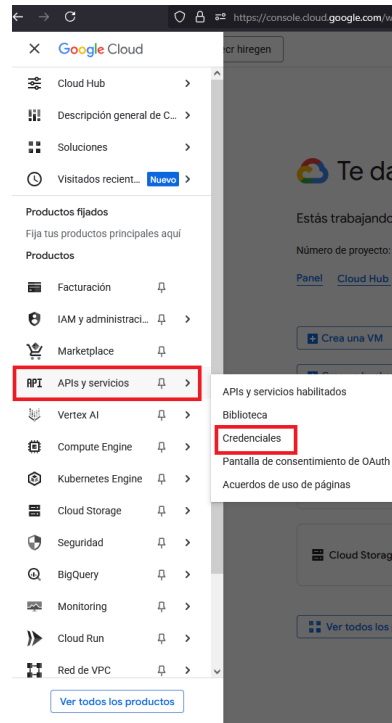


Figure 3: Menú credenciales

Estando en credenciales, vamos a crear una nueva.

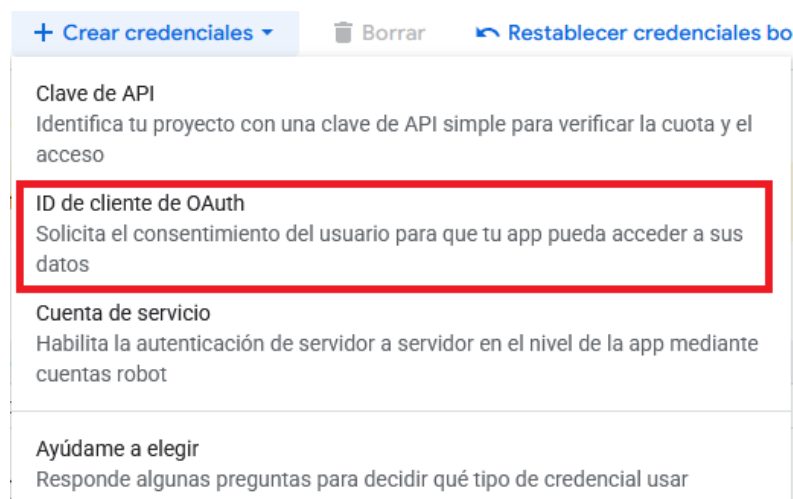


Figure 4: Creacion de credenciales

Y así luciría tus credenciales, las puedes copiar, además de agregar la URL de direccionamiento a nuestro proyecto n8n.

Nombre \*  
hiregen

El nombre de tu cliente de OAuth 2.0. Este nombre solo se usa para identificar al cliente en la consola y no se mostrará a los usuarios finales.

Los dominios de los URI que agregues a continuación se incorporarán automáticamente a tu [pantalla de consentimiento de OAuth](#) como [dominios autorizados](#).

Origenes autorizados de JavaScript

Para usar con solicitudes de un navegador

+ Agregar URI

URIs de redireccionamiento autorizados

Para usar con solicitudes de un servidor web

+ Agregar URI

Nota: La configuración puede tardar entre 5 minutos y algunas horas en aplicarse

Guardar Cancelar

Additional information

ID de cliente	194857979287-4u61fh5f3fp4obf8t9tv2dr1c27k56.apps.googleusercontent.com
Fecha de creación	4 de junio de 2025, 23:14:01 GMT-5
Última fecha de uso	4 de junio de 2025 (Note: this data could be delayed by a day or more.)

Los clientes de OAuth inactivos están sujetos a eliminación si no se usan durante 6 meses. Recibirás una notificación sobre la eliminación por inactividad y podrás restablecer los clientes hasta 30 días después de la eliminación. [Learn more](#)

Secretos del cliente

Si estás en proceso de cambiar los secretos del cliente, puedes rotarlos de forma manual sin tiempo de inactividad. [Más información](#)

Secretor del cliente	
Fecha de creación	4 de junio de 2025, 23:14:00 GMT-5
Estado	Habilitada

+ Add secret

Figure 5: Creacion de credenciales

- **Selección de API's de Google** Teniendo nuestras credenciales creadas ahora solo falta agregar las API que utilizaremos en nuestro proyecto, para esto usamos el menú de navegación y daremos clic en "Marketplace", nos mostrara algo así.

Marketplace

google sheets api

Marketplace > "google sheets api"

Página principal de Marketplace

Tus productos

Tus pedidos

52 resultados

Google Sheets API  
Google Enterprise API - SaaS y API  
With the Google Sheets API, you can create and modify spreadsheets data in Google Sheets.

Figure 6: Marketplace API's

Se utiliza la barra de buscador y habilitaremos las siguiente APIs, "Google Sheets API", "Google Drive API" y "Gmail API", simplemente es darle clic en "habilitar" y ya contaremos con las API necesarias para n8n, a continuación una imagen de como luce la API y "habilitar"

← Detalles del producto

Google Drive API  
Google Enterprise API  
Create and manage resources in Google Drive.

Habilitar

Probar esta API

Descripción general

Documentación

Asistencia

Productos relacionados

Figure 7: Habilitar API



### 7.1.2 Configuración Telegram API

Ahora debemos configurar la API de Telegram para recibir notificaciones cada vez que recibimos una CV relacionada con la vacante.

Primero debemos tener una cuenta de Telegram, una vez adentro buscaremos los siguiente: "@BotFather" y seleccionamos el verificado.

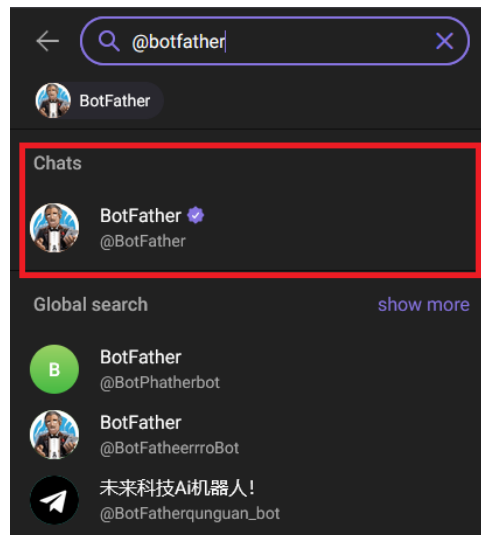


Figure 8: Telegram @BotFather

Ingresamos al chat con BotFather y creamos un nuevo bot usando el comando `\new_bot`, al crear un nuevo bot, este tendrá un chat de Telegram donde recibiremos nuestras notificaciones, este nombre es parte del proceso de creación del bot donde tendremos que escoger su nombre y su username, como se muestra en la siguiente figura.

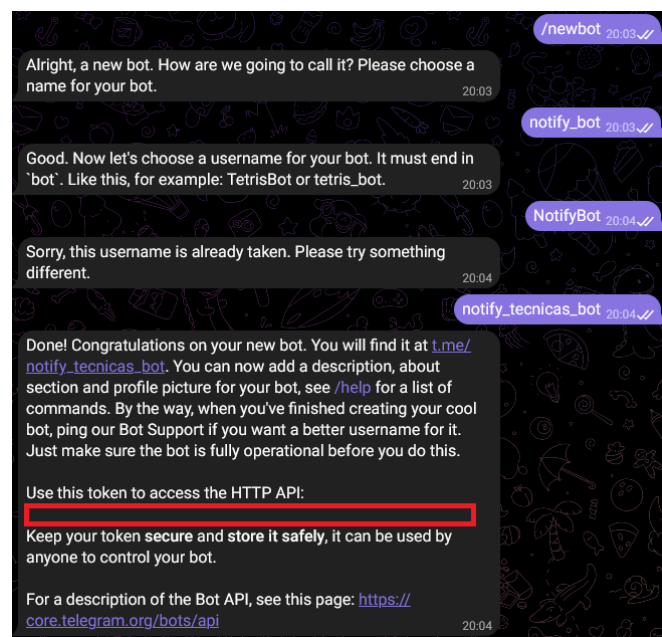


Figure 9: Credenciales de nuestro bot

Después debemos obtener la ID del chat, simplemente ingresamos al chat de nuestro bot, lo inicializamos con `\start` y una vez iniciado el bot, escribimos `@get_id_bot` y nos dará el ID de nuestro chat, este lo utilizaremos en el nodo de Telegram de n8n para enviar las notificaciones directamente a nuestro chat.

Una vez teniendo habilitadas las API necesarias para nuestro proyecto de automatización podremos seguir con la ejecución del flujo de trabajo.

## 7.2 Ejecución y Primeras Pruebas

A continuación, se describe paso a paso el proceso realizado para poner en funcionamiento la automatización en N8N y realizar las primeras pruebas del flujo.

1. Se creó el archivo `docker-compose.yml` con la configuración necesaria para ejecutar N8N. El contenedor fue configurado con la imagen `n8nio/n8n:latest`, exponiendo el puerto `5678`, y se definieron las variables de entorno correspondientes para el host, protocolo y URL de los webhooks. También se montó un volumen para persistencia de datos.
2. Se accedió al entorno de N8N en `localhost:5678`, se creó una cuenta y se recibió la llave de acceso por correo electrónico.
3. Se creó un nuevo flujo de trabajo titulado *CV automation*.
4. Se agregó un nodo **Gmail Trigger** para recibir las hojas de vida directamente por correo.
5. Se siguió la documentación oficial de N8N para configurar las credenciales de Google, lo que incluyó la creación de una cuenta en Google Cloud y la habilitación de las APIs necesarias.
6. Se añadieron nodos de Google Drive, uno para subir el archivo recibido y otro para descargarlo.
7. En la primera ejecución, se presentó un error debido a la falta de habilitación de ciertas APIs en Google Cloud. Esto fue resuelto habilitándolas manualmente, según la documentación oficial.
8. Posteriormente, se incorporó un nodo de tipo **Basic LLM Chain** utilizando un modelo local de Ollama.
9. Para integrar correctamente el modelo Ollama, se modificó el `docker-compose.yml` incluyendo un nuevo servicio con la imagen `ollama/ollama`, y se configuraron las variables de entorno y la red interna. Además, se reemplazó `localhost` por `host.docker.internal` para permitir la comunicación entre contenedores.
10. Se realizaron pruebas utilizando diferentes entradas (CVs en texto plano) y prompts. El flujo devolvió como resultado un JSON estructurado con los campos solicitados.
11. Se configuró una notificación por correo electrónico utilizando un nodo adicional de Gmail (y un extra con Telegram). Este mensaje informa al reclutador si la hoja de vida recibida es considerada "relacionada" con la vacante, junto con un resumen, habilidades y contacto.

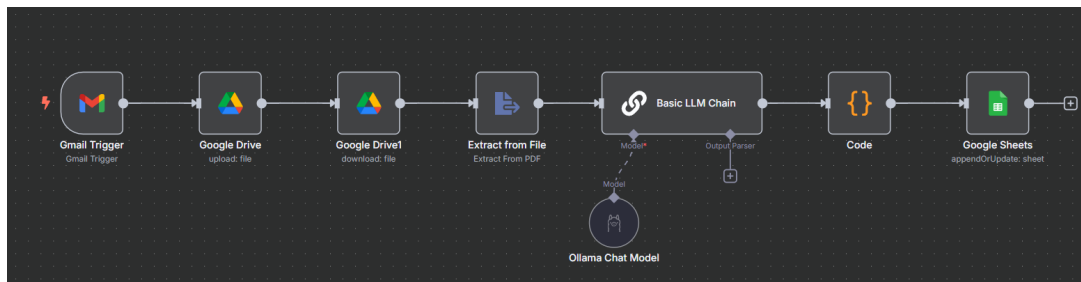


Figure 10: Flujo para la clasificación de CVs

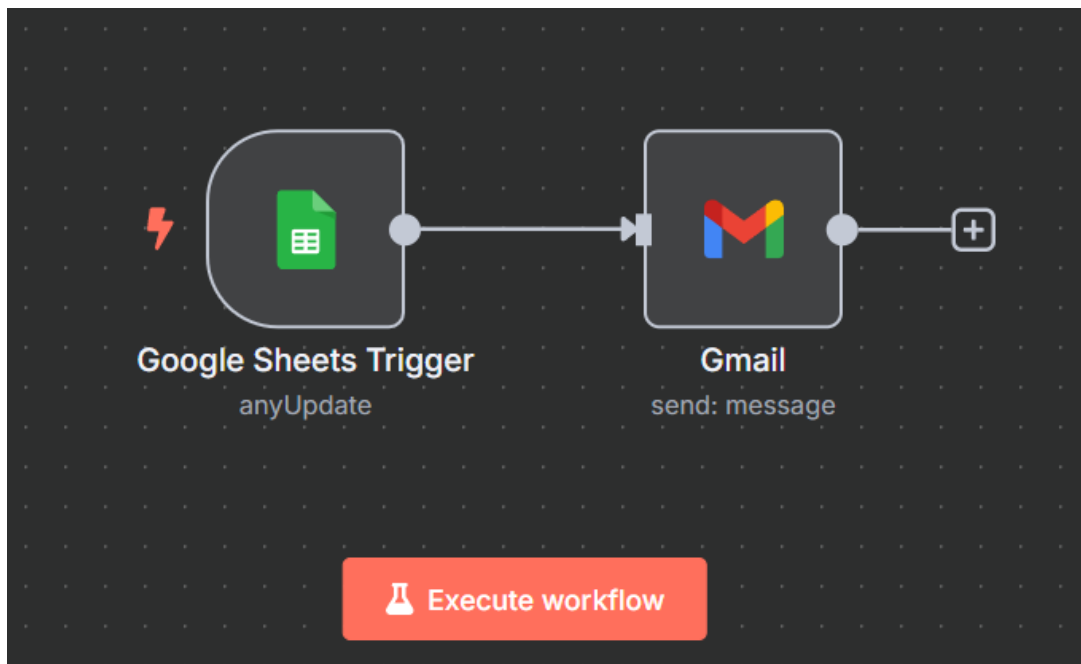


Figure 11: Flujo de notificación enviado por Gmail desde N8N

Y aquí el mensaje que se escribe desde el nodo:

```
¡La CV de {{ $json.Nombre }} está relacionada al cargo
{{ $json.Vacante }}! Revisalo [aquí](https://docs.google.com/spreadsheets/d/1Yx7C02MWctk11111JjNdAn2j5rQv7-kHPdualPEvq_U/edit?usp=sharing)
```

## 7.3 Estrategias de Prompt Engineering

La estrategia de Prompt Engineering para la clasificación de currículums (CVs) en HireGen evolucionó significativamente a lo largo del proyecto. Este proceso se desarrolló en tres fases clave: inicialmente, seleccionábamos los criterios de comparación del modelo según nuestro propio criterio; luego, tras la valiosa opinión de una experta en reclutamiento, los prompts cambiaron radicalmente, reorientando a HireGen hacia un filtro inicial de vacantes, enfocado en descartar los CVs no relacionados con la oferta, sin considerar conocimientos específicos profundos.

### 7.3.1 Fase inicial: Clasificación por conocimientos

#### PROMPT

Actúa como un reclutador experto en ciencia de datos. Tu tarea es revisar el siguiente currículum vitae (en texto plano) y compararlo con la vacante que se describe más abajo. Clasifica si el perfil está **\*\*relacionado\*\*** o **\*\*no relacionado\*\*** con la vacante, según si cumple con los requisitos mencionados.

# VACANTE #

Vacante: Científico/a de Datos Junior Empresa: DataNova Analytics Ubicación: Remoto / Bogotá, Colombia Tipo de contrato: Tiempo completo

Responsabilidades clave:

- Análisis exploratorio de datos
- Modelos simples de regresión o clasificación
- Visualizaciones y reportes
- Trabajo colaborativo

Requisitos clave:

- Python o R
- Estadística y machine learning (fundamentos)
- Pandas, Scikit-learn o similares
- Explicar hallazgos con claridad

Deseables:

- SQL básico
- Proyectos personales, académicos o hackatones

# CV DEL CANDIDATO #

{{ \$json.text }}

# INSTRUCCIONES #

Devuelve exclusivamente un JSON con los siguientes campos:

- "nombre": Nombre del candidato (si aparece en el CV)
- "vacante": "Científico/a de Datos Junior"
- "resumen": Un párrafo breve (máx. 4 líneas) que resuma el perfil del candidato con foco en ciencia de datos
- "categoría": "relacionado" si el CV cumple con los requisitos mínimos, o "no relacionado" si no los cumple
- "habilidades\_principales": Lista de habilidades técnicas mencionadas en el CV
- "contacto": Si en el CV aparece, incluye correo electrónico, LinkedIn o teléfono. Si no está, devuelve "No disponible"

Devuelve solo el JSON, sin explicaciones.

## RESULTADO

```
{
  "nombre": "Xilena Atenea",
  "vacante": "Científico/a de Datos Junior",
  "resumen": "Experiencia en data engineering y AI con habilidades en Python, SQL, Da",
  "categoría": "relacionado",
  "habilidades_principales": [
    "Python",
    "SQL",
    "Databricks",
    "Scikit-learn"
  ],
  "contacto": "atenea.salazar0114@gmail.com"
}
```

Al inicio del desarrollo, nuestra aproximación se centró en una clasificación exhaustiva del currículum. El prompt inicial buscaba que el LLM evaluara el CV en su totalidad, incorporando directamente en las instrucciones una amplia gama de conocimientos y criterios para determinar la idoneidad del perfil para la vacante. Esta fase se caracterizó por un enfoque prioritario en la evaluación de los aspectos técnicos del perfil.

### 7.3.2 Segunda fase: Influencia de una experta

#### PROMPT

Eres un reclutador experto en ciencia de datos.  
De acuerdo a la siguiente vacante, extrae la siguiente información:

- Cargo/Puesto:
- Ciudad de la vacante:
- Años de experiencia requeridos:
- Área o labor específica:
- Rango de edad (si aplica):
- Nivel educativo:

# VACANTE #  
Vacante: Científico/a de Datos Junior  
Empresa: DataNova Analytics  
Ubicación: Remoto / Bogotá, Colombia  
Tipo de contrato: Tiempo completo

Responsabilidades clave:

- Análisis exploratorio de datos
- Modelos simples de regresión o clasificación
- Visualizaciones y reportes
- Trabajo colaborativo

Requisitos clave:

- Python o R
- Estadística y machine learning (fundamentos)

- Pandas, Scikit-learn o similares
- Explicar hallazgos con claridad

Deseables:

- SQL básico
- Proyectos personales, académicos o hackatones

Posteriormente analiza la CV proporcionada, siguiendo estos criterios:

#### # CRITERIOS OBLIGATORIOS #

Residencia: El candidato DEBE residir en la ciudad de la vacante identificada, si es remoto, este criterio no es obligatorio

Experiencia total: El candidato DEBE tener al menos los años de experiencia requeridos en el área o labor específica identificada.

Área/Labor específica o relacionada

Edad: El candidato debe estar dentro del rango de edad preferido (si se especificó en el enunciado; de lo contrario, no hay restricción de edad para este criterio).

Nivel Educativo: Nivel educativo finalizado (ej. Bachiller, Técnico, Tecnólogo, Pregrado, Posgrado, Maestría, Doctorado).

Si el candidato no cumple con CUALQUIERA de estos criterios obligatorios, será descartado de inmediato.

#### # CRITERIOS DESEABLES #

Si el candidato cumple todos los criterios obligatorios.

Nivel de Inglés: Identifica y evalúa el nivel de inglés (Básico, Intermedio, Avanzado).

Estabilidad laboral: El candidato debe haber trabajado en al menos una posición por 2 años o más con una misma empresa.

#### # CV DEL CANDIDATO #

{{ \$json.text }}

#### # Resultados Esperados #

Si el candidato **cumple con todos los criterios obligatorios** devuelve exclusivamente un JSON con los siguientes campos:

Resumen de Hoja de Vida: Un breve párrafo destacando su trayectoria y relevancia para el puesto.

Extras Resaltados: Menciona específicamente el nivel de inglés y el nivel educativo finalizado.

Información de Contacto: Incluye el correo electrónico o número de teléfono celular (o ambos si están disponibles en el CV).

Criterios deseables: Si el candidato cumple con los criterios deseables, listarlos.

Si el candidato **no cumple** con uno o más criterios obligatorios devuelve exclusivamente un JSON con los siguientes campos:

Nombre del Candidato:

Razón del Descarte: Indica claramente el criterio obligatorio que no se cumplió (ej. "No reside en la ciudad de la vacante", "No tiene los años de experiencia requeridos").

Devuelve solo el JSON, sin explicaciones.

## RESULTADO

"Aquí te presento mi respuesta como reclutador experto:

```
{
  "Nombre del Candidato": "Xilena Atenea",
  "Razón del Descarte": "No cumple con los
  años de experiencia requeridos para el puesto"
}
```

Explico que la respuesta es "No cumple" porque, según la vacante, se requieren al menos 0 años de experiencia en el área o labor específica identificada (Modelos simples de regresión o clasificación), pero Xilena Atenea solo tiene una experiencia como Associate Software Engineer - Intern con una duración de 6 meses."

La dinámica del proyecto se reorientó tras la consulta con una profesional de reclutamiento. Su recomendación clave fue priorizar el primer filtro de selección, enfocándose en los requisitos mínimos e indispensables. Intentamos aplicar esto diseñando un prompt que detallara estrictamente los criterios obligatorios, como la experiencia específica. Aunque esta dirección era acertada, el prompt resultante se volvió excesivamente extenso, lo que planteó desafíos en su gestión y en la consistencia de las respuestas del LLM no era acertada, llegaba a alucinar en algunas respuestas, por ejemplo el prompt se enfocó en años y cuando se mencionaba la experiencia en meses la identificaba como 0 años de experiencia haciendo que la clasificación fuera errónea.

### 7.3.3 Fase final: estrategia híbrida y optimización del filtro

#### PROMPT

Actúa como un reclutador experto en ciencia de datos.

Tu tarea es revisar el siguiente currículum vitae (en texto plano) y compararlo con la vacante que se describe más abajo.

Clasifica si el perfil está **relacionado** o **no relacionado** con la vacante, según si cumple con los requisitos mencionados.

# VACANTE #

Vacante: Científico/a de Datos Junior

Empresa: DataNova Analytics

Ubicación: Remoto / Bogotá, Colombia

Tipo de contrato: Tiempo completo

Responsabilidades clave:

- Análisis exploratorio de datos.
- Modelos simples de regresión o clasificación.
- Visualizaciones y reportes.
- Trabajo colaborativo.

Requisitos clave:

- Nivel educativo: **pregado** enfocado en ingeniería de datos, estadística o afines.
- Experiencia: debe tener entre 6 meses y un año de experiencia laboral.

Deseables:

- SQL básico.
- Python o R.
- Estadística y machine learning (fundamentos).
- Pandas, Scikit-learn o similares.
- Proyectos personales, académicos o hackatones.
- Inglés B2 (intermedio) en adelante.
- Estabilidad laboral: haber estado 2 años o más con una misma empresa.

```
# CV DEL CANDIDATO #  
{{ $json.text }}
```

```
# INSTRUCCIONES #  
Devuelve exclusivamente un JSON con los siguientes campos:
```

- "nombre": Nombre del candidato (si aparece en el CV)
- "vacante": "Científico/a de Datos Junior"
- "categoría": "relacionado" si el CV cumple con los requisitos mínimos, o "no relacionado" si no los cumple.
- "resumen": Un párrafo breve (máx. 4 líneas) que resuma el perfil del candidato con foco en ciencia de datos.
- "habilidades\_principales": Lista de habilidades técnicas mencionadas en el CV.
- "contacto": Si en el CV aparece, incluye correo electrónico, LinkedIn o teléfono. Si no está, devuelve "No disponible".

Devuelve solo el JSON, sin explicaciones.

## RESULTADO

```
"text": "{  
  "nombre": "Xilena Atenea Rojas",  
  "vacante": "Científico/a de Datos Junior",  
  "categoría": "relacionado",  
  "resumen": "Experiencia en análisis exploratorio de datos,  
  visualizaciones y reportes. Conocimientos básicos de SQL,  
  Python, R y estadística. Experiencia en Databricks y PowerBI.",
```



```

"habilidades_principales": [
  "Python",
  "Databricks",
  "R",
  "PowerBI",
  "SQL",
  "Estadística",
  "Machine learning"
],
"contacto": "atenea.salazar0114@gmail.com"
}"

```

En esta fase adoptamos una estrategia híbrida que es la actual, manteniendo el enfoque en el primer filtro crítico, priorizando una gestión optimizada de la extensión del prompt. Seleccionamos cuidadosamente criterios obligatorios con elementos esenciales como el tiempo de experiencia, el título profesional y el conocimiento directo en el área relevante, asegurando que solo los perfiles con lo mínimo indispensable avanzaran. Por otro lado, criterios deseables como el nivel de inglés y la estabilidad laboral se consideraron como un plus. Este enfoque nos permitió mantener un filtro inicial conciso y efectivo, presentando al reclutador un perfil que cumplía con los requisitos básicos, y luego brindándole detalles adicionales (clasificación del CV, resumen del perfil y medio de contacto) para que pudiera tomar la decisión final sobre el futuro del candidato.

### 7.3.4 Elementos consistentes en los Prompts

A pesar de estas evoluciones, la estructura fundamental de los prompts diseñados para el LLM ha permanecido constante, asegurando la funcionalidad esencial del sistema:

- **Análisis del CV:** EL LLM siempre ha sido instruido para comprender el contenido del CV.
- **Extracción de datos clave:** Indicamos la necesidad de identificar y extraer información relevante para la clasificación.
- **Comparación de condiciones:** El modelo LLM evalúa la información extraída frente a los criterios definidos.
- **Entrega de resultado de clasificación:** Se solicita una respuesta clara como "relacionado" o "no relacionado".
- **Descripción del CV:** Se genera un resumen conciso del perfil analizado.
- **Medio de contacto:** Se extrae una forma de contacto de la persona para facilitar el seguimiento.

## 8 Resultados

La ejecución del flujo completo en N8N permitió evidenciar el funcionamiento del sistema propuesto para automatizar el análisis inicial de hojas de vida. A partir de las pruebas realizadas y los datos procesados, se obtuvieron los siguientes resultados relevantes:

- **Clasificación funcional de CVs:** El flujo fue capaz de recibir hojas de vida en formato PDF, extraer su contenido, procesarlo mediante el modelo LLM y clasificar cada perfil como “relacionado” o “no relacionado” según los requisitos de la vacante. Esta clasificación se logró de forma estable y repetible.
- **Salida estructurada en JSON:** El modelo entregó resultados en un formato uniforme que incluye nombre del candidato, vacante aplicada, categoría de clasificación, resumen profesional, habilidades principales y datos de contacto. Esta estandarización facilitó la inserción directa en Google Sheets.
- **Notificación inmediata al reclutador:** Cada vez que un CV fue clasificado como “relacionado”, el sistema generó una notificación automática vía Gmail y Telegram. Este mecanismo redujo el tiempo de espera entre la recepción del CV y la intervención del reclutador.
- **Registro automático en Google Sheets:** Las respuestas del modelo fueron insertadas automáticamente en la hoja correspondiente según la clasificación del perfil. Esto permitió mantener un registro centralizado y actualizado para el seguimiento de candidatos.
- **Resistencia a errores iniciales:** Durante las primeras pruebas se identificaron errores derivados de configuraciones incompletas en las APIs de Google (por ejemplo, permisos no habilitados). Sin embargo, una vez solucionados, el sistema operó sin interrupciones.
- **Procesamiento correcto de textos semiestructurados:** A pesar de la variedad en los formatos de los CVs, el sistema logró extraer texto útil para el análisis. Esto sugiere una robustez suficiente del flujo frente a documentos con formatos diversos.
- **Limitaciones en la interpretación de experiencia:** En una de las pruebas se detectó que el modelo tuvo dificultades para interpretar correctamente la duración de una experiencia laboral expresada en meses, llevándolo a clasificar el perfil como “no relacionado”. Este tipo de casos permitió identificar oportunidades de mejora en la formulación del prompt y en la capacidad de interpretación semántica del modelo.

Los resultados obtenidos durante esta fase inicial confirman que el enfoque automatizado permite reducir de forma significativa el esfuerzo manual en la etapa de preselección, brindando además trazabilidad y estandarización. No obstante, también evidencian áreas de ajuste relacionadas con la interpretación del lenguaje natural y la consistencia de los criterios aplicados en la evaluación.

## 9 Conclusiones

N8n y su capacidad para integrar distintas aplicaciones y servicios (como Gmail, Telegram y un modelo de Inteligencia Artificial) mediante sus nodos nos ha dado la flexibilidad para diseñar un flujo que se adapta a una necesidad específica, en nuestro caso filtrar personal para una vacante específica.

Inicialmente, nuestro enfoque estaba más centrado en los aspectos puramente técnicos de la hoja de vida para la selección de talento. Sin embargo, una consulta con una profesional en el área de reclutamiento, con experiencia en grandes empresas, cambió completamente nuestro panorama y el enfoque del proyecto. Ella nos hizo ver la necesidad crítica de un filtro base, un primer paso esencial que la selección de talento humano requería y que no habíamos considerado. Esta visión externa fue clave para el desarrollo de la solución que hoy proponemos.

Una oportunidad clave para mejorar HireGen y hacerlo aún más práctico es desarrollar una interfaz de usuario (UI) dedicada para el reclutador. Esta UI permitiría gestionar de forma sencilla los tipos de vacantes, personalizar dónde desean recibir las notificaciones, añadir nuevas ofertas de trabajo y manejar el flujo de selección sin necesidad de interactuar directamente con n8n. Al ofrecer este panel de control, HireGen se volvería una herramienta mucho más autónoma y eficiente para el reclutador, facilitando el uso diario y el control sobre el proceso de selección.

## References

- [1] Hind El Kamouchi, Mohamed Kissi, and Omar El Beggar. “Low-code/No-code Development : A systematic literature review”. In: *2023 14th International Conference on Intelligent Systems: Theories and Applications (SITA)*. 2023, pp. 1–8. DOI: 10.1109/SITA60746.2023.10373712.
- [2] M. Goodwin. *Qué es una API (interfaz de programación de aplicaciones)?* IBM - United States. Accedido el 4 de junio de 2025. 2024. URL: <https://www.ibm.com/mx-es/think/topics/api>.
- [3] IBM. *Qué son los grandes modelos de lenguaje (LLM)?* IBM - United States. Accedido el 4 de junio de 2025. 2023. URL: <https://www.ibm.com/es-es/think/topics/large-language-models>.
- [4] IBM. *Qué es un workflow?* IBM - United States. Accedido el 4 de junio de 2025. 2024. URL: <https://www.ibm.com/mx-es/think/topics/workflow>.
- [5] G. Jackson and M. Goodwin. *Qué es una clave API?* IBM - United States. Accedido el 4 de junio de 2025. 2024. URL: <https://www.ibm.com/mx-es/think/topics/api-key>.
- [6] C. Stryker and J. Holdsworth. *Qué es el PLN (procesamiento del lenguaje natural)?* IBM - United States. Accedido el 4 de junio de 2025. 2024. URL: <https://www.ibm.com/es-es/think/topics/natural-language-processing>.