

Guía 1 — Hello World AI con Python y la API de OpenAI

Curso: IA en el Aula — Nivel Avanzado

Profesor: Luis Daniel Benavides Navarro

Fecha: 22 de octubre de 2025

Esta guía introduce los conceptos básicos para conectarse a un modelo de lenguaje de OpenAI usando Python. El objetivo es construir un entorno funcional, comprender la estructura de la API y ejecutar un ejemplo tipo 'Hello World' con inteligencia artificial.

1. Requisitos previos

Antes de comenzar, asegúrese de cumplir los siguientes requisitos:

- Python 3.10 o superior instalado en su equipo.
- Cuenta activa en la plataforma de OpenAI.
- Conexión a Internet.
- Editor de texto o entorno como Visual Studio Code o Jupyter Notebook.

2. Creación del entorno virtual

Se recomienda crear un entorno virtual de Python para aislar las dependencias del proyecto. Ejecute los siguientes comandos en su terminal o consola:

```
# Crear carpeta de trabajo
mkdir hello_ai
cd hello_ai

# Crear entorno virtual
python -m venv venv

# Activar el entorno virtual
# En Windows:
venv\Scripts\activate
# En macOS/Linux:
source venv/bin/activate

# Instalar dependencias
pip install openai python-dotenv
```

3. Configuración de la clave API

La API de OpenAI requiere autenticación mediante una clave personal. Cree un archivo llamado ` `.env` en la carpeta del proyecto y agregue la siguiente línea:

```
OPENAI_API_KEY=su_clave_aqui
```

Puede obtener su clave iniciando sesión en <https://platform.openai.com/account/api-keys>. Nunca comparta esta clave públicamente ni la suba a repositorios.

4. Primer script: Hello World AI

Cree un archivo llamado `hello_ai.py` con el siguiente contenido:

```
import os
from openai import OpenAI
from dotenv import load_dotenv

# Cargar variables de entorno
load_dotenv()

# Inicializar cliente con la clave API
client = OpenAI(api_key=os.getenv("OPENAI_API_KEY"))

# Solicitud al modelo
prompt = "Escribe un saludo tipo 'Hello World' con un toque creativo de IA."
response = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[{"role": "user", "content": prompt}],
    temperature=0.6
)

# Mostrar respuesta
print("Respuesta del modelo:")
print(response.choices[0].message.content)
```

5. Explicación del código

A continuación se explica el propósito de cada bloque del script:

1. Importaciones: se usan las librerías os, dotenv y openai para manejar variables de entorno y conectarse a la API.
2. `load_dotenv()`: carga la clave API almacenada en el archivo `.env`.
3. Inicialización del cliente: crea una instancia de conexión con la API mediante `OpenAI(api_key=...)`.
4. `prompt`: texto de entrada enviado al modelo. En este caso, una instrucción simple tipo 'Hello World'.
5. `temperature`: controla la creatividad de la respuesta. Valores bajos (0.1–0.3) hacen salidas más predecibles; valores altos (0.7–1.0) aumentan la variabilidad.
6. `response.choices[0].message.content`: obtiene el texto generado por el modelo.

6. Ejecución del programa

Una vez configurado el entorno y creado el archivo `hello_ai.py`, ejecute el siguiente comando en la terminal:

```
python hello_ai.py
```

El modelo responderá con un mensaje tipo 'Hello World' generado por IA. Por ejemplo:
'¡Hola mundo! Hoy la inteligencia artificial también quiere saludarte.'

7. Conclusión

Ha completado su primera conexión a la API de OpenAI. En las próximas guías aprenderá a estructurar salidas en JSON, automatizar tareas docentes y crear asistentes personalizados.