

## Guía 2 — Hello World AI en Jupyter Notebook en VS Code

*Curso: IA en el Aula — Nivel Avanzado*

*Profesor: Luis Daniel Benavides Navarro*

*Fecha: 22 de octubre de 2025*

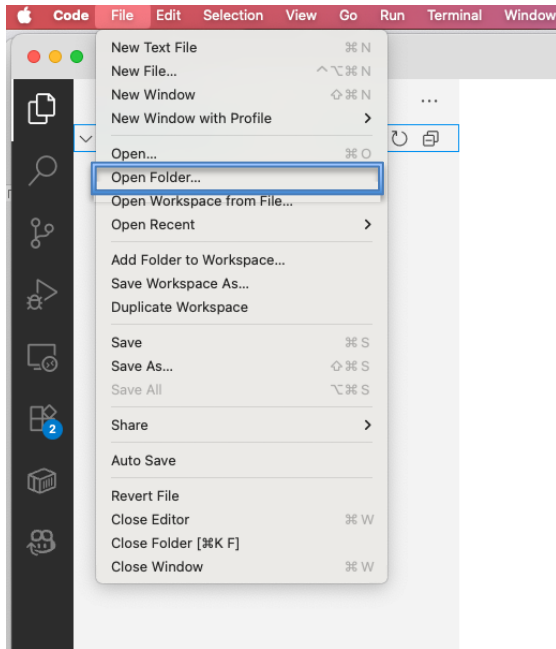
Esta guía explica cómo crear un proyecto de inteligencia artificial desde Visual Studio Code (VS Code), configurando el entorno Jupyter, conectándose a la API de OpenAI y ejecutando un ejemplo 'Hello World AI'. Todo el proceso se realiza dentro de VS Code, sin usar la terminal externa.

### 1. Requisitos previos

- Visual Studio Code instalado (versión 1.85 o superior).
- Extensiones instaladas: Python (Microsoft) y Jupyter (Microsoft).
- Python 3.10 o superior instalado en su equipo.
- Cuenta y clave API de OpenAI.
- Conexión a Internet estable.

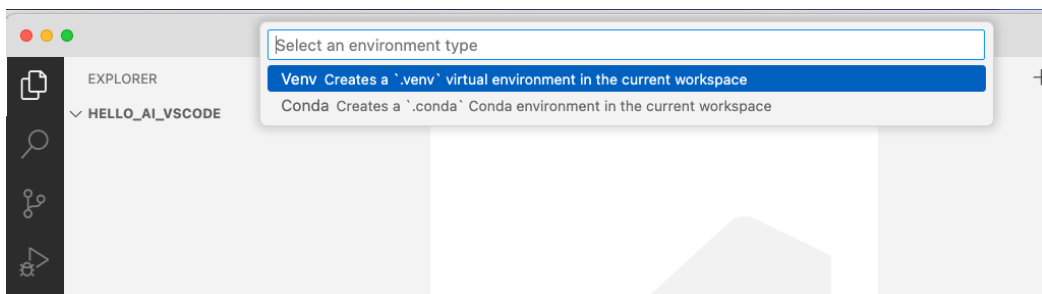
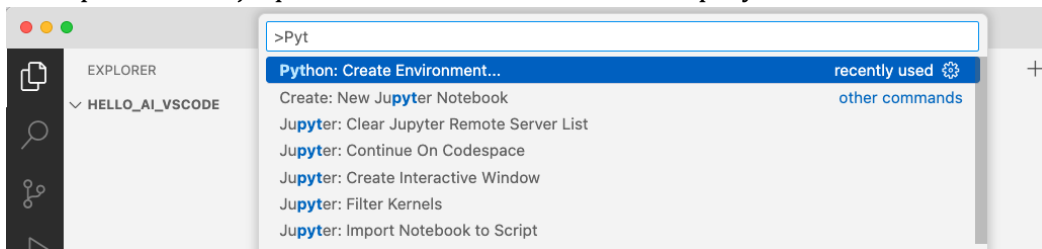
### 2. Crear carpeta del proyecto desde VS Code

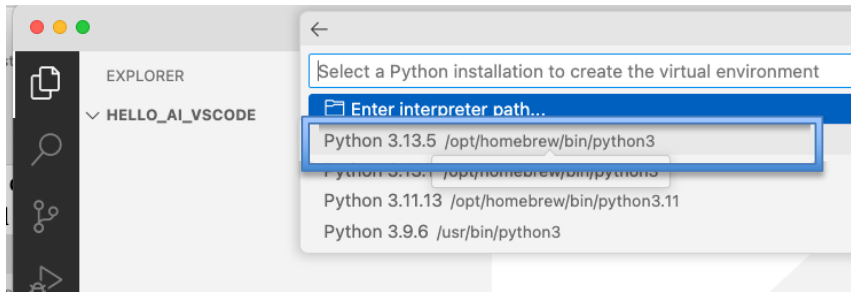
1. Abra Visual Studio Code.
2. En el menú superior, seleccione **File > Open Folder...** y elija la ubicación donde desea crear su proyecto.
3. Cree una nueva carpeta llamada **hello\_ai\_vscode** y ábrala.



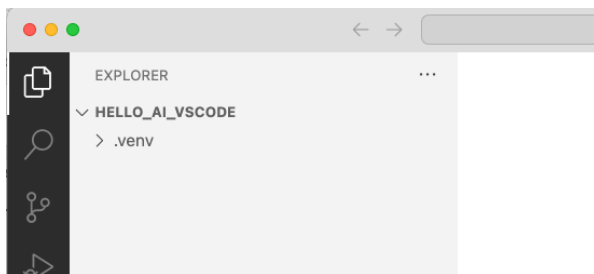
### 3. Crear y activar el entorno virtual dentro de VS Code

1. Abra el panel de comandos con **Ctrl+Shift+P** (Windows/Linux) o **Cmd+Shift+P** (macOS).
2. Escriba **Python: Create Environment** y presione **Enter**.
3. Seleccione el tipo de entorno **Venv**.
4. Elija el intérprete de Python (por ejemplo, Python 3.10).
5. Espere a que VS Code cree el entorno automáticamente.
6. Acepte el mensaje que ofrece activar el entorno en el proyecto.





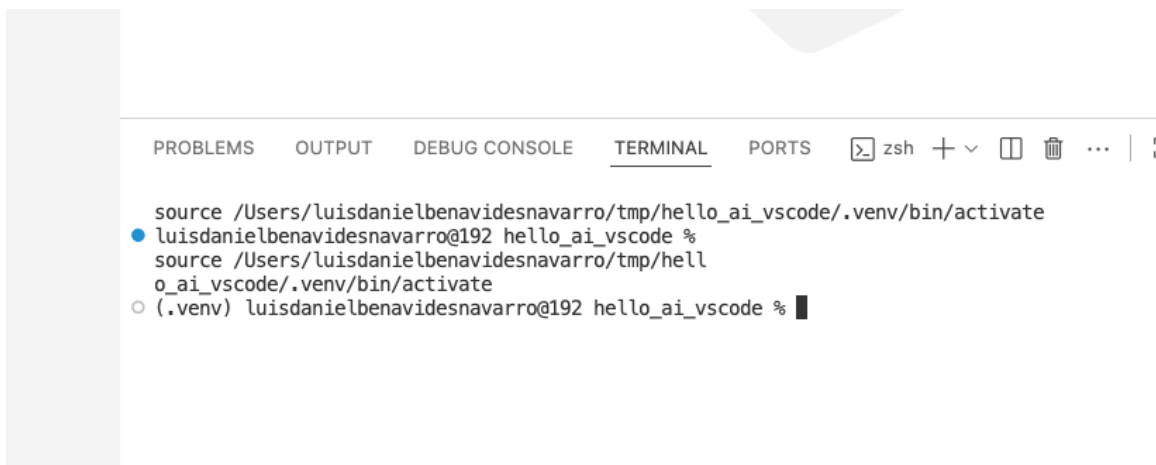
Una vez creado el entorno, VS Code mostrará en la esquina inferior derecha un mensaje confirmando que el entorno se activó correctamente.



#### 4. Instalar dependencias desde el terminal integrado

Abra el terminal integrado con **Ctrl+Ñ** o desde **View > Terminal** y ejecute los siguientes comandos:

```
pip install openai python-dotenv jupyter ipykernel
python -m ipykernel install --user --name hello_ai_vscope
```

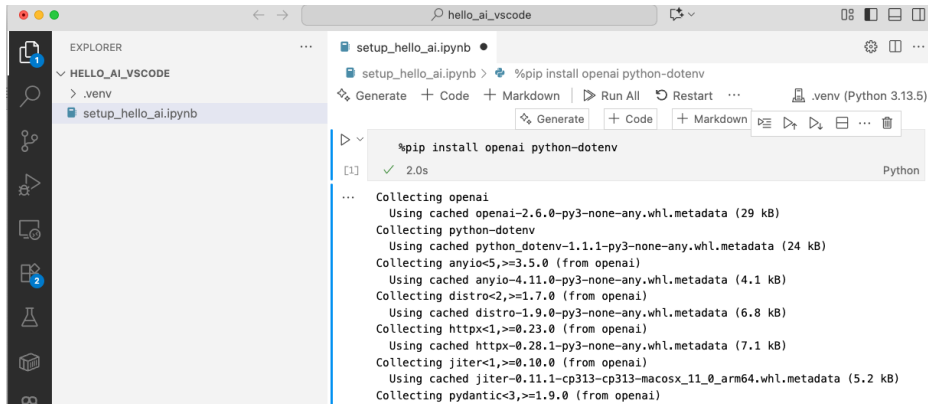


#### 4a. Crear un archivo Jupyter Notebook para instalar y configurar dependencias

En lugar de usar la terminal, puede instalar todas las dependencias directamente desde un archivo Jupyter Notebook dentro de VS Code.

1. Presione **Ctrl+Shift+P** y seleccione **Jupyter: Create New Jupyter Notebook**.
2. Guarde el archivo como **setup\_hello\_ai.ipynb** dentro de la carpeta del proyecto.
3. En la parte superior derecha, seleccione el kernel correspondiente al entorno que creó (**hello\_ai\_vscode**).
4. Cree una primera celda con los siguientes comandos para instalar las librerías necesarias:

```
%pip install openai python-dotenv
```



VSCode le pedirá instalar las librerías de jupyter en su entorno “.venv”, autorice la instalación.

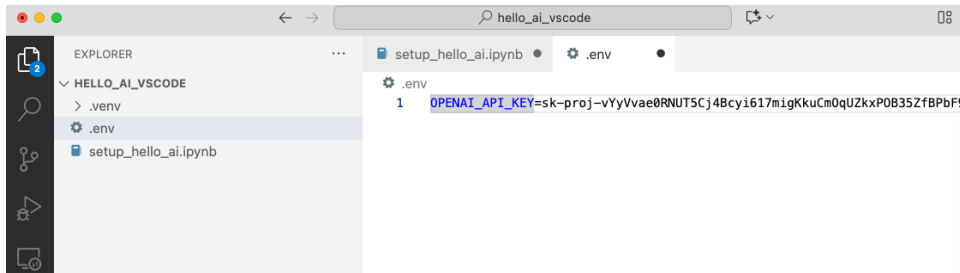
Una vez completada la instalación, el entorno quedará configurado y podrá ejecutar celdas de código Python dentro del mismo notebook.

## 5. Configurar la clave API (.env)

1. En el explorador lateral de VS Code, haga clic derecho en la carpeta del proyecto y seleccione **New File**.
2. Nombre el archivo como **.env**.
3. Agregue la siguiente línea (reemplace con su propia clave):

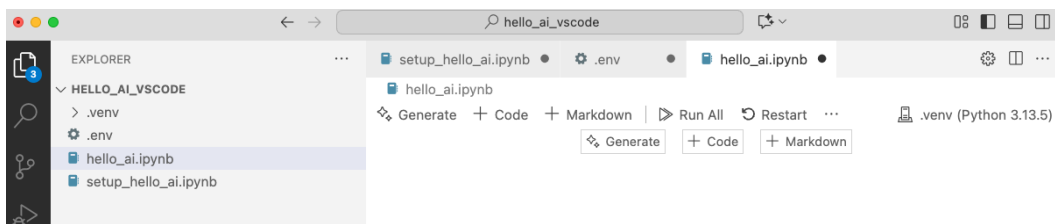
```
OPENAI_API_KEY=su_clave_aqui
```

4. Guarde el archivo. VS Code cargará automáticamente las variables del **.env** en su entorno cuando ejecute código.
5. No suba este archivo a GitHub ni lo comparta públicamente.



## 6. Crear y configurar el Notebook en VS Code

1. Presione **Ctrl+Shift+P** y ejecute el comando **Jupyter: Create New Jupyter Notebook**.
2. Guarde el archivo como **hello\_ai.ipynb** dentro de la carpeta del proyecto.
3. En la parte superior derecha, seleccione el kernel correspondiente al entorno creado (**hello\_ai\_vscode**).
4. Verifique que la barra inferior muestre el entorno activo.

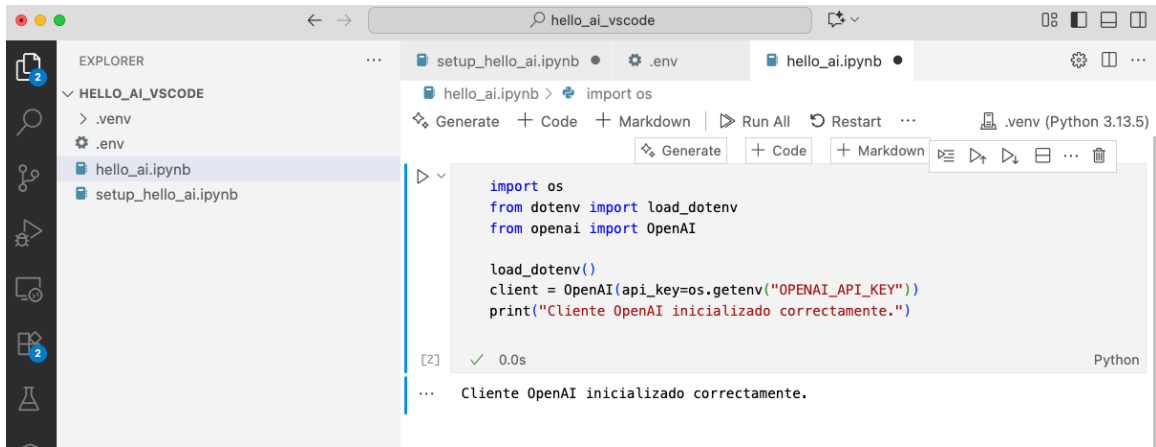


## 7. Celdas principales del notebook

### Celda 1 — Importación y configuración del entorno

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

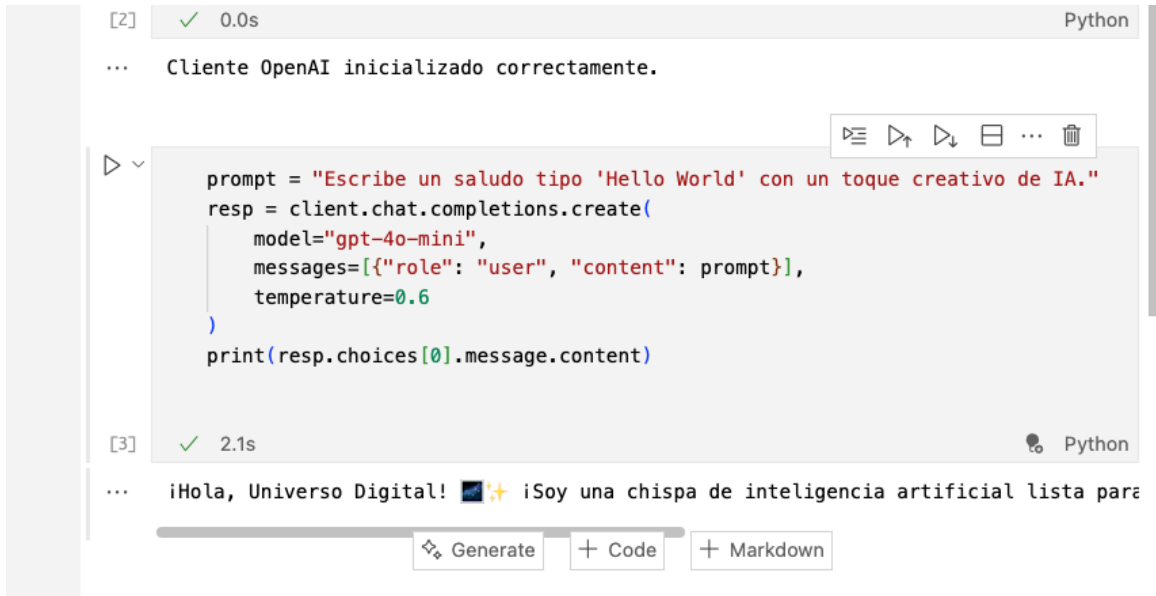
load_dotenv()
client = OpenAI(api_key=os.getenv("OPENAI_API_KEY"))
print("Cliente OpenAI inicializado correctamente.")
```



## Celda 2 — Ejemplo Hello World AI

```
prompt = "Escribe un saludo tipo 'Hello World' con un toque creativo de IA."
resp = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[{"role": "user", "content": prompt}],
    temperature=0.6
)
print(resp.choices[0].message.content)
```

Ejecute la celda (Ctrl+Enter o Shift+Enter). El modelo responderá con un mensaje generado por IA.



## 8. Parámetros principales de la API

- **\*\*model:\*\*** identifica el modelo usado (por ejemplo, gpt-4o-mini).

- **messages:** estructura del diálogo; cada mensaje tiene un rol ('system', 'user', 'assistant').
- **temperature:** controla la creatividad de las respuestas. Valores bajos (0.1–0.3) dan salidas consistentes; altos (0.7–1.0) más originales.
- **max\_tokens:** define el número máximo de tokens (palabras/piezas de texto) generados en la respuesta.
- **top\_p:** alternativa a temperature; controla la diversidad mediante muestreo probabilístico.

## 9. Ejercicio adicional

Cree una nueva celda y pruebe distintas configuraciones de `temperature` y `max\_tokens`. Observe cómo cambia la respuesta del modelo.

```
for t in [0.1, 0.5, 0.9]:
    response = client.chat.completions.create(
        model="gpt-4o-mini",
        messages=[{"role": "user", "content": "Describe brevemente qué es la IA."}],
        temperature=t,
        max_tokens=50
    )
    print(f"--- temperature={t} ---")
    print(response.choices[0].message.content)
```



The image shows a VS Code Jupyter Notebook interface. The top part is a code editor with a Python script. The script uses the OpenAI API to generate text for different temperature values. The bottom part shows the output of the script, which displays the generated text for each temperature value. The output is truncated on the right side.

```
for t in [0.1, 0.5, 0.9]:
    response = client.chat.completions.create(
        model="gpt-4o-mini",
        messages=[{"role": "user", "content": "Describe brevemente qué es la IA"}],
        temperature=t,
        max_tokens=150
    )
    print(f"--- temperature={t} ---")
    print(response.choices[0].message.content)
```

[5] ✓ 7.4s Python

... --- temperature=0.1 ---  
La inteligencia artificial (IA) es un campo de la informática que se centra en la c  
--- temperature=0.5 ---  
La inteligencia artificial (IA) es un campo de la informática que se centra en la c  
--- temperature=0.9 ---  
La inteligencia artificial (IA) es un campo de la informática que se centra en crea

## 10. Cierre y próximos pasos

Ha completado la configuración del entorno Jupyter dentro de VS Code y ejecutado su primer programa de IA. En la siguiente guía aprenderá a crear salidas estructuradas en JSON y a usar la API para automatizar tareas docentes.