SPARK

Descargar Spark desde la web oficial.

Descomprimir y abrir carpeta en terminal.

Entrar en sbin y ejecutar:

```
./start-master.sh
```

Si no está instalado Java:

```
sudo apt-get install default-jdk
```

```
java –version
```

Ahora, si todo va bien y se ejecuta el archivo start-master.sh, el servidor maestro de Spark estará arrancado y se podrá acceder en un navegador web a la interfaz gráfica que provee Spark en la dirección

```
//127.0.0.1:8080
```

El maestro está ejecutándose, pero tenemos que crear los esclavos. Para ello, necesitamos copiar la dirección del maestro que vemos en la interfaz gráfica. Después, vamos a la consola y ejecutamos el comando:

```
./start-worker.sh spark://profe-VirtualBox:7077
```

Si vamos a la interfaz y actualizamos, veremos el esclavo en estado activo, aunque sin trabajos encomendados.

Vamos a trabajar con la Shell de Spark para enviar trabajos. Pero antes, debemos descargar datos con los que trabajar:

```
Cd Downloads
```

```
Mkdir Data
```

```
Cd Data
```

```
Wget http://www.gutemberg.org/cache/epub/2489/pg2489.txt
```

```
Wget https://data.cityofchicago.org/api/views/xzkq-xp2w/rows.csv?accessType=DOWNLOAD
```

Le cambiamos el nombre a este archivo último

```
Mv rows.csv?accessType=DOWNLOAD data.csv
```

Ahora ejecutamos el comando de la Shell de Spark. Cambiamos al directorio /bin

```
Cd ..

Cd bin

./spark-shell --master spark://profe-VirtualBox:7077
```

Dentro del Shell, vamos a probar a trastear con el archivo de texto del libro Moby Dick:

```
Val dataset = spark.read.textFile("/home/profe/Downloads/Data/pg2489.txt")

Dataset.show()

Dataset.count()

Val dataset2 = dataset.filter(line => line.contains("Moby"))

Dataset2.show()

Dataset2.show(4)

Dataset.summary().show()
```

Ahora vamos a trabajar con el fichero CSV:

```
Val df = spark.read.option("header", "true").csv("/home/profe/Downloads/Data/data.csv")

Df.head()
```

Ahora usaremos el lenguaje SQL para consultas sobre el DataFrame:

```
Df.select("Hourly Rate").summary().show()
```

A continuación ejecutaremos un archivo de ejemplo que trae Spark para probar una ejecución desde una aplicación (en lugar que hasta ahora, que ha sido por consola):

```
./run-example --master spark://profeVirtualBox:7077 JavaWordCount
/home/profe/Downloads/Data/pg2489.txt
```

Ahora enviaremos nosotros los trabajos que queremos ejecutar a los workers como se haría de verdad, si fuese una aplicación creada por nosotros (no un ejemplo):

./spark-submit --master spark://profeVirtualBox:7077 --class org.apache.spark.examples.JavaPageRank ../examples/jars/spark-examples_2.12-3.3.2.jar ../data/mllib/pagerank_data.txt 10