

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

# 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 rectangular segments.

# Conjuntos de Datos

Tema 3: Procesamiento con Apache Spark

**UCO**  
ONLINE

# Índice de la sección

- Introducción
- DataFrames
- Datasets

# Introducción

- Cuando estamos trabajando con conjuntos de datos muy grandes, es necesario que estos datos tengan una estructura.
- Los RDD tienen una gran complejidad y no presentan estructura.
- Es por ello que Spark introduce los conceptos de Dataset y Dataframe.

# Dataset (I)

- Un Dataset es una colección de datos distribuidos que tienen una estructura.
- Características:
  - Se introdujo a partir de Spark 1.6
  - Posee más beneficios que los RDD ya que permite por ejemplo, crear estructuras tipo clases de objetos, para que a la hora de obtener esos datos, estén estructurados y el trabajar con ellos sea más fácil y amigable.
  - Actualmente están disponibles en la API de Java y Scala.

## Dataset (II)

- Los conjuntos de datos son similares a los RDD, sin embargo, en lugar de utilizar la serialización de Java o Kryo, utilizan un codificador especializado para serializar los objetos para su procesamiento o transmisión a través de la red.

## Dataset (III)

- Si bien tanto los codificadores como la serialización estándar son responsables de convertir un objeto en bytes, los codificadores son códigos generados dinámicamente y usan un formato que permite a Spark realizar muchas operaciones como filtrar, clasificar y codificar sin deserializar los bytes nuevamente en un objeto.

# Dataframe (I)

- Un Dataframe es un Dataset que está organizado en columnas.
- Toda funcionalidad de SparkSQL depende del objeto SparkSession, es similar a SparkContext.
- Usamos SparkContext para RDD y SparkSession para DataFrame y DataSet.



## Dataframe (II)

- Con otras palabras, un DataFrame es un conjunto de datos organizado en columnas con nombre.
- Es conceptualmente equivalente a una tabla en una base de datos relacional o un marco de datos en R/Python, pero con optimizaciones más ricas bajo esa apariencia.
- Los dataframes se pueden construir a partir de una amplia gama de fuentes, como: archivos de datos estructurados, tablas en Hive, bases de datos externas o RDD existentes.

## Dataframe (III)

- La API de DataFrame está disponible en Scala, Java, Python y R.
- En Scala y Java, un DataFrame está representado por un conjunto de datos de filas.
- En la API de Scala, DataFrame es simplemente un alias de tipo de Dataset[Row]. Mientras que, en la API de Java, los usuarios deben usar Dataset<Row> para representar un DataFrame.

# Ejemplo Dataframe (I)

- Lo primero que hacemos es realizar las instalaciones oportunas

Los Dataframe están organizados por columnas

```
54 ✓ !pip install pyspark
s
Collecting pyspark
  Downloading pyspark-3.2.1.tar.gz (281.4 MB)
    [redacted] 281.4 MB 35 kB/s
Collecting py4j==0.10.9.3
  Downloading py4j-0.10.9.3-py2.py3-none-any.whl (198 kB)
    [redacted] 198 kB 50.4 MB/s
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.2.1-py2.py3-none-any.whl size=281853642 sha256=65927cb1066594e97f728f8f2617f7801758a459b07e9f9d5337b091417564bd
  Stored in directory: /root/.cache/pip/wheels/9f/f5/07/7cd8017084dce4e93e84e92efd1e1d5334db05f2e83bcef74f
Successfully built pyspark
Installing collected packages: py4j, pyspark
Successfully installed py4j-0.10.9.3 pyspark-3.2.1
```

## Ejemplo Dataframe (II)

- Se crea la SparkSession

✓  
7s

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

# Ejemplo Dataframe (III)

- Y se crea el Dataframe a partir de un fichero JSON

Creamos un dataframe a partir de un archivo JSON

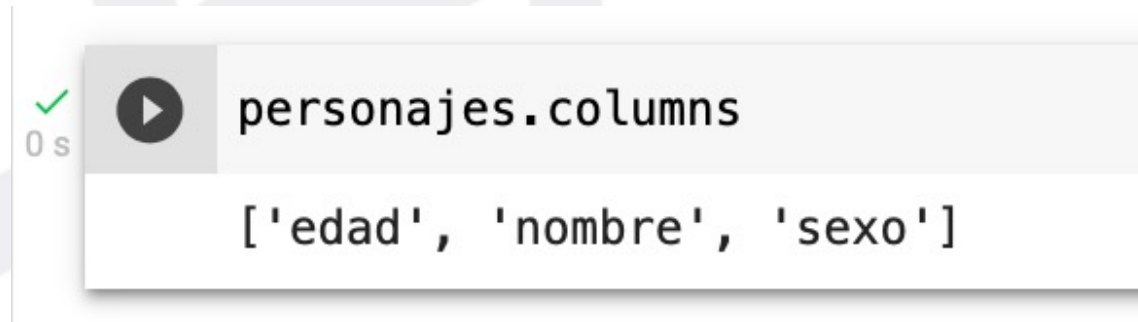
✓  
0 s

```
personajes=spark.read.json("strangersCharacters.json")  
personajes.show()
```

edad	nombre	sexo
12	Eleven	M
45	Hopper	H
11	Will	H
11	Max	M
11	Mike	H
11	Lucas	H
11	Dustin	H
12	Eleven	M
23	Nancy	M
24	Jonathan	H

## Ejemplo Dataframe (IV)

- Una vez cargado se puede operar sobre ese Dataframe y obtener información,



A screenshot of a Jupyter Notebook cell. On the left, there is a green checkmark and the text '0 s'. The cell contains the code `personajes.columns`. Below the code, the output is displayed as `['edad', 'nombre', 'sexo']`.

# Ejemplo Dataframe (V)

- Se puede ver el esquema del Dataframe,

✓  
0 s




```
personajes.printSchema()
```



```
root
|-- edad: long (nullable = true)
|-- nombre: string (nullable = true)
|-- sexo: string (nullable = true)
```

## Ejemplo Dataframe (VI)

- Mostrar los datos de una de las columnas (o campos) de nuestro Dataframe

```
✓ 0 s  nombresPersonajes=personajes.select("nombre").show()
```

nombre
Eleven
Hopper
Will
Max
Mike
Lucas
Dustin
Eleven
Nancy
Jonathan



## Ejemplo Dataframe (VII)

- Mostrar los datos de dos de las columnas (o campos) de nuestro Dataframe

```
✓ [10] nombresEdadPersonajes=personajes.select("nombre","edad").show()  
0 s
```

nombre	edad
Eleven	12
Hopper	45
Will	11
Max	11
Mike	11
Lucas	11
Dustin	11
Eleven	12
Nancy	23
Jonathan	24

## Ejemplo Dataframe (VIII)

- Aplicar filtros sobre los datos de los distintos campos de nuestro Dataframe

```
✓ 0 s ▶ menoresEdad=personajes.filter("edad<15")  
menoresEdad.show()
```

edad	nombre	sexo
12	Eleven	M
11	Will	H
11	Max	M
11	Mike	H
11	Lucas	H
11	Dustin	H
12	Eleven	M

## Ejemplo Dataframe (IX)

- Recoger información del mismo, como seleccionar el primer elemento

First devuelve el primero

✓  
0 s



```
menoresEdad.first()
```

```
Row(edad=12, nombre='Eleven', sexo='M')
```

# Ejemplo Dataframe (X)

- Recuperar algunos elementos, no solo el primero, con la función `head()`, que es similar a `take` con la que se trabaja en los RDD.

Similar a Take de los RDD

✓  
0 s



```
menoresEdad.head(3)
```

```
[Row(edad=12, nombre='Eleven', sexo='M'),  
Row(edad=11, nombre='Will', sexo='H'),  
Row(edad=11, nombre='Max', sexo='M')]
```

# Ejemplo Dataframe (XI)

- Se tiene la función `count()` para contar el número de elementos del dataframe.

Contar el número de elementos

✓  
0 s




```
menoresEdad.count()
```


7

## Ejemplo Dataframe (XII)

- Se tiene la función `groupBy()` para realizar agrupamientos y conteo de grupos del dataframe.

Función `groupBy` para agrupar

✓ 0 s  `personajes.groupBy("edad").count().show()`



edad	count
12	2
11	5
23	1
45	1
24	1

# Ejemplo Dataframe (XIII)

- Incluso estadísticas o resúmenes de los datos almacenados en nuestro Dataframe

Pequeñas estadísticas o resumen

✓  
1 s

▶ `personajes.describe().show()`

summary	edad	nombre	sexo
count	10	10	10
mean	17.1	null	null
stddev	11.049886877249017	null	null
min	11	Dustin	H
max	45	Will	M

# Descarga Ejemplos

- El ejemplo trabajado en esta sección está disponible para su descarga en la siguiente dirección:
- [https://drive.google.com/file/d/1ETKFiLO3ppn7Qdjhue9fQRB1\\_0BcwAeb/view?usp=sharing](https://drive.google.com/file/d/1ETKFiLO3ppn7Qdjhue9fQRB1_0BcwAeb/view?usp=sharing)
- El archivo de datos JSON para trabajar sobre el ejemplo está disponible en la dirección:
- <https://drive.google.com/file/d/1Myl-HL2uXrLNcuwWEcyDFHDUvTCOLjIR/view?usp=sharing>



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.