

# Tutorial de instalación y configuración para el proyecto ML

Visual Studio + Anaconda

## Requisitos mínimos para Anaconda

- **Procesador:** Un procesador de 64 bits de doble núcleo a 1.6 GHz o superior.
- **Memoria RAM:** 8 GB de RAM (se recomiendan 16 GB o más para proyectos grandes).
- **Almacenamiento:** 20 GB de espacio libre en disco (se recomiendan 50 GB o más si vas a trabajar con conjuntos de datos grandes).
- **Sistema operativo:** Windows 10 u 11 (64 bits), macOS 10.15 o posterior, o una distribución de Linux compatible.

## Recomendaciones adicionales

- **Procesador:** Se recomienda un procesador de cuatro núcleos o más para un rendimiento óptimo, especialmente si vas a trabajar con modelos de Machine Learning complejos.
- **Memoria RAM:** Si vas a trabajar con conjuntos de datos grandes o modelos que requieren mucha memoria, 16 GB de RAM o más son esenciales.
- **Almacenamiento:** Un disco de estado sólido (SSD) puede mejorar significativamente los tiempos de carga y procesamiento de datos en comparación con un disco duro tradicional (HDD).
- **Tarjeta gráfica (GPU):** Si vas a trabajar con Deep Learning, una tarjeta gráfica dedicada con soporte para CUDA (NVIDIA) o ROCm (AMD) puede acelerar el entrenamiento de modelos.

## Consideraciones adicionales

- **Uso de entornos virtuales:** Es altamente recomendable utilizar entornos virtuales en Anaconda para aislar las dependencias de cada proyecto y evitar conflictos entre librerías.
- **Actualización de paquetes:** Mantén tus paquetes de Anaconda actualizados para asegurar el mejor rendimiento y compatibilidad.
- **Optimización del código:** Escribe código eficiente y optimizado para evitar el uso innecesario de recursos.

## En resumen

Si tu computadora cumple con los requisitos mínimos y sigues las recomendaciones adicionales, Anaconda debería funcionar de manera fluida para la mayoría de los proyectos de Machine Learning. Sin embargo, si vas a trabajar con proyectos muy grandes o complejos, es posible que necesites un equipo más potente.

Para trabajar proyectos de Machine Learning en Python usando Visual Studio Code (VS Code), te recomiendo las siguientes configuraciones y extensiones:

## 1. Descarga e instalación de Anaconda

- Ve al sitio web de Anaconda: <https://docs.anaconda.com/anacondaorg/>
- Elige la versión de Anaconda que corresponda a tu sistema operativo (Windows, macOS o Linux).
- Descarga el instalador.
- Ejecuta el instalador descargado.
- Sigue las instrucciones en pantalla. Generalmente, deberás aceptar los términos de licencia y elegir la ubicación de instalación.
- **Importante:** Durante la instalación, asegúrate de marcar la opción para añadir Anaconda a tu variable de entorno PATH. Esto te permitirá ejecutar comandos de Anaconda desde la línea de comandos.

Para computadores con menores recursos (menos de 8GB de RAM) usar **Miniconda** en vez de Anaconda.

## Qué es Miniconda?

Miniconda es un instalador mínimo de Conda. A diferencia de Anaconda, que instala una gran cantidad de paquetes predefinidos, Miniconda solo instala Conda, Python y las dependencias básicas. Esto lo hace más ligero y rápido de instalar, ideal si solo necesitas Conda para gestionar tus entornos y paquetes.

## Pasos para instalar Miniconda en Windows

### 1. Descarga Miniconda:

- Ve al sitio web de Miniconda:  
<https://www.anaconda.com/docs/getting-started/miniconda/install>

- Elige la versión de Miniconda que corresponda a tu sistema operativo (Windows) y a la versión de Python que desees (por ejemplo, Python 3.9).
- Descarga el instalador.

## 2. Ejecuta el instalador:

- Ejecuta el archivo instalador descargado.
- Sigue las instrucciones en pantalla. Generalmente, deberás aceptar los términos de licencia y elegir la ubicación de instalación.
- **Importante:** Durante la instalación, asegúrate de marcar la opción para añadir Miniconda a tu variable de entorno PATH. Esto te permitirá ejecutar comandos de Conda desde la línea de comandos.

## 3. Verifica la instalación:

- Abre una nueva ventana de línea de comandos (símbolo del sistema o PowerShell).
- Escribe el siguiente comando y presiona Enter:

```
Bash
conda --version
```

## 4.

- Si Miniconda se instaló correctamente, mostrará la versión de Conda.

## Ventajas de usar Miniconda

- **Ligero:** La instalación es más pequeña y rápida que Anaconda.
- **Personalizable:** Solo instalas los paquetes que necesitas.
- **Control:** Tienes más control sobre tu entorno.

## Desventajas de usar Miniconda

- **Menos paquetes preinstalados:** Tendrás que instalar manualmente las librerías que necesites (NumPy, pandas, scikit-learn, etc.).
- **Requiere más configuración:** Puede requerir un poco más de configuración inicial.

## En resumen

Si prefieres tener un control más preciso sobre los paquetes que instalas y no necesitas la gran cantidad de librerías que vienen con Anaconda, Miniconda es una excelente opción. Te permite tener Conda y Python en tu sistema de manera más ligera y personalizada.

## 2. Verificación de la instalación

- Abre una nueva ventana de línea de comandos (terminal en macOS o Linux).
- Escribe el siguiente comando y presiona Enter:

Bash

conda --version

- Si Anaconda se instaló correctamente, mostrará la versión de conda.

### 3. Instalación de Python (si no se instaló con Anaconda)

- Si por alguna razón no tienes Python instalado, puedes descargarlo desde <https://www.python.org/>

**Versiones de python recomendadas: 3.9 en adelante.**

### Por qué elegir Python 3.13.2 para Machine Learning?

- **Estabilidad:** Es una versión estable, lo que significa que ha sido probada y corregida para minimizar errores y asegurar un funcionamiento confiable.
- **Soporte:** Cuenta con soporte completo de la comunidad de Python y de las principales librerías de Machine Learning.
- **Actualizaciones:** Incluye mejoras de rendimiento, seguridad y nuevas características que pueden ser útiles para tus proyectos.
- **Compatibilidad:** Es compatible con la mayoría de las librerías y herramientas de Machine Learning más populares, como TensorFlow, PyTorch, scikit-learn, pandas y NumPy.

### ¿Dónde descargar Python 3.13.2?

Puedes descargar la versión 3.13.2 de Python desde el sitio web oficial:

- <https://www.python.org/downloads/>

### 4. Instalación de Visual Studio Code

- Descarga e instala VS Code desde <https://code.visualstudio.com/>

### 5. Extensiones esenciales

- **Python (Microsoft):** Esta extensión proporciona soporte completo para Python en VS Code, incluyendo IntelliSense, depuración, formateo de código, etc.
- **Pylance (Microsoft):** Esta extensión mejora el rendimiento y la precisión de IntelliSense para Python.
- **Jupyter (Microsoft):** Esta extensión permite trabajar con notebooks de Jupyter (.ipynb) dentro de VS Code.
  - Para instalar estas extensiones, abre VS Code, ve a la pestaña de extensiones (Ctrl+Shift+X) y busca cada una por su nombre.

## 6. Configuración del entorno virtual (recomendado)

- Es altamente recomendable crear un entorno virtual para cada proyecto de Machine Learning. Esto aísla las dependencias del proyecto y evita conflictos entre librerías.
- Puedes crear un entorno virtual usando **venv** (incorporado en Python 3) o **conda**.

### venv:

Bash  
`python3 -m venv .venv # Crea el entorno virtual`  
`source .venv/bin/activate # Activa el entorno virtual (Linux/macOS)`  
`.venv\Scripts\activate # Activa el entorno virtual (Windows)`

○

### conda:

Bash  
`conda create --name mi_entorno python=3.9 # Crea el entorno virtual`  
`conda activate mi_entorno # Activa el entorno virtual`

○

Dentro del entorno virtual activado, instala las librerías necesarias:

Bash  
`pip install numpy pandas matplotlib scikit-learn tensorflow # Librerías básicas`

•

## 7. Configuración de VS Code

- Abre la configuración de VS Code (Archivo > Preferencias > Configuración o Ctrl+,).
- Busca las siguientes opciones y configúralas según tus preferencias:
  - `"python.defaultInterpreterPath"`: Ruta al intérprete de Python dentro de tu entorno virtual.
  - `"python.formatting.provider"`: Elige un formateador de código (e.g., autopep8, yapf).
  - `"editor.formatOnSave"`: Formatea el código al guardar el archivo.
  - `"files.autoSave"`: Guarda automáticamente los archivos.

## 8. Uso de VS Code para Machine Learning

- Abre la carpeta de tu proyecto en VS Code.
- Crea un nuevo archivo Python (.py) o un notebook de Jupyter (.ipynb).
- Escribe tu código de Machine Learning utilizando las librerías instaladas.
- Ejecuta el código directamente en VS Code o en el notebook de Jupyter.
- Utiliza las herramientas de depuración de VS Code para identificar y corregir errores.

## Consejos adicionales

- **Explora las extensiones:** Hay muchas otras extensiones útiles para VS Code, como GitLens para trabajar con Git, o Bracket Pair Colorizer para resaltar los paréntesis en el código.
- **Aprende los atajos de teclado:** Los atajos de teclado te ayudarán a ser más productivo en VS Code.
- **Personaliza la apariencia:** Puedes personalizar la apariencia de VS Code con temas y configuraciones de color.