

Timothée Aberturas

### PRIMER MÉTODO:

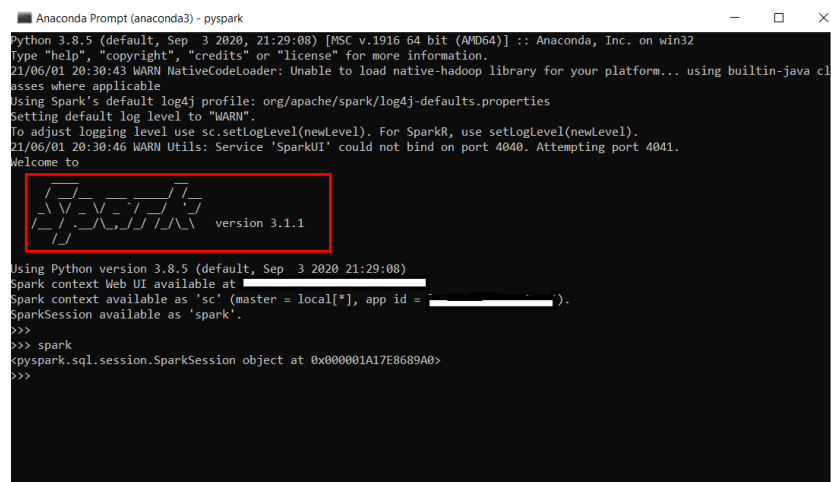
Si queremos instalarlo desde Anaconda, quizá el problema se pueda solucionar actualizando el pip escribiendo en el Anaconda Prompt:

```
python -m pip install --user --upgrade pip
```

puesto que el pip probablemente no esté actualizado.

Una vez hecho esto, escribimos `pip install pyspark` y se instalaran los módulos necesarios.

Cuando termine de instalar, escribimos en el **Anaconda Prompt**: `pyspark` y aparecerá algo como lo siguiente:



Si escribimos `spark` en la consola, nos da la dirección de la memoria donde está alojado `SparkSession`. El `local[*]` indica que utilizamos todos los cluster (en este caso, núcleos) que tengamos en nuestro PC.

### SEGUNDO MÉTODO:

#### FASE I:

Antes de nada, se recomienda instalar Python manualmente, es decir, descargar Python 3.7.0 de la página oficial <https://www.python.org/downloads/release/python-370/> y NO usar Anaconda. En caso de tener instalado Anaconda y Python-Spyder, no interfiere la instalación de Python con Anaconda.

Al instalar Python 3.7.0 tenemos que seleccionar la casilla `Add Python 3.7 to PATH`, esto es importante para instalar paquetes desde la consola de Windows (importante para hacer cosas como en el paso 10 de la **FASE II**).



Esta obra está bajo una [licencia de Creative Commons Reconocimiento 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/).

## FASE II:

Nos sale un error al instalar *pyspark*, nos dice que nos faltan archivos. Si, por ejemplo, en Python escribimos *run 20\_graph\_degree\_simple.py g0.txt* del archivo grafos del Moodle, nos sale una lista de errores inmensa. Esto es porque nos falta instalar al menos una de estas tres componentes en el disco duro local C. A saber, Java (de <https://www.oracle.com/es/java/technologies/javase/javase-jdk8-downloads.html>), spark-3.1.1-bin-hadoop2.7 (de la página oficial de Spark <https://spark.apache.org/downloads.html>) y winutils (el archivo de GitHub: <https://github.com/steveloughran/winutils>)

## PROCEDIMIENTO:

1. Instalar *Java* en el disco local C.

2. Descomprimir el archivo *spark-3.1.1-bin-hadoop2.7.zip* en el disco local C. Revisando la carpeta, vemos que hay una carpeta, dentro de la cual hay a su vez otra carpeta que contiene todos los archivos, por tanto, hay que cortar y pegar esos archivos en la carpeta inicial. Es decir, lo que queremos es que al entrar en la carpeta *spark-3.1.1-bin-hadoop2.7* encontremos ya los archivos directamente y no otra carpeta donde se almacenan todos los archivos (esto es por comodidad, para el **paso 6**, pero se puede dejar como está).

3. Vamos al disco local C, al mismo sitio donde hayamos instalado Java y descomprimido *spark-3.1.1-bin-hadoop2.7*. En mi caso:

*Disco local C -> Archivos de programa*

y creamos una carpeta llamada **winutils** y dentro de la misma otra carpeta que llamaremos **bin**.

4. Abrimos **winutils.zip**, recordemos que hemos instalado la última versión estable de hadoop, que es *hadoop.2.7*, luego tenemos que ir a:

*winutils-master -> hadoop-2.7.1 -> bin*

y arrastramos el ejecutable *winutils.exe* dentro de la carpeta **bin** del paso anterior.

5. Ahora solo falta crear unas variables de entorno y añadirlas a Spark. Para ello, vamos a:

*Equipo -> Propiedades -> Configuración avanzada del sistema -> Variables de entorno...*

6. Se despliega una nueva ventana. Procedemos a crear las variables presionando **Nueva...**:



Esta obra está bajo una [licencia de Creative Commons Reconocimiento 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/).

#### Creación de la variable para *Java*:

Nombre de la...: JAVA\_HOME

valor de la...: Escribimos la ruta a la carpeta donde esté alojado Java,  
en mi caso: C:\Program Files\Java\jre1.8.0\_291

#### Creación de la variable para *Spark*:

Nombre de la...: SPARK\_HOME

valor de la...: Escribimos la ruta a la carpeta donde esté alojado Spark  
en mi caso: C:\Program Files\spark-3.1.1-bin-hadoop2.7 (por eso eliminábamos lo de la doble carpeta)

#### Creación de la variable para *Hadoop*:

Nombre de la...: HADOOP\_HOME

valor de la...: Escribimos la ruta a la carpeta donde esté alojado winutils.exe  
en mi caso: C:\Program Files\winutils

7. Vamos a editar el camino, para ello seleccionamos **Path** presionándolo y clickando en **Editar...**

Aquí vamos a añadir la carpeta *bin* de Java y Spark. Creamos una nueva ruta clickando en **Nuevo** y escribimos:

```
%JAVA_HOME%\bin  
%SARK_HOME%\bin
```

Y aceptamos todo.

8. Creamos una carpeta en el disco C (donde estén alojados *Java*, *Spark* y *winutils*) llamada **tmp** y, dentro de la misma, otra carpeta llamada **hive**.

9. Iniciamos una terminal de Windows (cmd) y vamos a la ruta donde está alojado el *winutils*. Para ello, escribimos, en mi caso **cd C:\Program Files\winutils\bin** para acceder a ese directorio desde la terminal.

Ahora escribimos en la terminal *winutils chmod 777* seguido de la ruta donde se encuentre la carpeta *hive* que creamos en el paso 8. En mi caso,

```
winutils chmod 777 C:\Program Files\tmp\hive
```

**Nota:** El *chmod 777* es para cambiar permisos de administrador. Otorga permisos



Esta obra está bajo una [licencia de Creative Commons Reconocimiento 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/).

de lectura, escritura y ejecución a todos los usuarios.

Otra forma de hacerlo es ir dentro de la carpeta *hive* e ir a **Propiedades** -> **Seguridad** -> **Editar...** y habilitamos la opción **Escritura**.

10. Escribimos en la consola `pip install pyspark findspark`, el *findspark* encontrará el *spark* en nuestro PC para trabajar con él.

Si al hacer el paso 10 no se instala todo como debería quizás sea porque la versión del *pip* es antigua, por lo que deberíamos actualizarlo. Para ello, se escribe lo siguiente en la consola de Windows:

```
python -m pip install --user --upgrade pip
```

Instalamos Jupyter Notebook desde la consola:

```
pip install jupyter
```

y para iniciarlo escribir en la consola de windows `jupyter notebook`.

Cuando se realizan todos los pasos anteriores, no debería aparecer ningún fallo de instalación de algún módulo.

Se abre *Jupyter Notebook* en el navegador y podemos comprobar que lo hemos instalado iniciando una sesión de Spark para comprobar que se inicia correctamente:

Ya tenemos iniciada una sesión de Spark alojada en nuestro PC.

```
Out[5]: SparkSession - in-memory
SparkContext

Spark UI
Version
v3.1.1
Master
local[*]
AppName
pyspark-shell
```



Esta obra está bajo una [licencia de Creative Commons Reconocimiento 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/).