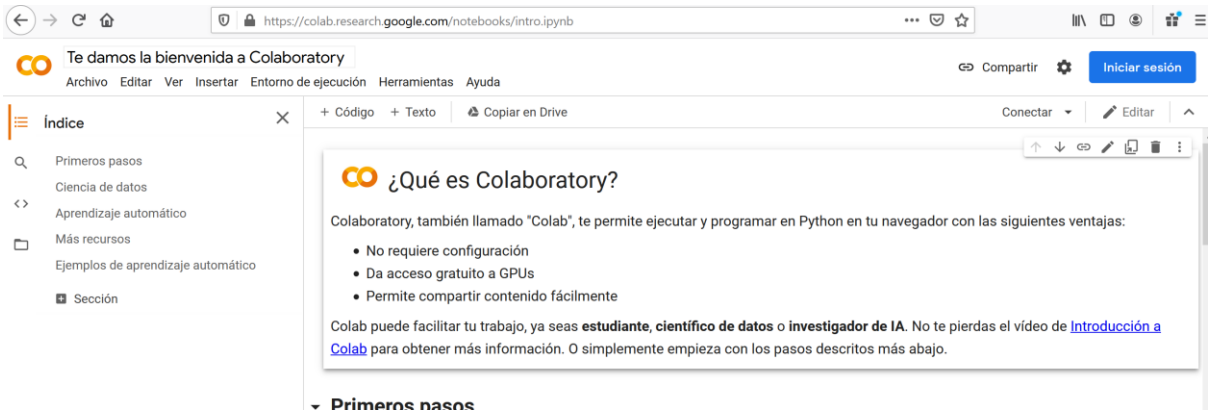
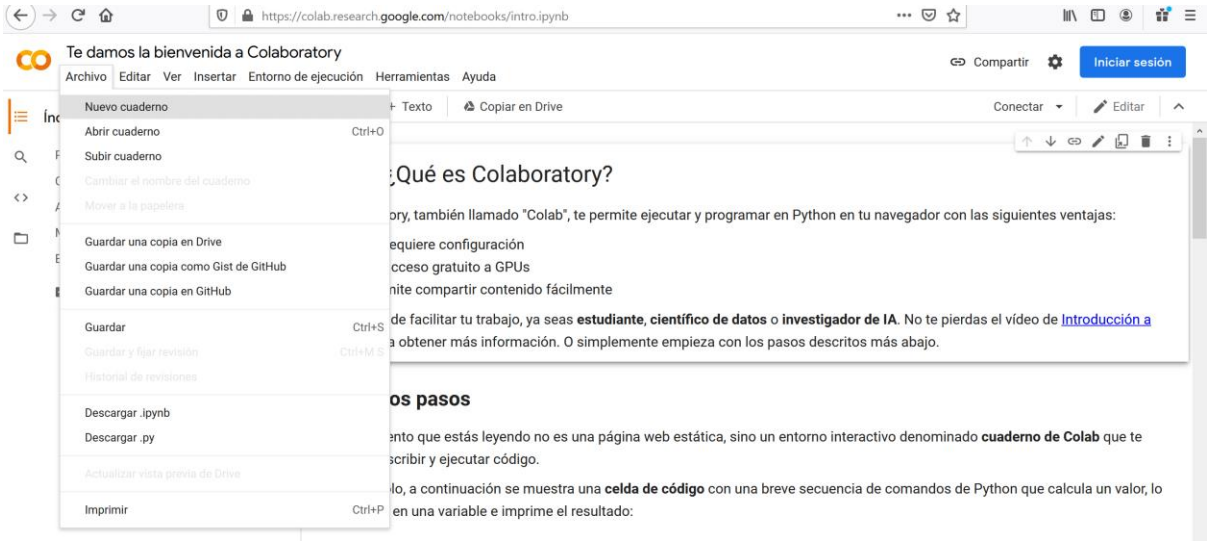
Instalar Spark en Google Colab

Google Colab permite crear un entorno de ejecución en la nube de Google de forma gratuita, siempre que Google tenga recursos para ello. Se basa en el notebook Jupyter.

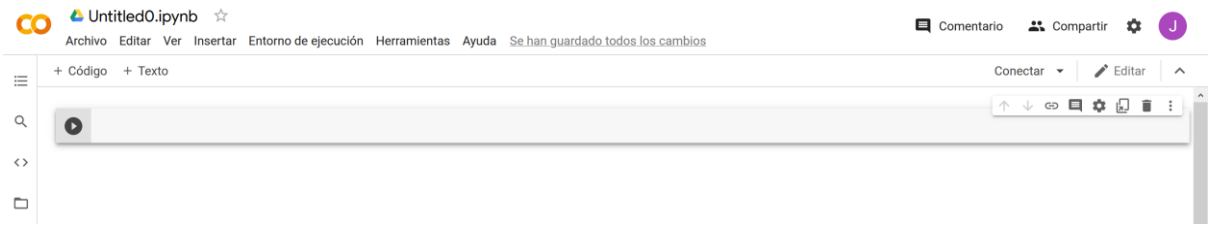
1. Entramos en la página de Google Colab: https://colab.research.google.com/



2. Entramos en Archivo -> Nuevo cuaderno:



3. Nos dice que tenemos que iniciar sesión con una cuenta de Google. Entonces iniciamos sesión:



Ya podemos empezar a escribir párrafos (celdas) con código Python, comandos, etc. Para ello tenemos que indicar cuál es el kernel sobre el que se ejecutará la nota

4. Instalamos java, para ello tenemos que indicar: (1) que queremos ejecutar un comando, y (2) que el comando es el de instalar. Es decir, escribimos:

```
!apt-get install openjdk-8-jdk-headless --quiet > /dev/null
```

```
[2]  !apt-get install openjdk-8-jdk-headless --quiet > /dev/null
```

4.1. Le damos a ejecutar para que instale java

5. Instalamos Spark:

   5.1. Creamos una nueva nota, para ello pulsamos en el botón “+ Código”:

   5.2. Descargamos Spark: para ello tenemos que indicar:

```
!wget --quiet https://dlcdn.apache.org/spark/spark-3.2.0/spark-3.2.0-bin-hadoop3.2.tgz
```

```
[3]  !wget --quiet https://dlcdn.apache.org/spark/spark-3.2.0/spark-3.2.0-bin-hadoop3.2.tgz
```

   5.3. Pulsamos el botón ejecutar celda (el del símbolo de Play).

   5.4. Ahora añadimos una nueva celda para indicar que descomprima Hadoop:

```
!tar xf spark-3.2.0-bin-hadoop3.2.tgz
```

```
[4]  !tar xf spark-3.2.0-bin-hadoop3.2.tgz
```

   5.5. Le damos a ejecutar a la última nota que acabamos de crear

6. Creamos las variables de entorno de java y Spark. Para ello creamos una nueva nota con lo siguiente:

```
import os
os.environ["JAVA_HOME"] =  "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.2.0-bin-hadoop3.2"
```

```
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.2.0-bin-hadoop3.2"
```

   6.1. Le damos a ejecutar la nota

7. Ejecutamos python de forma interactiva como si estuviésemos en pyspark. Para ello:

   7.1. Instalamos findspark para que nos permita importar pyspark de forma sencilla como una librería:

```
!pip install --quiet findspark
```

```
[10] !pip install --quiet findspark
```

   7.2. Iniciamos findspark:

```
import findspark

findspark.init()
```

```
[11] import findspark
     findspark.init()
```

7.3. Creamos el SparkContext sc tal y como está en pyspark:

```
from pyspark import SparkContext

sc = SparkContext("local", "Ejemplo interactivo")
```

```
[12] from pyspark import SparkContext
     sc = SparkContext("local", "Ejemplo interactivo")
```

7.4. Ejecutamos algún código de forma interactiva:

```
miRDD = sc.parallelize(["uno", "dos", "tres",
                        "cuatro", "cinco", "seis",
                        "siete", "ocho", "nueve"])

print("\nmiRDD:", miRDD.collect())
```

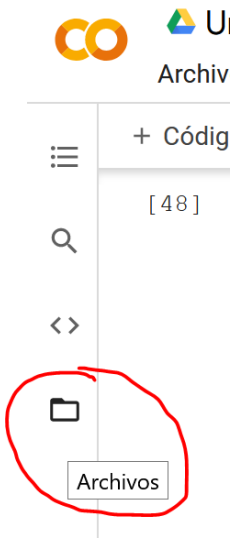
```
miRDD = sc.parallelize(["uno", "dos", "tres",
                        "cuatro", "cinco", "seis",
                        "siete", "ocho", "nueve"])
print("\nmiRDD:", miRDD.collect())

miRDD: ['uno', 'dos', 'tres', 'cuatro', 'cinco', 'seis', 'siete', 'ocho', 'nueve']
```

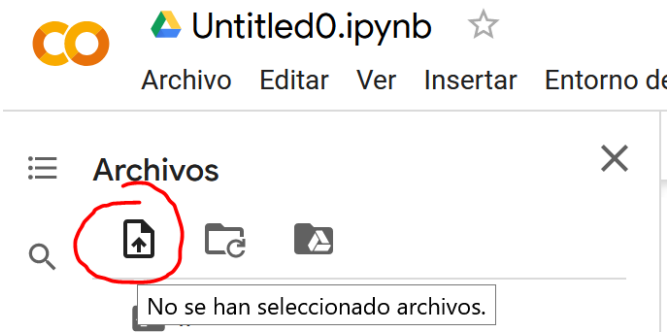
8. Ejecutamos un programa .py. Para ello:

8.1. Subimos el programa a Colab:

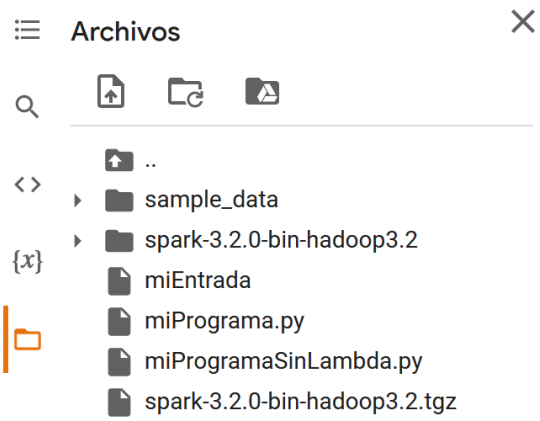
8.1.1. Pulsamos el botón Archivos:



8.1.2. Pulsamos el botón subir:



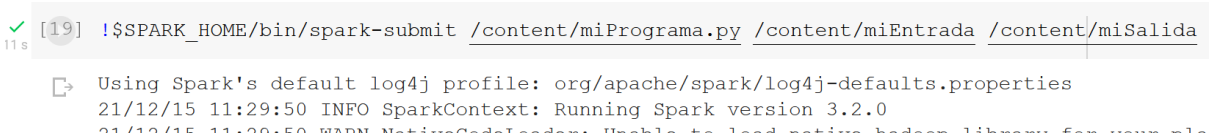
8.1.3. Se añaden los programas (ej. miPrograma.py) y archivos de entrada (ej.miEntrada):



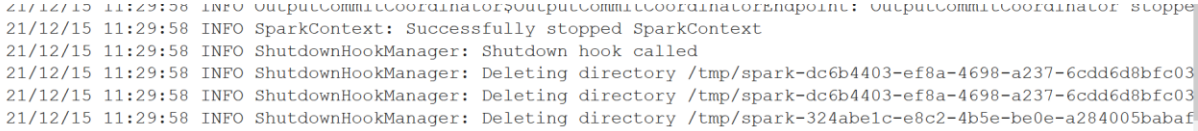
8.2. Ejecutamos el programa:

```
!$SPARK_HOME/bin/spark-submit /content/miPrograma.py
/content/miEntrada /content/miSalida
```

8.3. Cuando ejecutamos la celda tendremos algo del estilo:



...



8.4. Comprobamos la salida, para ello creamos una nota con lo siguiente:

```
!head miSalida/*
```

```
!head miSalida/*
```

```
==> miSalida/part-00000 <==
('Esto', 1)
('prueba', 1)
('para', 1)
('contar', 1)
('', 2)
('archivo', 1)
('varias', 1)
('programa', 1)
('el', 1)
('numero', 1)

==> miSalida/part-00001 <==
('es', 1)
('una', 1)
('palabras', 1)
('El', 2)
('tiene', 1)
('líneas', 1)
('cuenta', 1)
('de', 2)
('apariciones', 1)
('palabra', 1)

==> miSalida/_SUCCESS <==
```