

Computación en la Biología 2025

Introducción a Jupyter Lab y los Notebooks



Objetivos

1. Comprender qué son los *notebooks* y para qué sirven.
2. Conectarse de forma remota al servidor del curso.
3. Familiarizarse con la interfaz de Jupyter Lab y sus principales funciones.

Entender qué son y para qué sirven

Un **Jupyter Notebook** es un documento interactivo que permite combinar en un mismo lugar:

- **Código ejecutable** (principalmente en Python, aunque admite decenas de lenguajes).
- **Texto enriquecido** en formato Markdown.
- **Resultados visuales** como tablas, gráficos e imágenes.

Esto lo convierte en una herramienta esencial para el **procesamiento de datos**, la **computación científica** y la **ciencia de datos**, ya que facilita documentar, compartir y reproducir análisis completos.

El proyecto Jupyter

Jupyter es un proyecto sin fines de lucro cuya misión es:

“Desarrollar software de código abierto, estándares abiertos y servicios para la computación interactiva en docenas de lenguajes de programación.”

El nombre proviene de la unión de **Julia**, **Python** y **R**, los tres lenguajes principales soportados al inicio.

A su vez, es un homenaje a los cuadernos de Galileo, en los que registró el descubrimiento de los satélites de Júpiter.



¿Qué es Jupyter Lab?

Jupyter Lab es la evolución de los notebooks clásicos. Ofrece un **entorno interactivo, moderno y flexible** que se ejecuta directamente en el navegador web mediante una interfaz gráfica.

Entre sus principales características:

- Permite combinar **código, texto y resultados** en un mismo espacio de trabajo.
- Soporta múltiples lenguajes de programación a través de *kernels* (en este curso trabajaremos principalmente con **Python**).
- Incorpora **elementos visuales interactivos**, como tablas y gráficos.
- Facilita la **gestión de archivos y proyectos** en el servidor.
- Permite **exportar** el contenido (por ejemplo en PDF o HTML) para compartir o entregar resultados.

En este curso, utilizaremos Jupyter Lab como la herramienta principal para realizar los análisis prácticos en genómica evolutiva y entregar las tareas de cada módulo.

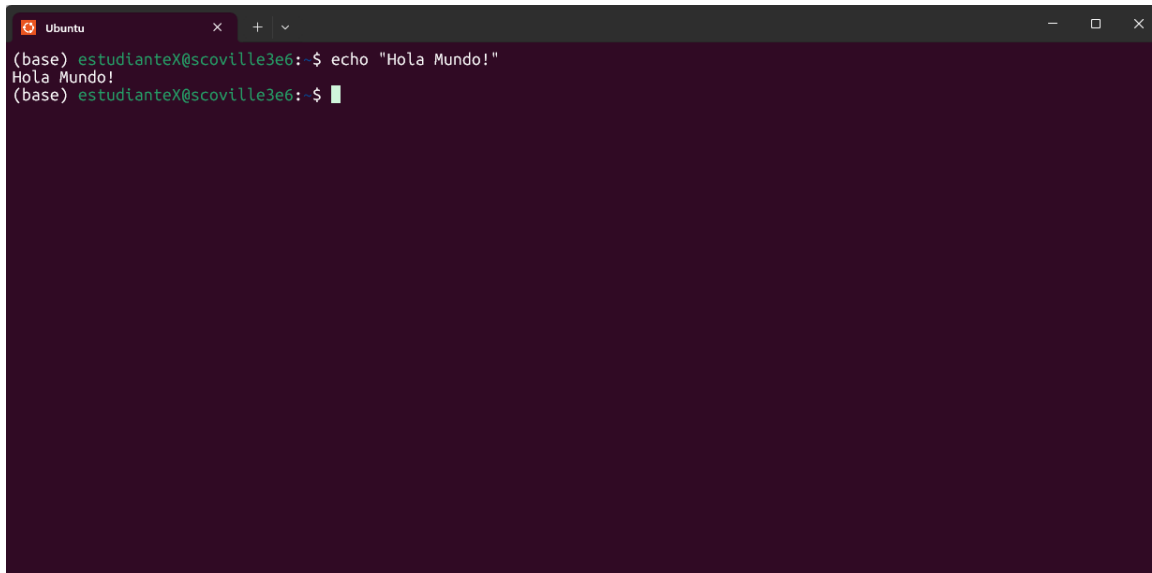
Línea de comando

La **línea de comando** (o terminal) es una herramienta que permite interactuar directamente con la computadora escribiendo instrucciones en texto.

Es fundamental en bioinformática, ya que la mayoría de los programas y análisis se ejecutan de esta forma. Durante el curso la utilizaremos frecuentemente para conectarnos al servidor, ejecutar programas y organizar archivos.

Cómo acceder a la terminal

- **Linux:** la terminal viene instalada por defecto. Se abre desde el menú de aplicaciones o con el atajo `Ctrl + Alt + T`.
- **macOS:** se encuentra en *Aplicaciones > Utilidades > Terminal*.
- **Windows:** de forma nativa se puede usar **PowerShell** o **Windows Terminal**.



```
(base) estudianteX@scoville3e6: ~$ echo "Hola Mundo!"
Hola Mundo!
(base) estudianteX@scoville3e6: ~$
```

Uso de MobaXterm en Windows

En **Windows** recomendamos usar [MobaXterm](#):

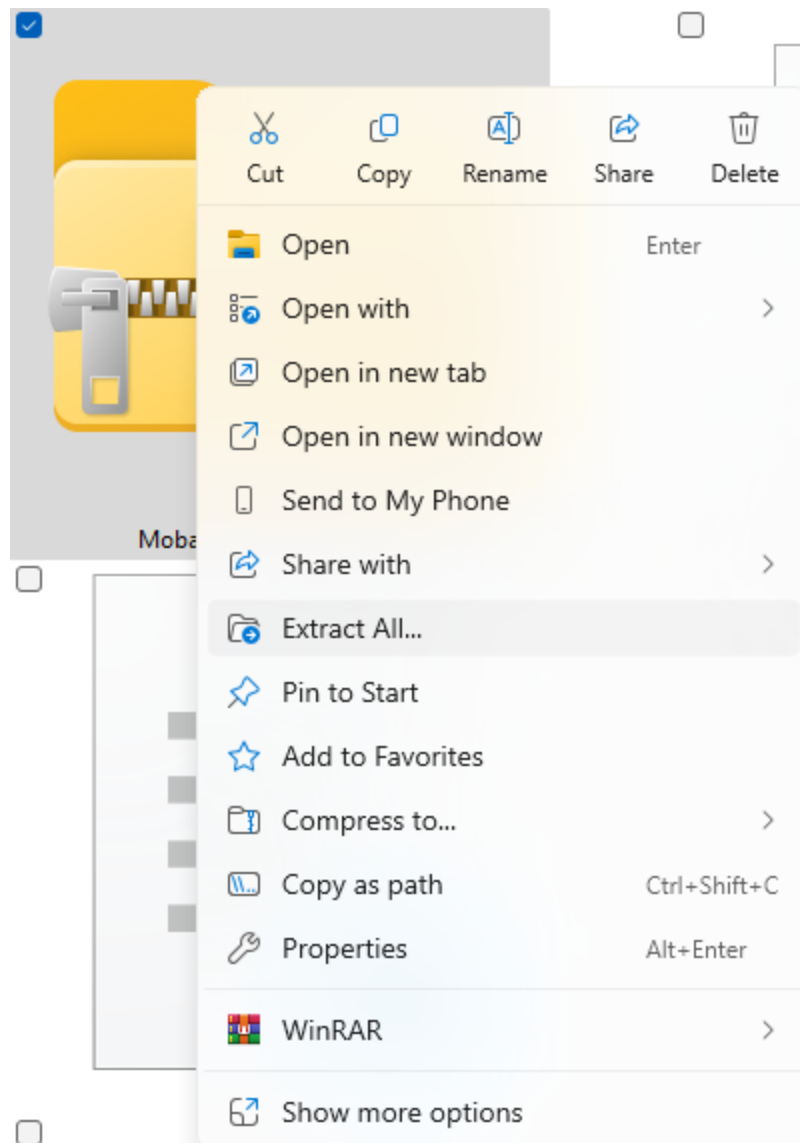
- Descargar la **Installer Edition** (botón verde).
- Si no puede instalar programas, usar la **Portable Edition** (botón azul).

MobaXterm Home Edition

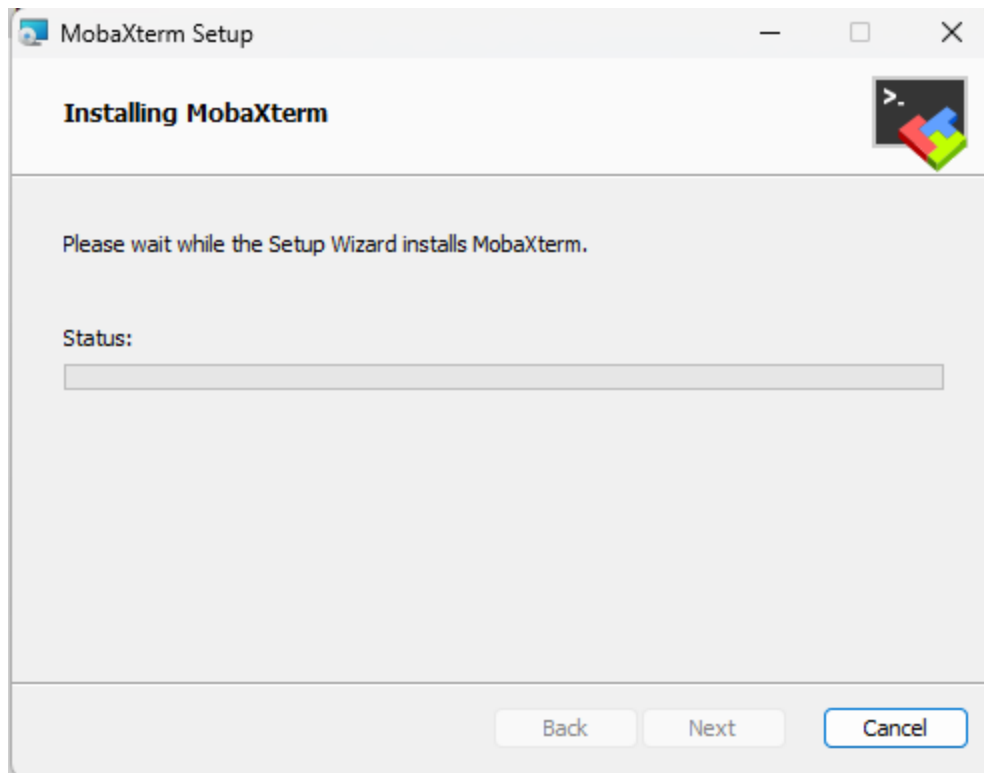
Download MobaXterm Home Edition (current version):



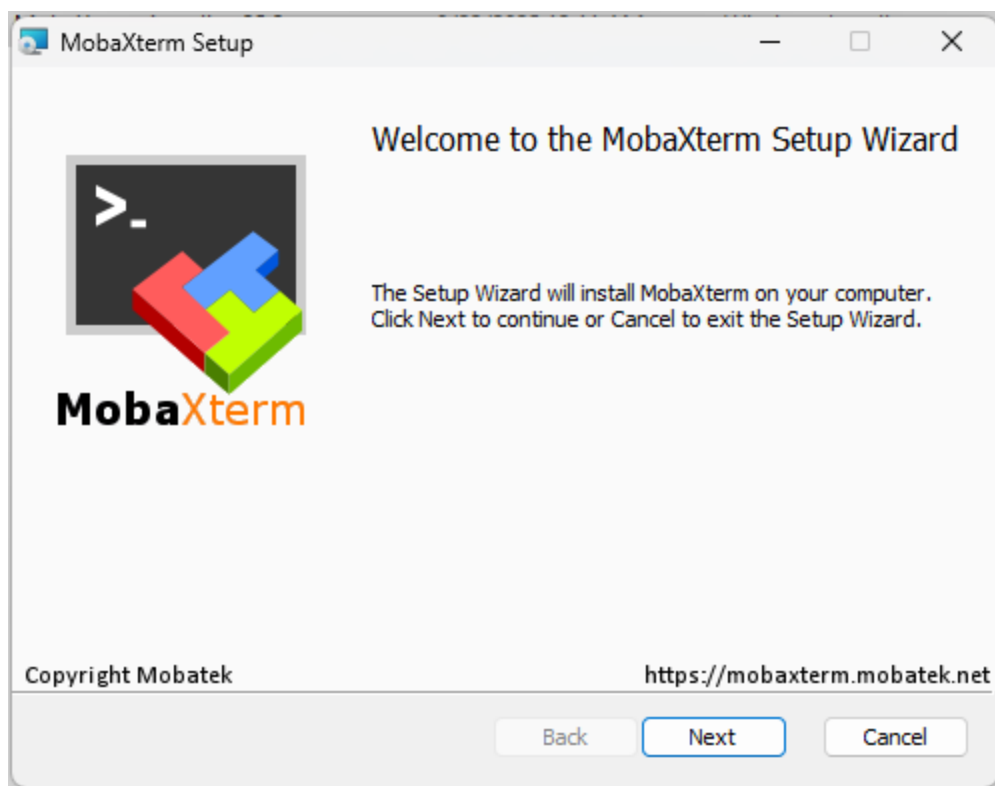
Una vez descargado, hay que descomprimir el archivo `MobaXterm_Installer_v25.2`.



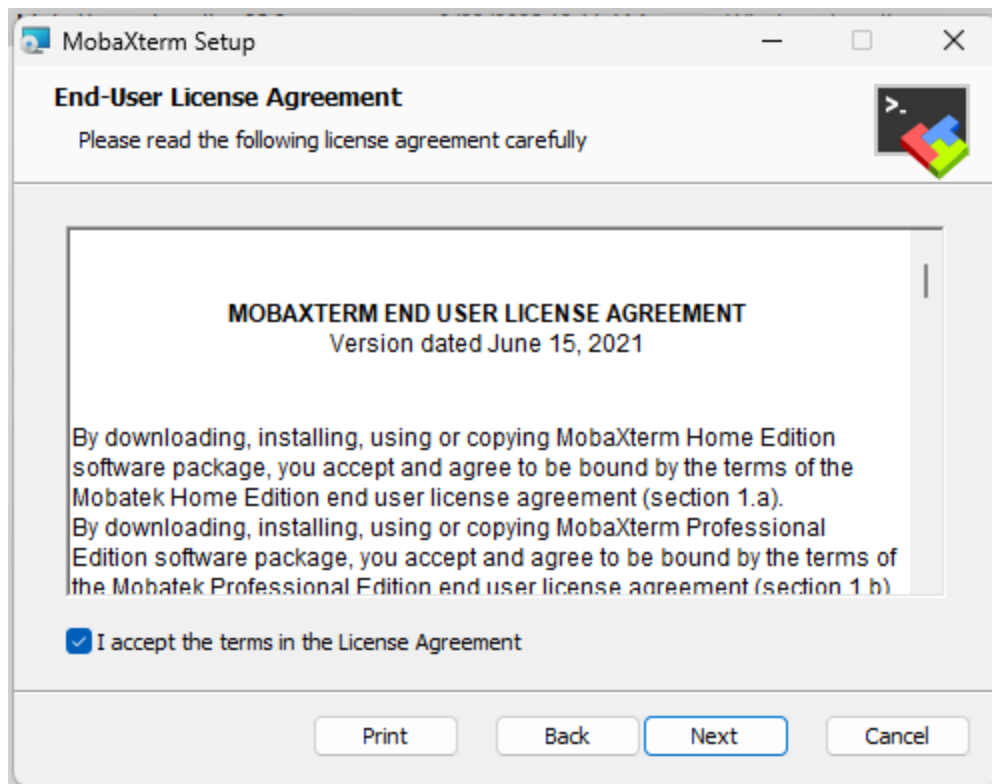
Luego se ejecuta el instalador.



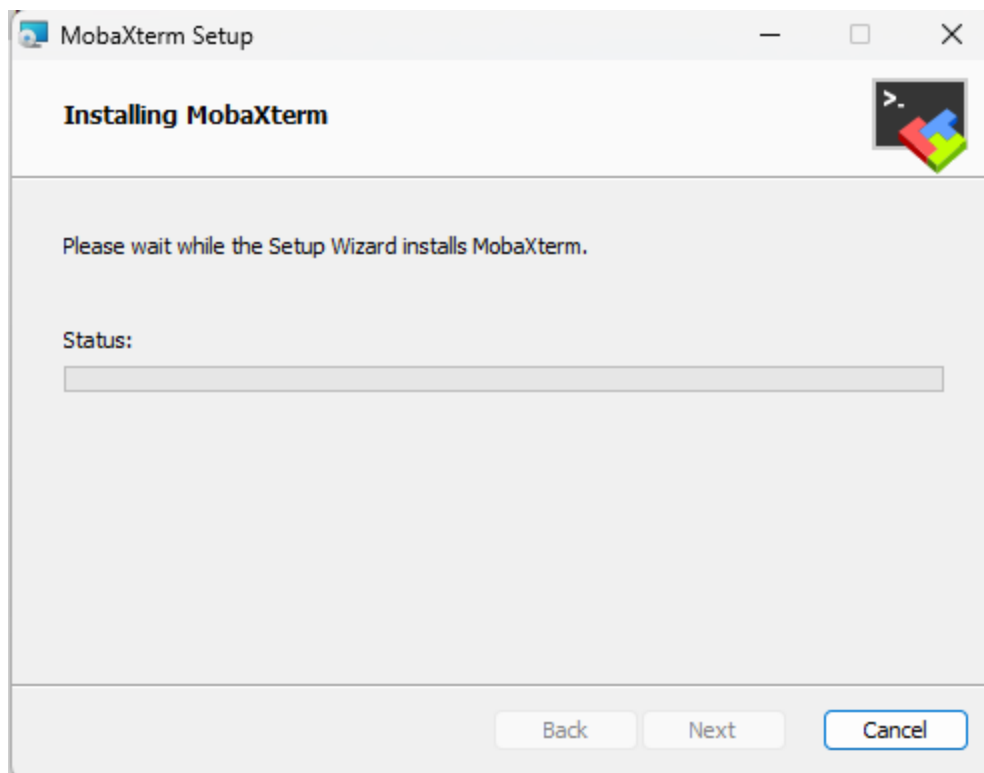
Y se inicia la instalación.



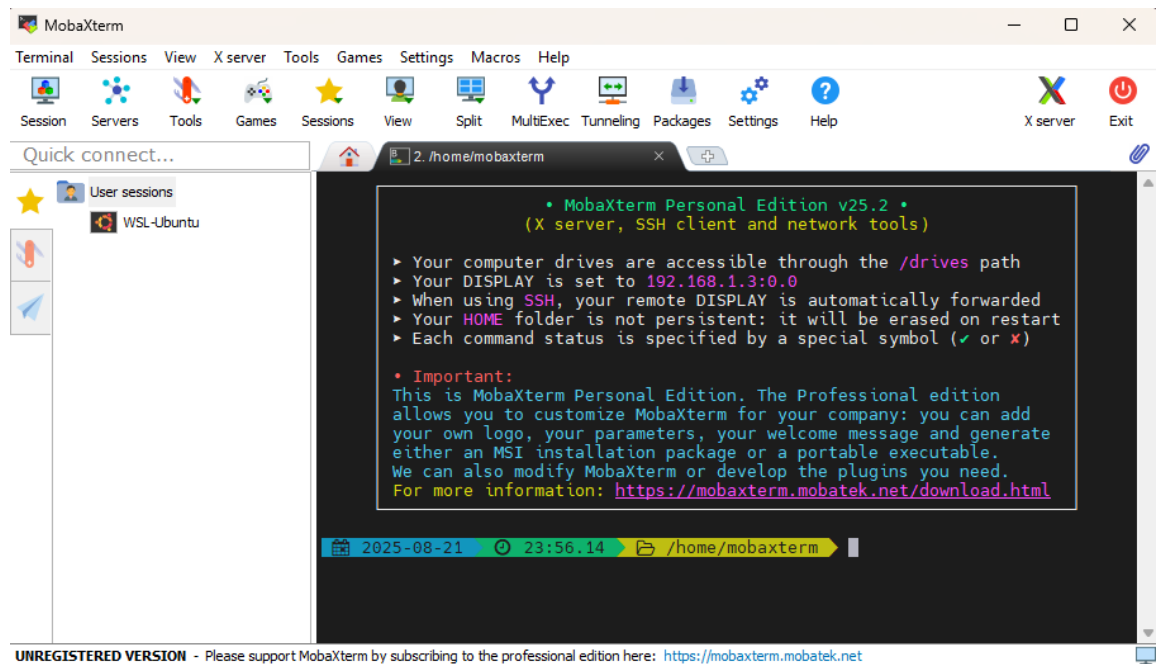
Hacemos click en "I accept the terms in the License Agreement".



Le damos siguiente, y continuamos hasta que inicie la instalación.



MobaXterm combina una terminal con herramientas gráficas adicionales y simplifica la conexión por SSH al servidor que utilizaremos en el curso.



Conexión a un servidor remoto por SSH



Durante el curso cada estudiante accederá al servidor **scoville3e6** para realizar las prácticas.

A este servidor nos conectaremos usando el protocolo SSH (de **Secure SHell**). Este es el nombre de un protocolo y del programa que lo implementa cuya principal función es el acceso remoto a un servidor por medio de un canal seguro.

Para iniciar sesión en un servidor necesitamos:

- un **usuario** registrado,
- y la **dirección IP** o el **hostname** del servidor.

El formato del comando es:

```
In [ ]: ssh usuario@servidor
```

Por ejemplo:

```
In [ ]: mi_usuario@pc:~$ ssh estudianteX@164.73.84.53
```

Al ejecutarlo, la máquina pedirá su contraseña. La primera vez que se conecte es probable que aparezca un mensaje como:

```
The authenticity of host 'scoville3e6.fcien.edu.uy (164.73.84.53)'
can't be established.
ECDSA key fingerprint is (...)
Are you sure you want to continue connecting (yes/no)?
```

Este aviso significa que su computadora todavía no conoce al servidor remoto. Como es la primera vez que se conecta, esto es **normal**: escriba **yes** para continuar.

Si la conexión es exitosa, el *prompt* cambiará, mostrando el nombre del servidor y su usuario. Por ejemplo:

Antes:

```
In [ ]: mi_usuario@pc:~$
```

Después:

```
In [ ]: estudianteX@scoville3e6:~$
```

A partir de ahora usted se estará conectando a una computadora remota: todo lo que escriba en la terminal se ejecutará en el servidor, no en su máquina local.

Para salir y volver a su máquina local, use el comando:

```
In [ ]: exit
```

Conexión remota a Jupyter Lab

Para acceder a Jupyter Lab en el servidor, debemos combinar la conexión por **SSH** con un **túnel** (*tunneling*).

La conexión se realiza en **tres pasos sucesivos**:

1. Conectarse al servidor e iniciar Jupyter Lab

Desde la terminal local, conectarse por SSH:

```
In [ ]: ssh estudianteX@164.73.84.53
```

Una vez dentro del servidor, iniciar Jupyter Lab con:

```
In [ ]: jupyter lab --no-browser --port=XXXX --notebook-dir=~
```

Donde `XXXX` es un número de puerto libre de 4 cifras (por ejemplo, 1248). El sistema mostrará un mensaje similar a:

```
To access the server, open this file in a browser:
file:///home/estudianteX/.local/share/jupyter/runtime/jpserver-
285986-open.html
Or copy and paste one of these URLs:
http://localhost:1250/lab?
token=5ce4ce82b72d5ec5999cbe05b60cdc74ceabfb4ea82569bb
```

El puerto que vamos a tener que usar en el siguiente paso va a aparecer en el segundo link después de `localhost:`, en este caso sería `1250`.

⚠ Nota: El puerto puede no coincidir con el que eligió (e.g. `1248`) si estaba ocupado por otro usuario. Use **siempre** el puerto que aparece en el mensaje (e.g. `1250`).

2. Crear el túnel SSH en otra terminal local

En una segunda terminal local (no desde dentro del servidor), crear el túnel que enlaza su computadora con el servidor. Debe sustituir `XXXX` por el mismo número de puerto que obtuvo en el paso anterior (e.g. `1250`):

```
In [ ]: ssh -N -f -L XXXX:localhost:XXXX estudianteX@164.73.84.53
```

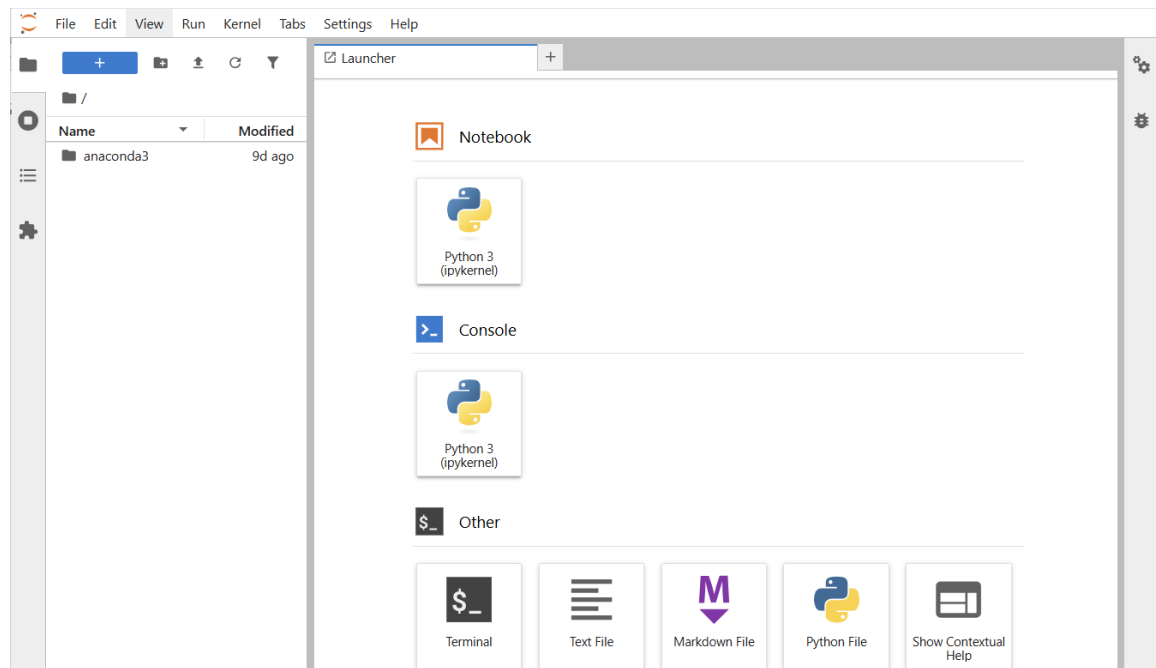
Recomendamos que ambos números de puerto sean iguales (ejemplo: 1250).

3. Abrir Jupyter Lab en el navegador

Copiar el enlace brindado en el paso 1, por ejemplo:

```
http://localhost:1250/lab?
token=5ce4ce82b72d5ec5999cbe05b60cdc74ceabfb4ea82569bb
```

Pegar este enlace en un navegador web (Firefox, Chrome, etc.). Debería abrirse la interfaz gráfica de **Jupyter Lab**:



Interfaz gráfica de Jupyter Lab

Jupyter Lab cuenta con una **interfaz gráfica moderna y flexible** que se ejecuta en el navegador web. Su diseño está pensado para que puedas **organizar tu trabajo científico o de análisis de datos en un solo lugar**, combinando código, archivos, terminales y documentación.

A diferencia de otros entornos como Jupyter Notebook clásico o RStudio, Jupyter Lab permite abrir **múltiples documentos y herramientas en pestañas o paneles divididos**, similar a IDEs modernos como Visual Studio Code o PyCharm. Esto facilita trabajar simultáneamente con notebooks, consolas interactivas, archivos de texto, o incluso ejecutar comandos en una terminal dentro del mismo entorno.

La filosofía de Jupyter Lab se basa en el concepto de **"literate programming"** la idea de combinar código ejecutable con documentación rica y visualizaciones en un mismo documento. Por ejemplo, puedes tener abierto simultáneamente un notebook con tu análisis principal, un archivo CSV en el visor de datos, una terminal ejecutando scripts de preprocesamiento, y documentación en Markdown explicando tu metodología, todo organizado en el mismo espacio de trabajo.

La interfaz se organiza en **varias secciones bien definidas**: barra de menús superior, panel lateral de navegación, área central de trabajo y Launcher para crear rápidamente nuevos notebooks, terminales y otros archivos.

Barra superior de menús

La **barra superior** de Jupyter Lab contiene los menús principales que organizan todas las funcionalidades del entorno. Similar a cualquier aplicación de escritorio, esta barra

proporciona acceso a las herramientas y configuraciones más importantes.

Menús principales

- **File (Archivo):** Gestiona la creación, apertura, guardado y exportación de documentos. Desde aquí puedes crear nuevos notebooks, abrir archivos existentes, guardar tu trabajo, y exportar notebooks a diferentes formatos como PDF, HTML o presentaciones. También incluye opciones para cerrar pestañas y administrar sesiones.
- **Edit (Editar):** Contiene las operaciones de edición estándar como deshacer, rehacer, cortar, copiar y pegar. Para notebooks específicamente, incluye funciones para buscar y reemplazar texto, así como operaciones de limpieza como "Clear All Outputs" (limpiar todas las salidas).
- **View (Ver):** Controla la apariencia y disposición de la interfaz. Permite mostrar u ocultar diferentes paneles (como el explorador de archivos o la paleta de comandos).
- **Run (Ejecutar):** Específicamente diseñado para notebooks, este menú controla la ejecución de código. Incluye opciones para ejecutar celdas individuales, ejecutar todas las celdas, reiniciar el kernel, o interrumpir procesos en ejecución.
- **Kernel:** Administra los kernels (los procesos que ejecutan el código). Permite cambiar de kernel (por ejemplo, de Python 3 a R), reiniciar kernels que no respondan.
- **Tabs (Pestañas):** Facilita la navegación entre múltiples documentos abiertos. Incluye opciones para cerrar pestañas, reorganizarlas, o abrir documentos en nuevos paneles divididos.
- **Settings (Configuración):** Accede a las preferencias generales del entorno, incluyendo configuración de editores, temas, extensiones instaladas, y ajustes avanzados del sistema.
- **Help (Ayuda):** Proporciona acceso a documentación, tutoriales, atajos de teclado, y enlaces a recursos de la comunidad. También incluye información sobre la versión actual de Jupyter Lab.

El panel lateral de navegación

El **panel izquierdo** de Jupyter Lab funciona como el centro de navegación y control del entorno. Las herramientas de este panel se organizan mediante **iconos en la barra lateral izquierda** que actúan como pestañas. Se puede **colapsar** el panel haciendo doble click para maximizar el espacio de trabajo cuando sea necesario.

Principales herramientas disponibles

- **File Browser** : Muestra la estructura de directorios del directorio de trabajo. Permite navegar por carpetas, crear nuevos archivos y directorios, renombrar archivos, etc. También puedes arrastrar y soltar los archivos para reorganizarlos o abrirlos en el área principal.
- **Running Terminals and Kernels** : Muestra todos los procesos en ejecución, incluyendo notebooks activos, terminales abiertos y consolas interactivas. Útil para monitorear el estado de tus kernels, cerrar procesos que no respondan, o cambiar rápidamente entre diferentes sesiones de trabajo.
- **Table of Contents** : Genera automáticamente un índice navegable basado en los encabezados de markdown en notebooks, facilitando la navegación en documentos largos y proporcionando una vista estructurada del contenido.
- **Extension Manager** : Permite ampliar las funcionalidades de Jupyter Lab mediante la instalación de complementos desarrollados por la comunidad. En la sección **discover** vas a encontrar un catálogo con las extensiones más populares: como integraciones con GIT, temas personalizados, entre otros. También puedes encontrar nuevas extensiones usando el buscador que se encuentra en la sección superior del panel. Antes de instalar cualquier extensión es necesario que aceptes los términos y condiciones de Jupyter Lab que se va a mostrar en

Área de trabajo y Launcher

El Área de trabajo

El **Área de trabajo** es la zona central donde se despliegan todos tus documentos, herramientas y visualizaciones. Su diseño flexible permite organizar múltiples elementos de forma simultánea y personalizada.

Cada documento abierto aparece como una **pestaña independiente** en la parte superior del área de trabajo. Las pestañas muestran el nombre del archivo, su tipo, y el estado de guardado. Puedes reorganizar las pestañas arrastrándolas o cerrarlas individualmente.

Desde Jupyter Lab tienes la capacidad de **dividir el área de trabajo** en múltiples paneles. Puedes arrastrar cualquier pestaña hacia los bordes del área para crear divisiones horizontales o verticales. Esto permite, por ejemplo, tener un notebook en una mitad de la pantalla y un archivo CSV abierto en la otra mitad. Esta funcionalidad es de mucha utilidad cuando trabajas con monitores grandes.

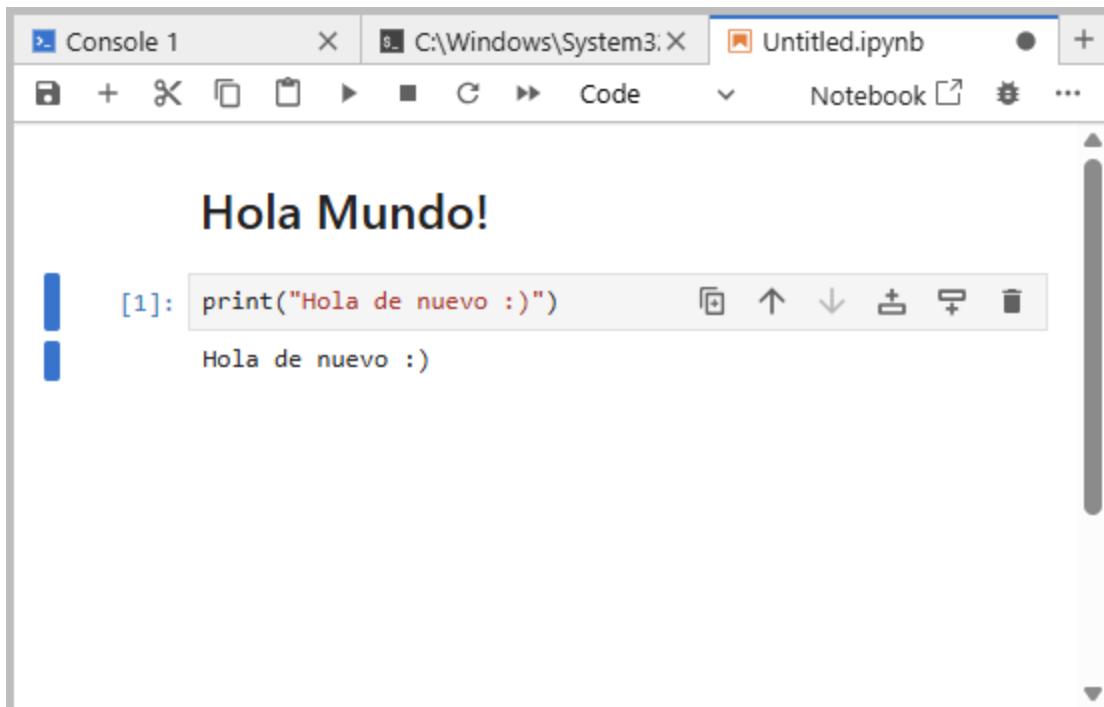
El Launcher

El **Launcher** es punto de partida para crear nuevos documentos y herramientas en Jupyter Lab. Aparece automáticamente cuando abres el entorno por primera vez y puede accederse

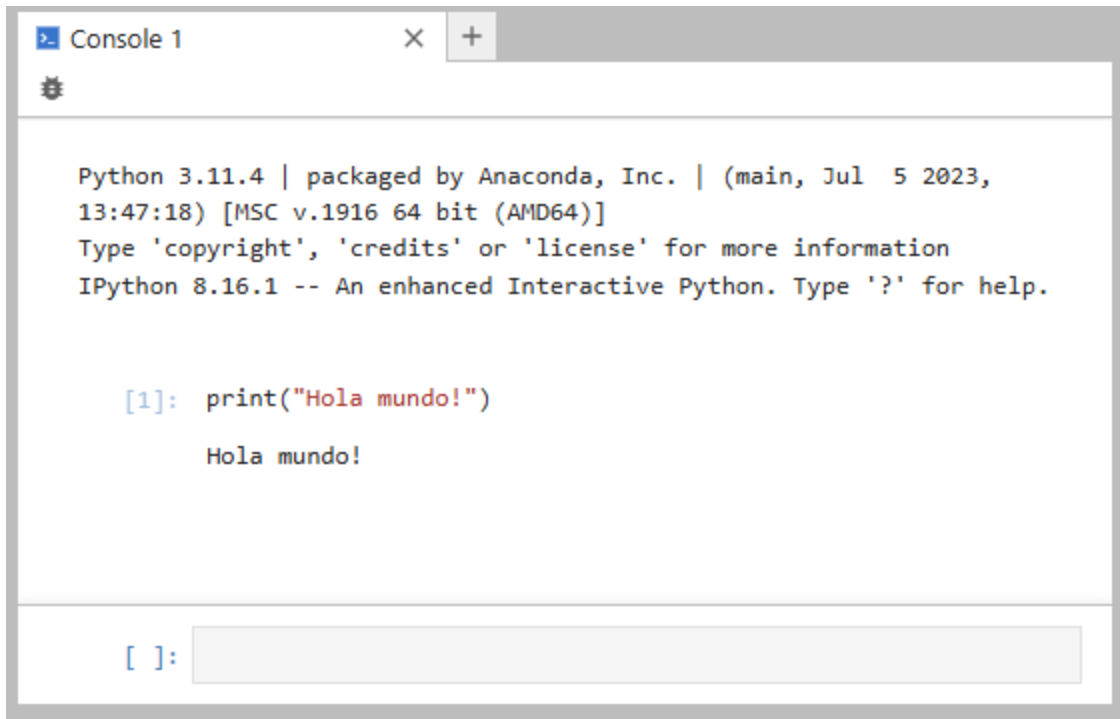
en cualquier momento a través del menú File > New Launcher o haciendo clic en el ícono "+" en la barra de pestañas.

El Launcher se organiza en **categorías visuales** que agrupan las diferentes opciones disponibles:

- **Notebook** : Permite crear nuevos notebooks con diferentes kernels instalados (Python 3, R, Julia, etc.). Cada opción muestra el logo del lenguaje correspondiente, facilitando la identificación rápida.



- **Console** : Crea consolas interactivas que funcionan como terminales de código para cada lenguaje. Son útiles para pruebas rápidas, debugging, o ejecutar comandos sin crear un notebook completo.

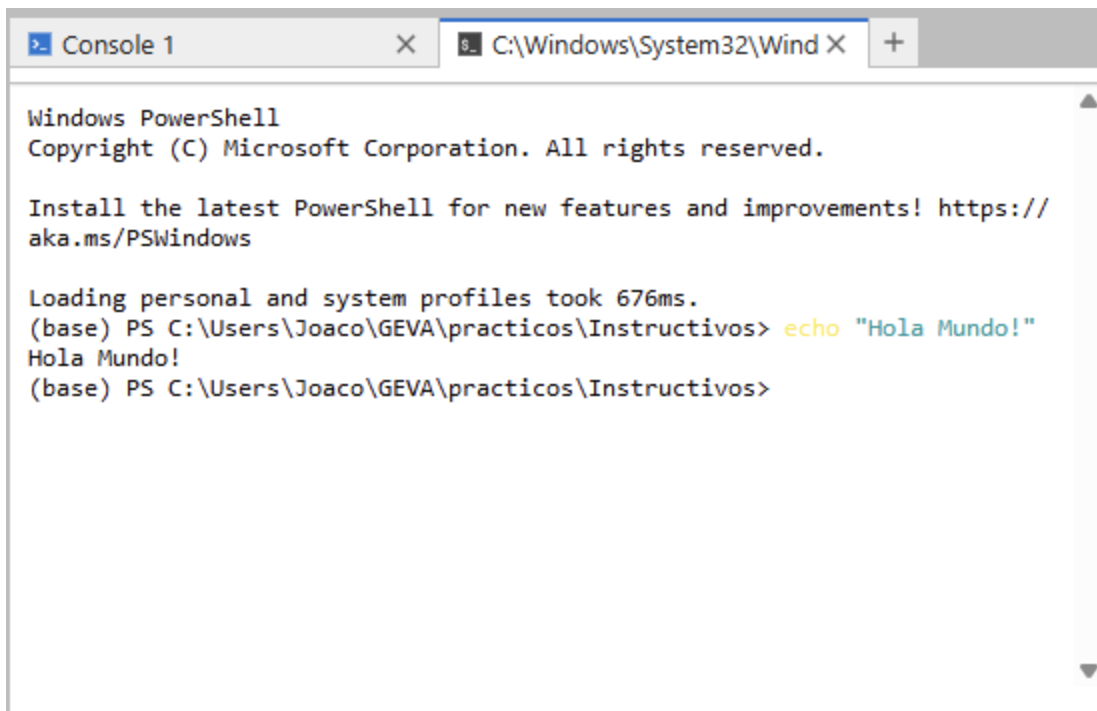


```
Python 3.11.4 | packaged by Anaconda, Inc. | (main, Jul 5 2023, 13:47:18) [MSC v.1916 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.16.1 -- An enhanced Interactive Python. Type '?' for help.

[1]: print("Hola mundo!")

      Hola mundo!
```

- **Other** : Permite abrir la terminal de sistema, una de las opciones más importantes para este curso, con la cual podremos ejecutar código de **shell bash** dentro de Jupyter Lab. Esto te permitirá ejecutar comandos, mover archivos, lanzar programas y gestionar tu entorno de trabajo en el servidor. También agrupa herramientas adicionales como editores de texto y visualizadores de archivos. Además incluye acceso a extensiones instaladas que pueden agregar nuevos tipos de documentos.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

Loading personal and system profiles took 676ms.
(base) PS C:\Users\Joaco\GEVA\practicos\Instructivos> echo "Hola Mundo!"
Hola Mundo!
(base) PS C:\Users\Joaco\GEVA\practicos\Instructivos>
```

Introducción general a Jupyter Notebook

¿Qué es un Jupyter Notebook?

Un **Jupyter Notebook** es un documento interactivo que combina código ejecutable, texto explicativo, visualizaciones y resultados en un mismo archivo. Imagínalo como un "cuaderno digital" donde puedes escribir código, ejecutarlo inmediatamente, ver los resultados, y documentar todo tu proceso de análisis paso a paso.

Los notebooks están organizados en **celdas**, que son bloques independientes donde puedes escribir código o texto explicativo. Esta estructura celular te permite ejecutar tu código de manera incremental, probando y refinando cada paso antes de continuar.

Filosofía de trabajo: Análisis iterativo

Los notebooks están diseñados para el **análisis exploratorio y iterativo**. A diferencia de un script tradicional que ejecutas de principio a fin, en un notebook puedes:

- Ejecutar una pequeña porción de código y ver inmediatamente el resultado
- Modificar y re-ejecutar celdas específicas sin afectar el resto
- Intercalar explicaciones entre código para documentar tu razonamiento
- Visualizar datos y gráficos directamente en el documento
- Experimentar con diferentes enfoques sin perder trabajo previo

Conceptos clave para empezar

Cada notebook está compuesto por **celdas** que pueden ser de dos tipos principales:

- **Celdas de código:** Donde escribes el código que se ejecuta.

```
# Ejemplo de celda de código
print("¡Hola, mundo!")
x = 2 + 3
print(f"El resultado es: {x}")
```
- **Celdas Markdown:** Donde escribes texto explicativo con formato. Estas sirven para documentación, explicaciones y formato de texto. Algunas de las opciones de formatos disponibles:
 - Texto formateado con ****negrita****, **cursiva**, ``código inline``
 - Títulos con `#`, `##`, `###`
 - Listas, enlaces, tablas
 - Fórmulas matemáticas con LaTeX: $E = mc^2$
 - HTML básico

El notebook funciona en dos modos de edición distintos:

- **Modo edición** (borde verde): Cuando estás escribiendo dentro de una celda, se activa con `Esc`
- **Modo comando** (borde azul): Cuando tienes una celda seleccionada pero no estás escribiendo

Para convertir una celda puede usar los siguientes atajos cuando estés en modo comando:

- **Code** → **Markdown**: Presiona **M**
- **Markdown** → **Code**: Presiona **Y**

El **kernel** es el proceso que ejecuta tu código (Julia, Python, R, etc). Mantiene en memoria todas las variables y funciones que defines, permitiendo que las celdas "se comuniquen" entre sí.

Barra de Herramientas

Esta es la **barra de herramientas principal** de Jupyter Notebook, que proporciona acceso rápido a las funciones más utilizadas:



La descripción de cada ícono junto con su acceso directo es el siguiente:

- : **Guardar notebook** - **Ctrl + S**
- : **Insertar celda debajo** - **B** (en modo comando)
- : **Cortar celda** - **X** (en modo comando)
- : **Copiar celda** - **C** (en modo comando)
- : **Pegar celda** - **V** (en modo comando)
- : **Ejecutar celda** - **Shift + Enter**
- : **Interrumpir kernel** - **I + I** (1 dos veces)
- : **Reiniciar kernel** - **0 + 0** (cero dos veces)
- : **Reiniciar y ejecutar todo** - Sin atajo directo

Atajos para la gestión de celdas

En modo comando también se pueden usar los siguientes atajos:

- : **Borrar celda** - **D + D** (presionar **D** dos veces)
- : **Deshacer borrado de celda** - **Z**
- : **Convertir a código** - **Y**
- : **Convertir a Markdown** - **M**
- : **Convertir a raw (texto plano)** - **R**

-   : **Moverse entre celdas** –  o 
-   : **Seleccionar varias celdas** –  +  / 
-  : **Mostrar lista completa de atajos** – 
-  : **Combinar celdas seleccionadas** –  + 

Comandos de sistema en notebooks

Además de ejecutar código en python, es posible usar  para ejecutar comandos de bash/terminal dentro de una celda de código:

```
In [ ]: # Ejemplos con !
!pwd          # Ver directorio actual
!ls -la       # Listar archivos
!pip install pandas # Instalar paquetes
!python mi_script.py # Ejecutar script externo
```

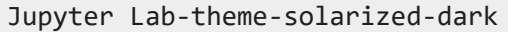
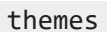
Para ejecutar múltiples comandos de bash en una celda, usa la "magic command"  :

```
In [ ]: %%bash
echo "Iniciando procesamiento..."
mkdir datos_temp
cd datos_temp
wget https://ejemplo.com/datos.csv
echo "Descarga completada"
```

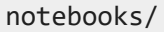
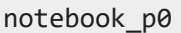
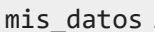
Otras magic commands útiles

```
In [ ]: %%time      # Medir tiempo de ejecución de la celda
%%html      # Renderizar contenido HTML
%%javascript # Ejecutar código JavaScript
%who        # Listar variables definidas
%whos       # Información detallada de variables
%history     # Ver historial de comandos
```

Ejercicios Jupyter Lab.

Ejercicio 1: Busca e instala la extensión . Luego ingresa a la sección  dentro de  y elige tu preferido

Ejercicio 2: Usando la interfaz de jupyter lab crea las siguientes carpetas dentro de tu directorio de trabajo: , , 

Ejercicio 3: Dentro de la carpeta  crea un notebook usando el launcher de Jupyter Lab. Renombrala a . Crea un archivo de texto en la carpeta  llamado .

Ejercicio 4: Ejecuta el siguiente código dentro de una celda del `notebook_p0` :

```
# Ejemplo de celda de código
print("¡Hola de nuevo, mundo!")
x = ((2 + 3) * 2) ** (1/2)
print(f"El resultado es aproximadamente: {x:.3f}")
```

Ejercicio 5: Mide el tiempo de ejecución del siguiente código

```
x = 11 ** 8
y = 13 ** 6
print(f"La proporción es de : {x/y}")
```

Ejercicio 6: Crea una nueva celda y escribe un texto que tenga un título, palabras en negrita

Ejercicio 7: Ejecuta el siguiente código de bash en una nueva celda

```
echo "Quien sos?"
echo "Yo soy $(whoami)"
```

Ejercicio 8: Guarda los resultados del notebook y luego exportalos en formato HTML usando `File -> Save and Export Notebook as -> HTML`