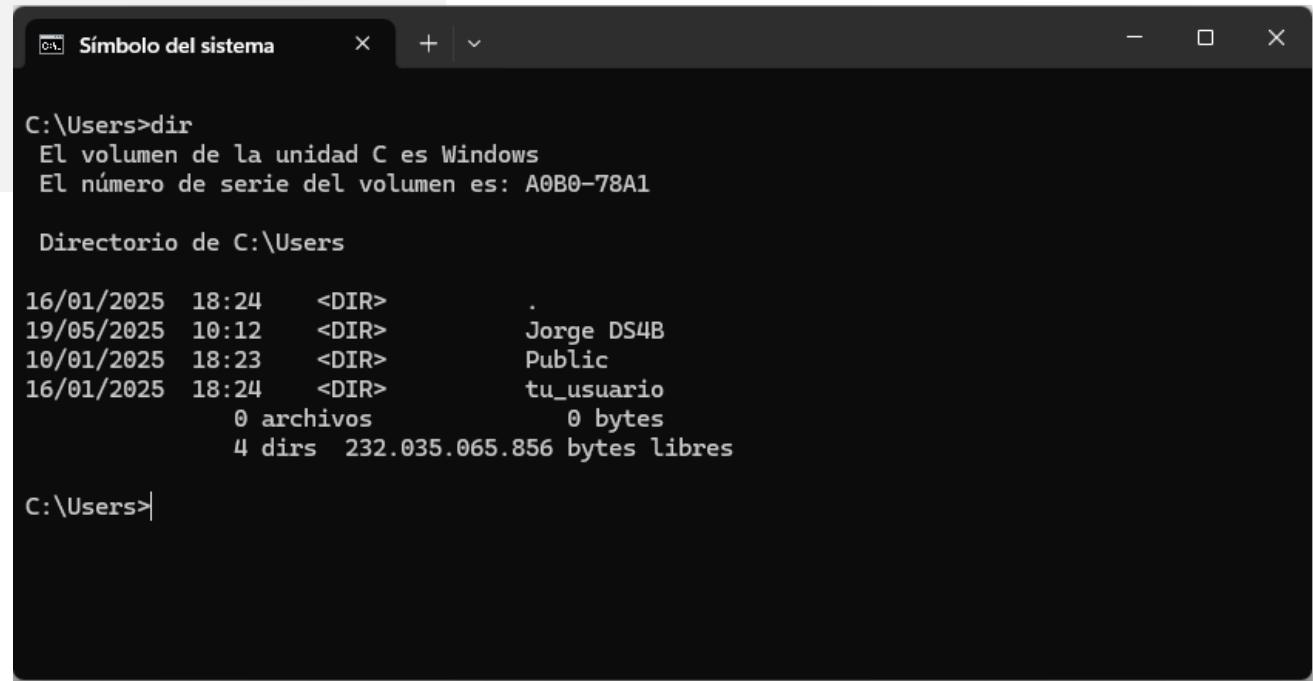


¿Para qué nos puede servir?

- **Instalación y gestión de librerías y entornos.**
- **Organización de proyectos - Estructuras de carpetas.**
- **Ejecución de scripts.**
- **Lanzar apps en local.**
- **Trabajar con Git**



```
C:\Users>dir
El volumen de la unidad C es Windows
El número de serie del volumen es: A0B0-78A1

Directorio de C:\Users

16/01/2025 18:24    <DIR>      .
19/05/2025 10:12    <DIR>      Jorge DS4B
10/01/2025 18:23    <DIR>      Public
16/01/2025 18:24    <DIR>      tu_usuario
                           0 archivos           0 bytes
                           4 dirs   232.035.065.856 bytes libres

C:\Users>
```

Inconveniente:

Existen diferentes tipos de terminal, en función del sistema operativo y cada una con comandos diferentes.



PowerShell (Windows):

Acceso rápido: (Win + R) y escribir powershell

Terminal avanzada de Windows con sintaxis orientada a objetos, ideal para automatización y scripting profesional.



Símbolo del sistema - CMD (Windows)

Acceso rápido: (Win + R) y escribir CMD

Terminal clásica con comandos básicos, útil para tareas sencillas de archivos y navegación.



Terminal (macOS/Linux)

Acceso rápido: (Cmd + espacio) y escribir terminal

Consola por defecto en sistemas basados en Unix, como macOS y Linux, muy utilizada para desarrollo, automatización y manejo del sistema.

1. Instalación y gestión de librerías

- Crear entornos

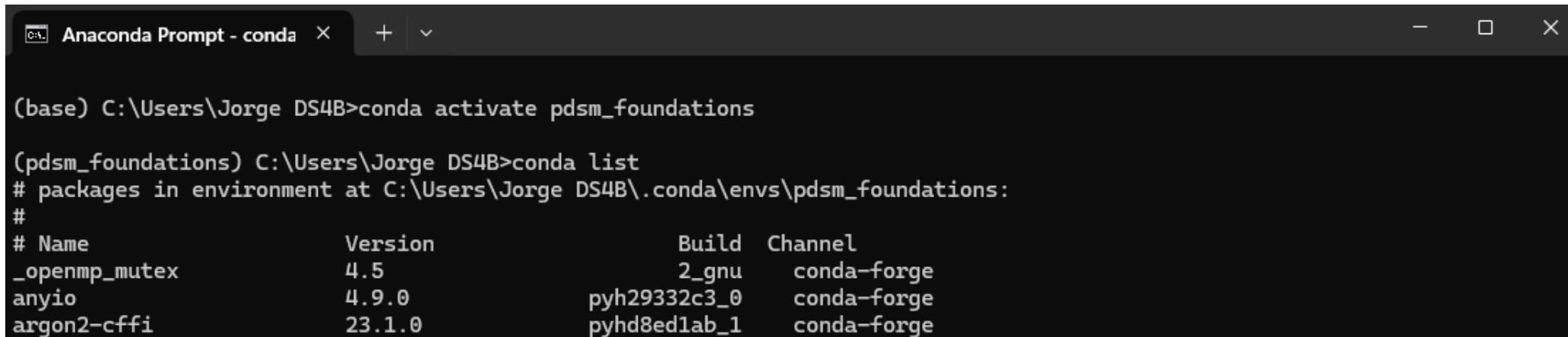
conda create -n entorno1 python=3.11

- Activar entornos

conda activate entorno1

- Instalar librerías:

conda install pandas matplotlib seaborn



```
(base) C:\Users\Jorge DS4B>conda activate pdsm_foundations

(pdsm_foundations) C:\Users\Jorge DS4B>conda list
# packages in environment at C:\Users\Jorge DS4B\.conda\envs\pdsm_foundations:
#
# Name           Version        Build  Channel
_openmp_mutex   4.5            2_gnu  conda-forge
anyio          4.9.0          pyh29332c3_0  conda-forge
argon2-cffi    23.1.0         pyhd8ed1ab_1  conda-forge
```

2. Organización de proyectos

- Crear y estructurar carpetas para un nuevo proyecto.
- Navegar y moverse entre carpetas.
- Ver y comprobar dónde estás (pwd, cd, ls, mkdir, tree, dir).

Ejemplos:

1. Crear una carpeta `projeto_ds` y después cambiar el directorio para trabajar sobre esta carpeta recién creada:

```
mkdir proyecto_ds && cd proyecto_ds
```

2. Crear 4 carpetas al mismo nivel dentro del directorio actual:

```
mkdir datos scripts notebooks resultados
```

3. Ejecución de scripts

- Ejecutar un script de Python sin necesidad de abrir Jupyter Notebook ni otro IDE

Ejemplo:

1. Ejecutar el script de entrenamiento del modelo de ML, ubicado en la carpeta procesos:

python procesos/entrenamiento.py

Variables de entorno - PATH

En este punto entra en juego el concepto de **PATH**, una variable de entorno que almacena una lista de rutas donde la terminal busca los programas o comandos que intentamos ejecutar.

Si el intérprete de Python, o herramientas como pip o conda, no están incluidas en el PATH, la terminal no podrá ejecutarlas, aunque estén correctamente instaladas en el sistema.

Variables de entorno PATH

¿Cómo solucionarlo?

En Windows existen dos alternativas:

- Incluir manualmente la ruta de instalación del programa (Python, conda, etc) a la variable de entorno PATH.
- Usar una terminal ya configurada, como Miniconda, donde Python y sus herramientas ya las tenemos disponibles directamente.

En macOS normalmente al instalar Miniconda , se añaden automáticamente sus rutas al PATH, lo que permite ejecutar los comandos sin problemas.

Si usamos VS Code, podremos configurar el entorno desde el editor para que reconozca automáticamente la ruta correcta del intérprete de Python.

4. Lanzar Apps en local

- Hacer pruebas de desarrollo simulando un servidor en local (Streamlit o FastApi)

Streamlit:

Librería que permite crear **Interfaces web interactivas** (formularios, visualizaciones).

Es ideal para compartir modelos, análisis o prototipos sin necesidad de programar HTML o JS.

FastAPI:

Es un framework que actúa como **puente entre aplicaciones**, permitiendo que se comuniquen de forma rápida y segura.

Ejemplo:

1. Lanzar la app comparador.py para probar las configuraciones efectuadas y comprobar su interfaz:

streamlit run comparador.py

5. Trabajar con Git

- Inicializar repositorio y versionar archivos (git init, git add, git commit).
- Subir código a GitHub (git push).
- Descargar proyectos (git clone).

Ejemplos:

1. Guardar el cambio del archivo analysis.py en el índice de Git:

```
git add scripts/analysis.py
```

2. Crear un punto de control en el historial de Git con todos los archivos añadidos y con el mensaje explicativo: “Versión final del análisis”:

```
git commit -m "Versión final del análisis"
```

Acción	macOS/Linux (Terminal)	Windows (CMD)	Windows (PowerShell)
Mostrar ubicación actual	pwd	cd	pwd
Listar archivos/directorios	ls	dir	ls, dir
Cambiar directorio	cd ruta	cd ruta	cd ruta
Limpiar pantalla	clear	cls	Clear
Ver carpeta del usuario actual	echo \$HOME	echo %USERPROFILE%	echo \$env:USERPROFILE
Crear carpeta	mkdir nombre	mkdir nombre	mkdir nombre
Crear archivo vacío	touch archivo	type nul > archivo	New-Item archivo
Añadir texto a un archivo	echo texto >> archivo	echo texto >> archivo	Add-Content archivo -Value "texto"
Mover/renombrar archivo	mv origen destino	move origen destino	Move-Item origen destino
Copiar archivo	cp origen destino	copy origen destino	Copy-Item origen destino
Visualizar contenido archivo	cat archivo	type archivo	Get-Content archivo
Ver árbol de carpetas y archivos	tree	tree /f	tree
Eliminar archivo	rm archivo	del archivo	Remove-Item archivo
Eliminar carpeta (vacía)	rmdir carpeta	rmdir carpeta	Remove-Item carpeta
Eliminar carpeta (con contenido)	rm -r carpeta	rmdir carpeta /s /q	Remove-Item carpeta -Recurse -Force
Buscar archivo por nombre	find . -name archivo	dir archivo /s	Get-ChildItem -Recurse archivo
Buscar contenido dentro de archivo	grep "texto" archivo.txt	find "texto" archivo.txt	Select-String -Pattern "texto" -Path archivo.txt