Preparación e instalación de herramientas de ciencia de datos

Gabriel Rengifo

Recomendaciones de hardware

Duración estimada de la instalación: 60–90 minutos (dependiendo del sistema operativo y privilegios de usuario).

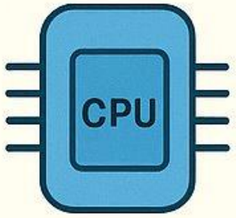
Requisitos mínimos de hardware:

•CPU moderna (2+ cores). 8+ GB RAM recomendado para trabajar con datasets medianos.
•Espacio en disco: 10 GB libre mínimo.
Sistemas operativos: Windows 10/11, macOS (10.15+), Linux (Ubuntu/Debian/CentOS).

Nota sobre seguridad: si trabajas con datos sensibles de la Armada, **no** uses Google Colab para esos datos; usa entornos locales con cifrado y repositorios privados en GitLab/GitHub Enterprise.

CPU vs GPU vs NPU vs TPU
The Real Differences for AI/ML

CPU

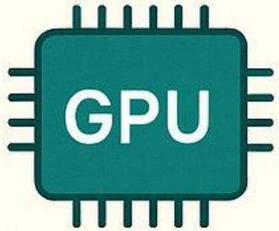


The classic processor in every computer, CPUs can run any sor-ware, plurring Al models, but are slower for de-ep learning due to fewer parallel cores.

Best for:

- Traditional machine learning (sckit-learm XGBoost)
- General-pmall models or prototypes

GPU

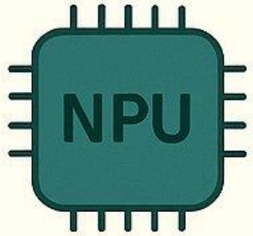


GPUs are built for parallel processing. They are the backbone of modern deep learning, perfect for train-ing and inference of mo-dels like CNNs, RNNs. and transformers (GPT, BERT)

Best for

- Training and running large deep learning models
- Flexible for many Al workloads

NPU

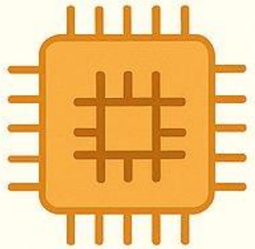


NPUs are specialized chips designed only for neural network operations. often embedded in smart-phones and loT devices They run efficient models for vision, speech, and edge

Best for

- On-device, real-time Al (face unlock language translation)
- Lightweight - fly Al in mobile and loT

TPU



TPUs are Google's custom Al accelerators, tuned for Tensor Flow and mass-ssive neural networks. lld for training and deploying large models at cloud scale

Best for

- Scalable deep learning in Google Cloud
- Training and inference for big models (BERT. GPT-2, EfficientNet)

Hoja de Ruta

Objetivo de la Sesión

1

Instalación y Configuración

Herramientas base para ciencia de datos.

2

Validación del Ecosistema

Python y librerías esenciales en funcionamiento.

3

Introducción a Git

Buenas prácticas para trabajo colaborativo.

El Corazón de tu Proyecto

El Ecosistema del Científico de Datos

1

1. Recolección

Obtener los datos necesarios.

2

2. Limpieza y Preparación

Transformar datos crudos en utilizables.

3

3. Análisis y Modelado

Construir y entrenar modelos.

4

4. Evaluación

Medir el rendimiento del modelo.

5

5. Despliegue

Poner el modelo en producción.

> 💡 Sin un entorno bien configurado, este ciclo virtuoso se rompe y los proyectos pueden estancarse.

Tu Caja de Herramientas

Herramientas Esenciales

Core

- **Python 3.x:** El lenguaje base para todo desarrollo en ML y PLN.
- **Anaconda / Miniconda:** Gestores de entornos para aislar proyectos y evitar conflictos de dependencias.
- **Google Colab:** Una alternativa poderosa en la nube, ideal para experimentación rápida y uso de hardware avanzado.

Entornos de Desarrollo

- **Jupyter Notebook / VSCode:** Tus espacios interactivos para codificar, documentar y visualizar resultados.

Librerías Clave

- **Numpy:** Computación numérica de alto rendimiento.
- **Pandas:** Manipulación y análisis de datos.
- **Matplotlib / Seaborn:** Visualización de datos.
- **Scikit-learn:** Algoritmos de Machine Learning listos para usar.

Manos a la Obra

Instalación Paso a Paso (Usando Conda)

- **1. Instalar Anaconda/Miniconda:** Descarga la versión adecuada para tu sistema operativo desde su sitio oficial.
- **2. Crear un Entorno Virtual:** Utiliza los siguientes comandos en tu terminal. Esto crea un espacio aislado para tu proyecto.

```
conda create -n diplomado python=3.10conda activate diplomado
```

- **3. Instalar Librerías Esenciales:** Ya dentro de tu entorno, instala las herramientas que usarás.

```
pip install numpy pandas matplotlib seaborn scikit-learn jupyter
```

- **4. Validar la Instalación:** Abre un intérprete de Python o un Jupyter Notebook y ejecuta:

```
import pandas as pdprint(pd.__version__)
```

Este proceso garantiza un entorno limpio y reproducible para todos tus proyectos de Machine Learning.

Sin Instalación, ¡Directo a la Nube!

Alternativa: Google Colab

- **Cero Instalación:** Acceso instantáneo a un entorno de desarrollo Python en la nube, sin configuraciones locales complejas.
- **Hardware Potente:** Permite el uso de GPUs y TPUs gratuitas, ideal para entrenar modelos complejos de Machine Learning y PLN.
- **Integración con Google Drive:** Facilita la carga y descarga de datasets, así como el guardado de tus notebooks y resultados.

⊗ **¡Atención!** No utilizar datos sensibles ni clasificados de la Armada en Google Colab, ya que es un servicio externo. Para estos casos, siempre usar entornos locales o plataformas aprobadas.

Tu Primer Logro

Notebook “Mi Primer Análisis Naval”

Objetivo:

Realizar un análisis exploratorio simple de un conjunto de datos y visualizarlo, demostrando que tu ecosistema está funcionando.

```
from sklearn.datasets
import load_irisimport pandas as pd
import matplotlib.pyplot as plt
# Cargar el dataset de Iris (ejemplo clásico)
iris = load_iris(as_frame=True)df = iris.frame
# Mostrar las primeras filas y estadísticas descriptivasprint("Primeras 5 filas del dataset:")print(df.head())print("\nEstadísticas descriptivas:")print(df.describe())# Generar un gráfico simpleplt.figure(figsize=(8, 6))plt.scatter(df['sepal length (cm)'], df['sepal width (cm)'], c=df['target'], cmap='viridis')plt.xlabel('Longitud del Sépalo (cm)')plt.ylabel('Ancho del Sépalo (cm)')plt.title('Diagrama de Dispersión de Longitud vs. Ancho del Sépalo')plt.colorbar(label='Especie de Iris')plt.grid(True)plt.show()
```

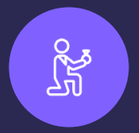
👉 Este ejercicio valida tu instalación y te da confianza para proyectos más complejos.

Claves del Éxito

Buenas Prácticas en Ciencia de Datos

Usar Entornos Virtuales

Aíslan las dependencias de cada proyecto, evitando conflictos y facilitando la reproducibilidad.



Documentar Notebooks

Un código claro y bien documentado es fundamental para la comprensión y el mantenimiento futuro.

Versionar con Git

Permite un control de cambios exhaustivo y facilita la colaboración en equipo.

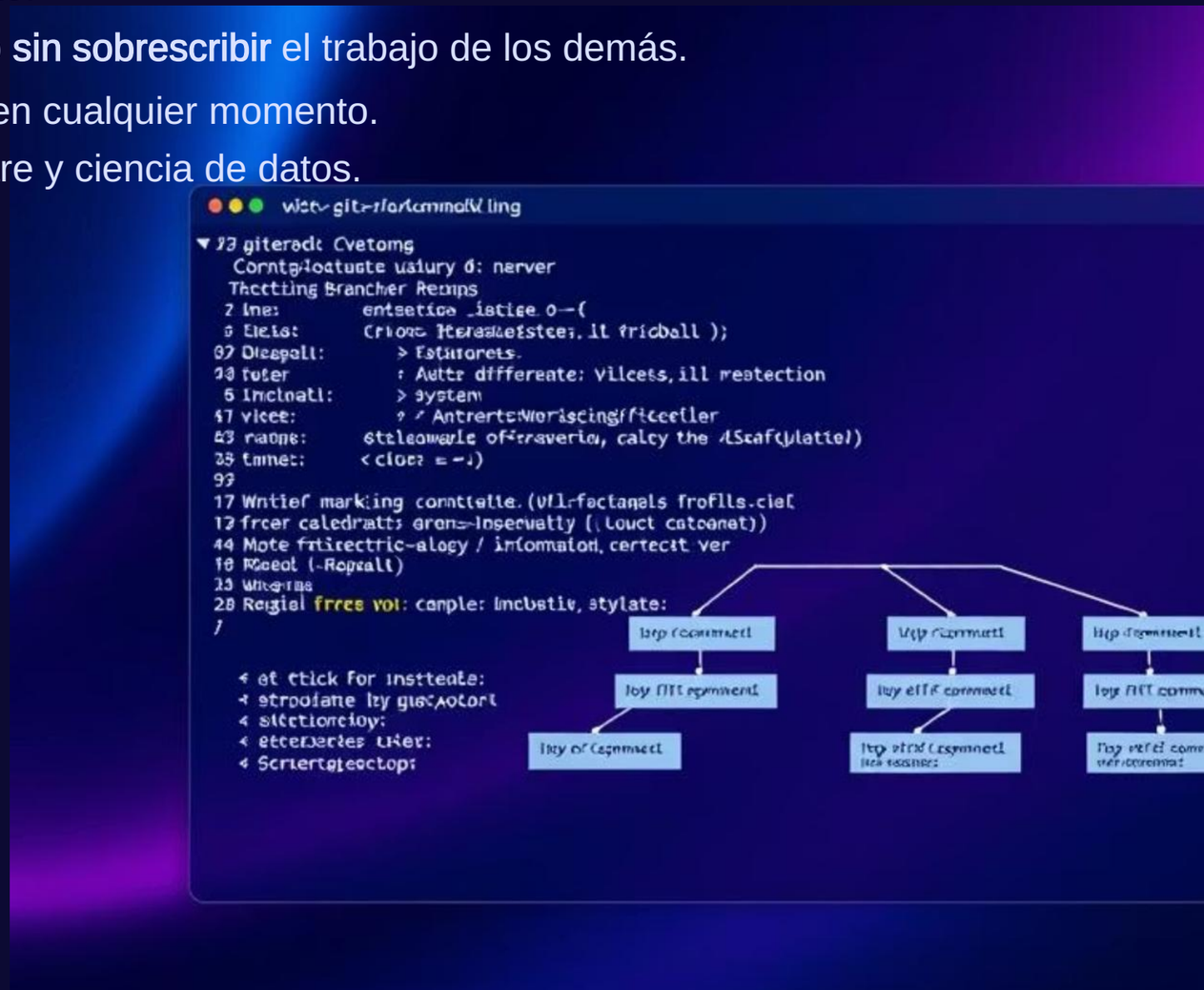
Seguridad y Confidencialidad

Manejar los datos con la máxima responsabilidad, especialmente en entornos críticos como la Armada.

Tu Aliado Colaborativo
Introducción a Git

¿Qué es Git?

- Un **Sistema de Control de Versiones** distribuido y de código abierto.
- Permite a múltiples desarrolladores trabajar en el mismo proyecto **sin sobrescribir** el trabajo de los demás.
- Registra cada cambio, permitiendo **volver a versiones anteriores** en cualquier momento.
- Esencial para la **trazabilidad y la auditoría** de proyectos de software y ciencia de datos.



Comandos Básicos de Git

- **git init:** Inicializa un nuevo repositorio Git en tu directorio actual.
- **git add .:** Agrega todos los archivos nuevos o modificados al área de preparación.
- **git commit -m "Mensaje":** Guarda los cambios preparados en el historial del repositorio con un mensaje descriptivo.
- **git status:** Muestra el estado de tu directorio de trabajo y el área de preparación.

Git en la Armada

Uso de GitHub/GitLab para Proyectos Institucionales

- **Repositorios Privados:** Imprescindibles para la seguridad y confidencialidad de los proyectos militares.
- **Centralización:** Unifica el desarrollo y permite la gestión de accesos.
- **Auditoría y Trazabilidad:** Cada cambio es registrado, facilitando la revisión y el cumplimiento de normativas.

Flujo Simple de Trabajo

1. Crear un repositorio privado en la plataforma institucional.
2. Clonar el repositorio a tu máquina local.
3. Subir tu notebook de prueba y los cambios realizados.

Actividad Práctica

- Valida tu entorno instalando las librerías y corriendo el notebook de ejemplo.
- Crea un repositorio local con Git.
- Sube un archivo `README.md` con el mensaje: "Mi ecosistema de datos está listo ".