

# M1B1T1. Herramientas de Gestión del Dato

## Actividad guiada. 3

### Codificación Primer Programa Spark

#### Exposición de la tarea

Necesitamos que codifiques un programa Spark con el lenguaje de programación Python, que leas un fichero de texto y mediante RDDs, transformaciones y acciones respondas a las siguientes preguntas:

- ¿Cuántas líneas tiene el fichero?
- ¿Cuántas palabras tiene el fichero?
- ¿En cuántas líneas aparece la palabra “spark”?
- Imprime, por pantalla, el nº de palabras de 5 líneas.

#### Objetivo

El objetivo de esta actividad es que aprendas a configurar el entorno de desarrollo para trabajar con Spark y Python y entiendas la estructura y opciones básicas de la programación Spark.

#### Pasos para la realización de la actividad

##### 1. Preparación del entorno

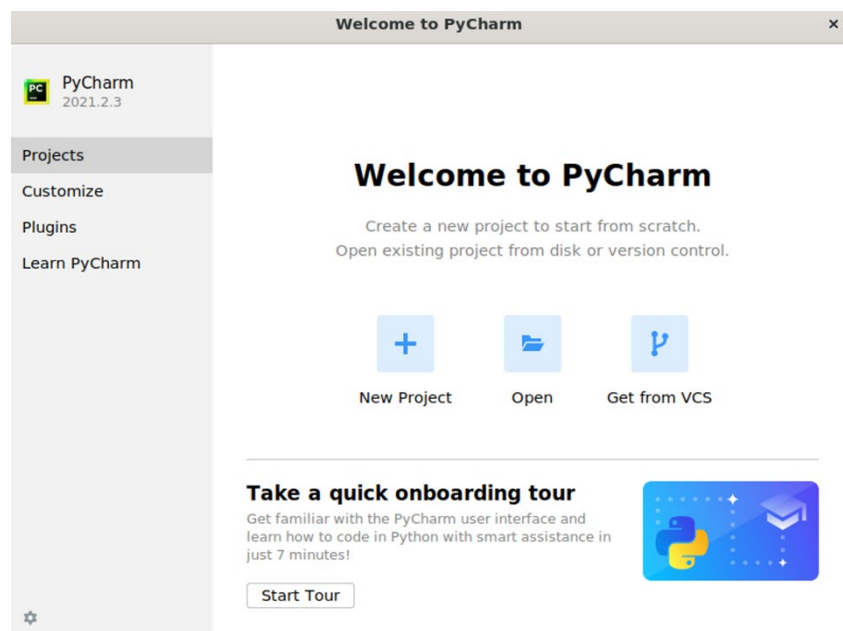
En este primer apartado, vas a preparar el entorno de desarrollo para poder codificar y ejecutar aplicaciones con Spark. Para ello, debes utilizar el lenguaje **Python 3** y el IDE de desarrollo **PyCharm**.

#### IMPORTANTE

Estos pasos están indicados para poder realizar la actividad en ambas máquinas virtuales. En la máquina virtual Windows o en la máquina virtual Ubuntu.

1.1. Busca y arranca la herramienta PyCharm

1.2. Pulsa “New Project”.



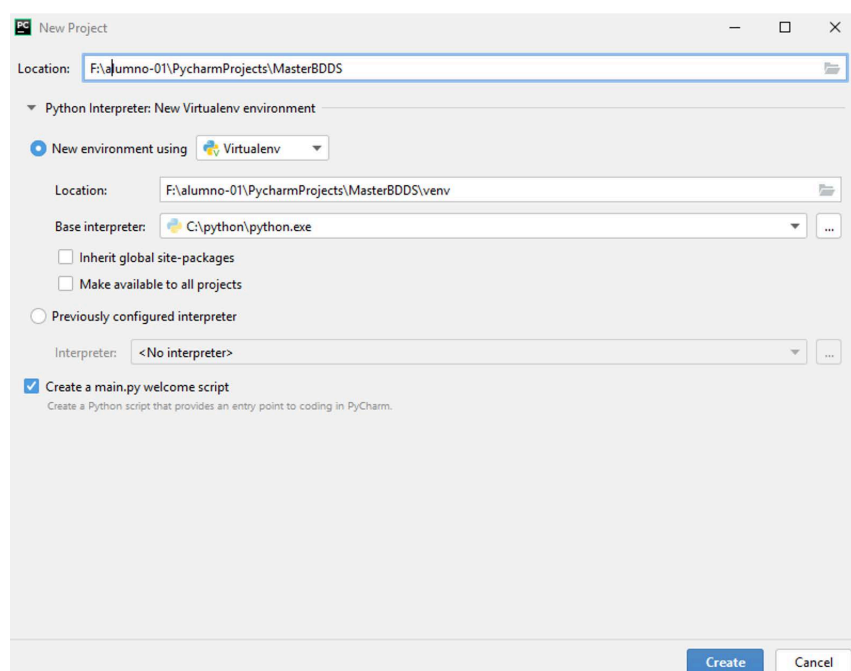
1.3. Configura el proyecto con las opciones indicadas en las figuras. MasterBDDS es el nombre que le tienes que dar a tu proyecto.

### IMPORTANTE

Cada alumno/a debe poner su ruta correspondiente. Las rutas serían:

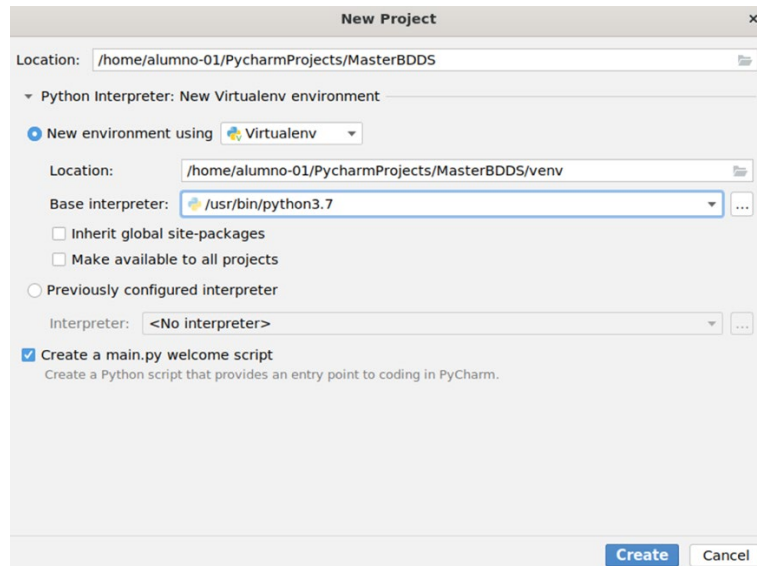
Windows:

**alumno-xx** -> F:\alumno-xx\PycharmProjects\MasterBDDS



Ubuntu:

**alumno-xx** -> /home/**alumno-xx**/PycharmProjects/MasterBDDS

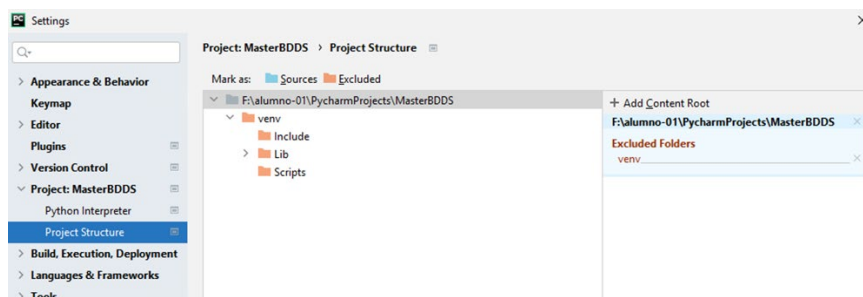


1.4. Una vez que tienes un proyecto para codificar con Python, debes configurarlo para añadir las librerías de Spark.

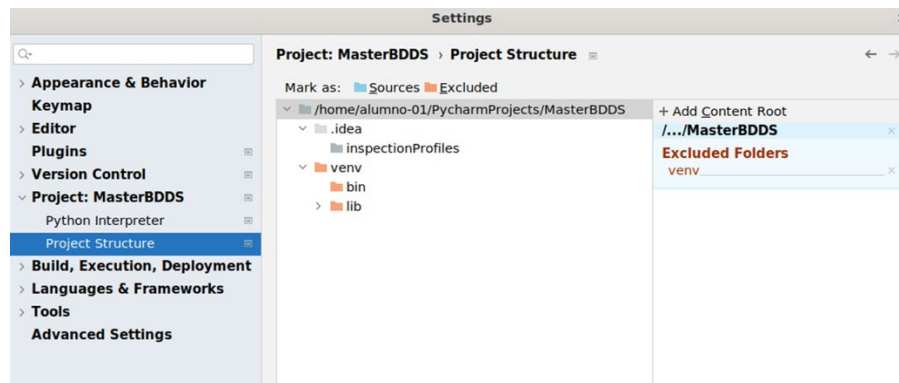
- Navega hasta llegar a la siguiente pantalla:

File → Settings ... → Project MasterBDDS → Project Structure

Windows:

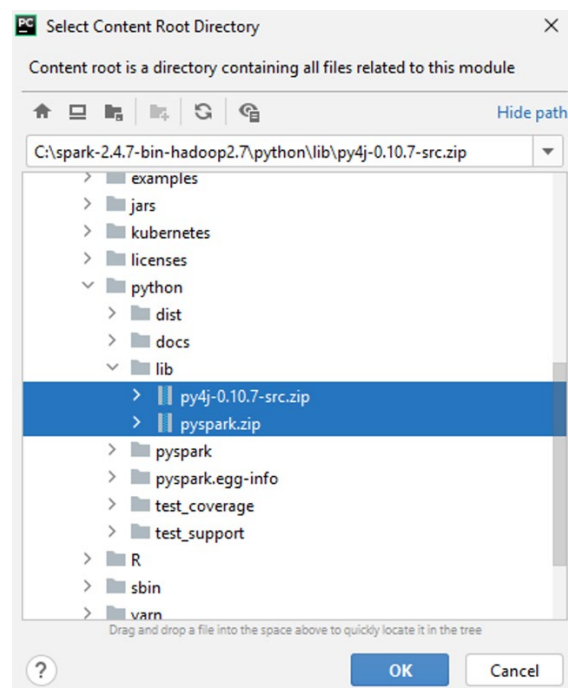


Ubuntu:

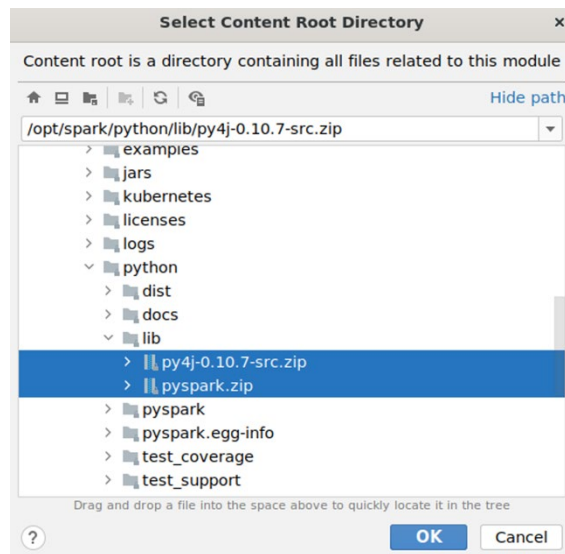


- Pulsa “Add Content Root” y selecciona **los dos archivos zip** que se encuentran en las rutas indicadas en las figuras:

Windows:



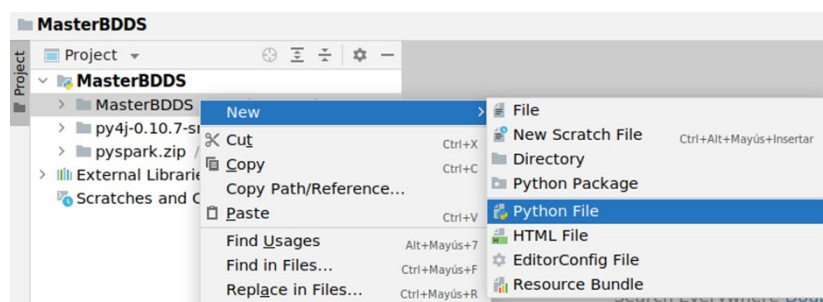
Ubuntu:



- Luego, pulsa “Ok” y en la ventana anterior pulsa, también, “Ok”:

#### 1.5. Crea tu fichero Python

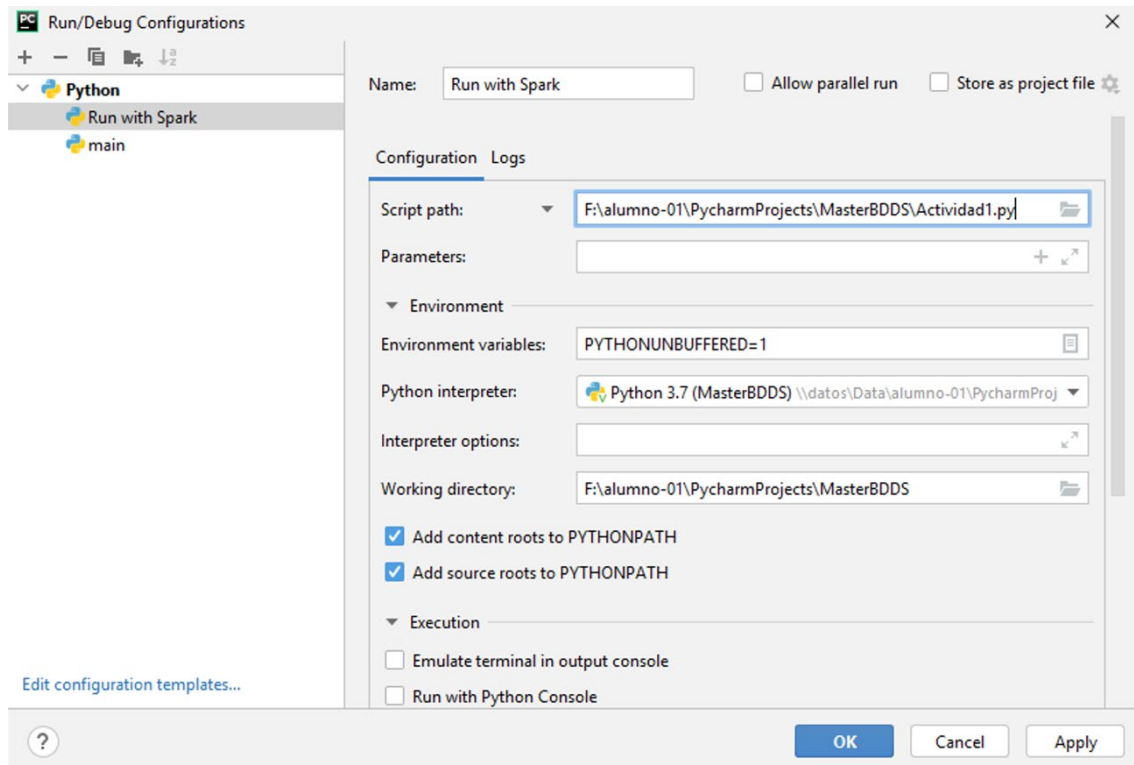
- Pulsa botón derecho sobre “MasterBDDS (segundo nivel) -> New -> Python File”.
- Ponle el siguiente nombre: Actividad1.



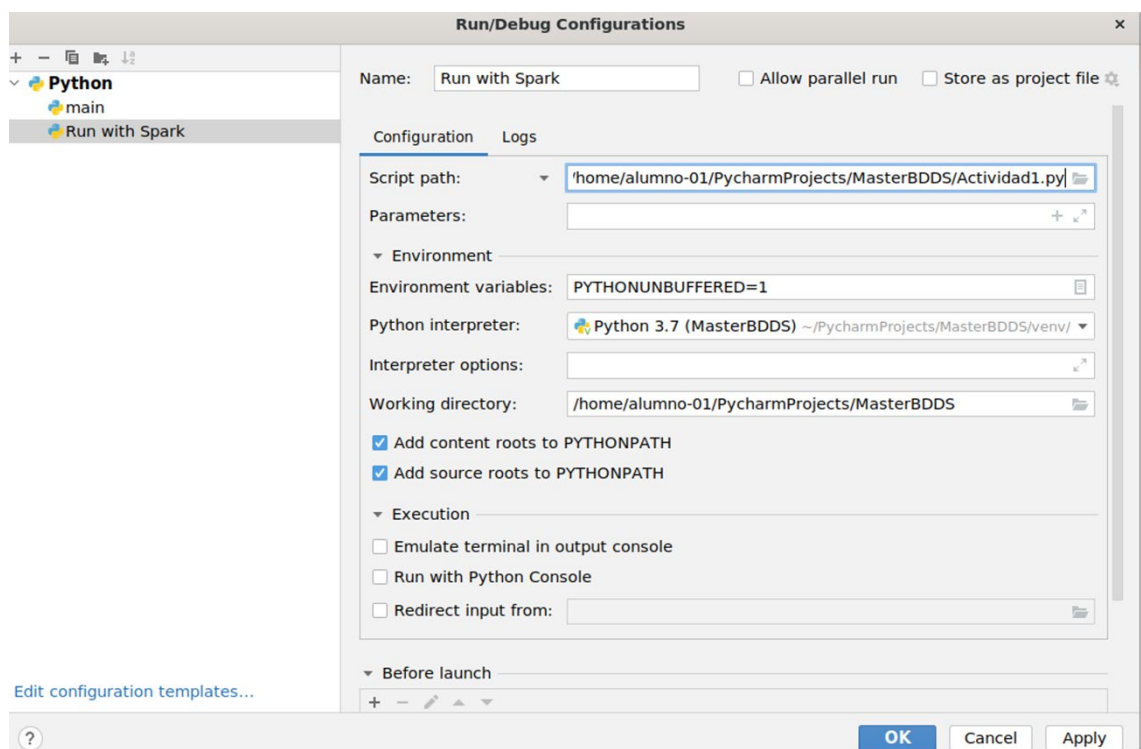
#### 1.6. Crea una nueva “Run configuration”.

- Selecciona la siguiente opción de Menú:  
Run → Edit Configurations → Botón + → Python.
- Ponle el siguiente nombre: Run with Spark.
- Selecciona el fichero que has creado en el paso anterior:

## Windows:

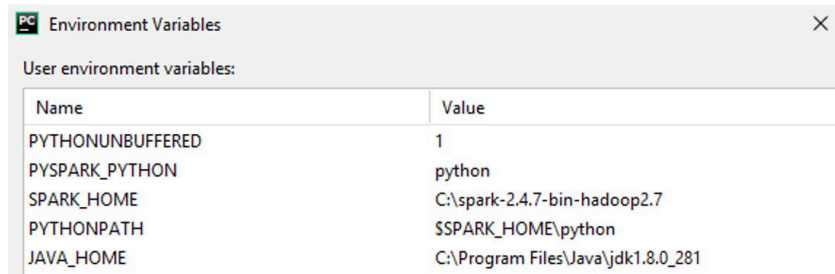


## Ubuntu:

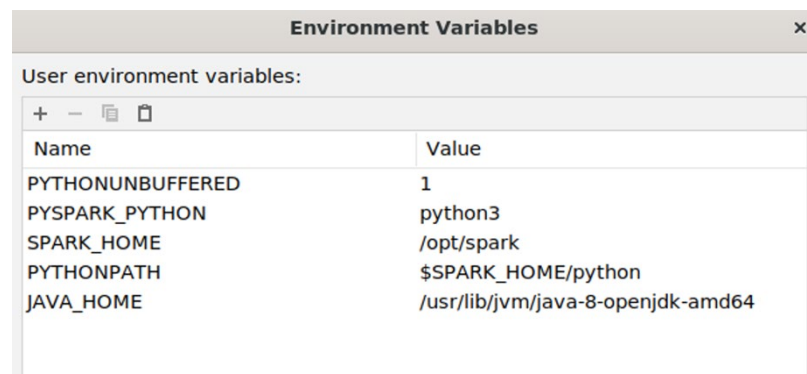


1.7. Añade las variables de entorno. Para ello, pulsa el botón que aparece a la derecha de “Environment variables”. Debe quedar como aparece en la siguiente figura:

Windows:



Ubuntu:



- Luego pulsa “Ok” y “Ok”

2 - Una vez que tienes configurado el entorno, codifica tu primer programa Spark con Python, respondiendo a las preguntas que se solicitan al inicio de esta actividad.

Copiar fichero README.md del directorio de Spark al directorio del proyecto.

Windows:

C:\spark-2.4.7-bin-hadoop2.7\README.md -> F:\alumno-xx\PycharmProjects\MasterBDDS\README.md

Ubuntu:

/opt/spark/README.md -> /home/alumno-xx/PycharmProjects/MasterBDDS

Para ejecutarlo utiliza la configuración "Run with Spark"

```
from pyspark import SparkContext

sc = SparkContext(appName="Actividad1")

fileRDD = sc.textFile("README.md")

# 1 - ¿Cuántas líneas tiene el fichero?
print("README.md tiene " + str(fileRDD.count()) + " líneas")

# 2 - ¿Cuántas palabras tiene el fichero?
num_words = fileRDD.flatMap(lambda line: line.split(" "))
print("README.md tiene " + str(num_words.count()) + " palabras")

# 3 - ¿En cuántas líneas aparece la palabra "spark"?
filterRDD = fileRDD.filter(lambda line: "spark" in line)
print("README.md tiene " + str(filterRDD.count()) + " líneas donde aparece la palabra spark")

# 4 - Imprimir por pantalla el nº de palabras de 5 líneas
num_words_5 = sc.parallelize(fileRDD.take(5)).flatMap(lambda line: line.split(" ")).count()
print("README.md tiene " + str(num_words_5) + " palabras en 5 líneas")
```

## Resultado

El resultado que debes obtener es el siguiente:



```
Run: Run with Saprk x
/Users/ignacio.perez.torres/PycharmProjects/MasterBDDS/venv/bin/python /Users/ignacio.perez.torres/PycharmProjects/MasterBDDS/Actividad1.py
20/09/01 21:11:15 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
README.md tiene 184 líneas
README.md tiene 557 palabras
README.md tiene 13 líneas donde aparece la palabra spark
README.md tiene 42 palabras en 5 líneas

Process finished with exit code 0
```