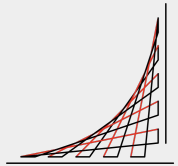
# PFSD

Programación Funcional en Sistemas Distribuidos
2022-2

Ingeniero Robert Camilo Martínez Páez

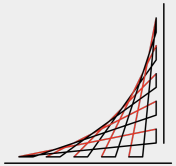
Octubre 8 de 2022



ESCUELA
COLOMBIANA
DE INGENIERÍA
JULIO GARAVITO

# Agenda

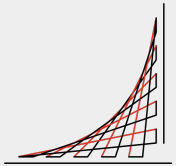
- Qué es Spark?
- Historia
- Componentes de Spark
- Módulos de Spark
- APIs ofrecidos
- RDDs - Resilient Distributed Datasets
- APIs estructurados
- Ambientes de despliegue
- Fuentes de Datos
- ¿Dónde correr Spark?
- Componentes de un clúster de Spark



ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

# Agenda

- Proyecto de curso
- Spark session
- Dataframes vs Datasets
- Dataframes
- Datasets
- Ejercicios



ESCUELA
COLOMBIANA
DE INGENIERÍA
JULIO GARAVITO

# Qué es Spark?


Es un motor de análisis cuyas características principales son:

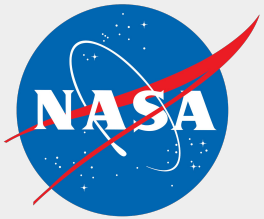
- Capacidad de procesamiento
- Analítica avanzada
- Facilidad de usar

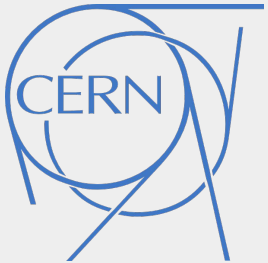
ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

# Qué es Spark?

Es uno de los proyectos open source más activo:

https://github.com/apache/spark







ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

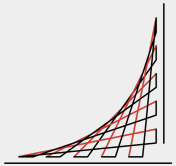
# Historia

## 2009

Para la época Hadoop era la solución más común para lograr procesamiento paralelo, esta tenía deficiencias que generaban periodos de procesamiento extensos.

Spark nace cómo un proyecto de investigación del equipo Amp Lab de la Universidad de Berkeley.

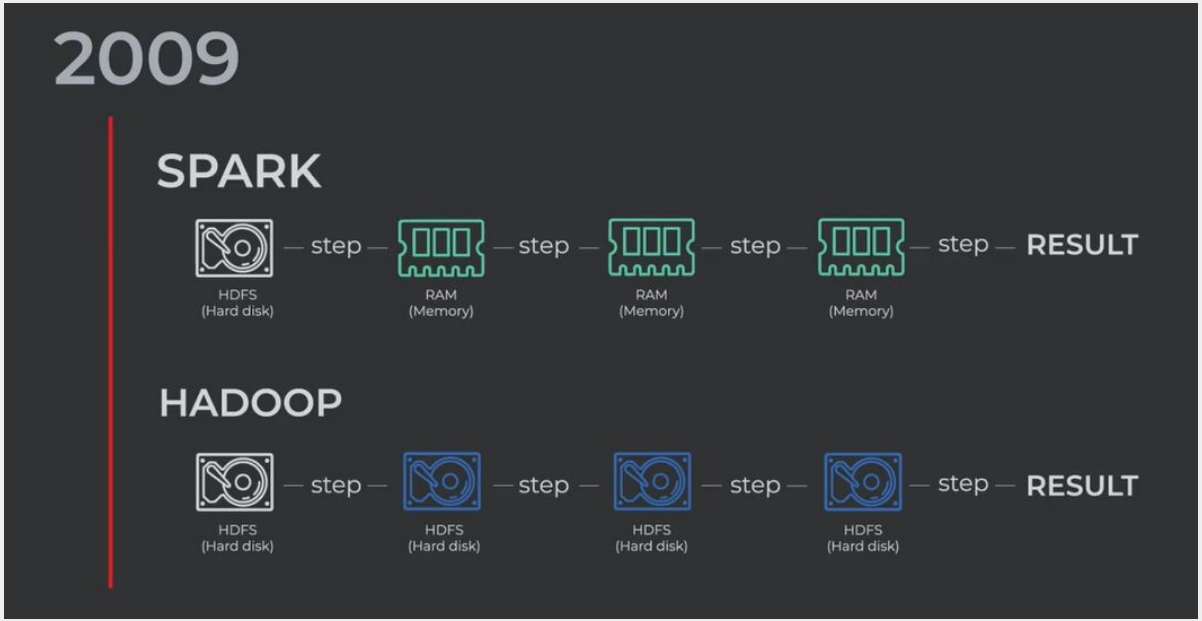
Diseño de un API en programación funcional que permitía crear aplicaciones basadas en múltiples pasos (steps).



ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

# Historia

El motor desarrollado tenía un rendimiento eficiente en memoria, con intercambios de información entre cada paso.

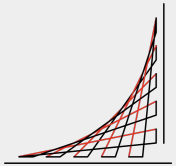



ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

# Historia

Spark resulto ser...

**100 veces más rápido que Hadoop!**

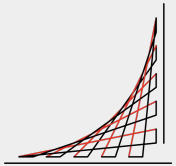


ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

# Historia

## 2013

- Spark es ampliamente aceptado
- Spark es agregado a la fundación Apache
- Amp Lab crea Databricks que es la encargada de hacer anualmente las conferencias de Spark Summit



ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

# Historia

## 2014

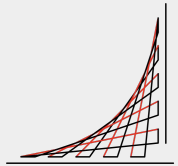
- Version 1.0

## 2016

- Version 2.0

## 2020

- Version 3.0

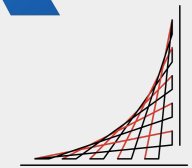


ESCUELA
COLOMBIANA
DE INGENIERÍA
JULIO GARAVITO

# Componentes de Spark

Librería disponibles en 4 lenguajes de programación



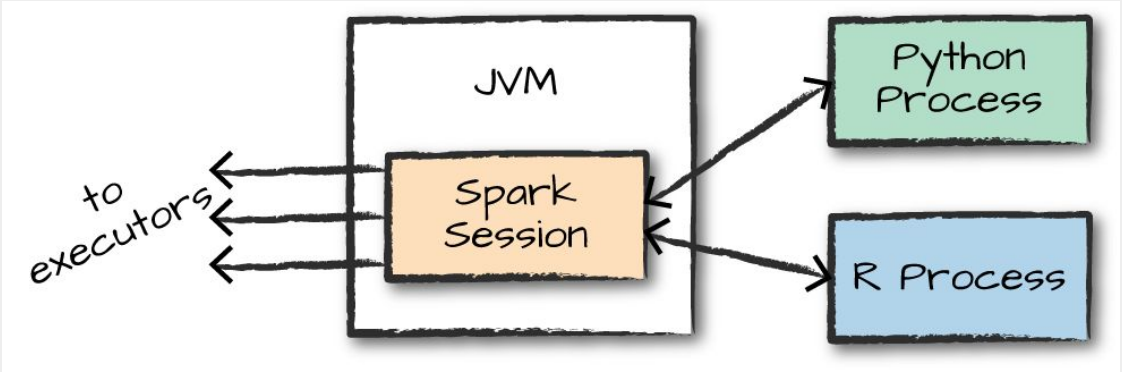



ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

# Componentes de Spark

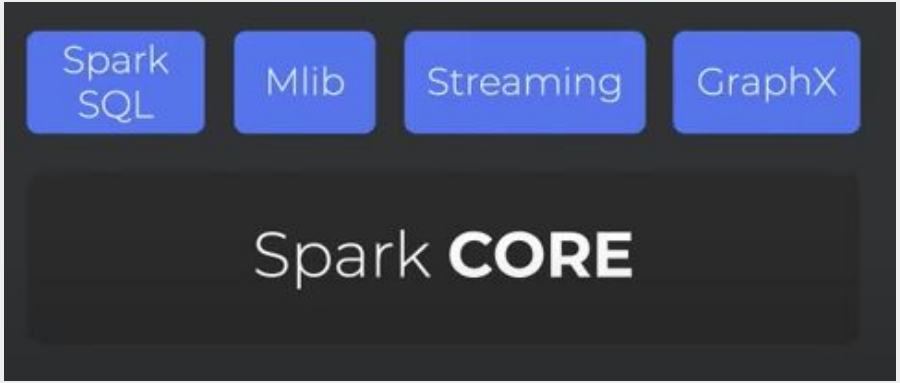
**Spark** está escrito en *Scala*, lo que quiere decir que funciona nativamente sobre la **JVM**. Así mismo, código **Spark** escrito en *Java*, va a funcionar sobre la misma máquina virtual.

Sí el lenguaje utilizado es *Python* o *R* va a existir un **proceso intermedio** que se encarga de poder interpretar dicho código desde la *JVM*.




ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

# Módulos de Spark




- ***Spark CORE*** - RDDs
- ***Spark SQL*** - APIs estructurados - Dataset, Dataframe y SQL
- ***Mlib*** - Machine learning
- ***Streaming*** - Aplicaciones en tiempo real
- ***GraphX*** - Computación Gráfica

# Módulos de Spark


ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

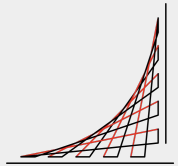
# APIs ofrecidos

**Spark** hoy en día provee dos APIs.

Uno basado en esquemas o estructuras definidas y otro que no.

- El API que no maneja estructura son los **RDDs** y hace parte de la versión 1.0.

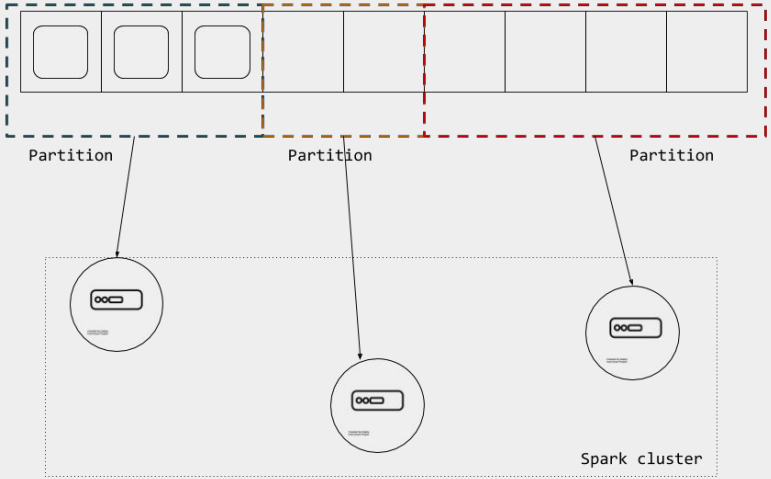
- Los APIs más recomendables son los **estructurados**, mirémoslo en detalle.



ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

# RDDs - Resilient Distributed Datasets

Podría entenderse cómo una colección inmutable, cuyos elementos son distribuidos en un conjunto de máquinas o clúster, que puede ser operada en paralelo.



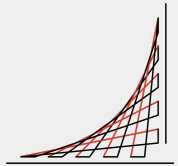

ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

# RDDs - Resilient Distributed Datasets

Al no existir esquema definido, se tiene completo control sobre cada elemento lo que hace que el uso de este **API sea de muy bajo nivel**, luego hay que tener pleno conocimiento de que se está haciendo en cada operación.

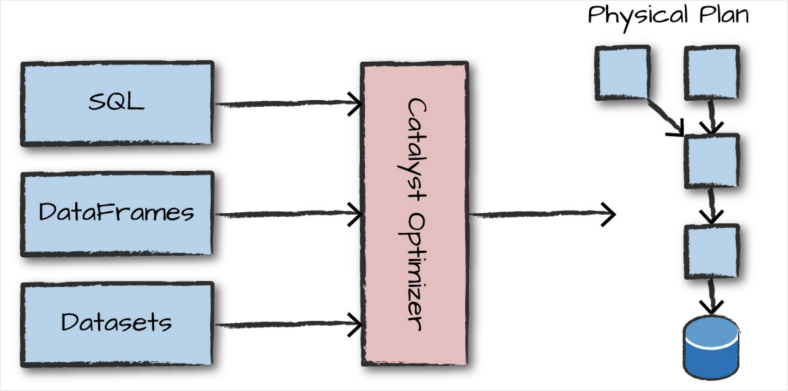
Hoy en día el uso de **RDDs** se hace para leer y/o manipular *información cruda*, pues al no tener un esquema definido Spark no puede hacer ninguna optimización automáticamente, que sí nos dan los **APIs estructurados**.

A no ser que se esté manteniendo código, lo recomendable es usar la segunda opción de APIs.



ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

# APIs estructurados



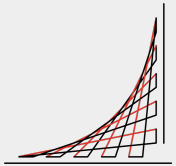

Son colecciones distribuidas con una estructura similar a tablas en bases de datos relacionales.

Dichas colecciones son *inmutables* y *lazy*.

ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

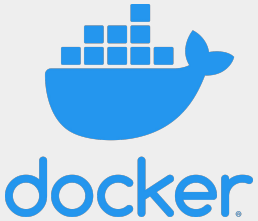
# APIs estructurados

Se consideran de alto nível porque libera al desarrollador de definir el plan de ejecución de ciertas operaciones, y se basa en el uso del motor de optimización ***Catalyst*** y es este quién se encarga de planear y procesar el trabajo de la mejor manera.



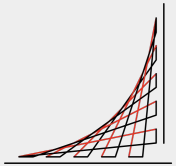
ESCUELA
COLOMBIANA
DE INGENIERÍA
JULIO GARAVITO

# Ambientes de despliegue









ESCUELA
COLOMBIANA
DE INGENIERÍA
JULIO GARAVITO

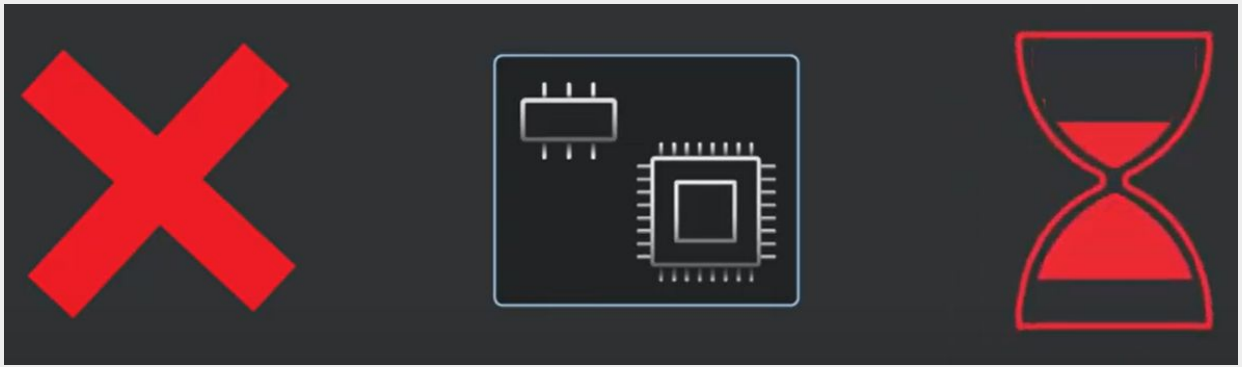
# Fuentes de Datos


ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

# ¿Dónde correr Spark?

Hoy en día se pueden adquirir máquinas con recursos de RAM y procesamiento bastante altos.

Sin embargo, una sola máquina no tiene la capacidad suficiente para realizar este procesamiento por sí sola o se tardaría demasiado.



ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

# ¿Dónde correr Spark?

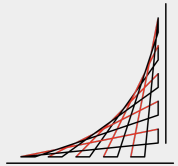
Un clúster nos permite agrupar recursos y procesamiento para lograr el objetivo cómo si se estuviera en una sola máquina.

ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

# ¿Dónde correr Spark?

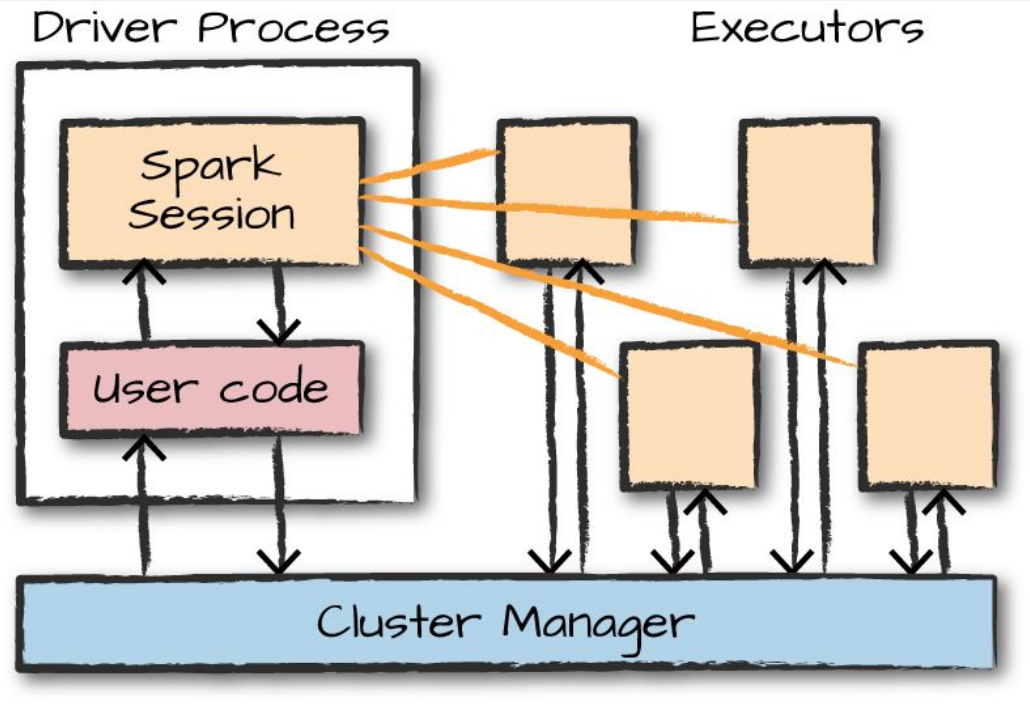
Spark se encarga de administrar y coordinar la ejecución de tareas sobre datos distribuidos en un clúster.

Las máquinas o nodos del clúster pueden estar definidas en la nube o en un datacenter On Premise.



ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

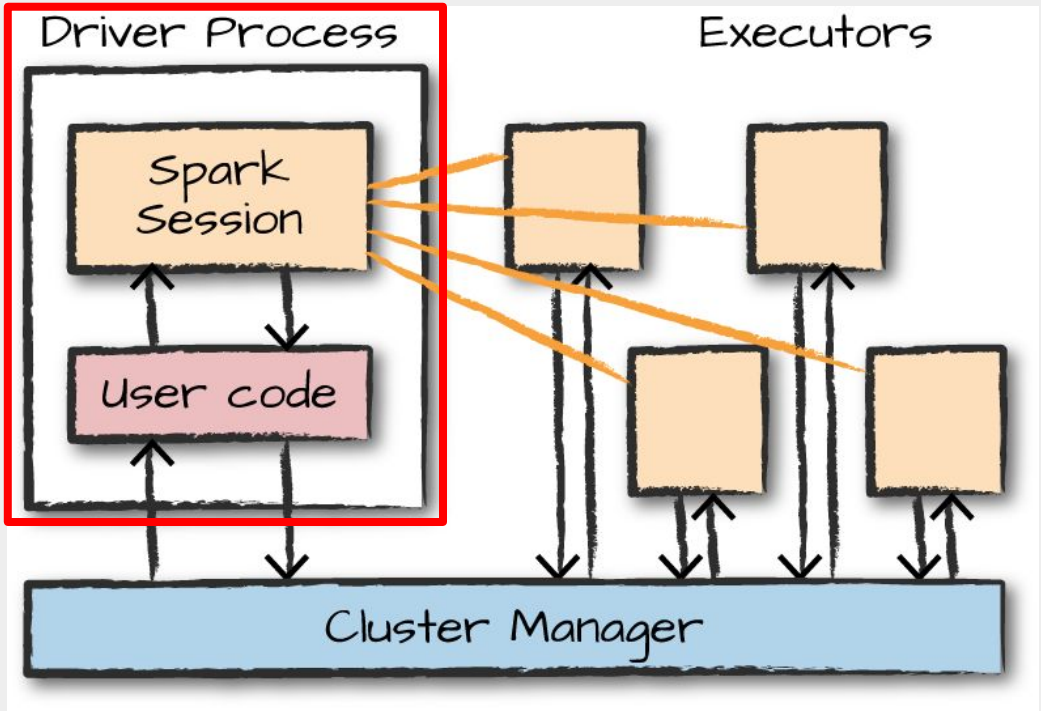
# Componentes de un clúster de Spark




ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

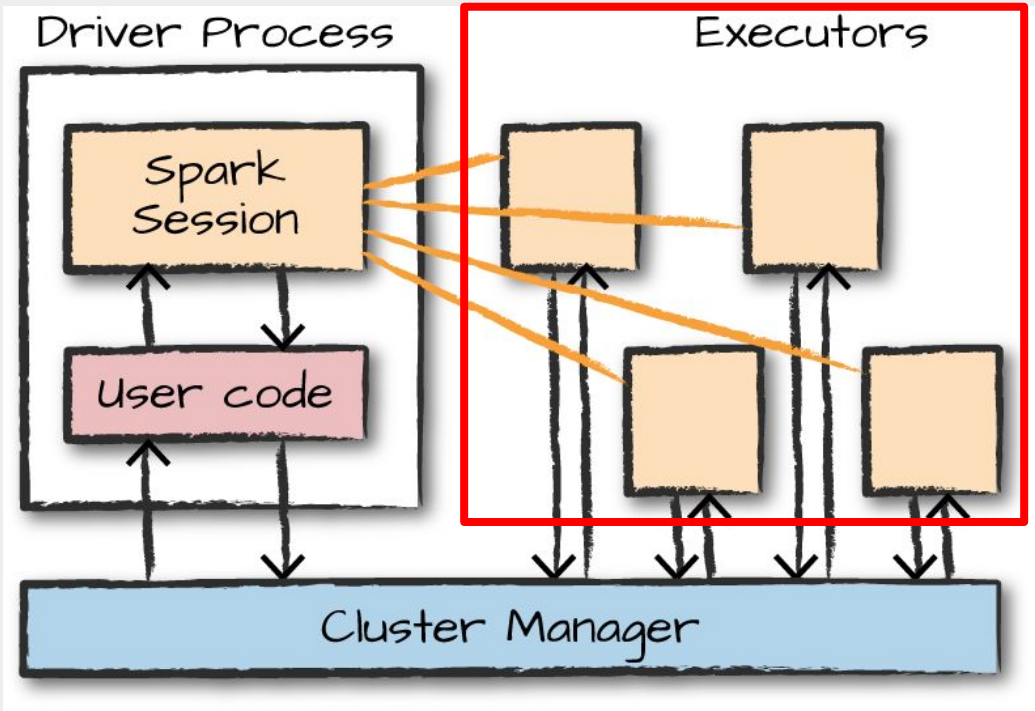
# Componentes de un clúster de Spark

- Mantiene información de la aplicación
- Analizar, distribuir y programar el trabajo entre los executors




ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

# Componentes de un clúster de Spark

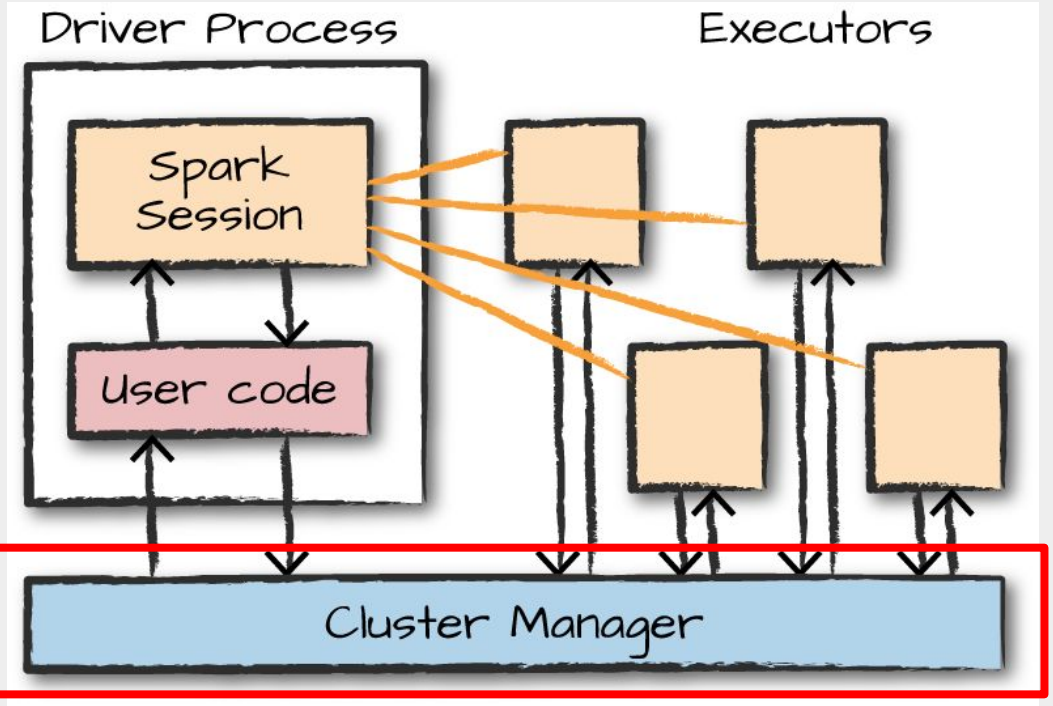



- Ejecutar código asignado
- Reporta estado de ejecución al driver
- Segmentación de datos en particiones, mayor número de executors, mayor trabajo en paralelo, ejecución de Spark más rápida.

ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

# Componentes de un clúster de Spark

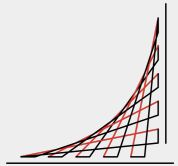
- Controlar las máquinas físicas
- Asignar recursos a la aplicación



# Proyecto de curso

Piensen en un problema que involucre

- Volúmenes de datos (Sería ideal una fuente de datos que les guste o que esté asociada a su labor)
- Que fin querrían tener con esos datos
  - ¿Necesariamente una base de datos?
  - ¿Qué tal una notificación?
  - ¿Un correo?
- ¿Cómo ven que la inmutabilidad creen beneficiaría el procesamiento de dichos datos?

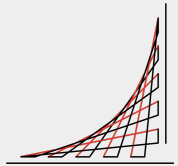


ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

# Proyecto de curso

Con lo que han pensado, podría enfocarse a

- Un pipeline de datos con Kafka y llegar a un fin
- Procesar datos con Spark (batches)
- Un híbrido? Que el fin sea Spark? Es decir un insumo para hacer algún procesamiento?

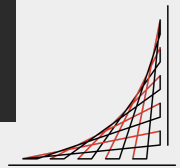


ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

# Spark session

Es el punto de entrada para una aplicación de *Spark*. Es a partir de dicha **sesión** que podemos empezar a realizar operaciones.

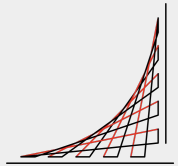
```scala
package eci.edu.co

import org.apache.spark.sql.SparkSession

trait SparkSessionWrapper extends Serializable {

  // Starting point of any application
  lazy val spark: SparkSession = {
    SparkSession
      .builder()
      .master( master = "local") //In a real cluster the ip of the master node
      .appName( name = "sparkLocalSession")
      .getOrCreate()
  }
}
```



ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

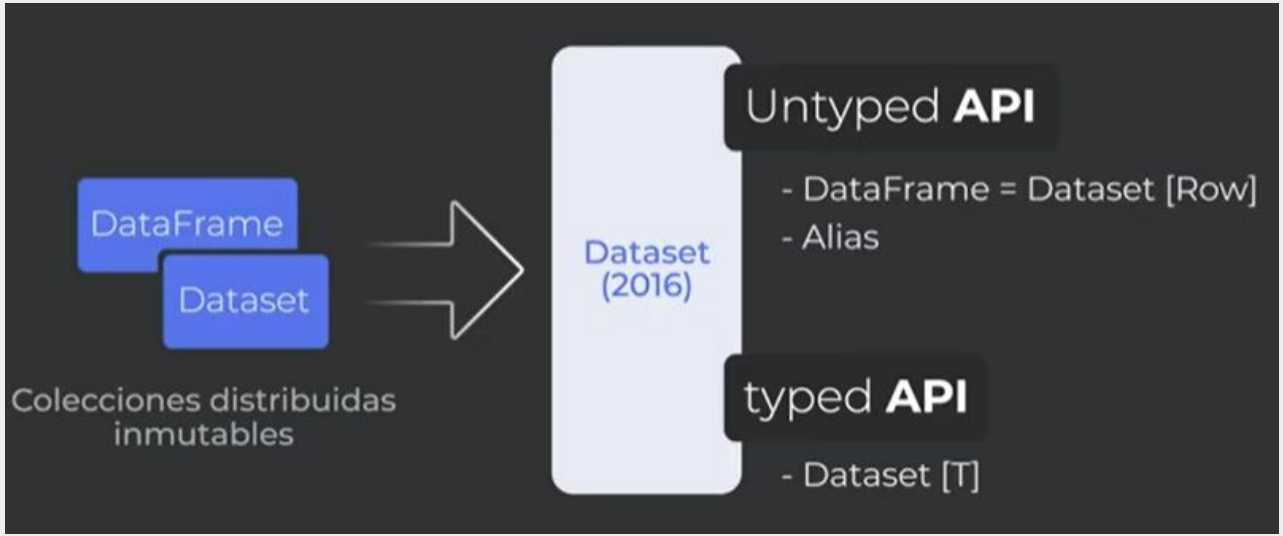
# Spark session

También en la **sesión** de *Spark* podemos cambiar parámetros de configuración por defecto, o agregar credenciales para autenticarse ante un clúster por ejemplo.



ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

# Dataframes vs Datasets



# Dataframes

Vamos a enfocarnos en los *Dataframes* y *Datasets*, qué son las abstracciones más utilizadas al programar en Spark.

Los *Dataframes* y *Datasets* hacen parte del módulo de **Spark SQL**, enfocado al trabajo sobre conjuntos de datos estructurados que cómo vimos anteriormente, incluye el motor de procesamiento y optimización ***Catalyst*** .


ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

# Dataframes

De la **sesión** de *Spark* podemos leer en este caso un archivo csv, y el tipo de retorno será un Dataframe, es decir genérico de tipo Row.

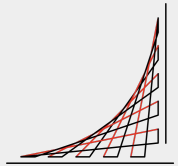
```
val transactions: DataFrame =
  spark
    .read
    .format( source = "csv")
    .option("header", "true")
    .load( path = "transactions.csv")
```

https://mungingdata.com/apache-spark/introduction-to-dataframes/

# Transformaciones y acciones

Las *transformaciones* actúan sobre colecciones lazy, y retornan así mismo resultados lazy.

Las *acciones* hacen que se materialicen dichas transformaciones, es decir desencadenan las operaciones necesarias que quedaron definidas para obtener un resultado.



ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

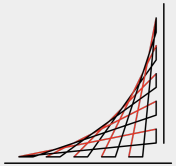
# Datasets

La **sesión** de *Spark* va a ser la misma, pero vamos a indicarle a qué clase queremos mapear la lectura del archivo Csv.

De esa manera vamos a obtener un Dataset tipado a ***Transaction***.

```
val transactions =
  spark
    .read
    .format( source = "csv")
    .option("inferSchema", "true")
    .option("header", "true")
    .load( path = "transactions.csv")
    .withColumn( colName = "created", $"created".cast( to = "timestamp"))
    .as[Transaction]
```

https://docs.databricks.com/spark/latest/dataframes-datasets/introduction-to-datasets.html

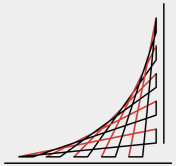


ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

# Datasets

Los atributos de la clase, deben corresponder a los nombres dispuestos en el header del Csv.

Convertir cada récord a un objeto específico tiene un **costo en desempeño** asociado, sin embargo, la flexibilidad que brinda el poder empezar a utilizar las operaciones sobre colecciones en Scala que hemos venido trabajando usualmente hace que valga la pena pagar ese costo.

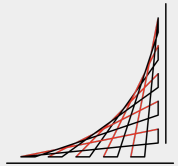


ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

# Ejercicios

A partir del proyecto: https://github.com/PFSD-ECI/2022-PFSD-Session-08

- Complete los *TODOs* dejados en los archivos:
  - DatasetExercise3
  - DatasetExercise4
- Realice la tarea descrita en el **README**



ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO