

# Guía de instalación y ejecución de Python

(En el marco del PROYECTO DE EXTENSIÓN DE BIOINFORMÁTICA EN EL AULA)

## El lenguaje de programación Python

Nuestro curso tiene como herramienta principal de programación al lenguaje **Python**. Tenemos distintas alternativas para usarlo. Podés instalar Python siguiendo los pasos mencionados a continuación, específicos del sistema operativo con el que trabajes.

*Nuestra recomendación es instalar la distribución de Python llamada Anaconda, ya que incluye por defecto muchas bibliotecas útiles y una interfaz amigable para el desarrollo. Si aceptás nuestra sugerencia, quizás quieras saltar lo que sigue y pasar a la siguiente sección.*

### Windows

Podés descargar la versión 3.8.5 de la distribución oficial de Python desde el sitio web oficial: <https://www.python.org/downloads/>. Después de descargar el archivo \*.exe, debés ejecutarlo y seguir las instrucciones.

### Linux

Es muy probable que tengas instalado Python. Para verificar si es así, y qué versión tenés, podés abrir una consola o terminal y tipear el siguiente comando:

```
python3 --version
```

Si no tenés Python o si querés instalar una versión diferente, podés hacerlo desde la terminal, tipeando en ella lo siguiente:

```
sudo apt-get install python3.8
```

### MacOS

Podés descargar la versión 3.8.5 de la distribución oficial de Python desde el sitio web oficial: <https://www.python.org/downloads/>. Después de descargar el archivo \*.pkg debés ejecutarlo y seguir las instrucciones.

Es importante recordar la ruta (el directorio) donde se ha instalado Python, para poder ejecutarlo después. Algo para tener en cuenta: en la segunda pantalla del asistente de instalación, llamada "Customize" o "Personalizar", asegurate de ir hacia abajo y elegir la opción "Add python.exe to the Path" o "Agregar python.exe a la variable de entorno PATH".

## Jupyter Notebook y Anaconda: Una interfaz amigable para correr Python

Jupyter es un intérprete de comandos para el lenguaje de programación Python. Se trata de un entorno amigable e interactivo, que facilita el proceso de programación. No vemos solo la terminal negra de Python, sino que podemos usar botones para navegar nuestra PC, ejecutar programas, copiar y pegar código, etc. Jupyter tiene como ventajas que posee incorporadas bibliotecas muy útiles para manipulación de datos científicos, un formato de navegación de datos amigable y celdas de ejecución por partes de nuestro código. Su instalación y uso son gratuitas y existen distribuciones para todos los sistemas operativos más comunes (DOCUMENTACION: <https://docs.anaconda.com/anaconda/install>).

Windows (Documentación: <https://docs.anaconda.com/anaconda/install/windows> )

- Descargar el programa de instalación desde:  
[https://repo.anaconda.com/archive/Anaconda3-2020.07-Windows-x86\\_64.exe](https://repo.anaconda.com/archive/Anaconda3-2020.07-Windows-x86_64.exe)  
(para sistemas operativos de 64 bits) o  
<https://repo.anaconda.com/archive/Anaconda3-2020.07-Windows-x86.exe>  
(para sistemas operativos de 32 bits).
- Hacer doble click sobre el archivo ejecutable que se acaba de descargar.
- Leer los términos de licencia y hacer click en “I Agree”.
- Seleccionar ‘install for all users’ y dar click al botón ‘Next’. **OJO!** Recordar que puede ser útil realizar la instalación para todos los usuarios del equipo, y usando un path de instalación sin espacios (por ejemplo: `C:\Users\Yo\Anaconda3`).
- Seleccionar la carpeta de destino y dar click al botón de ‘Next’.
- Elegir el PATH como default Python 3.8
- Clickear el botón ‘Install’ y luego ‘Next’.
- Luego clickear el botón ‘Skip’ para instalar Anaconda sin VS Code.
- Finalizar la instalación clickeando ‘Finish’.

La interfaz se abre buscando en el menú de Inicio la aplicación Anaconda Prompt. Haciendo click se abrirá una consola (una ventana negra), donde debés escribir `jupyter notebook` y dar enter. Se abrirá una nueva pestaña en el navegador de Internet con la interfaz de Jupyter.

■ Anaconda Prompt

```
(base) C:\Users\Leand>jupyter notebook
```

Linux (Documentación: <https://docs.anaconda.com/anaconda/install/linux> )

- Descargar el script de instalación desde:  
[https://repo.anaconda.com/archive/Anaconda3-2020.07-Linux-x86\\_64.sh](https://repo.anaconda.com/archive/Anaconda3-2020.07-Linux-x86_64.sh)
- Abrir una terminal de Ubuntu e ir hasta la carpeta donde se descargó el archivo, por ejemplo:  
`cd Descargas`
- Ejecutar luego los siguientes comandos, uno a continuación del otro:  
`chmod +x Anaconda3-2020.07-Linux-x86_64.sh`  
`./Anaconda3-2020.07-Linux-x86_64.sh`
- Leer los términos de licencia y hacer click en “I Agree” e ingresar ‘YES’ y ENTER.
- Seguir la instalación e ingresar ENTER cuando solicite instalar en la carpeta por defecto.
- Cuando el instalador solicita “¿Desea que el instalador prepare la ubicación de instalación anaconda <2 o 3> en PATH en su /home/<user>/bashrc?”, ingresar “Sí”.
- Puede ingresar “No” si el instalador pregunta si deseamos instalar el programa VSCode.
- Luego cerrar y reabrir la terminal y ejecutar:  
`source ~/.bashrc`

En Linux la interfaz puede ser abierta ejecutando en la terminal de Unix:

```
jupyter notebook
```

Se abrirá en una nueva pestaña en el navegador de Internet con la interfaz de Jupyter.

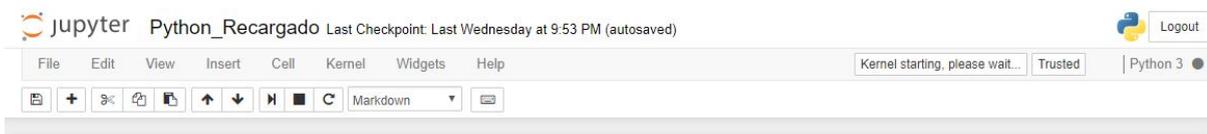
La interfaz de Jupyter Notebooks en el navegador tiene el siguiente aspecto:



Como podés ver, la interfaz te permite explorar los archivos que tengas en tu PC (botón: ‘Files’), generar archivos nuevos de texto (botón: ‘New → Text File’) o de tipo Notebook, que son archivos con extensión ‘\*.ipynb’ (botón: ‘New → Python 3’), donde se almacena la información de la sesión en la que se trabajó.



Accediendo a una notebook, el entorno de Jupyter nos permite trabajar con estos archivos de forma interactiva: creando nuevas celdas donde ejecutar el código que queremos probar (botón: ‘+’), eliminando celdas que no queramos conservar en el archivo (botón: ‘cut’, las tijeras), o guardando todo lo escrito (botón: ‘guardar’, el disquete), etc. Para ejecutar una celda con el código que hemos escrito solo tenemos que hacer SHIFT + ENTER. Aquí no veremos el prompt de Python, pero sabremos cuando está corriendo nuestra celda porque veremos al costado ‘In[\*]’ y al finalizar la corrida de la celda se verá entre corchetes el número de celdas ejecutadas, por ejemplo ‘In[1]’.



Además de código ejecutable, una celda puede contener sólo texto. Jupyter resulta muy útil cuando se utiliza python para el análisis de datos científicos, ya que la combinación de celdas de texto y celdas de código nos permite registrar cada uno de los pasos que vamos llevando a cabo junto con los resultados intermedios que obtenemos. Así, actúa como entorno de desarrollo y al mismo tiempo como cuaderno (‘notebook’) de laboratorio.

## Y si tenés un teléfono inteligente y un rato libre en el tren...

¡Buena noticia! Existen aplicaciones gratuitas que instalan una consola de Python en tu teléfono celular. Te permitirán correr todos los comandos de Python, aunque tienen limitaciones para el uso de bibliotecas externas. Solo debés buscar en tu tienda 'Python' y descargar la app correcta. Según el sistema operativo que uses, recomendamos:

- Android: QPython 3L, Pydroid 3
- iOS y Windows Phone: Python 3

## ¿Y si no puedo instalar nada?

Existen las consolas en línea, que te permiten correr Python como si estuviese instalado en tu PC, pero online. Son completamente gratis (bueno, siempre que tengas internet!).

Recomendamos usar la plataforma **Colaboratory** de Google, que pone a disposición la mayoría de las bibliotecas interesantes para bioinformática en un entorno Jupyter online muy cómodo y accesible. Se puede acceder con una cuenta de Google registrándose a través de <https://colab.research.google.com/>

Como opciones más sencillas y útiles para pruebas rápidas, tenemos repl.it (<http://repl.it/languages/python3>) y CodingGround ([https://www.tutorialspoint.com/execute\\_python3\\_online.php](https://www.tutorialspoint.com/execute_python3_online.php)), aunque hay otras.

## "Aún el camino más largo siempre comienza con el primer paso" (Lao Tse)

El primer paso para poder hacer tu primer programa es abrir la consola de Python, tu App del teléfono o consola en línea, lo que tengas a mano para arrancar! Si estás desde tu PC, seguí estos pasos.

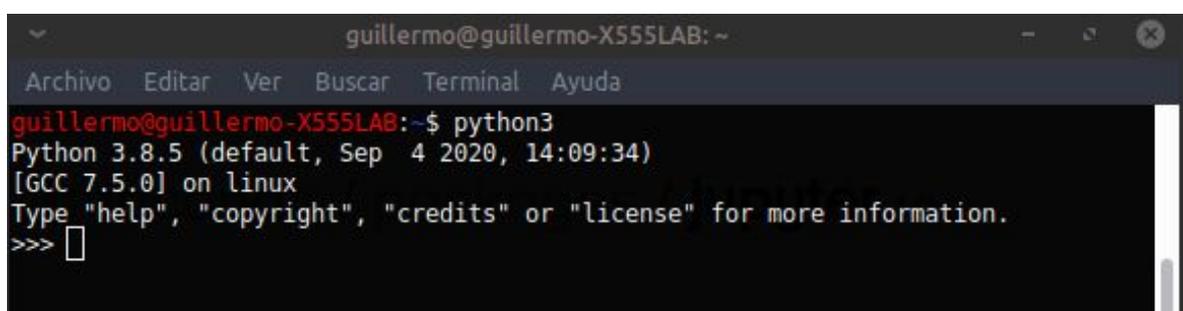
### Windows

La interfaz se abre buscando en el menú de Inicio la aplicación Anaconda Prompt y haciendo click se abrirá una consola (una ventana negra), donde debés escribir *jupyter notebook* y dar ENTER. Se abrirá una nueva pestaña en el navegador de Internet con la interfaz de Jupyter. Abrió un nuevo notebook (botón: 'New → Python 3') y ¡listo! Si querés trabajar desde la consola, sin la interfaz interactiva, buscá en el inicio Python 3.8 y hacé click en la aplicación Python Prompt. Verás lo que se denomina 'prompt', un símbolo que indica que el programa está listo para recibir una nueva orden; en el caso de Python, el prompt está formado por tres signos mayor '>>>'.  
>>>

### Linux

Para abrir la interfaz, basta con abrir una terminal y escribir:

`python3`



```

guillermo@guillermo-X555LAB: ~
Archivo Editar Ver Buscar Terminal Ayuda
guillermo@guillermo-X555LAB:~$ python3
Python 3.8.5 (default, Sep 4 2020, 14:09:34)
[GCC 7.5.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>

```

## ¿Cómo ejecutar scripts de Python?

Por convención, los archivos de los scripts de python tienen extensión '.py'. Para ejecutar un script de python en la consola simplemente debemos escribir:

```
python3 <nombre_del_archivo.py> <argumentos>
```

Si el script requiere de argumentos para ser ejecutado, estos deben ser especificados luego del nombre del script. Sin embargo, no todos los scripts de python requieren argumentos. En Linux puede no ser necesario escribir 'python' antes del nombre del script. Esto es posible cuando la primera línea del archivo \*.py es `#!/usr/bin/env python3`, indicando que se trata de un archivo a correr con Python. Luego de hacer el archivo ejecutable (corriendo en consola de Linux la orden `chmod +x nombre_del_archivo.py`) podemos llamar al programa así:

```
./<nombre_del_archivo.py> <argumentos>
```

## INSTALACIÓN DE BIBLIOTECAS

### Comprobar que las bibliotecas están instaladas

La instalación de Anaconda incluye muchas bibliotecas, además de versiones de IPython y Jupyter. Sin embargo durante el transcurso de este curso utilizaremos algunas bibliotecas que son útiles en el tratamiento de datos científicos y que no se encuentran incluidas entre las instaladas por Anaconda o en Python oficial.

En este curso en particular requerimos de la instalación de Pandas, Seaborn, Scipy y Biopython. Algunas de ellas, como Pandas, pueden estar incluidas en las versiones de Python que hemos instalado. Antes de proceder a la instalación de cualquier biblioteca, siempre es necesario comprobar que éstas no estén ya instaladas. Para ello podemos abrir Python (por consola o en un notebook) y observar que no haya ningún mensaje de error al importar las bibliotecas con el siguiente comando:

```
import scipy, pandas, matplotlib, seaborn, Bio
```

### Linux - usando Anaconda

Las versiones de Python posteriores a 2.7 suelen incluir una herramienta para instalar bibliotecas denominada pip3. Sin embargo, siempre es importante verificar que esté instalada, lo cual puede hacerse desde la consola tipeando:

```
pip3 --version
```

De no contar con el paquete pip, pueden instalarlo siguiendo las instrucciones disponibles en la [documentación](#). Si pip ya está instalado, las bibliotecas requeridas se instalan fácilmente desde consola:

```
pip3 install <nombre_de_la_biblioteca>
```



### Windows - usando Anaconda

Casi todas las bibliotecas necesarias ya vienen pre-instaladas en esta distribución, siendo generalmente la excepción Biopython. Pueden verificar que esté instalada corriendo el siguiente comando dentro de Python:

```
import Bio
```

Si no está instalado, pueden obtener Biopython ejecutando desde la consola de Anaconda:

```
pip3 install biopython
```

Anaconda tiene otro instalador de paquetes llamado conda, que es similar a pip, pero no funciona bien con Biopython.

## Windows - usando Python oficial

Para instalar los paquetes necesarios hay que abrir una línea de comandos e ir a la carpeta de instalación (p.ej. `C:\Users\mi_usuario\AppData\Local\Programs\Python\Python3.8`). Luego se instalan las bibliotecas con pip. La instalación de algunas bibliotecas puede requerir herramientas adicionales que pueden descargarse desde la página de Microsoft: [.NET framework 4.5 o superior](#) (que suele estar ya instalado en cualquier PC que se use cotidianamente) y [Microsoft Visual C++ 14.2 standalone: Build Tools para Visual Studio 2019](#). Con estas bibliotecas instaladas, sólo resta hacer:

```
Scripts\pip.exe install pandas
Scripts\pip.exe install scipy
Scripts\pip.exe install matplotlib
Scripts\pip.exe install seaborn
Scripts\pip.exe install biopython
```

El mismo método de instalación mediante pip puede usarse para instalar Jupyter

```
Scripts\pip.exe install jupyter
```

Luego de la instalación de Jupyter podríamos usarlo ejecutando el siguiente comando desde una consola abierta en la carpeta de instalación de python:

```
Scripts\jupyter-notebook.exe
```

Al hacerlo, se abrirá en el navegador de internet predefinido la interfaz de jupyter. ¡Atención! Conviene ejecutar jupyter-notebook.exe desde otra carpeta o con el argumento '--notebook-dir' para que nuestro código no quede guardado en la carpeta de instalación de python.

## Bibliotecas que te pueden ser útiles para análisis de datos Biológicos

Existen algunas bibliotecas útiles para tareas más complejas o específicas. Estas pueden ser instaladas usando los siguientes tutoriales:

- ETE Toolkit: <http://etetoolkit.org/download/>
- Biopython: <https://biopython.org/>
- pandas: <https://pandas.pydata.org/>
- seaborn: <https://seaborn.pydata.org/>
- scikit-learn: <http://scikit-learn.org/stable/install.html>

### Nos veremos otra vez...

Hasta acá llegamos con esta guía ¡Esperamos tu participación en el próximo Concurso de Bioinformática para Escuelas Secundarias!

**¡Gracias totales!**

Confeccionaron este material: Ana Julia Velez Rueda, Nicolás Palopoli, Guillermo I. Benitez, Leandro Matías Sommese.