

**Reto 6**  
**Mariena Rodriguez Oquendo**  
**CC. 1085331871**

---

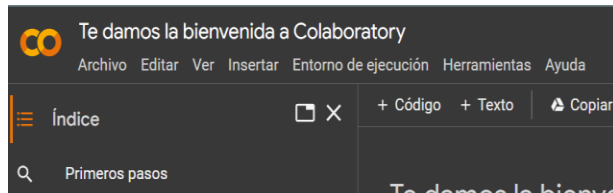
**1. Alistamiento de un entorno de código**

**1. Instalación de Numpy**

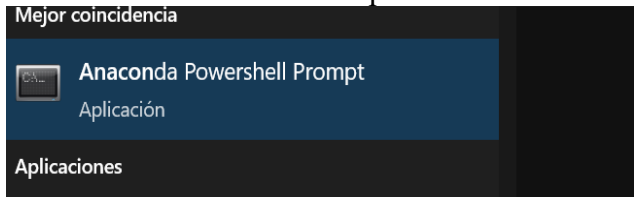
Para comenzar con la instalación de la librería **numpy** se deben seguir los siguientes pasos:

- a. **Medio:** abrir Colab (figura 1) o un programa para trabajar desde escritorio como Anaconda Powershell Prompt (figura 2).

**Figura 1**  
Interfaz Colab

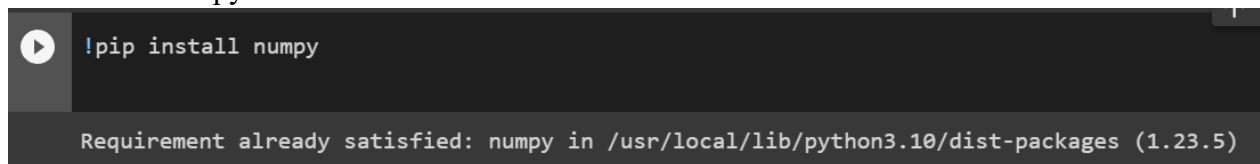


**Figura 2**  
Anaconda Powershell Prompt



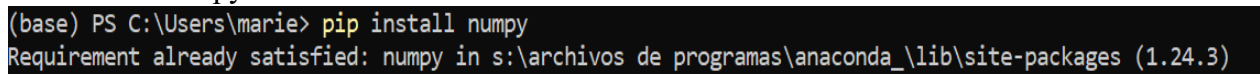
- b. **Instalación de numpy:** el proceso para la instalación para Colab es mediante la escritura en la consola de la sentencia: **!pip install numpy** (Ingenia, 2023), obteniendo así el siguiente resultado:

**Figura 3**  
Instalación numpy en Colab



Para el caso de Anaconda es un proceso similar, solo que en este caso en la sentencia se elimina el interrogante, así **pip install numpy** (Ingenia, 2023), obteniendo un mensaje de esta manera:

**Figura 4**  
Instalación numpy en Anconda



Para hacer uso de la librería se escribe en Colab o Spyder (Anaconda) la sentencia **import numpy** acompañado de **as np** de manera que se pueda utilizar las funciones y estructuras de la librería de una manera más rápida, pero esto último es opcional.

## 2. Instalar el paquete del toolbox de redes neuronales tensorflow, versión 2.7.0.

En este caso para Colab y Anaconda se tiene la misma estructura de sentencia que en el apartado anterior, pero con algunas variación como se logra observar en la nueva sentencia se pondrá para la instalación en Colab **!pip install tensorflow==2.7.0** y para Anaconda **pip install tensorflow==2.7.0** (Ingenia, 2023). El proceso se muestra a continuación, pero se ha encontrado que no existe una versión 2.7.0 para *tensoflow*, por lo cuál se recomienda las versiones que se muestran en las figuras para cada uno de los programas.

**Figura 5**

Instalación tensorflow en Colab

```
[4] pip install tensorflow==2.7.0
ERROR: Could not find a version that satisfies the requirement tensorflow==2.7.0 (from versions: 2.8.0rc0, 2.
ERROR: No matching distribution found for tensorflow==2.7.0
```

**Figura 6**

Instalación tensorflow en Anaconda


```
(base) PS C:\Users\marie> pip install tensorflow==2.7.0
ERROR: Could not find a version that satisfies the requirement tensorflow==2.7.0 (from versions: 2.12.0rc0, 2.12.0rc1,
.12.0, 2.12.1, 2.13.0rc0, 2.13.0rc1, 2.13.0rc2, 2.13.0, 2.13.1, 2.14.0rc0, 2.14.0rc1, 2.14.0)
ERROR: No matching distribution found for tensorflow==2.7.0
```

## 3. Actualizar la librería matplotlib en su última versión.

La actualización de la librería se realiza bajo la estructura **!pip install -- upgrade matplotlib** (Ingenia, 2023), para Colab y **pip install -- upgrade matplotlib** para Anaconda. En este caso se tiene que en el caso se Colab no se ha encontrado una versión de actualización, por lo cuál actualmente se estaba trabajando con la última versión, caso contrario a lo que muestra en Anaconda que la actualización de la librería se realizó de manera correcta, ya que al parecer no estaba actualizada.

**Figura 7**

Actualización de matplotlib en Colab

```
 !pip install -- upgrade matplotlib
ERROR: Could not find a version that satisfies the requirement upgrade (from versions: none)
ERROR: No matching distribution found for upgrade
```

**Figura 8**

Actualización de matplotlib en Anaconda

```
(base) PS C:\Users\marie> pip install -- upgrade matplotlib
Usage:
pip install [options] <requirement specifier> [package-index-options] ...
pip install [options] -r <requirements file> [package-index-options] ...
pip install [options] [-e] <vcs project url> ...
pip install [options] [-e] <local project path> ...
pip install [options] <archive url/path> ...
```

#### 4. Seleccionar en la librería matplotlib el método figure

Para este paso será necesario abrir el programa de Spyder asociado a Anaconda para lograr importar el método figure de la librería, mientras en el caso de Colab se podrá seguir usando esa ventana sin inconvenientes, pero se sugiere crear un nuevo bloque de código para este apartado, la estructura que se tendrá como código será el mismo para ambos casos correspondiente a como se muestra en la figura 9 y 12 (Hunter et al., 2012):

**Figura 9**

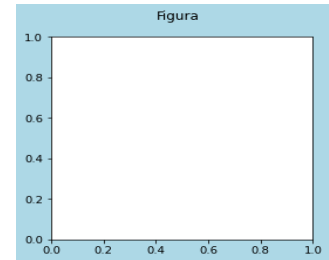
Selección de método figure en matplotlib en Spyder

```
# Importar la librería Matplotlib
import matplotlib.pyplot as plt

# Creación de una figura en Matplotlib
fig, ax = plt.subplots(figsize=(4, 4), facecolor='lightblue')
fig.suptitle('Figura')
```

**Figura 10**

Figura obtenida en Spyder



Con el proceso anterior, se logra obtener la figura observada, donde se pueden hacer cambios como el tamaño de 4 x 4 o el color de fondo que en este caso es de azul claro, así mismo se puede cambiar el título de la Figura, así como otras características como dpi: puntos por pulgada, edgecolor: color del borde, entre otros que se pueden apreciar en la figura 11.

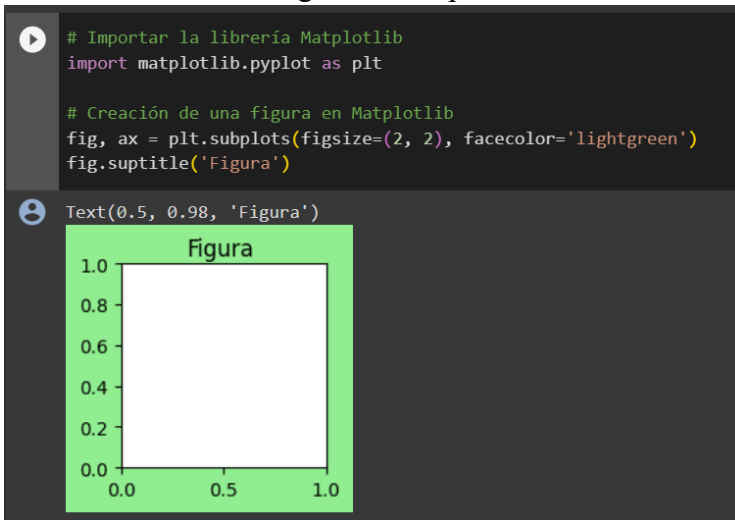
**Figura 11**

Características asociadas a el método figure

```
Definition : Figure(figsize=None, dpi=None, *, facecolor=None, edgecolor=None,
linewidth=0.0, frameon=None, subplotpars=None, tight_layout=None,
constrained_layout=None, layout=None, **kwargs)
```

**Figura 12**

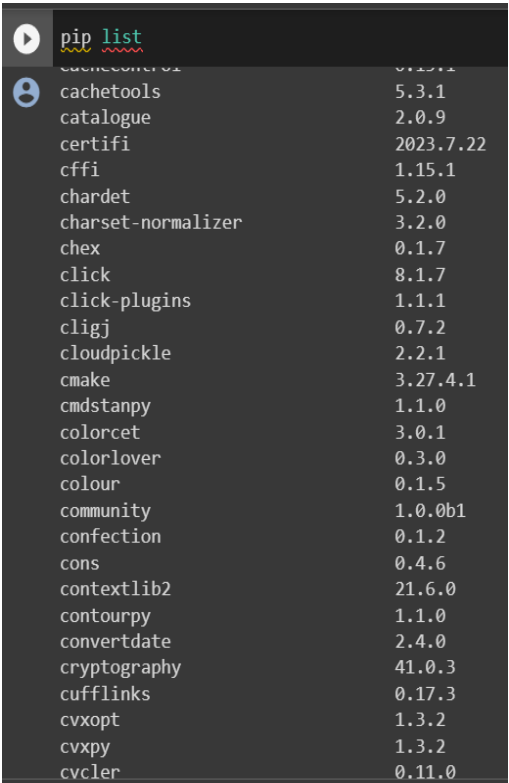
Selección de método figure en matplotlib en Colab



5. **Mostrar todo el stack de librerías del entorno de ejecución utilizado**

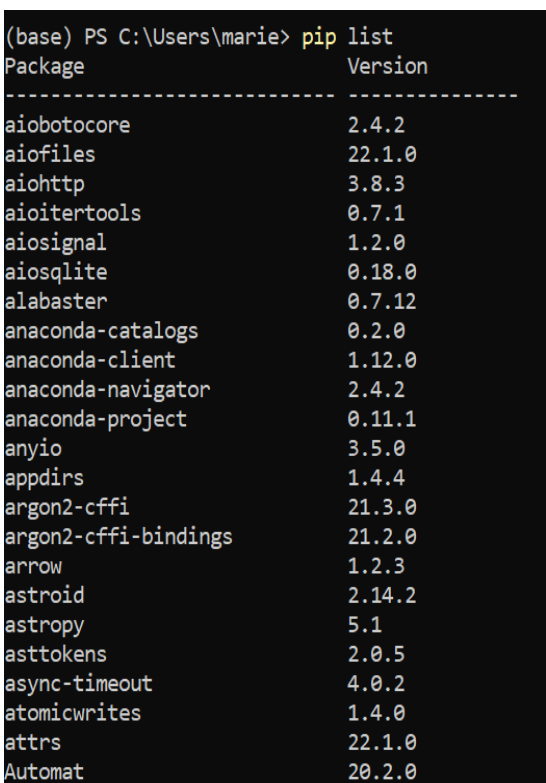
Para mostrar todo el stack de librerías en los casos de Colab y Anaconda se sigue el mismo procedimiento, con ayuda de la sentencia **pip list** (Ingenia, 2023), obteniendo así la lista de las librerías instaladas en cada uno de los entornos como se muestra en las figuras 13 y 14, donde se puede observar el nombre de algunas librerías seguidas de su versión actual.

**Figura 13**  
Lista de librerías en Colab



cachetools	5.3.1
catalogue	2.0.9
certifi	2023.7.22
cffi	1.15.1
chardet	5.2.0
charset-normalizer	3.2.0
chex	0.1.7
click	8.1.7
click-plugins	1.1.1
cligj	0.7.2
cloudpickle	2.2.1
cmake	3.27.4.1
cmdstanpy	1.1.0
colorcet	3.0.1
colorlover	0.3.0
colour	0.1.5
community	1.0.0b1
confection	0.1.2
cons	0.4.6
contextlib2	21.6.0
contourpy	1.1.0
convertdate	2.4.0
cryptography	41.0.3
cufflinks	0.17.3
cvxopt	1.3.2
cvxpy	1.3.2
cvcler	0.11.0

**Figura 14**  
Lista de librerías en Anaconda



Package	Version
aiobotocore	2.4.2
aiofiles	22.1.0
aiohttp	3.8.3
aioitertools	0.7.1
aiosignal	1.2.0
aiosqlite	0.18.0
alabaster	0.7.12
anaconda-catalogs	0.2.0
anaconda-client	1.12.0
anaconda-navigator	2.4.2
anaconda-project	0.11.1
anyio	3.5.0
appdirs	1.4.4
argon2-cffi	21.3.0
argon2-cffi-bindings	21.2.0
arrow	1.2.3
astroid	2.14.2
astropy	5.1
asttokens	2.0.5
async-timeout	4.0.2
atomicwrites	1.4.0
attrs	22.1.0
Automat	20.2.0

**2. Aplicación con Mapas**

Entre las librerías más comunes para representar información sobre un mapa en Phyton se encuentran **Geopandas** y **Folium**, en este caso se trabajará con la librería **Folium**, esto debido a que facilita la visualización de datos manejados en Python mediante mapas interactivos (Github, 2013), además la interfaz es muy amigable, por lo que puede ser muy llamativo y claro para los usuarios.

Para comenzar a trabajar con **Folium** es necesario conocer si se cuenta con la librería instalada, este proceso se puede realizar desde la revisión de la lista de librerías instaladas con la sentencia **pip list** (Ingenia, 2023) en el programa de **Anaconda Powershell Prompt** o instalando directamente desde este programa la librería **pip install folium**, así como se muestra en la figura 15.

**Figura 15**  
Instalación de **Folium** en Anaconda

```
(base) PS C:\Users\marie> pip install folium
Collecting folium
  Downloading folium-0.14.0-py2.py3-none-any.whl (102 kB)
----- 102.3/102.3 kB 2.0 MB/s eta 0:00:00
Collecting branca>=0.6.0 (from folium)
  Downloading branca-0.6.0-py3-none-any.whl (24 kB)
Requirement already satisfied: Jinja2>=2.9 in s:\archivos de programas\anaconda\lib\site-packages (from folium) (3.1.2)
Requirement already satisfied: numpy in s:\archivos de programas\anaconda\lib\site-packages (from folium) (1.24.3)
Requirement already satisfied: requests in s:\archivos de programas\anaconda\lib\site-packages (from folium) (2.31.0)
Requirement already satisfied: MarkupSafe>=2.0 in s:\archivos de programas\anaconda\lib\site-packages (from Jinja2>=2.9->folium) (2.1.1)
Requirement already satisfied: charset-normalizer<4,>=2 in s:\archivos de programas\anaconda\lib\site-packages (from requests->folium) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in s:\archivos de programas\anaconda\lib\site-packages (from requests->folium) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in s:\archivos de programas\anaconda\lib\site-packages (from requests->folium) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in s:\archivos de programas\anaconda\lib\site-packages (from requests->folium) (2023.7.22)
Installing collected packages: branca, folium
Successfully installed branca-0.6.0 folium-0.14.0
```

Para el uso de la librería **Folium** el primer paso es importar la librería en el entorno o programa que se desee trabajar, en este caso se realizará con la ayuda Colab, después de esto según García (2022) se define una variable *m* donde se guardará la locación deseada junto con características, como el nivel de zoom con el cuál se mostrara la figura, en este caso se utilizó la ubicación de 6.280, -75.504 correspondiente a el Parque Arví y se configuro con un zoom de 13 el cuál se puede modificar de acuerdo a los gustos o necesidades del usuario. También se podrá usar herramientas como *tooltip*, este permitirá mostrar el texto indicado, además de utilizar un marcador agregado mediante la sentencia *folium.Marker*, en este se incluye la ubicación escogida, seguido del texto o nombre del lugar que se desea mostrar tal como se muestra en la figura 16 donde además se utilizará *display(m)* en lugar de *print(m)*, porque *print* no mostrará los objetos complejos sino que mostrará el texto “<folium.folium.Map object at 0x00000270C425BED0>”, por ello se utilizó *display*, ya que con este si logrará tener el mapa con las características que se le dio como se puede observar en la figura 17.

**Figura 16**  
Código para usar **Folium** con Colab

```
import folium

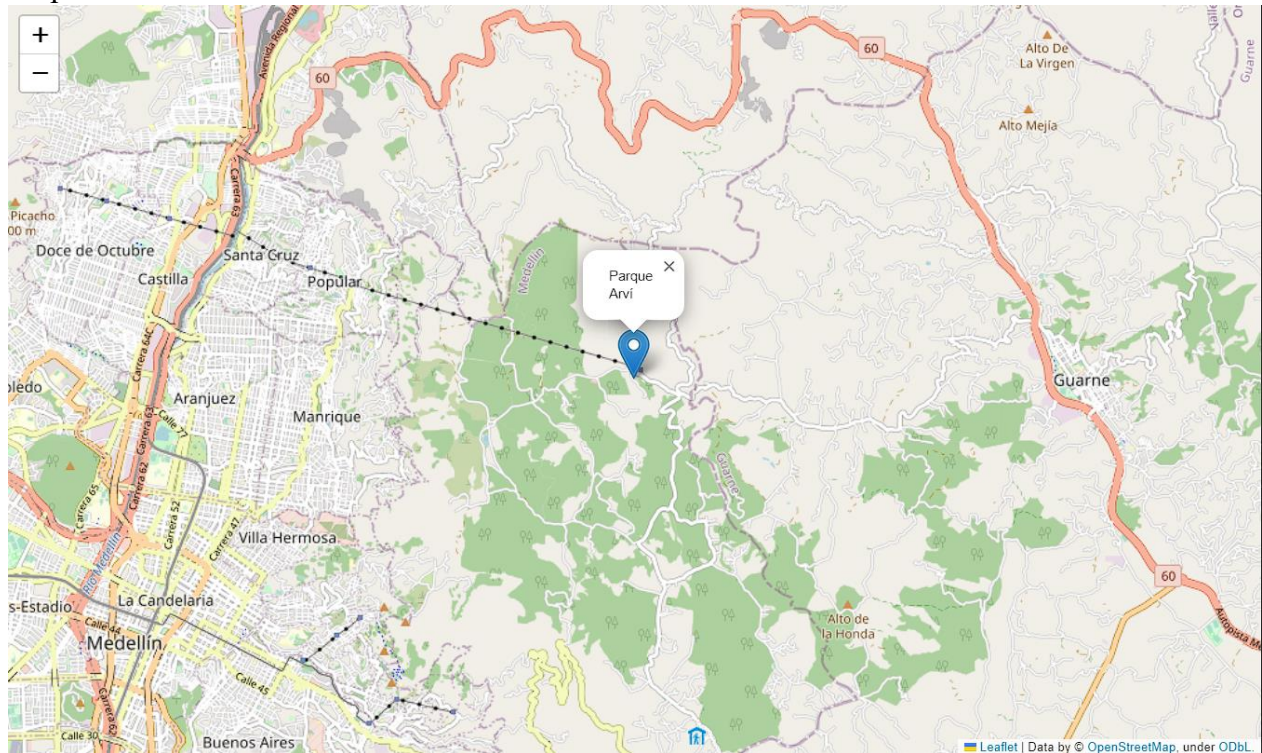
m = folium.Map(location=[6.280, -75.504], zoom_start=13)

tooltip = 'Parque Arví'
folium.Marker([6.280, -75.504], popup='Parque Arví', tooltip=tooltip).add_to(m)

display(m)
```

**Figura 17**

Mapa obtenido con **Folium** con Colab



## Referencias

- Garcías Grandes, J. L. (2022). *Folium: creación de mapas web interactivos con Python y Leaflet*. MappingGIS. <https://mappinggis.com/2022/09/folium-utilizando-leaflet-con-python/>
- Github. (2013). *Folium*. <https://python-visualization.github.io/folium/latest/>
- Hunter, J., Dale, D., Firing, E., Droettboom, M., & the Matplotlib Development Team. (2012). *Matplotlib.figure*. [https://matplotlib.org/stable/api/figure\\_api.html](https://matplotlib.org/stable/api/figure_api.html)
- Ingenia, Talento Especializado para el Valle del Software. (2023). *Machine Learning*. [Presentación de Ingenia, Universidad de Antioquia].