**Para el desarrollo de los programas se utilizó la versión de Python 3.9.18 ya que es la última compatible con Tensor Flow a la fecha de realizado el proyecto.**

**Las Librerías usadas fueron:**

* **os:** Para interactuar con el sistema operativo.
* **re:** Para expresiones regulares.
* **pandas:** Para la manipulación y análisis de datos.
* **sklearn.preprocessing.MinMaxScaler:** Para escalar características.
* **numpy:** Para computación numérica.
* **tensorflow:** Para construir y entrenar redes neuronales.
* **tensorflow.keras.models.Sequential:** Para construir modelos secuenciales.
* **tensorflow.keras.layers.LSTM, tensorflow.keras.layers.Dense, tensorflow.keras.layers.Input, tensorflow.keras.layers.RepeatVector, tensorflow.keras.layers.TimeDistributed:** Para construir diferentes capas en redes neuronales.
* **tensorflow.keras.models.load\_model:** Para cargar modelos pre-entrenados.
* **matplotlib.pyplot:** Para trazar gráficos y visualizaciones.
* **datetime:** Para trabajar con fechas y horas.
* **itertools:** Para crear iteradores para bucles eficientes.
* **sklearn.metrics.mean\_squared\_error:** Para calcular el error cuadrático medio.

**Comandos de Instalación:**

Puede instalar las bibliotecas necesarias usando pip. Así es como puede hacerlo:

pip install pandas scikit-learn numpy tensorflow matplotlib

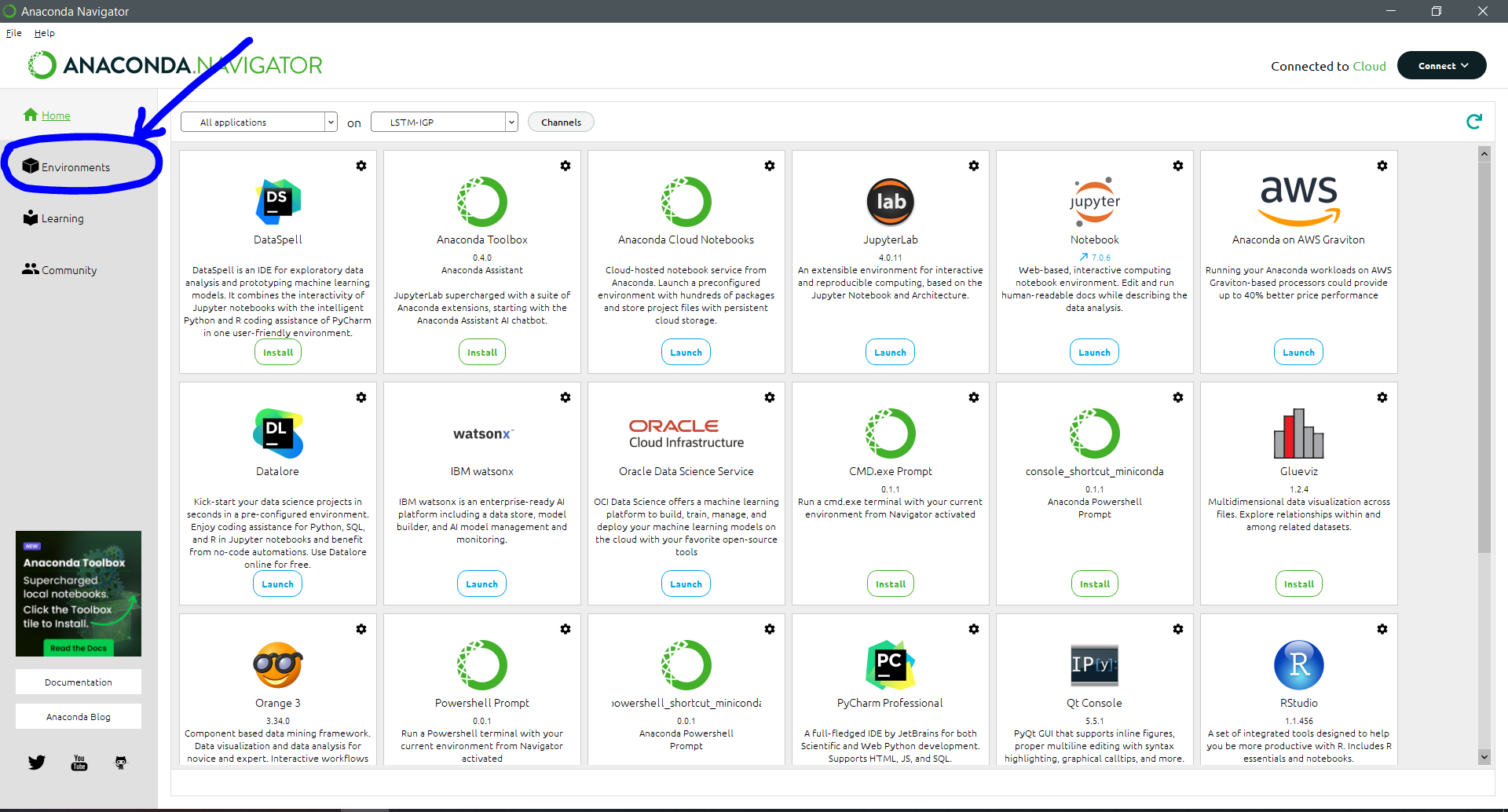
**Observación:** Para que no interfiera con otras versiones se sugiere crear un nuevo entorno en donde se pueda trabajar sin problemas.

**Si se utiliza en Windows:**

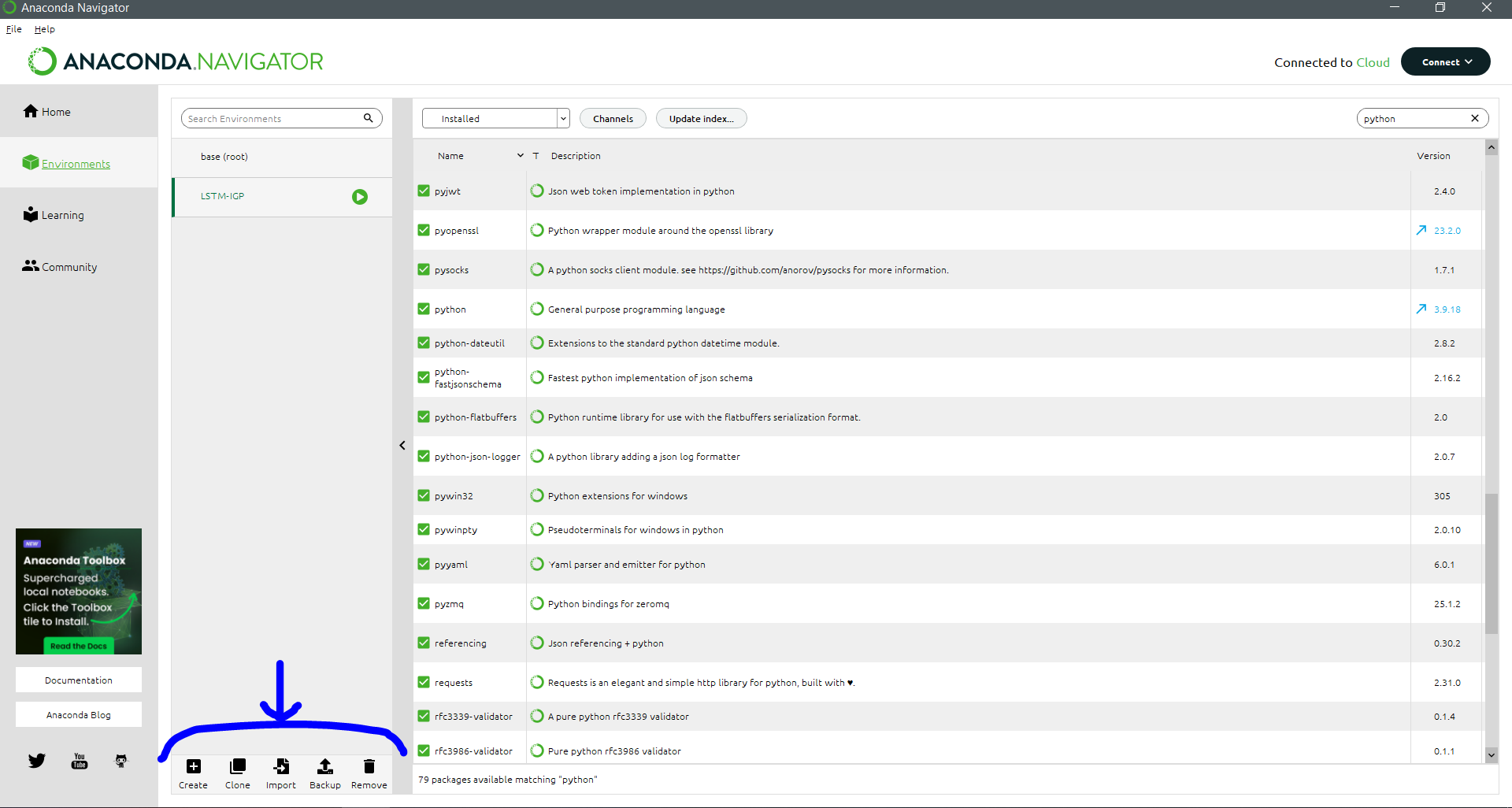
Si se está usando un sistema operativo Windows, puede usarse Anaconda, que puede descargarse de:

https://www.anaconda.com/download

En él es sencillo crear un entorno:

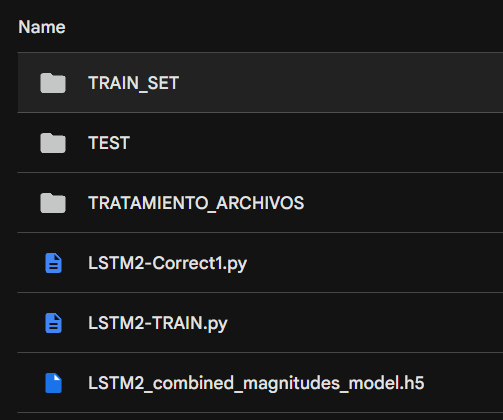


Se puede crear un entorno o importarlo. Por ello se adjunta una carpeta llamada LSTM-Environment en donde se encuentra el archivo del entorno usado en el desarrollo.



**CONTENIDO:**En cuanto a los archivos, hay carpetas llamadas: LSTM1\_model, LSTM2\_model, LSTM3\_model y LSTM4\_model

En cada una de ellas se tiene:



En donde se da el script usado para entrenar dicho modelo, asi como el TRAIN\_SET usado, además del modelo guardado en un archivo de tipo h5 y con ello el archivo que se encarga de usar el modelo, en el ejemplo: LSTM2-Correct1.py

**LSTM1:**

Modelo básico de predicción de un punto a partir de 10 previos, es decir, si se desea usar, tiene que hacerse componente por componente.

**LSTM2 (mejor):**

Modelo de predicción de un punto a partir de 10 previos, con la diferencia que lo hace en tres componentes a la vez, es decir, a partir de 30 puntos genera 3, siendo para cada componente 10 datos previos - 1 predicho.

**LSTM3:**

Modelo planteado para detectar saltos de línea de base y corregirlos. El planteamiento esta sobre la premisa que el salto de línea de base ocurre en media toma de datos, es decir repentinamente.

En la realidad eso no es común, usualmente cuando ocurre un salto de linea de base hay unos minutos en donde no hay datos para darse el salto cuando se tiene datos de nuevo. Por ello este modelo no se siguió entrenando ni probando.

**LSTM4:**

Este modelo se planteo para igual que el modelo LSTM1, funcionar para una sola componente, tomando 10 valores pero prediciendo 3. No se continuo en el desarrollo debido a que era para cada componente es decir si se desea usar para las 3 componentes, se tendría que ejecutar 3 veces.

**Uso de los modelos:**

En cada una de las carpetas, hay un archivo donde se usa el modelo, también se incluye la carpeta de Test que se puede usar para las pruebas. Al final de cada uno de estos archivos, se ubica la funcion main() en donde se tiene que cambiar la ruta en donde se encuentre la carpeta TEST para asi poder ejecutarse sobre los archivos de esta carpeta.

En el caso de la carpeta del modelo LSTM2, se tiene una subcarpeta llamada TRATAMIENTO\_ARCHIVOS, en ella se tienen tres scripts, uno para descomprimir los archivos (al descargarse de la pagina de LISN vienen comprimidos, primero se tiene que descomprimir el zip y despues puede usarse el script), otro para clasificar los archivos min si es que estan completos, incompletos o vacios y finalmente un script para graficar los datos, puede graficarse por día o por mes.