

A stylized sunburst graphic in shades of purple and blue, located on the left side of the slide.

Tema 3: Procesamiento con Apache Spark

Máster en Ciencia de Datos (Universidad de Córdoba)

UCO
ONLINE

A decorative horizontal bar at the bottom of the slide, consisting of alternating yellow and red segments.

Introducción

Tema 3: Procesamiento con Apache Spark

UCO
ONLINE

Índice de la sección

- ¿Qué es Apache Spark?
- Características
- Componentes Principales.
- Trabajo Spark con Google Colab.
- Instalación Spark en Google Colab.
- SparkContext vs SparkSession.
- Creación de SparkContext
- Creación de SparkSession

¿Qué es Spark?

- Apache Spark es un sistema de computación de datos basado en Hadoop Map Reduce.
- Es un framework de computación (entorno de trabajo) en clúster open-source.
- **Spark** fue desarrollado en sus inicios por Matei Zaharia en el AMPLab de la UC Berkeley en 2009.
- La última versión es la 3.2.1.



Características (I)

- Spark está integrado con Hadoop.
- A diferencia de Hadoop trabaja en memoria, lo que supone un rendimiento comparado de unas 100 veces más rápido.
- Permite también trabajar en disco con la persistencia de datos
- Presenta varios lenguajes de programación que pueden operar mediante sus frameworks como son API Java, Scala, Python o R.

Características (II)

- Spark permite el procesamiento en tiempo real.
- Presenta los tipos de datos RDD (Resilient Distributed Dataset).
- Usa evaluación perezosa.

Componentes principales

Spark Core

Es la base o núcleo donde se apoya el resto de componentes.

Spark SQL

Procesamiento de datos estructurados y semi-estructurados.

Spark Streaming

Procesamiento de datos en tiempo real.

Spark MLlib

Librería de Machine Learning.

Spark Graph

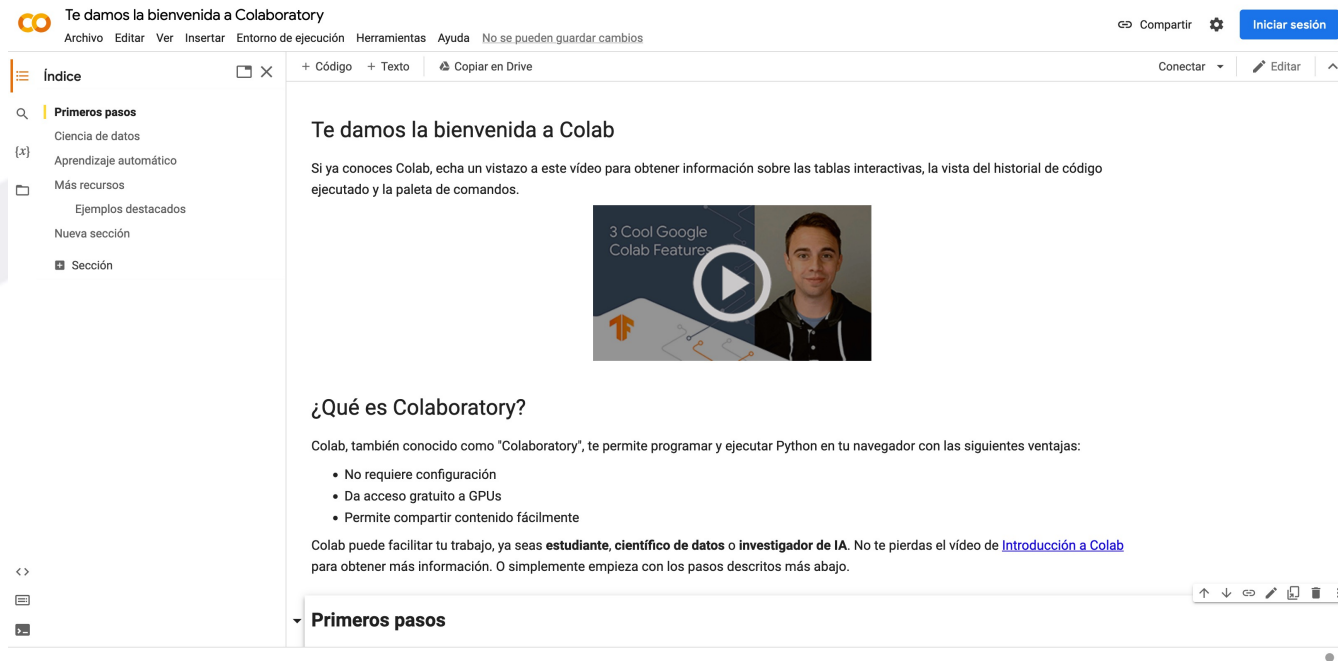
Procesamiento de Grafos.

Trabajo Spark con Google Colab (I)

- En nuestro caso, para facilitar el trabajo y no depender de instalaciones diferentes, trabajaremos con Google Colab.
- <https://colab.research.google.com/notebooks/welcome.ipynb?hl=es>
- Debemos tener una cuenta de Google.
- Nos servirá para realizar pruebas y posteriormente las tareas de este tema.

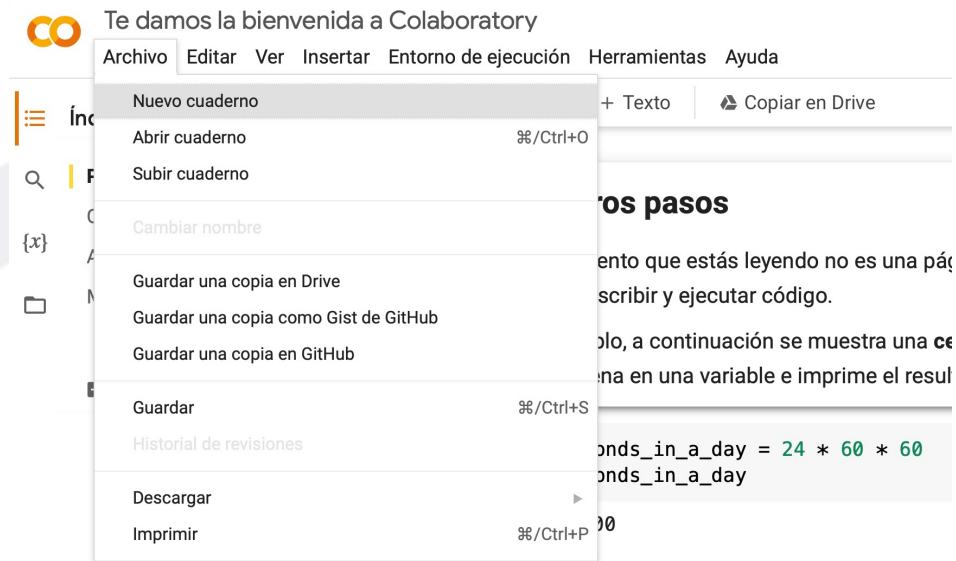
Trabajo Spark con Google Colab (II)

- Entorno interactivo Cuaderno de Colab, que permite escribir y ejecutar código



Instalación Spark en Google Colab (I)

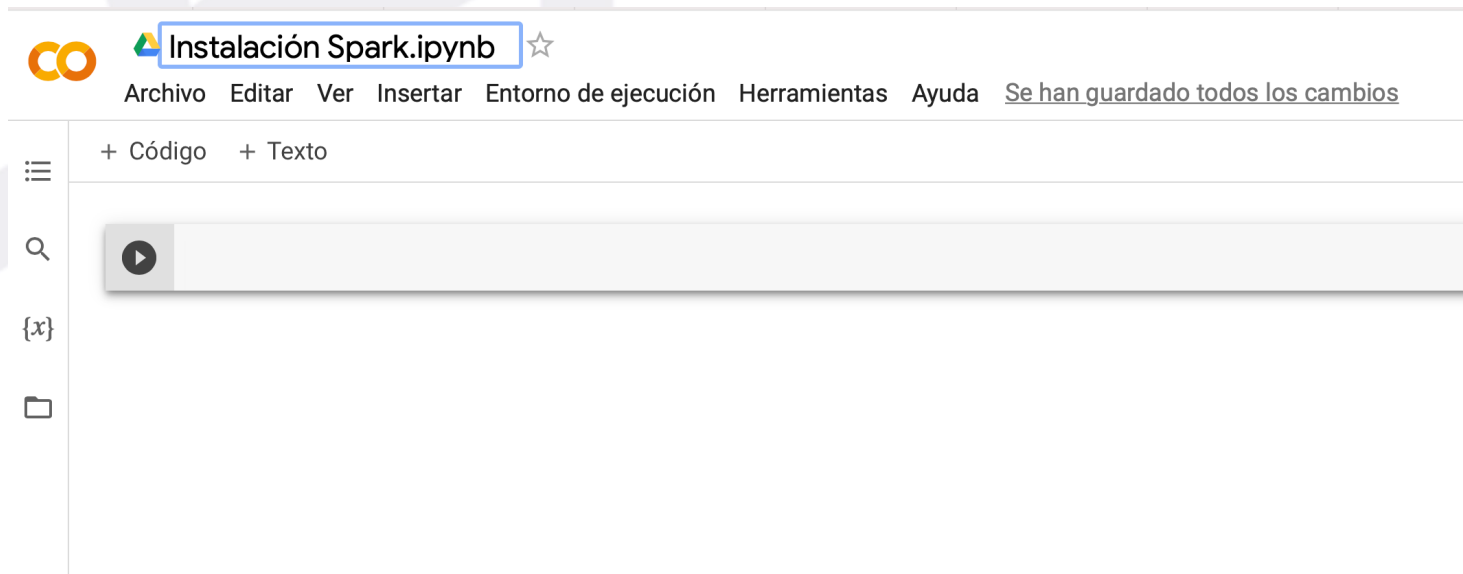
- Para realizar las pruebas simplemente creamos un nuevo cuaderno



Si quieres ejecutar el código de la celda anterior, haz clic en el icono de ejecución o usa la combinación de teclas "Comar

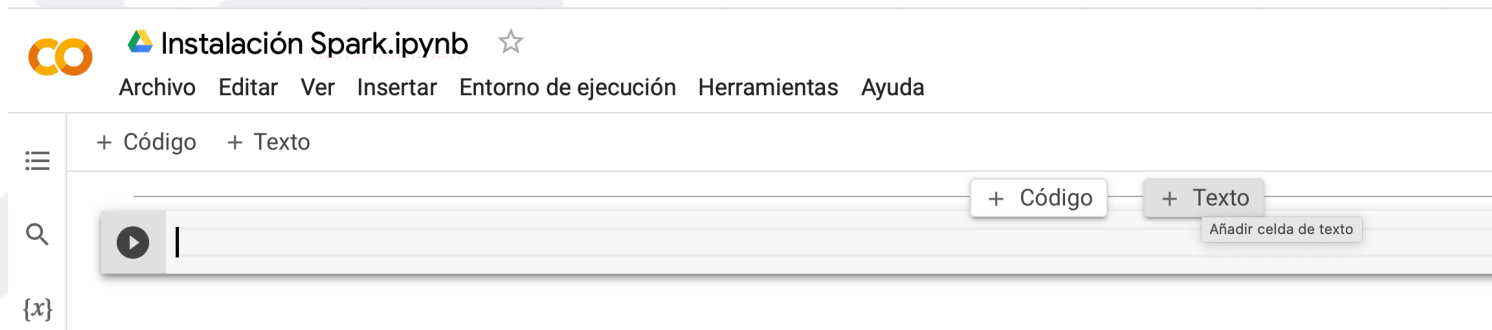
Instalación Spark en Google Colab (II)

- Cambiamos el nombre.



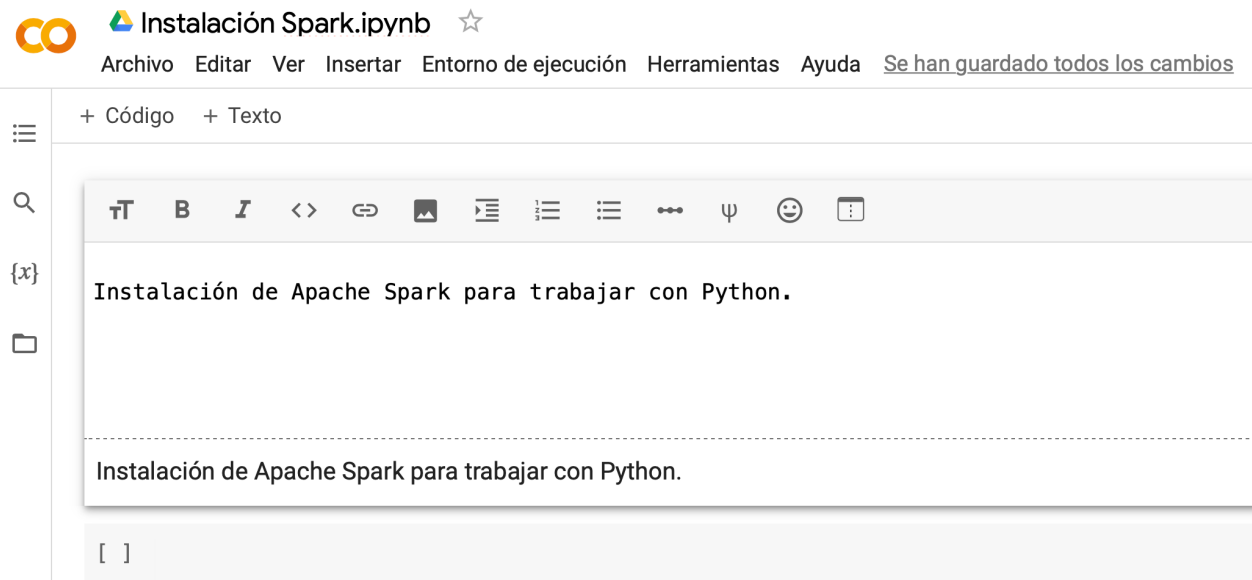
Instalación Spark en Google Colab (III)

- Añadimos un bloque de comentario “Texto” antes de l código.



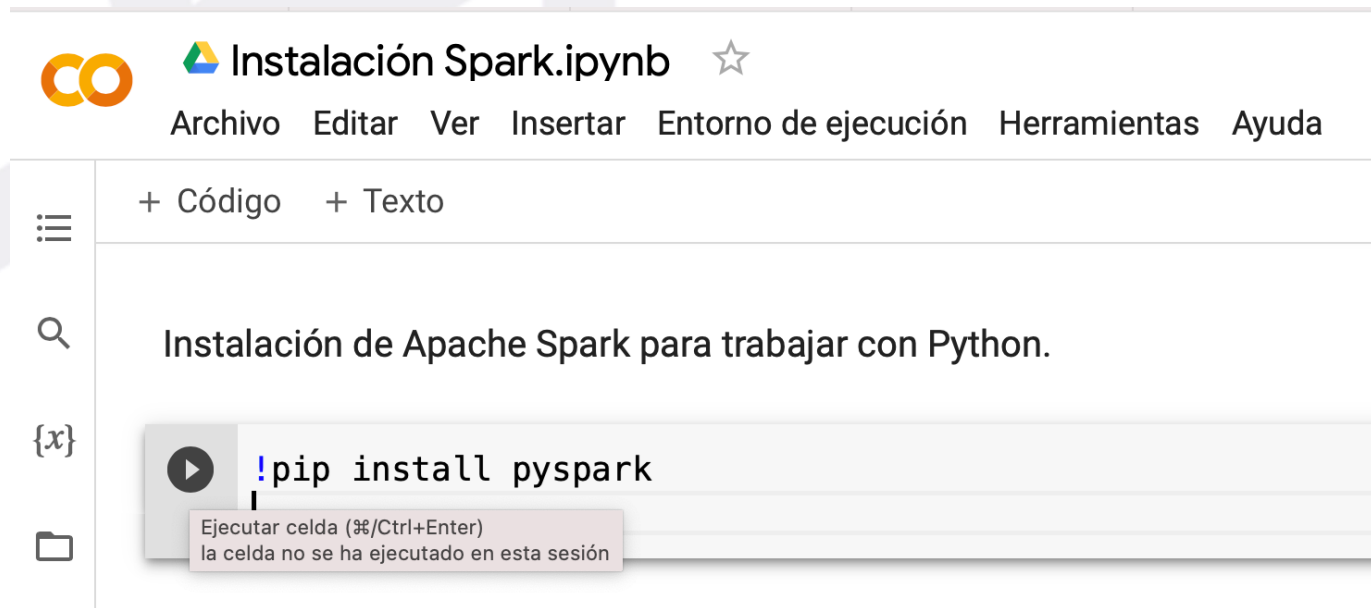
Instalación Spark en Google Colab (IV)

- Insertamos el texto en el bloque creado.



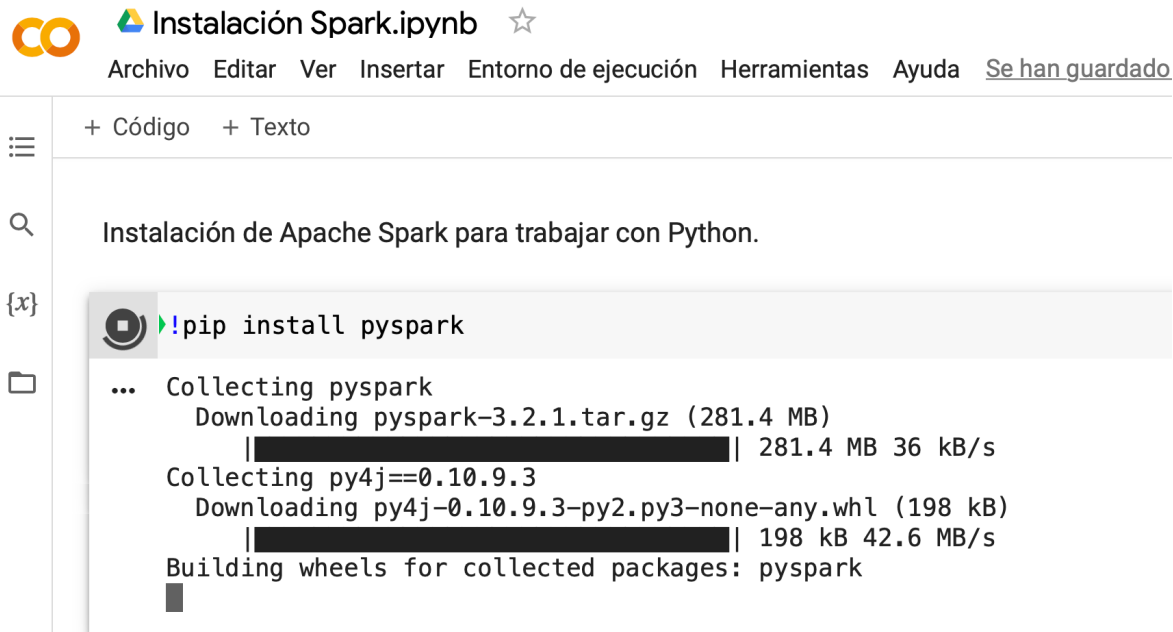
Instalación Spark en Google Colab (V)

- Insertamos la sentencia en el bloque de código debajo del bloque de texto.



Instalación Spark en Google Colab (VI)

- Pulsamos “play”, y en unos segundos se descargará, e instalará nuestro entorno.



The screenshot shows the Google Colab interface for a notebook titled "Instalación Spark.ipynb". The top menu bar includes "Archivo", "Editar", "Ver", "Insertar", "Entorno de ejecución", "Herramientas", "Ayuda", and a link "Se han guardado". The left sidebar has icons for file explorer, search, and code execution. The main area shows a code cell with the command `!pip install pyspark` and its output. The output indicates the collection and downloading of `pyspark-3.2.1.tar.gz` (281.4 MB) and `py4j-0.10.9.3` (198 kB), followed by the building of wheels.

```
!pip install pyspark

... Collecting pyspark
  Downloading pyspark-3.2.1.tar.gz (281.4 MB)
    |████████████████████████████████████████| 281.4 MB 36 kB/s
Collecting py4j==0.10.9.3
  Downloading py4j-0.10.9.3-py2.py3-none-any.whl (198 kB)
    |████████████████████████████████████████| 198 kB 42.6 MB/s
Building wheels for collected packages: pyspark
█
```

SparkContext Vs SparkSession

- **SparkContext:** Antes de Spark 2.0.0, se usaba `sparkContext` como un canal para acceder a todas las funciones de Spark. El programa del controlador Spark utiliza el contexto Spark para conectarse al clúster a través de un administrador de recursos (YARN o Mesos...). Se requiere `SparkConf` para crear el objeto de contexto Spark, que almacena parámetros de configuración como `appName` (para identificar su controlador Spark), aplicación, número de núcleos y tamaño de memoria del ejecutor que se ejecuta en el nodo trabajador. Para usar las API de SQL, HIVE y Streaming, se deben crear contextos separados.
- **SparkSession** proporciona un único punto de entrada para interactuar con la funcionalidad subyacente de Spark y permite programar Spark con API de `DataFrame` y `Dataset`. Toda la funcionalidad disponible con `sparkContext` también está disponible en `sparkSession`. Para usar las API de SQL, HIVE y Streaming, no es necesario crear contextos separados, ya que `sparkSession` incluye todas las API. Una vez que se crea una instancia de `SparkSession`, podemos configurar las propiedades de configuración de tiempo de ejecución de Spark.

Creación de SparkContext (I)

- A la hora de trabajar con RDDs en Spark necesitamos definir nuestra SparkContext.
- La clase SparkContext, es el punto de entrada principal para la funcionalidad de Spark. Un SparkContext representa la conexión a un clúster de Spark y se puede usar para crear RDD, acumuladores y variables de transmisión en ese clúster..

Importar la librería de SparkContext

```
[ ] from pyspark.context import SparkContext  
    SparkContext
```

```
pyspark.context.SparkContext
```

Creación de SparkContext (II)

- Algunas veces podríamos necesitar adicionalmente añadir la librería SparkConf

Importar la librería de SparkConf

```
▶ from pyspark.conf import SparkConf  
SparkConf
```

Creación de SparkSession(I)

- A la hora de trabajar con dataframes y datasets en Spark necesitamos definir nuestra SparkSession.
- La clase SparkSession, es el punto de entrada a la programación de Spark con la API Dataset y DataFrame.

```
[ ] from pyspark.sql import SparkSession
    spark = SparkSession.builder\
        .master("local")\
        .appName("Colab")\
        .config('spark.ui.port', '4050')\
        .getOrCreate()
```

Creación de SparkSession(I)

- Tras la importación de SparkSession, y la creación de la misma, podemos comprobar su instalación de la siguiente forma.

```
✓ 9 s [3] from pyspark.sql import SparkSession  
      spark = SparkSession.builder\  
        .master("local")\  
        .appName("Colab")\  
        .config('spark.ui.port', '4050')\  
        .getOrCreate()
```

```
✓ 0 s ▶ print("Apache Spark version: ", spark.version)
```

```
Apache Spark version: 3.2.1
```

A stylized sunburst graphic in shades of purple and blue, located in the top-left corner of the slide. It features a semi-circle on the left with several rays extending outwards to the right.

¡Gracias!

UCO
ONLINE

A decorative horizontal bar at the bottom of the slide, consisting of alternating yellow and red rectangular segments.