

Instalación de python y principales librerías

Francisco Gárate Santiago - fgarate@ucm.es

UCM

Master en Ciencias Actuariales y Financieras (2024-2025)



PIGS

Python, Interface, Git y Shell

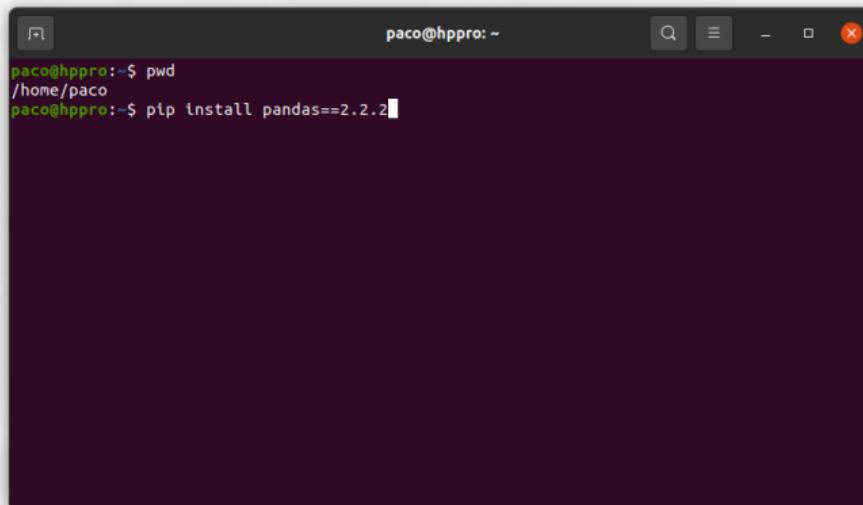


Shell

Terminal, Consola, Línea de comandos, etc.

La terminal¹ va a ser imprescindible. Con ella vamos a:

- Conocer si está correctamente instalado python
- Instalar todas las librerías necesarias
- Usar o crear nuestro propio repositorio de código



A screenshot of a terminal window titled "paco@hppro: ~". The window contains the following text:

```
paco@hppro:~$ pwd  
/home/paco  
paco@hppro:~$ pip install pandas==2.2.2
```

Realmente: **Terminal** es ventana que permite introducir comandos y mostrar su salida (emula), la **consola** es cualquier interfaz de control y monitorización (concepto mas amplio que la terminal) y **shell** es el programa que interpreta y ejecuta los comandos (bash, cmd.exe, powershell)

Python

comandos previos de testeo

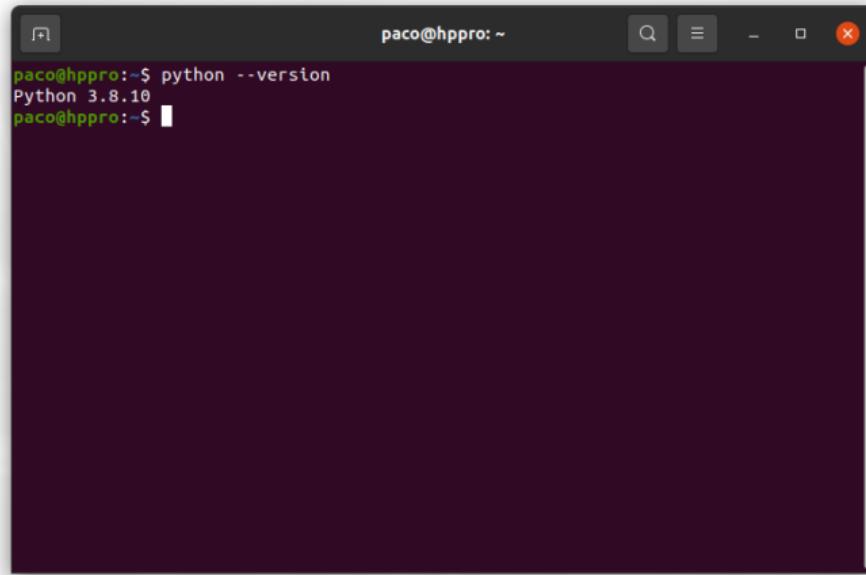
Comprobamos si tenemos ya instalado python desde la terminal:

- Chequear version: `python --version` o `python3 --version`
- Ver si, aunque esté instalado (solo para windows), está incluida su ruta en el path: `path` (*precaución: no añadir ningún parámetro a este comando*)
- Si no es la versión que deseamos, la recomendación es desinstalar python e instalar la versión deseada.

Python

comandos previos de testeo

Chequeo de versión:

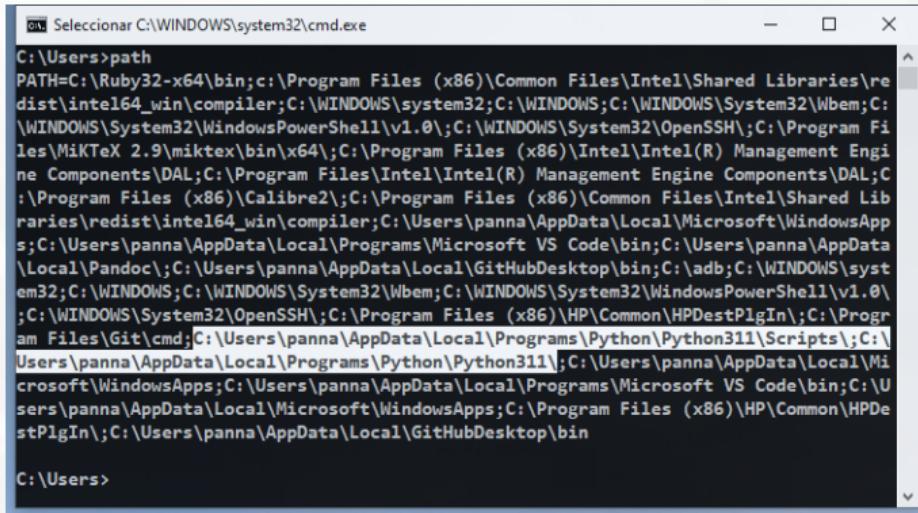


A screenshot of a terminal window titled "paco@hppro: ~". The window contains the following text:
paco@hppro:~\$ python --version
Python 3.8.10
paco@hppro:~\$ █

Python

comandos previos de testeo

Chequeo de path:



```
Selezionare C:\WINDOWS\system32\cmd.exe
C:\Users>path
PATH=C:\Ruby32-x64\bin;c:\Program Files (x86)\Common Files\Intel\Shared Libraries\redist\intel64_win\compiler;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;C:\WINDOWS\System32\OpenSSH\;C:\Program Files\MiKTeX 2.9\miktex\bin\x64\;C:\Program Files (x86)\Intel\Intel(R) Management Engine Components DAL;C:\Program Files\Intel\Intel(R) Management Engine Components DAL;C:\Program Files (x86)\Calibre2\;C:\Program Files (x86)\Common Files\Intel\Shared Libraries\redist\intel64_win\compiler;C:\Users\panna\AppData\Local\Microsoft\WindowsApps;C:\Users\panna\AppData\Local\Programs\Microsoft VS Code\bin;C:\Users\panna\AppData\Local\Pandoc\;C:\Users\panna\AppData\Local\GitHubDesktop\bin;C:\adb;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;C:\WINDOWS\System32\OpenSSH\;C:\Program Files (x86)\HP\Common\HPDestPlugIn\;C:\Program Files\Git\cmd;C:\Users\panna\AppData\Local\Programs\Python\Python311\Scripts\;C:\Users\panna\AppData\Local\Programs\Python\Python311\;C:\Users\panna\AppData\Local\Microsoft\WindowsApps;C:\Users\panna\AppData\Local\Programs\Microsoft VS Code\bin;C:\Users\panna\AppData\Local\Microsoft\WindowsApps;C:\Program Files (x86)\HP\Common\HPDestPlugIn\;C:\Users\panna\AppData\Local\GitHubDesktop\bin

C:\Users>
```

Python

Descargar python desde python.org

Instalación fácil: descargar la última versión del ejecutable desde www.python.org

The screenshot shows the Python.org homepage with a dark blue header. The navigation bar includes links for Python, PSF, Docs, PyPI, Jobs, and Community. The main content area features the Python logo and a search bar with a 'GO' button. A sidebar on the left contains a snippet of Python code for generating a Fibonacci sequence and a list of download options: All releases, Source code, Windows, macOS, Other Platforms, License, and Alternative Implementations. A prominent call-to-action button labeled 'Download' is visible. Below it, a box highlights 'Download for Windows' with a link to 'Python 3.10.4'. A note states that Python 3.9+ cannot be used on Windows 7 or earlier. A message at the bottom encourages users to explore other platforms and view the full list of downloads.

Python 3: Fib

```
>>> def fib(n):  
>>>     a, b =  
>>>     while a <  
>>>         pri  
>>>         a,  
>>>         print()  
>>>         fib(1000)  
0 1 1 2 3 5 8 1
```

All releases
Source code
Windows
macOS
Other Platforms
License
Alternative Implementations

Download for Windows

Python 3.10.4

Note that Python 3.9+ cannot be used on Windows 7 or earlier.

Not the OS you are looking for? Python can be used on many operating systems and environments.

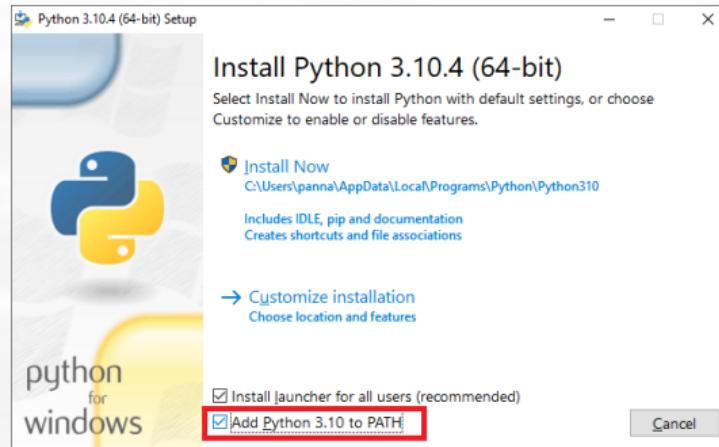
View the full list of downloads.

Python is a programming language that lets you work quickly

Instalación

Descargar python desde python.org

IMPORTANTE: Seleccionar **Add Python 3.XX to PATH**



En caso de no haber realizado este paso se debe añadir manualmente la ruta de Python en el PATH de windows (ver anexo de esta presentación), aunque la recomendación es desinstalar y volver a instalar marcando esta opción.

Librerías

pip

pip es el instalador recomendado, ya que instala los paquetes publicados en el Python Package Index o PyPi (<https://pypi.org>). Instalar una librería (o biblioteca²) con pip es muy simple. Ejemplo: `pip install numpy`

- Para actualizar: `pip install numpy --upgrade` o
`pip install numpy -U`
- Para desinstalar: `pip uninstall numpy`
- En el caso de querer instalar una versión específica de una librería:
`pip install numpy==1.19.0`
- Si queremos instalar varias librerías, estas pueden estar en un fichero externo e instalarse de la siguiente forma: `pip install -r requirements.txt`

Por último, ejecutando `pip list` muestra todas las librerías (y sus versiones) instaladas en nuestro equipo.

Librerías

- Listado de las principales librerías de python utilizadas en matematicas financieras y actuariales: **jupyter, matplotlib, notebook, numpy, numpy-financial, pandas, xlrd, xlwt, lxml, openpyxl, pyliferisk, scipy, statsmodels, sklearn, scikit-learn, dateutils, kedro, folium, geopy, prophet.**
- Para instalarlas, ejecutar en la línea de comandos del sistema:
`pip install nombre_libreria`
- O bien, para facilitar la instalación se puede ejecutar el siguiente comando³:
`pip install -r http://www.garpa.net/assets/requirements.txt -U`

Si **pip install** diera un “Error Fatal”, puede ser debido a que previamente hubieras instalado Anaconda en tu equipo. En este caso, ejecutar el comando **pip** desde la siguiente ruta: `cd C:\ProgramData\Anaconda3\Scripts`

Librerías



Librerías

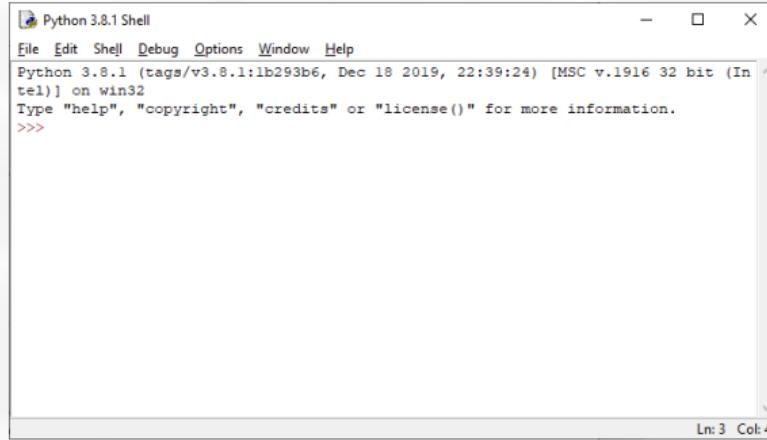
Crear mi propio requeriments.txt

- Para evitar que los cambios de versiones en las librerías causen incidencias en tu código, sobre todo en el tratamiento de fechas, la recomendación es definir y fijar las librerías que vamos a utilizar.
- Así, la actualización podrá ser una a una, chequeando que nuestro código siga funcionando.
- La recomendación es tener siempre a mano nuestro fichero de librerías (`requirements.txt`) y alojarlo en la carpeta donde estemos programando.
- Para ello, **git** nos será especialmente útil (lo veremos más adelante)

Entorno gráfico

IDE

- Python posee su propio **IDE** (Entorno de desarrollo integrado) que se incluye en la instalación inicial, denominado IDLE (Integrated Development Environment for Python).
- IDLE es un entorno gráfico muy elemental que permite editar y ejecutar código en Python.



Entorno gráfico

IDE

No obstante, lo habitual es instalar un entorno gráfico más amigable. Existen múltiples opciones:

- Visual Studio Code
- Jupyter
- Spyder (el más parecido a RStudio)
- Atom
- Sublime Text (gratuito aunque no de código libre)

Si bien, todos son muy parecidos, la tendencia es utilizar los conocidos notebooks originarios de Jupyter, los cuales permiten añadir código, texto y gráficos. Desde hace muy poco, Visual Studio Code soporta estos notebooks (formato **.ipynb**), así como servicios en la nube como Databricks.

Entorno gráfico

Jupyter Notebook

- El entorno gráfico utilizado en el curso será Jupyter, aunque puede ser utilizado cualquier otro que sea conocido por el alumno.
- El **proyecto Jupyter** (acrónimo de Julia, Python y R) engloba un conjunto de recursos orientados a la ciencia de datos y computación científica.
- Jupyter es 100 % software libre (BSD).
- El recurso más conocido es **Jupyter Notebook** (antes IPython notebook).
Jupyter Lab es la versión moderna del notebook.
- Son **consolas interactivas** accesibles desde el navegador web, con posibilidad de introducir texto, fórmulas y código.
- Los **navegadores** compatibles son: Firefox, Chrome, Safari y Opera.
- Los campos de texto utilizan formato Markdown con soporte de fórmulas en **LATEX**

Entorno gráfico

Jupyter: Instalación y uso

- Se instala igual que cualquier otra biblioteca: `pip install jupyter`
- Y se llama desde la linea de comando: `jupyter notebook` o `jupyter lab`
- Por defecto, se abrirá el navegador predeterminado con la ruta de la carpeta desde donde ha sido ejecutado jupyter.
- No es necesario disponer de conexión a internet.
- Es posible usar Jupyter desde su página oficial sin necesidad de instalación: <https://jupyter.org/try> o instalarse en un servidor compartido (Jupyterhub).

Entorno gráfico

Jupyter: Instalación y uso

The screenshot shows a Jupyter Notebook interface with the following details:

- Header:** jupyterhub Repaso Last Checkpoint: el viernes pasado a las 20:00 (autosaved)
- Toolbar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Run, Code, Logout, Control Panel, Trusted, Python 3, Memory: 165.4 MB
- Cells:**
 - In [1]: `import numpy as np`
 - In [2]: `np.pi`
Out[2]: 3.141592653589793
 - In [3]: `a = np.linspace(0,10,5)`
a
Out[3]: array([0., 2.5, 5., 7.5, 10.])
 - In [4]: `from funciones import guitarprimeraletra`
 - In [5]: `guitarprimeraletra('A123')`
Out[5]: '123'
 - In [6]: `from prueba import prueba`
 - In [7]: `prueba(8)`
Out[7]: 100.0
 - In [8]: `datos = np.random.normal(5, 0.5, 10015)`
datos
Out[8]: array([4.95299326, 4.64041255, 5.08828029, 4.70559452, 4.39180046, 5.58683945, 4.7452246 , 4.76162775, 4.61153497, 4.80425741, 5.1428888 , 4.58894283, 5.55275566, 5.43143908, 5.06759416, 4.95657354, 4.65135995, 4.97991867, 4.78919119, 5.31488087, 5.09160179, 5.04170889, 5.27462203, 5.03046101, 4.39417366, 5.35691724, 5.58055392, 5.80949338, 4.78649569, 3.88690749, 5.15763879, 4.90156976, 5.25132653, 4.5173623 , 5.04661651, 4.50494518, 4.47249727, 4.59589948, 4.62184807, 4.62681141, 4.66460831, 4.31850394, 4.59075055, 4.84425222, 4.86572727, 5.0246851 , 4.84426254, 4.78251211, 5.11381151, 4.86629472, 5.00000000, 4.00000000, 5.00000000, 5.00000000])

Entorno gráfico

Spyder IDE

Otro IDE recomendado (especialmente a aquellos usuarios acostumbrados a R-Studio por su similitud) es **Spyder**, gratuito y de código abierto :
<https://www.spyder-ide.org>.



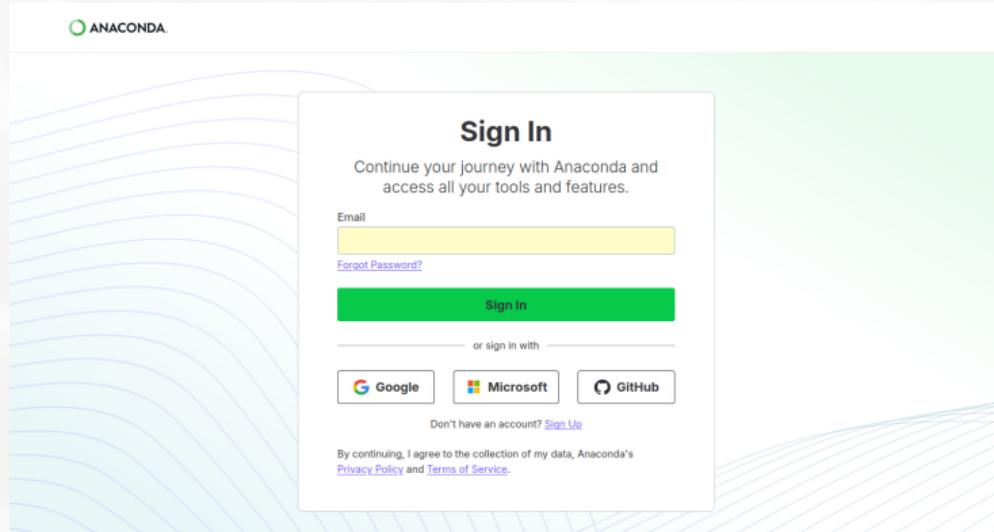
Entorno gráfico

Anaconda Cloud

El proyecto Anaconda, facilita una versión gratuita de JupyterHub:

<https://nb.anaconda.cloud>

Así, no hace falta la instalación de python si no desea ya que puede accederse desde esta página.



Entorno gráfico

Anaconda Cloud

The screenshot shows the Anaconda Cloud dashboard. At the top, there's a navigation bar with the Anaconda logo, a search bar, and user profile information. Below the header, there are three main tabs: "Org Profile" (Organization Dashboard), "Users" (Invite and manage your users), and "Subscriptions" (Manage payments and details). The main content area is titled "Explore Anaconda" and features a grid of nine icons, each representing a different tool or service:

- Notebooks (Icon: Jupyter logo)
- Learning (Icon: Graduation cap)
- Community (Icon: Earth globe)
- Docs (Icon: Document with lines)
- AI Navigator (Icon: Brain with circuit board)
- Distribution (Icon: Green circular logo)
- PyScript (Icon: Ghost-like face)
- EduBlocks (Icon: Blue block with 'edu' text)
- PythonAnywhere (Icon: Python logo with a person icon)
- Toolbox for Excel (Icon: Green 'Q' with a wrench)

Git

Repositorio

Git es un sistema de control de versiones distribuido que permite gestionar y rastrear los cambios en el código fuente de un proyecto de manera eficiente y colaborativa.

- Instalar git
- Github es el repositorio elegido por la amplia mayoría de desarrolladores.
- Desde hace tiempo, ya no existe limitaciones para crear repositorios privados, así que si queremos podemos usar para uso propio o compartir nuestros repositorios.
- No hace falta utilizar todas las funcionalidades de git (como fork, branch...)
- En esta sesión haremos uso de la función clone para copiar un repositorio

Kedro

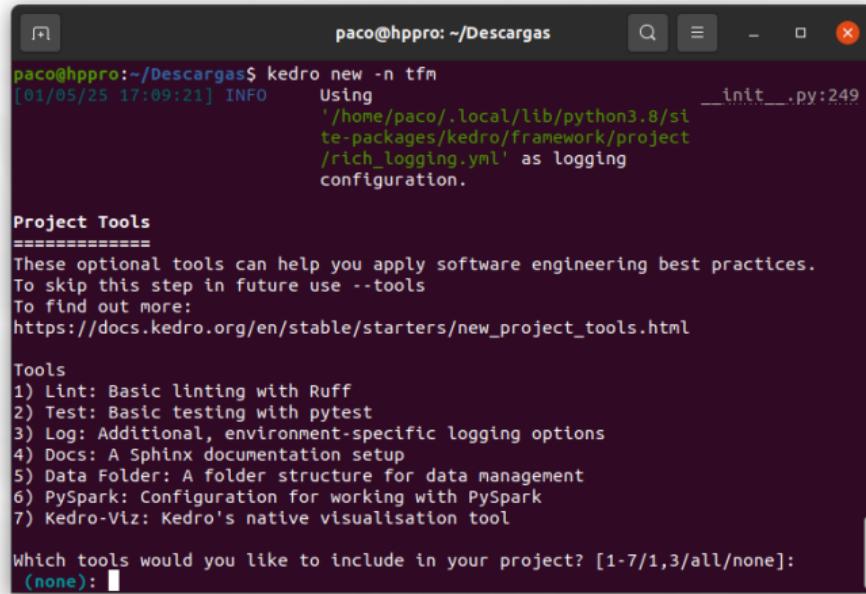
Poner orden en los datos

Kedro es una librería de Python diseñada para facilitar la creación de proyectos de ciencia de datos. Proporciona una estructura modular y coherente para gestionar flujos de trabajo, integrar datos de manera eficiente y asegurar la trazabilidad y reproducibilidad de los experimentos, promoviendo buenas prácticas en el desarrollo de proyectos de datos.



Kedro

Poner orden en los datos



```
paco@hpro:~/Descargas$ kedro new -n tfm
[01/05/25 17:09:21] INFO    Using      __init__.py:249
                     '/home/paco/.local/lib/python3.8/si
                     te-packages/kedro/framework/project
                     /rich_logging.yml' as logging
                     configuration.

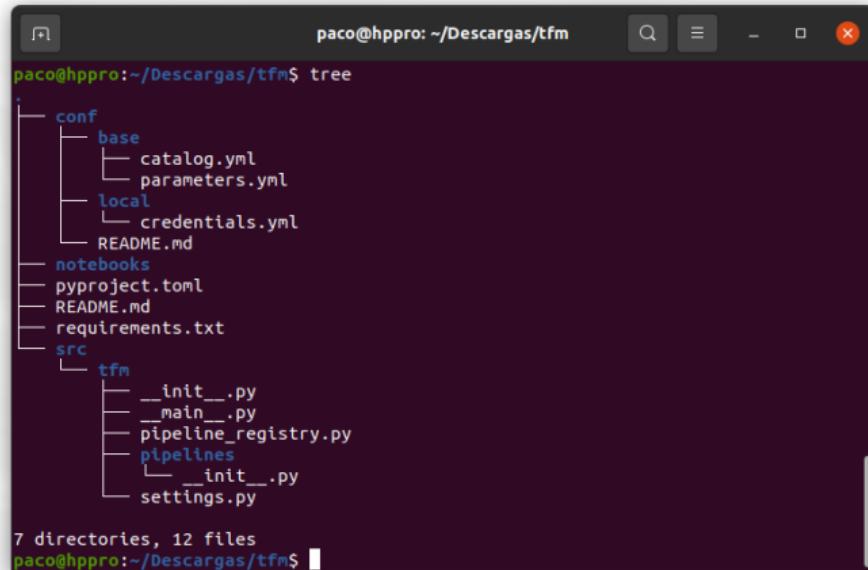
Project Tools
=====
These optional tools can help you apply software engineering best practices.
To skip this step in future use --no-tools
To find out more:
https://docs.kedro.org/en/stable/starters/new\_project\_tools.html

Tools
1) Lint: Basic linting with Ruff
2) Test: Basic testing with pytest
3) Log: Additional, environment-specific logging options
4) Docs: A Sphinx documentation setup
5) Data Folder: A folder structure for data management
6) PySpark: Configuration for working with PySpark
7) Kedro-Viz: Kedro's native visualisation tool

Which tools would you like to include in your project? [1-7/1,3/all/none]:
(none):
```

Kedro

Poner orden en los datos



```
paco@hppro:~/Descargas/tfm$ tree
.
├── conf
│   ├── base
│   │   └── catalog.yml
│   │   └── parameters.yml
│   └── local
│       └── credentials.yml
│       └── README.md
└── notebooks
└── pyproject.toml
└── README.md
└── requirements.txt
└── src
    └── tfm
        ├── __init__.py
        ├── __main__.py
        ├── pipeline_registry.py
        └── pipelines
            └── __init__.py
        └── settings.py

7 directories, 12 files
paco@hppro:~/Descargas/tfm$
```

Git

Git: `git clone https://github.com/user/repo` (*ojo, crea carpeta repo*)

- ver cambios: `git status`
- descargar cambios: `git pull`
- Descartar cambios: `git reset -hard`
- añadir al lanzador: `git add code.py` o `git add .`
- crear commit: `git commit -m 'nuevo desarrollo'`
- subida (*pide user y token*): `git push`

Fichero `.gitignore`:

- ignora solo 1 fichero: `/file.txt`
- ignora todo: `/*`
- excluir de la ignorancia a un fichero: `!conf.txt`
- para ignorar todas las carpetas: `*/`

Instalación en macOS

- **macOS** dispone de la versión 2.7 que no debe ser desinstalada.
- Para instalar python 3, descargar el .pkg desde www.python.org
- Ver instrucciones en el siguiente link:
<http://www.garpa.net/installar-python-en-macos.html>



Anexo

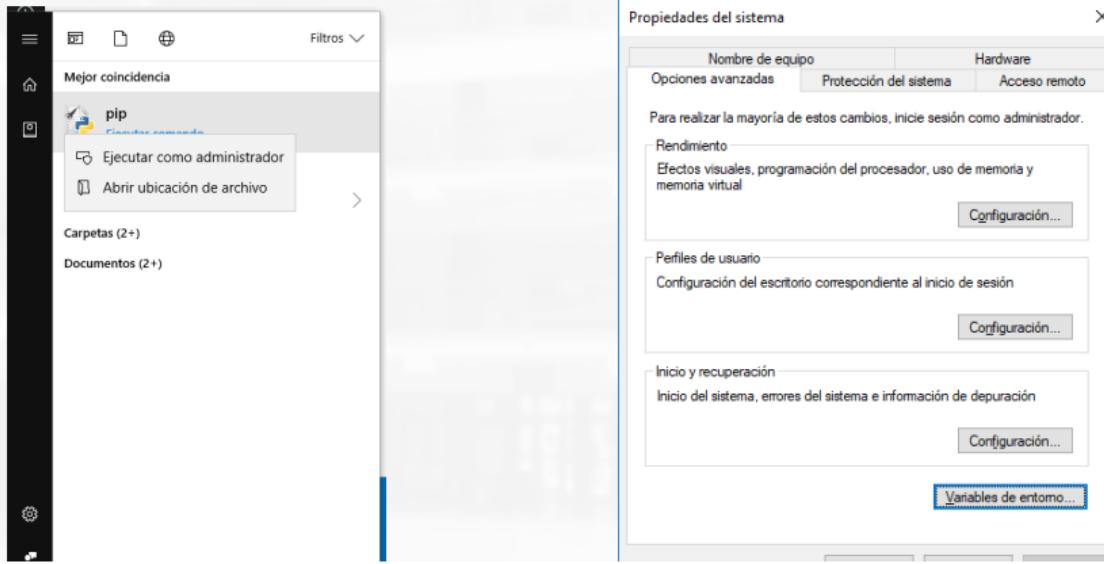
Incluir python en el \$PATH\$ de Windows

Añadir al PATH (sólo si no lo hemos hecho al inicio de la instalación)

Añadir al PATH (sólo si no lo hemos hecho al inicio de la instalación)

Incluir python en el \$PATH\$ de Windows

Copiar la ruta donde esta instalado python (botón derecho “Abrir ubicación de archivo”), y en “Windows > Editar las variables de entorno > Variables de Entorno”:



Añadir al PATH (sólo si no lo hemos hecho al inicio de la instalación)

Incluir python en el \$PATH\$ de Windows

“Path > Editar > Nuevo” y copiar la ruta de python. Normalmente:

%USERPROFILE%\AppData\Local\Programs\Python\Python38-32\

%USERPROFILE%\AppData\Local\Programs\Python\Python38-32\Scripts\

