Laboratory #4 *(optional)*

Spark

What to do:

1. Download the data and the notebook from Campus Virtual.
2. Use Docker and the attached instructions to set up your working environment.
3. Complete the notebook and write the pieces of code required whenever you find the following label "`### YOUR CODE HERE ###`" or write your text answer whenever you find "`### YOUR ANSWER HERE ###`".

What to deliver:

- The notebook completed and executed in PDF format
  [File > Download as > PDF via LaTeX].

How:

- In pairs
- Only one delivery per pair

Delivery January 23rd (23:55) at UB's CV

Instructions

To work with Spark, there are a few possibilities. It is written in Scala and works natively over JVM. Thus, there is an interface for Java, but also for R and Python.

In this laboratory, we will use Spark's Python API. Specifically, we will run it from a Jupyter Notebook, an interactive web application that allows us to create and share documents with live code that can be executed, and its partial results explored step-by-step.

To set up everything, we will use a Docker container which already contains all the required dependencies. First, download the image:

```
docker pull jupyter/pyspark-notebook
```

Then, run the container:

```
docker run -d -p 8888:8888 --name notebook-pyspark
jupyter/pyspark-notebook
```

Now, we just move into the container the zip you downloaded from CV:

```
docker cp ./adb_lab4.zip notebook-pyspark:/home/jovyan/
```

Note that this previous command only will work if you have the zip in your current directory.

Then, run the container:

```
docker exec -it notebook-pyspark bash
```

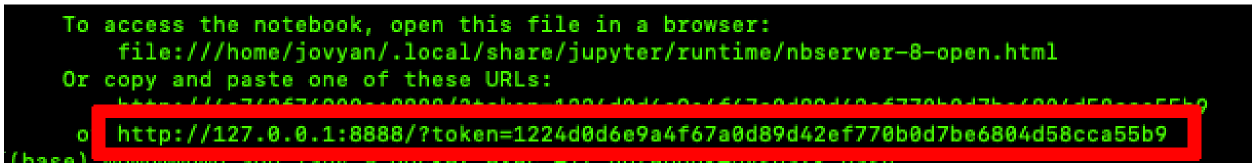
Now, we are in the container. We just need to unzip the file with the lab's material:

```
unzip adb_lab4.zip
```

Up to this point, if everything is working, we should be able to run our notebook. Let's see where the server is listening:

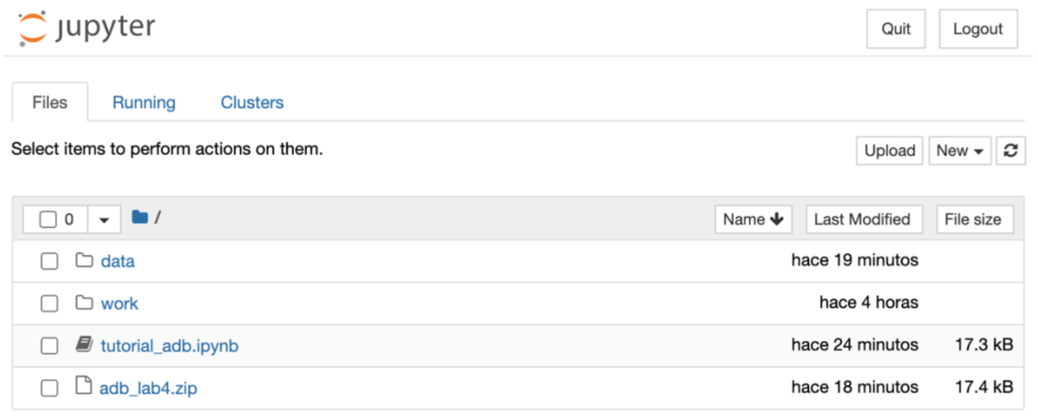
```
docker logs notebook-pyspark
```

You'll get something like:



```
    To access the notebook, open this file in a browser:
        file:///home/jovyan/.local/share/jupyter/runtime/nbserver-8-open.html
    Or copy and paste one of these URLs:
     o  http://127.0.0.1:8888/?token=1224d0d6e9a4f67a0d89d42ef770b0d7be6804d58cca55b9
```

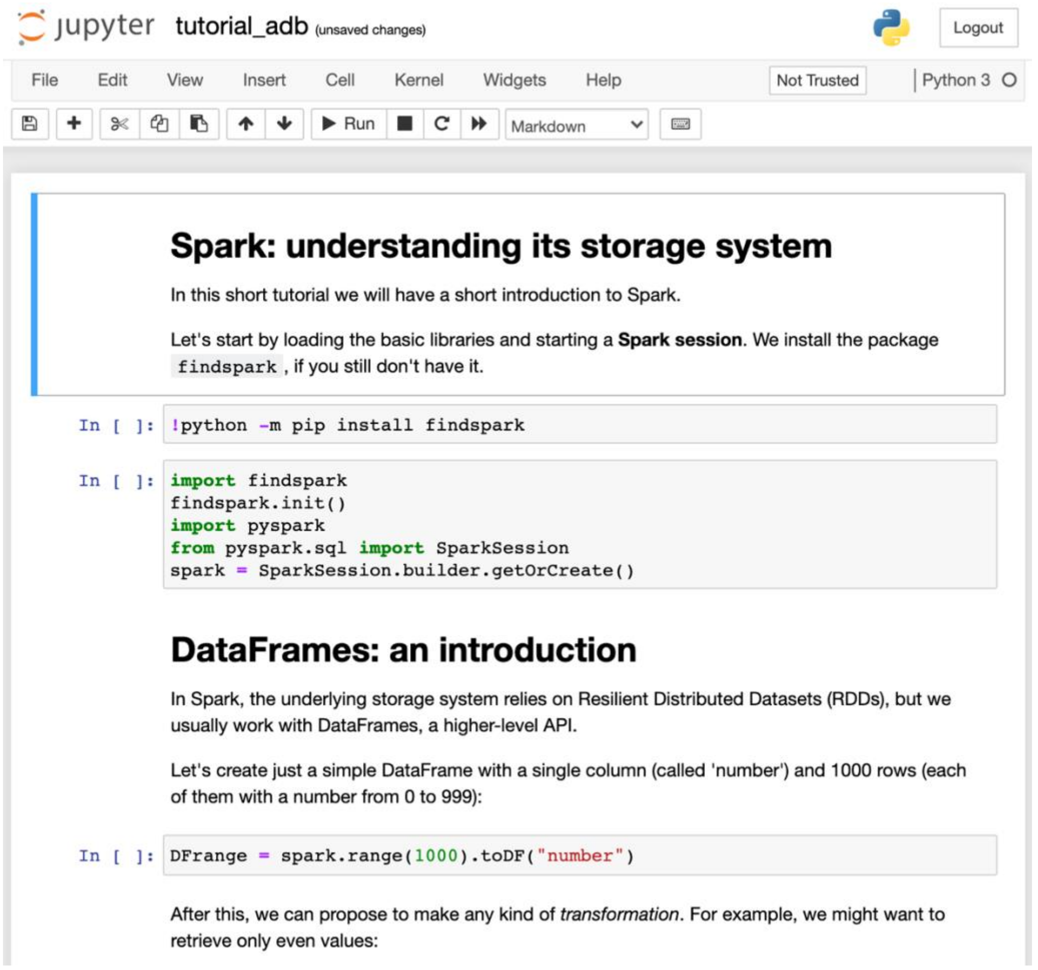
Copy the highlighted link into your browser and you'll see the Jupyter Notebook server:



Select items to perform actions on them.

data	hace 19 minutos	
work	hace 4 horas	
tutorial_adb.ipynb	hace 24 minutos	17.3 kB
adb_lab4.zip	hace 18 minutos	17.4 kB

Finally, you just need to select `tutorial_adb.ipynb` and you'll get into the notebook. Now, you are in position to start working within the lab! Use the button `Run` to execute the code step-by-step (one cell at a time).



Spark: understanding its storage system

In this short tutorial we will have a short introduction to Spark.

Let's start by loading the basic libraries and starting a **Spark session**. We install the package `findspark`, if you still don't have it.

```
In [ ]: !python -m pip install findspark
```

```
In [ ]: import findspark
        findspark.init()
        import pyspark
        from pyspark.sql import SparkSession
        spark = SparkSession.builder.getOrCreate()
```

DataFrames: an introduction

In Spark, the underlying storage system relies on Resilient Distributed Datasets (RDDs), but we usually work with DataFrames, a higher-level API.

Let's create just a simple DataFrame with a single column (called 'number') and 1000 rows (each of them with a number from 0 to 999):

```
In [ ]: DFrange = spark.range(1000).toDF("number")
```

After this, we can propose to make any kind of *transformation*. For example, we might want to retrieve only even values: