

GitHub y tu primer entorno de trabajo

Manual de la sesión — clonaremos el repositorio del curso y crearemos tu primer entorno virtual, todos juntos y paso a paso.

Modalidad: práctica guiada en clase **Fecha:** martes, 14 de julio de 2026

Trae: tu computadora con todo lo de la sección «Antes de la clase»

Antes de la clase — requisitos, no tarea

Esta sesión es 100 % práctica y avanza en bloque: quien llegue sin los requisitos no podrá seguirla. **No repetiremos aquí ningún paso de instalación ni de creación de cuentas;** verifica esta lista y, si te falta algo, resuélvelo **antes** de la clase por el canal de ayuda.

⚠ Verifica que ya tienes:

- **Las tres herramientas instaladas y verificadas**
— VS Code, Python y Git, con las extensiones **Python** y **Jupyter** y tu identidad configurada en Git. Todo esto está, paso a paso, en el **Manual de instalación** (también disponible en el aula virtual, Blackboard).
- **Una cuenta de Gmail activa**
— la misma que usas para Google Colab. Si no tienes: accounts.google.com/signup.
- **Una cuenta de GitHub creada y con el correo verificado**
— puedes registrarte con tu Gmail. Si no tienes: github.com/signup (elige un nombre de usuario profesional, tipo `nombre-apellido`, y anota tu contraseña).

Tres ideas antes de tocar el teclado

Repositorio

Clonar

Entorno virtual

Una carpeta de proyecto con memoria: además de los archivos, guarda todo su historial de cambios. El repositorio del curso vive en GitHub y contiene los materiales que usaremos.

Descargar una copia completa de un repositorio a tu computadora. A diferencia de bajar un ZIP, la copia queda *conectada* al original: podrás traer las novedades de cada semana con un solo comando.

Una caja de herramientas propia de cada proyecto: las librerías que instales dentro no se mezclan con las de otros proyectos. Trabajar así desde el día uno es el estándar profesional.

EN LA CLASE · LO HAREMOS JUNTOS

Práctica guiada: clonar el repositorio y crear el entorno

1

Prepara tu carpeta de trabajo

1. Con el explorador de archivos (Windows) o el Finder (Mac), crea dentro de **Documentos** una carpeta llamada exactamente **propedeutico**.
2. Abre VS Code y ve a **File → Open Folder** (Archivo → Abrir carpeta), selecciona **propedeutico** y pulsa **Seleccionar carpeta**.
3. Si VS Code pregunta si confías en los autores de la carpeta, marca la casilla y pulsa **Yes, I trust the authors**.
4. Abre la terminal integrada: menú **Terminal → New Terminal**. Fíjate en el detalle clave: la terminal se abre **“parada” dentro de tu carpeta** — la ruta antes del cursor termina en **propedeutico**. Todo lo que ejecutes ocurre ahí.

⚠ Regla de oro para nombres de carpetas y archivos

Siempre en minúsculas y **sin espacios, tildes ni ñes** (usa guiones: **mi-proyecto**). Los espacios y caracteres especiales en las rutas son una fuente clásica de errores difíciles de detectar.

2

Copia la dirección del repositorio del curso

1. Entra al repositorio del curso: github.com/franperezec/propedeutico-flacso.

2. Pulsa el botón verde `<> Code`, verifica que esté seleccionada la pestaña `HTTPS` y pulsa el icono de copiar junto a la dirección (termina en `.git`).

La dirección tiene siempre la forma `https://github.com/usuario/repositorio.git`. Ese mismo botón sirve para clonar *cualquier* repositorio público de GitHub.

3

Clona el repositorio en tu computadora

En la terminal de VS Code escribe `git clone`, un espacio, y pega la dirección (clic derecho → Pegar, o `Ctrl+V` / `⌘V`). Presiona Enter:

```
git clone https://github.com/usuario/repositorio.git
```

RESULTADO ESPERADO (LOS NÚMEROS VARÍAN)

```
~/Documentos/propedeutico> git clone https://github.com/usuario/repositorio.git
Cloning into 'repositorio'...
remote: Enumerating objects: 87, done.
remote: Counting objects: 100% (87/87), done.
Receiving objects: 100% (87/87), 1.24 MiB | 2.10 MiB/s, done.
Resolving deltas: 100% (31/31), done.
```

1. Al terminar, dentro de `propedeutico` apareció una carpeta nueva con el nombre del repositorio: es tu copia local, con todos los materiales.
2. Ábrela como carpeta de trabajo: `File → Open Folder`, selecciona la carpeta recién creada y confía en los autores. De aquí en adelante trabajamos **dentro del repositorio**.

💡 Para las próximas semanas

Cuando el docente publique materiales nuevos, no hay que clonar otra vez: con el repositorio abierto, ejecuta en la terminal `git pull` y las novedades bajan solas. Esa es la ventaja de clonar frente a descargar un ZIP.

4

Crea tu primer entorno virtual

Lo haremos por la vía integrada de VS Code, que es **idéntica en Windows y Mac**:

1. Abre la paleta de comandos: `Ctrl+Shift+P` (Windows) o `⌘P` (Mac).
2. Escribe **Python: Create Environment** y presiona Enter.
3. Elige `Venv` como tipo de entorno.

4. Selecciona el intérprete de Python que instalaste (versión 3.1x).
5. Si VS Code detecta un archivo `requirements.txt` en el repositorio, marca su casilla: instalará automáticamente las librerías del curso. Pulsa `OK`.
6. Espera la notificación de que el entorno fue creado. En el explorador de archivos de VS Code (barra izquierda) verás una carpeta nueva llamada `.venv`: **esa carpeta es tu entorno**. No la toques ni la borres.

Comprueba que está activo: abre una **terminal nueva** (`Terminal → New Terminal`). El indicador `(.venv)` al inicio de la línea confirma que estás dentro de la caja:

```
TERMINAL — ENTORNO ACTIVO
```

```
(.venv) ~/Documentos/propedeutico/repositorio>
```

Activar y desactivar el entorno a mano

Las terminales nuevas de VS Code activan el entorno solas. Pero si abres una terminal fuera de VS Code, o el `(.venv)` no aparece, actívalo tú desde la carpeta del repositorio — `WIN` `.venv\Scripts\activate` · `MAC` `source .venv/bin/activate` — y para salir del entorno, en ambos sistemas: `deactivate`. Si Windows responde «la ejecución de scripts está deshabilitada», la solución está en P2.

▼ Vía alternativa · Crear el entorno desde la terminal

El mismo resultado sin la paleta de comandos. Primero, **parado dentro de la carpeta del repositorio clonado** (la terminal integrada de VS Code ya se abre ahí), crea el entorno — **WIN** :

```
python -m venv .venv
```

MAC (fuera del entorno, el Python moderno se llama `python3`):

```
python3 -m venv .venv
```

Después actívalo — **WIN** :

```
.venv\Scripts\activate
```

MAC :

```
source .venv/bin/activate
```

Cuando aparezca el `(.venv)` al inicio de la línea, listo. Por esta vía no hay casilla de `requirements.txt`, así que nada se instala solo: el paso 5 es obligatorio. Si Windows responde «la ejecución de scripts está deshabilitada», la solución está en P2.

Última comprobación del paso: con el `(.venv)` visible, pide la lista de librerías instaladas. El comando es el mismo en Windows y Mac:

```
pip list
```

Si al crear el entorno marcaste la casilla de `requirements.txt`, en la lista deben aparecer las siete librerías del curso — `ipykernel`, `matplotlib`, `numpy`, `openpyxl`, `pandas`, `scipy` y `sympy` — junto a otras que se instalan solas como acompañantes:

RESULTADO ESPERADO (FRAGMENTO — LAS VERSIONES PUEDEN VARIAR)

```
(.venv) ~/Documentos/propedeutico/repositorio> pip list
```

```
Package          Version
```

```
-----
```

```
...
```

```
ipykernel        7.3.0
matplotlib        3.11.0
numpy             2.5.1
```

```
openpyxl      3.1.5
pandas        3.0.3
...
scipy         1.18.0
sympy         1.14.0
...
```

¿No aparecen, o la lista sale casi vacía? No pasa nada: el paso 5 lo resuelve con un solo comando.

5

Instala las librerías dentro del entorno

Si al crear el entorno marcaste la casilla de `requirements.txt`, este paso ya ocurrió solo — y `pip list` te lo confirmó. Pero si no la marcaste, creaste el entorno por la terminal, o algo falló y las librerías no aparecen, esta es la red de seguridad. Con el `(.venv)` visible, este comando es **el mismo en Windows y Mac**:

```
python -m pip install -r requirements.txt
```

`requirements.txt` es la lista de compras del proyecto: un archivo que vive en el repositorio y declara las siete librerías del curso con sus versiones exactas. Ese único comando las instala todas de una vez (la terminal ya está parada en la carpeta del repositorio, donde vive el archivo) — y como viaja con el repositorio, cualquiera que lo clone puede reconstruir el mismo entorno.

¿Por qué “dentro del entorno”? Con el `(.venv)` activo, las librerías se instalan *dentro de la caja del proyecto*, no en el Python “general” de tu máquina. Cada entorno lleva sus propias herramientas — por eso los proyectos no se estorban entre sí.

6

Prueba de fuego: ejecuta un notebook con tu entorno

1. En el explorador de VS Code, abre el notebook del repositorio:

```
clases/clase01_conjuntos_producto_cartesiano.ipynb .
```

2. Arriba a la derecha, pulsa `Select Kernel` → `Python Environments` → elige `.venv`.

3. Ejecuta la primera celda (botón ▶ o `Ctrl/⌘+Enter`, como en Colab).

Si la celda corre, acabas de reproducir en tu propia máquina todo el flujo profesional: **repositorio clonado** → **entorno propio** → **código ejecutando**. Esto es exactamente lo que hacías en Colab, pero ahora el laboratorio es tuyo.



Salida de la clase

Antes de cerrar la laptop, verifica que puedes marcar todo:

- ☐ Cloné el repositorio del curso y lo tengo abierto como carpeta en VS Code.
- ☐ Existe la carpeta `.venv` dentro del repositorio.
- ☐ Al abrir una terminal nueva aparece `(.venv)` al inicio de la línea.
- ☐ Instalé (o se instalaron) las librerías sin errores en rojo.
- ☐ Ejecuté una celda del notebook de prueba con el kernel `.venv`.
- ☐ Sé que `git pull` trae los materiales nuevos de cada semana.

Comandos clave de terminal para orientarte

Cinco comandos para no perderte nunca en la terminal. Los usaremos todo el semestre:

Comando	Qué hace
<code>pwd</code>	Muestra en qué carpeta estás "parado" ahora mismo.
<code>ls</code> (en Windows también sirve <code>dir</code>)	Lista el contenido de la carpeta actual.
<code>cd nombre-carpeta</code>	Entra a una carpeta que esté dentro de la actual.
<code>cd ..</code>	Regresa a la carpeta anterior (un nivel arriba).
<code>mkdir nombre-carpeta</code>	Crea una carpeta nueva donde estás parado.

Si algo se atasca

▼ P1 · `git clone` responde “repository not found” o “could not resolve host”

Casi siempre es la dirección mal copiada o incompleta. Vuelve al repositorio en el navegador, botón `<> Code` → pestaña `HTTPS` → icono de copiar, y pega de nuevo. “Could not resolve host” indica además un problema de internet: verifica tu conexión (o el wifi del aula).

▼ P2 · `WIN` “la ejecución de scripts está deshabilitada en este sistema”

Es una protección de PowerShell que bloquea la activación del entorno. Ejecuta en la terminal:

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

Responde `S` (o `Y`) si pregunta, cierra esa terminal y abre una nueva. El `(.venv)` debería aparecer.

▼ P3 · No aparece la opción “Python: Create Environment”

Falta la extensión **Python** de Microsoft (sección D del [Manual de instalación](#)) o VS Code necesita reiniciarse tras instalarla. Instálala desde el panel de extensiones, cierra y vuelve a abrir VS Code.

▼ P4 · La terminal no muestra `(.venv)`

Las terminales abiertas *antes* de crear el entorno no se enteran de que existe: ciérralas y abre una nueva. Si persiste, abre la paleta (`Ctrl+Shift+P` / `⌘P`) → **Python: Select Interpreter** → elige el que dice `.venv`, y abre otra terminal.

▼ P5 · GitHub pide usuario, contraseña o autorización al clonar

El repositorio del curso es público y clonar no debería pedir nada. Si aparece una ventana del navegador pidiendo autorizar, pulsa `Authorize` e inicia sesión con tu cuenta de GitHub. Si la terminal pide usuario y contraseña en texto, primero revisa la dirección (P1); si la dirección es correcta, avisa al docente.

▼ P6 · El notebook no encuentra el kernel o pide instalar ipykernel

Pulsa `Select Kernel` → `Python Environments` → `.venv`. Si VS Code ofrece instalar `ipykernel`, acepta; equivale al paso 5.

▼ P7 · Cloné en el lugar equivocado o no encuentro la carpeta

Ejecuta `pwd` para ver dónde estabas parado al clonar: la copia se creó ahí. No pasa nada grave: puedes **mover la carpeta completa** del repositorio a `Documentos/propedeutico` con el explorador o el Finder sin romper nada, y volver a abrirla desde VS Code.

💡 ¿Tu problema no está en la lista? Pídele ayuda a una IA

Un asistente de IA (ChatGPT, Claude, Gemini o Copilot) resuelve muy bien este tipo de atascos **si le das contexto suficiente**. Copia esta plantilla, rellena los huecos [entre corchetes] y pégala en el chat:

```
Estoy siguiendo un manual del curso propedéutico de FLACSO
y me atasqué en [paso o sección del manual donde estabas].
Mi sistema operativo es [Windows o Mac].
Escribí este comando o código:
[pega aquí exactamente lo que escribiste]
Y obtuve este mensaje de error o problema:
[pega aquí el mensaje completo, copiado desde la terminal]
Por favor: primero explícame por qué ocurre el problema,
en lenguaje sencillo (no tengo experiencia en programación),
y después dame los pasos para resolverlo, uno por uno.
```

Tres consejos para que la respuesta sirva:

- **Copia el error completo como texto** — selecciónalo en la terminal y cópialo — mejor que una captura de pantalla o un fragmento suelto.
- **Lee cada comando que la IA te sugiera antes de ejecutarlo**; si no entiendes qué hace, pregúntale. Nunca ejecutes a ciegas comandos que borren archivos o carpetas.
- Si la primera respuesta no funciona, **cuéntale qué pasó al intentarlo**: la conversación de ida y vuelta es parte normal del proceso.

Qué sigue

Hoy *recibiste* código: clonaste un repositorio ajeno y montaste tu laboratorio local. En la próxima sesión harás el camino inverso: crear **tu propio repositorio**, guardarle cambios con `git add` y `git commit`, y publicarlos en tu GitHub con `git push` — tu portafolio público empieza ahí. El paso a paso completo ya está publicado en el **Manual de la clase 3 — Tu propio repositorio en GitHub**.

Propedéutico de Matemáticas y Programación · FLACSO Ecuador — Manual de sesión: GitHub y primer entorno, versión julio 2026.

Complementa al **Manual de instalación** (VS Code, Python y Git), que contiene todos los pasos previos y su solución de problemas. Manual de consulta: el **Mapa de las herramientas**.