

LENGUAJE vs HERRAMIENTAS

Python es un **lenguaje de programación** sencillo, rápido y liviano y es ideal para aprender, experimentar, practicar, trabajar y compartir aplicaciones orientadas a las Finanzas y Ciencia de Datos.

Jupyter Notebook, Anaconda, la Terminal (o Consola o Línea de comandos) y GIT/GitHub, son **herramientas** que usamos para desarrollar código en Python. Ninguna de estas es indispensable, sino que nos asisten a la hora de programar, brindando diferentes funcionalidades.

A continuación definiremos con más detalle cada una, y veremos cómo usarlas. Al final de cada sección dejo un enlace con tutoriales en video en español e inglés que me parecieron los más adecuados.

Terminal

Los sistemas operativos (Windows, macOS, Linux, etc), cuentan con un programa que es normalmente conocido como **terminal**. Hoy en día toma la forma de una ventana en la que escribimos y leemos texto, para indicar diferentes comandos que la máquina ejecuta. Cualquier cosa que podemos hacer gráficamente (con el mouse y las ventanas de los programas), podemos hacerla escribiendo comando en la terminal.

El ejemplo más claro de esto, y que debemos comprender para poder desenvolvernos cómodamente en los módulos futuros, es navegar el **sistema de archivos**.

El sistema de archivos, es la “estructura” en la que están ordenados todos nuestros archivos y carpetas. De la misma forma que podemos crear, modificar, y borrar archivos y carpetas usando el mouse, el teclado y viendo todo gráficamente en la pantalla; podemos hacerlo usando comandos desde la terminal.

De igual forma podemos navegar entre esas carpetas (que se llaman directorios). Por ejemplo: La carpeta `icda-python-para-finanzas`, en mi máquina está adentro de la carpeta `Documents`. A su vez, la carpeta `icda-python-para-finanzas` tiene carpetas adentro suyo, como `tarea2-presentaciones`, `tarea3-variables`, etc.

Estos conceptos van a ser importantes para quienes quieran aprender a usar Git sin problemas, y aclaran un poco las diferencias entre las posibles maneras de abrir y trabajar Jupyter Notebooks.

[Tutorial recomendado para Windows](#)

[Tutorial recomendado para macOS](#)

Anaconda

Anaconda es un “kit de herramientas” de Python. Descargando Anaconda, contamos con muchas librerías y programas (como Jupyter Notebook) que podemos aprovechar para desarrollar código en Python.

Anaconda está orientada a la Ciencia de Datos, por lo que ya nos trae por defecto librerías para el análisis y visualización de datos, que vamos a usar de el Módulo 2 en adelante.

Jupyter Notebook

Jupyter Notebook es un programa que nos permite crear libretas interactivas basadas en la web. En estas libretas podemos ejecutar código en Python de manera secuencial e interactiva, sumando la posibilidad de agregar texto enriquecido para agregar información y contexto a nuestro código.

Cuando iniciamos Jupyter Notebook, “despertamos” un programa servidor que es el que va a realizar todo el procesamiento de datos, y un programa cliente, que es el que usamos para trabajar; “lo que vemos”.

Para iniciar Jupyter Notebook, podemos hacerlo de 2 formas principales:

- Buscando el programa desde el inicio de nuestro sistema operativo y haciendo click en el ícono. Esto “despierta” al servidor, y abre una nueva pestaña en nuestro navegador Web predeterminado. Tener en cuenta que se abre en el **directorio raíz de nuestro sistema operativo**; luego desde el navegador web podremos navegar hasta la carpeta donde se encuentran nuestros archivos (por ejemplo podemos ir desde el directorio raíz -> Documentos -> icda-python-para-finanzas).
- Desde la Terminal, con el comando `jupyter-notebook`. Se abre en el directorio en el que nos encontramos parados, por lo que podemos primero navegar hasta el directorio donde tenemos los archivos con los que queremos trabajar; o ejecutar el comando y buscar los archivos desde el navegador web.

[Tutorial recomendado](#)

GIT y GitHub

GIT es un Sistema de Control de Versiones. Lo que nos permite es trabajar de forma colaborativa en un mismo proyecto de software.

Al trabajar con GIT, creamos un repositorio, que sería una carpeta común y corriente, pero con unos atributos especiales que hacen que se pueda compartir y contribuir a la misma desde diferentes lugares, manteniendo siempre un registro de todo lo que se hace sobre esta carpeta.

GitHub, es una plataforma que nos permite crear y compartir repositorios. Como si fuera un “Facebook para GIT” en el que contamos con un usuario.

En el curso, trabajamos con un repositorio GIT, mantenido en la nube gracias a GitHub. Este repositorio es propiedad del docente y por el momento está orientado a que ustedes puedan interactuar con él en “modo lectura”.

Cuando usan el comando

`git clone https://github.com/fedeturi/icda-python-para-finanzas.git` están creando un **clon del repositorio** de la nube en sus maquinas. En sus máquinas lo ven como una carpeta más. Es importante reconocer donde estan parados cuando ejecutan este comando.

Por ejemplo si escriben este comando en la terminal adentro de la carpeta Documentos, el repositorio se va a clonar dentro de esa carpeta, entonces la estructura de archivos les quedaria asi:

```
raiz
|--- Documentos
    |--- icda-python-para-finanzas
```

Para ejecutar cualquier comando de GIT (los que hacen falta para actualizarlo siempre que se suba algo nuevo), hay que tener en cuenta dos cosas:

- Deben ejecutar el comando ADENTRO del repositorio, sino no tendrá efecto.
- Para no tener conflictos, les recomiendo que no modifiquen ninguno de los archivos del repositorio; y que trabajen con sus notebooks en una carpeta SEPARADA; porque cuando ejecuten el siguiente comando, van a descargar los archivos del repositorio, TAL CUAL aparecen en la nube. Es decir que si ustedes hacen algun cambio en el repositorio, y despues actualizan para ver los archivos nuevos, podrian perder los suyos, Por eso trabajen por separado.

El comando que deben usar para actualizar su clon del repositorio siempre que se suba material nuevo es `git pull` (desde ADENTRO del repositorio local).

[Tutorial recomendado](#) (en el minuto 7:00 explica este comando, que es el unico que van a usar por el momento).