Python

ACTIVIDAD PRÁCTICA

Reto 6

# Título: Librerías

# Modalidad: equipos de trabajo de 3 a 5 personas

# Resultado de aprendizaje

* El estudiante entiende qué son las librerías y sabe la importancia de estas en la programación en el lenguaje Python.
* El estudiante es capaz de encontrar una librería ideal para solucionar un problema específico y configurarla para ser usada en el lenguaje de programación Python.
* El estudiante es capaz de usar una librería en la solución de un problema específico en el lenguaje de programación Phyton.
* El estudiante es capaz de identificar cuándo es necesario usar un condicional o un ciclo en la solución de un problema dado.

# Recursos

* “Material fundamental”: “Librerías”
* Ejercicios prácticos del “material fundamental”: “Librerías”
* Tutorías sincrónicas

# Indicaciones

* Revise y estudie el “material fundamental”: “Librerías”.
* Invierta un poco de tiempo haciendo pruebas en los **ejercicios prácticos**.
* Revise los ejemplos dados por el tutor en el taller.
* Trabaje en equipos de tres a 5 personas.
* Intente responder las siguientes **preguntas de interés** por medio de pruebas que realice usted mismo, debatiéndolas con sus compañeros o investigándolas:
  + ¿En que situaciones se deben usar librerías?
  + ¿Puedo crear mi propia librería, con mi código?
  + ¿Las librerías son gratuitas todas?
* Si se le dificulta la búsqueda de información para solucionar problemas, puede emplear el foro **Preguntas al tutor**, en el que el tutor o uno de sus compañeros puede ayudarle, o puede haber una pregunta similar a la suya con solución previa que haya logrado resolver la misma duda que usted tiene. Finalmente, el tutor lo guiará y ayudará a resolver su dificultad.
* Lea atentamente el siguiente enunciado del problema e intente reconocer los elementos presentados en el tema “Librerías” que le pueden servir para crear la aplicación (recuerde que una aplicación no es más que el nombre que se le da a un archivo de código que cumple con una tarea).

**RETO**

Lea cuidadosamente el requerimiento y responda con las instrucciones para cada necesidad planteada.

Una empresa de *software* requiere del perfil de un profesional con experiencia en ciencia de datos. Se requiere realizar una labor de alistamiento de un entorno de código, con las siguientes librerías y paquetes:

1. Escriba un informe sobre cómo usaría estas librerías en la solución. Incluya las sentencias que utilizará como respuesta al requerimiento:
   * Instalar numpy con referencia np.
   * Instalar el paquete del toolbox de redes neuronales tensorflow, versión 2.7.0.
   * Actualizar la librería matplotlib en su última versión.
   * Seleccionar en la librería matplotlib el método figure.
   * Mostrar todo el stack de librerías del entorno de ejecución utilizado.

Tenga en cuenta que las instrucciones deben escritas de manera clara y con lenguaje técnico, para que cualquier programador pueda seguirlas y obtener los resultados esperados.

1. Construya una aplicación con mapas, es decir, que involucre georreferenciación. Investigue al menos dos librerías que puedan ser compatibles con Phyton y que sirvan para representar información sobre un mapa. Seleccione una de las dos librerías. Diga en el documento cuál librería eligió y por qué. Entregue el nombre de las librería, el sitio de descarga y un tutorial o ejemplo sobre cómo usarla.

Finalmente, desarrolle una solución al problema creando la aplicación en el lenguaje de programación Python de manera que cumpla con los requerimientos que se piden y siguiendo las indicaciones que se le dan en el enunciado.

# Criterios de valoración de la evidencia

Punto 1. Calidad técnica del documento entregado (50%)

El lenguaje usado es correcto.

Las instrucciones pueden ser seguidas por personal técnico.

Las instrucciones son claras y siguen un orden lógico.

Punto 2. Proceso de indagación efectivo (50%)

Logra encontrar librerías que satisfacen la necesidad.

Los criterios de selección son fuertes desde el punto de vista técnico.

En los criterios de selección usa argumentos de terceros.

Explica claramente cómo usar la nueva librería.

# Protocolo de entrega

* **Formato**: comprima el archivo en .zip con el siguiente nombre:

[NOMBREINTEGRANTE1,NOMBREINTEGRANTE2]\_Reto6.zip

donde [NOMBREINTEGRANTE] es su nombre y apellidos empezando por mayúscula cada uno. Ej.:

AndresRestrepoG,DianaLopez\_Reto6.zip

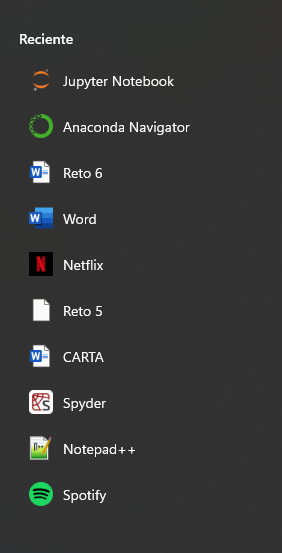
* **Medio**: suba el archivo .zip en la plataforma de Moodle en la sección correspondiente para la “Entrega de Trabajos del Reto 6”.
* **Plazo máximo de entrega**: de acuerdo con el cronograma de actividades.

**SOLUCIÓN**

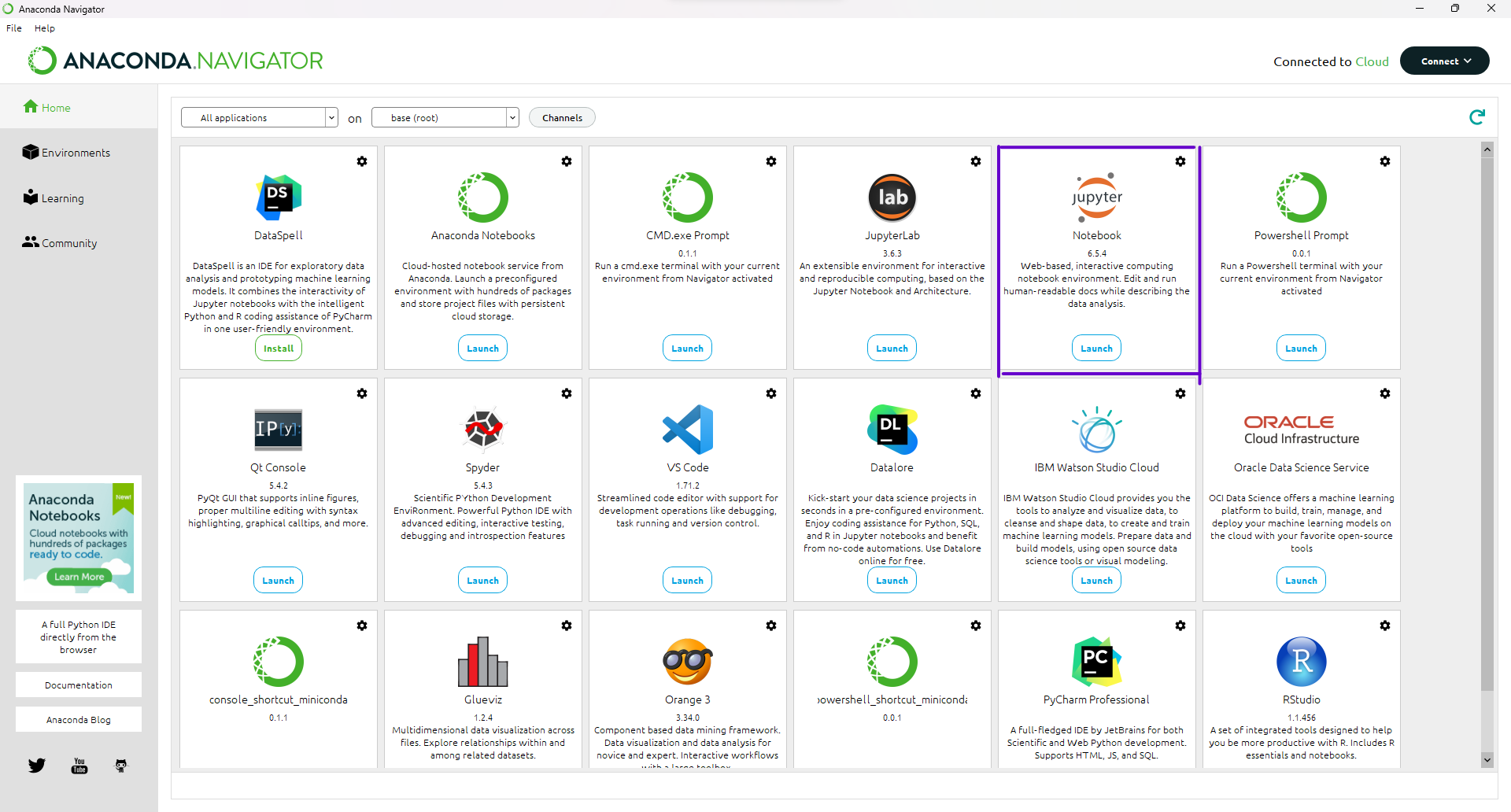
1. **Instalar numpy con referencia np**:

Para instalar las librerías del ejercicio, lo vamos a trabajar desde jupyter.

Primer paso: Abrir desde el escritorio el programa anaconda



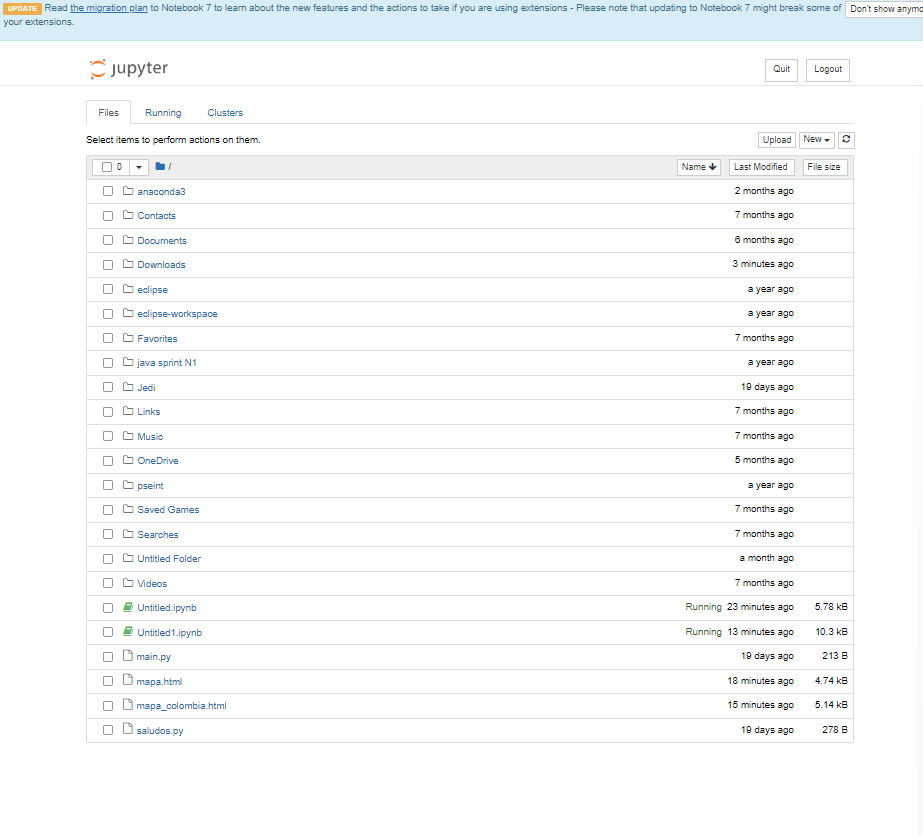
Segundo paso: Seleccionar la opción Jupyter notebook



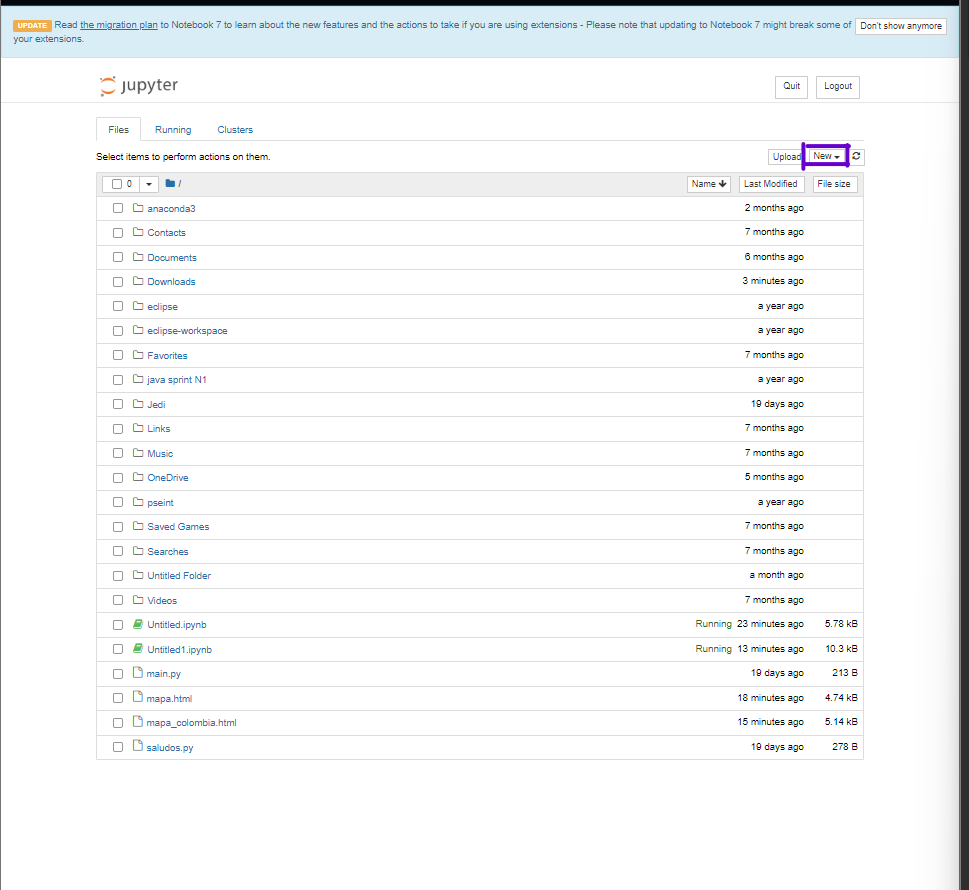
Tercer paso: dar click en “launch”.

Cuarto paso: se abre una pestaña en el buscador de internet que generalmente usa cada estudiante

(<http://localhost:8889/tree#notebooks>)



Quinto paso: seleccionar la opción “New”



Sexto paso: se abre una lista desplegable

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Séptimo paso: selecciones la opción Python 3 (ipkernel)

Interfaz de usuario gráfica, Aplicación, PowerPoint

Descripción generada automáticamente

Octavo paso: se abre una nueva pestaña

<http://localhost:8889/notebooks/%20Reto%206%20Grupo%201%20-%20David%20Villa%20Sanchez%20y%20Daniela%20Castano%20Henao.ipynb%20Reto%206%20Grupo%201%20-%20David%20Villa%20Sanchez%20y%20Daniela%20Castano%20Henao.ipynb#>

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Noveno paso: Instalar Numpy:

Numpy es una librería fundamental en ciencia de datos para realizar cálculos numéricos eficientes. Para instalarlo y utilizarlo en Python, primero debemos asegurarnos de tener un entorno virtual configurado (recomendado) y luego ejecutar los siguientes comandos:

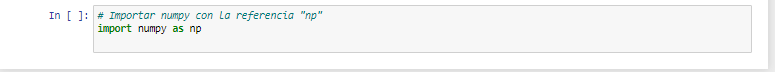
# Instalar numpy usando pip

!pip install numpy



# Importar numpy con la referencia "np"

import numpy as np



Decimo paso: Instalar el paquete del toolbox de redes neuronales tensorflow, versión 2.7.0:

TensorFlow es una librería popular para machine learning y redes neuronales. Para instalar la versión específica 2.7.0, podemos utilizar pip:

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

#Se instala esta versión porque la 2.7 no la encuentra

!pip install tensorflow==2.13.0

import tensorflow as tf

print("------------------------------------------")

print(tf.\_\_version\_\_)

!pip show tensorflow

print("------------------------------------------")Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Onceavo paso: Actualizar la librería matplotlib en su última versión:

Matplotlib se utiliza para crear visualizaciones en Python. Para actualizarlo a la última versión, ejecutamos el siguiente comando:

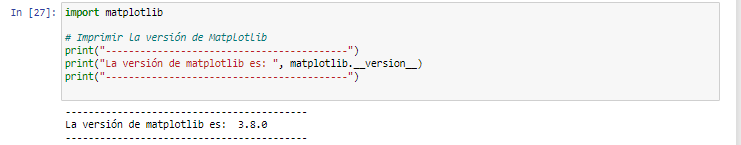
import matplotlib

# Imprimir la versión de Matplotlib

print("------------------------------------------")

print("La versión de matplotlib es: ", matplotlib.\_\_version\_\_)

print("------------------------------------------")



Doceavo paso: El método figure se utiliza para crear una nueva figura o lienzo en Matplotlib. Aquí está cómo se utiliza:

#Seleccionar en la librería matplotlib el método figure.

# Crear una nueva figura

import matplotlib.pyplot as plt

fig = plt.figure()

# Agregar un eje a la figura

ax = fig.add\_subplot(1, 1, 1)

# Graficar algunos datos en el eje

x = [1, 2, 3, 4, 5]

y = [2, 4, 6, 8, 10]

ax.plot(x, y)

# Mostrar la figura

plt.show()

print("------------------------------------------")

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Treceavo paso: Mostrar todo el stack de librerías del entorno de ejecución utilizado:

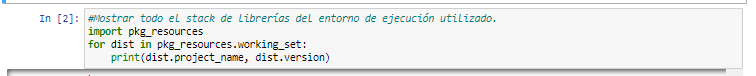
Para listar todas las librerías y sus versiones en su entorno de ejecución, puede utilizar el siguiente comando:

#Mostrar todo el stack de librerías del entorno de ejecución utilizado.

import pkg\_resources

for dist in pkg\_resources.working\_set:

print(dist.project\_name, dist.version)



Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Catorceavo paso: geolocalizador

Para representar información en un mapa en Python, dos de las librerías populares son Folium y Basemap. Para esta tarea, seleccionaremos Folium debido a su facilidad de uso y capacidades de personalización.

Folium es una librería de Python que permite crear mapas interactivos de manera sencilla y se integra bien con Jupyter Notebooks. Es ampliamente utilizada en la comunidad de ciencia de datos y es fácil de aprender.

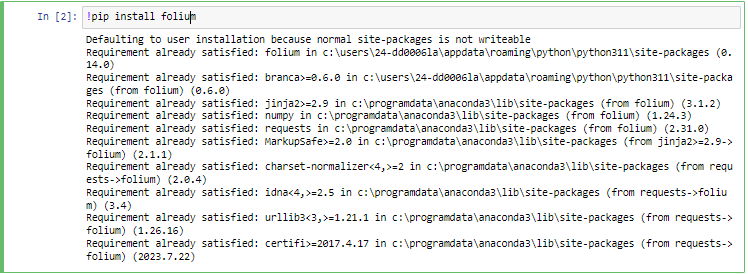
Sitio de descarga y tutorial:

Puede encontrar más información sobre Folium en su sitio oficial de descarga: https://python-visualization.github.io/folium/

Puede instalar Folium usando pip:

!pip install folium





import folium



# Crear un mapa centrado en una ubicación específica

mapa = folium.Map(location=[latitud, longitud], zoom\_start=16)

ejemplo:

folium.Map(

location = [40.4165,-3.10256],

zoom\_start=16

)

Mapa

Descripción generada automáticamente

# Agregar un marcador al mapa

folium.Marker([latitud, longitud], tooltip="Mi marcador").add\_to(mapa)

# Guardar el mapa en un archivo HTML

mapa.save("mapa.html")

Ejemplo:

import folium

# Crear un mapa centrado en una ubicación específica

mapa = folium.Map(location=[40.7128, -74.0060], zoom\_start=10) # Ubicación de Nueva York

# Agregar marcadores para ubicaciones específicas

folium.Marker([40.7128, -74.0060], tooltip='Nueva York').add\_to(mapa)

folium.Marker([34.0522, -118.2437], tooltip='Los Ángeles').add\_to(mapa)

folium.Marker([51.5074, -0.1278], tooltip='Londres').add\_to(mapa)

folium.Marker([48.8566, 2.3522], tooltip='París').add\_to(mapa)

# Guardar el mapa como un archivo HTML

mapa.save('mapa.html')

# Abrir el mapa en el navegador predeterminado

import webbrowser

webbrowser.open('mapa.html')

Texto

Descripción generada automáticamente

Mapa

Descripción generada automáticamente

Ejemplo de geolocalizador:

import folium

mapa = folium.Map(location=[4.5709, -74.2973], zoom\_start=16)

ciudades\_colombia = [

{"nombre": "Bogotá", "coordenadas": [4.6097, -74.0817]},

{"nombre": "Medellín", "coordenadas": [6.2442, -75.5812]},

{"nombre": "Cali", "coordenadas": [3.4516, -76.5320]},

{"nombre": "Barranquilla", "coordenadas": [10.9632, -74.7970]},

{"nombre": "Cartagena", "coordenadas": [10.3910, -75.4794]}

]

for ciudad in ciudades\_colombia:

folium.Marker(

location=ciudad["coordenadas"],

tooltip=ciudad["nombre"]

).add\_to(mapa)

# Guardar el mapa como un archivo HTML

mapa.save('mapa\_colombia.html')

# Abrir el mapa en el navegador predeterminado

import webbrowser

webbrowser.open('mapa\_colombia.html')

