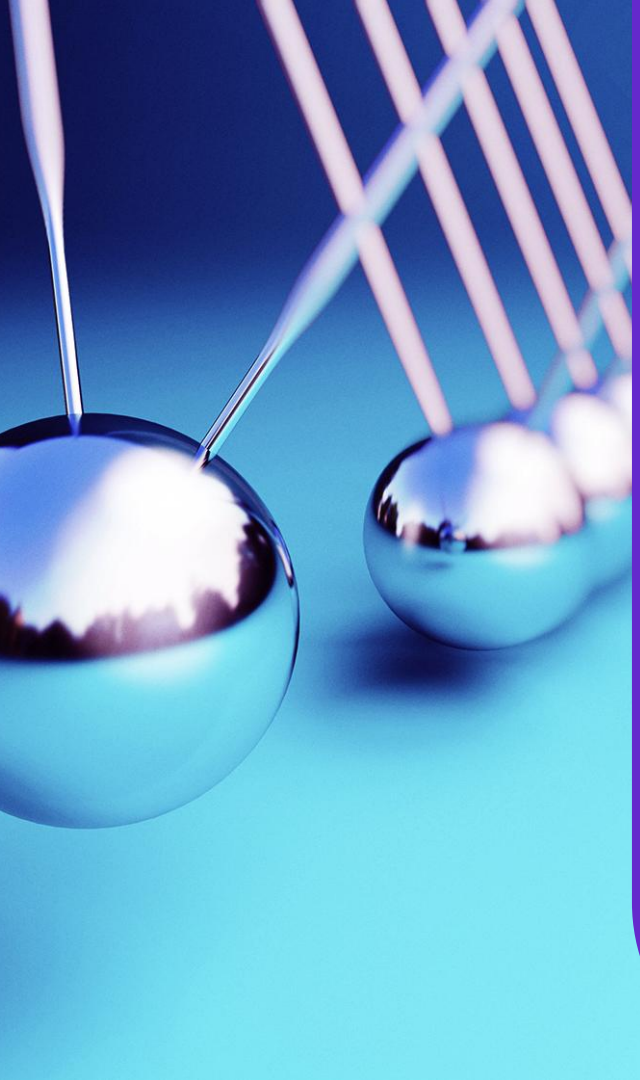


pragma | Keep moving

# Pipelines de datos con **AWS Glue** y Apache Spark

Mayo 2024

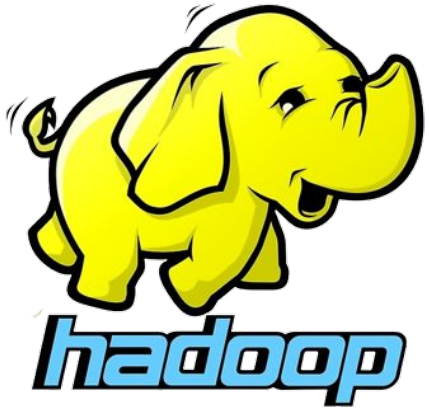
# Contenido

1. Introducción - Apache Hadoop
2. Introducción - Apache Spark
3. Taller práctico

# 1 Frameworks para el procesamiento de **Big Data**

## Hadoop

# ¿Qué es Hadoop?



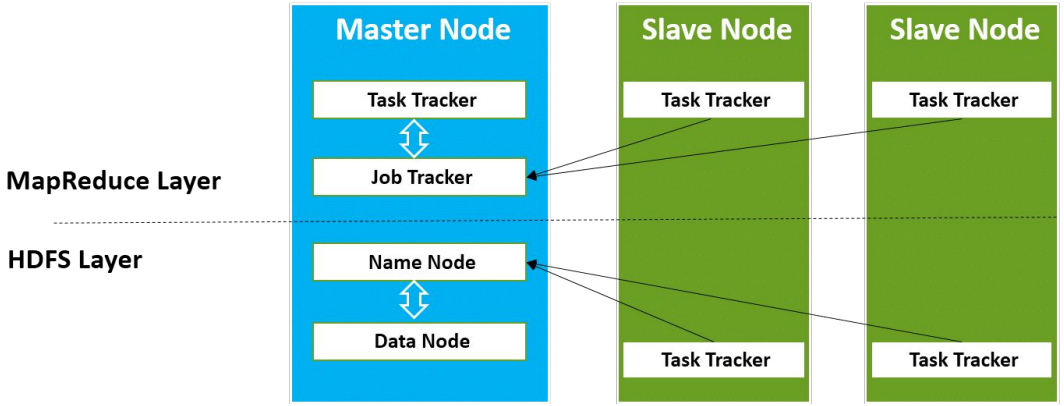

Apache Hadoop es un framework open source, para programar aplicaciones distribuidas que manejen **grandes volúmenes de datos**. Permite a las aplicaciones trabajar con miles de nodos en red y petabytes de datos. Hadoop se inspiró en las investigaciones de Google sobre **MapReduce** y **Google File System** (GFS).

# Arquitectura Hadoop
## Sistema de archivos HDFS

HDFS, o **Hadoop Distributed File System**, es un sistema de archivos distribuido que se utiliza en Hadoop para almacenar grandes conjuntos de datos. Funciona de la siguiente manera:

### Almacenamiento de datos

- Los datos se dividen en bloques de tamaño fijo (por defecto, 128 MB).
- Estos bloques se replican en varios nodos del clúster Hadoop para mayor seguridad y disponibilidad.
- Los metadatos de los archivos (ubicación de los bloques, tamaño, etc.) se almacenan en un nodo central llamado NameNode.




**Fuente:** Introduction to Hadoop

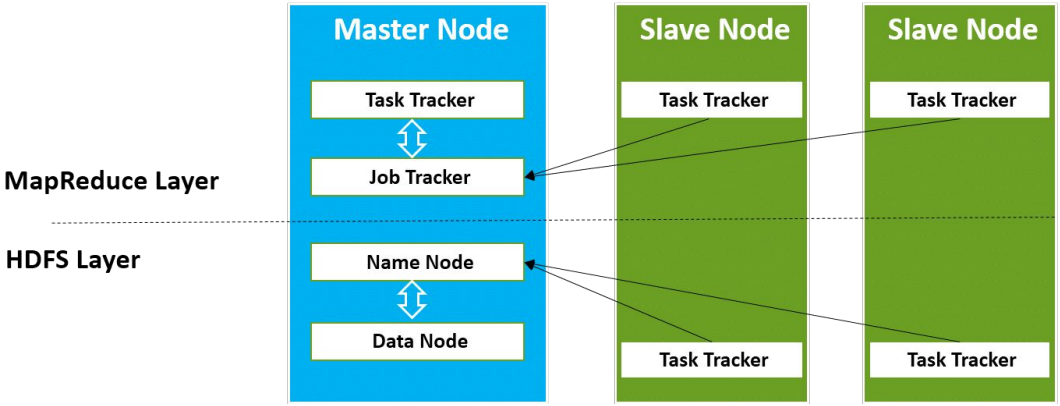
# Arquitectura Hadoop
## Sistema de archivos HDFS

### Acceso a datos

- Cuando se lee un archivo, el NameNode consulta la ubicación de los bloques y los envía al cliente.
- Los bloques se transfieren al cliente en paralelo desde varios nodos del clúster para aumentar la velocidad de transferencia.
- Los datos se pueden leer y escribir en HDFS utilizando diferentes APIs, como Java, Python y Hive.

### Manejo de fallos

- Si un nodo del clúster falla, HDFS replica automáticamente los bloques perdidos en otros nodos.
- Esto asegura que los datos siempre estén disponibles, incluso si algunos nodos fallan.




**Fuente:** Introduction to Hadoop

# MapReduce
## Procesamiento de datos distribuido

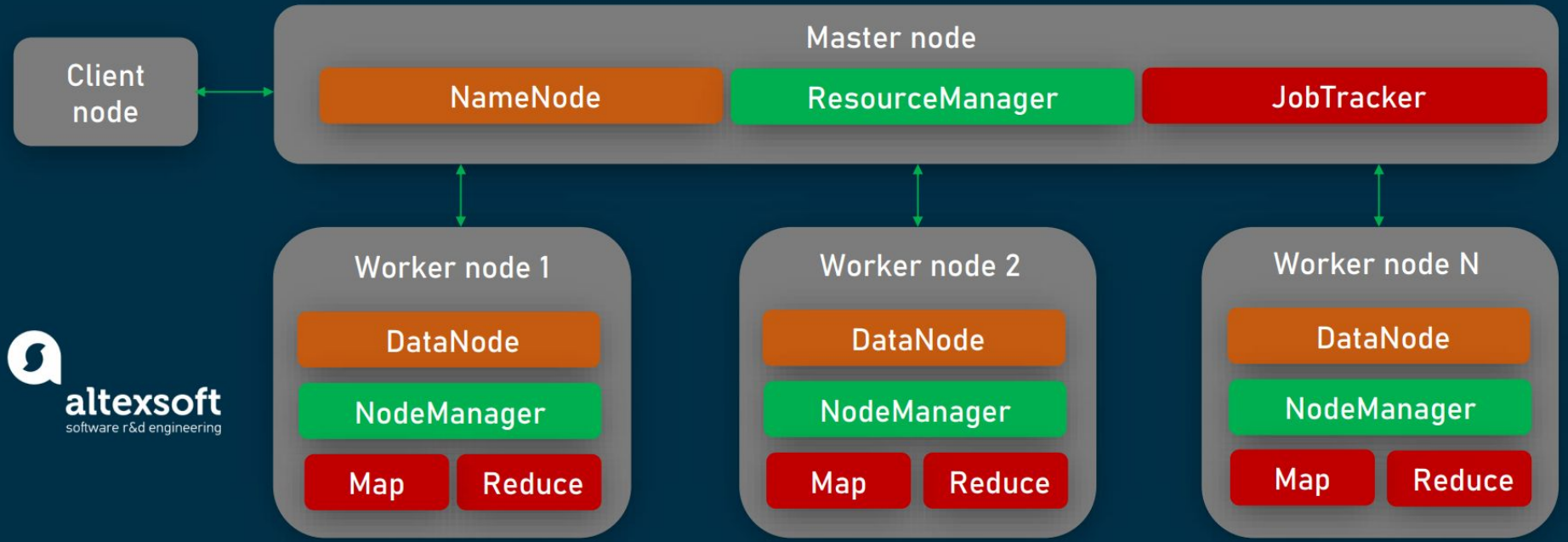
Es un marco de trabajo para procesar problemas **de forma paralela** a través de grandes conjuntos de datos utilizando un gran número de ordenadores (nodos), denominados colectivamente como un **clúster**. Está compuesto por 3 etapas:


- La fase Map toma como entrada un conjunto de datos y los divide en bloques más pequeños.
- Cada bloque se procesa en paralelo por una función Map.
- La función Map genera pares clave-valor.

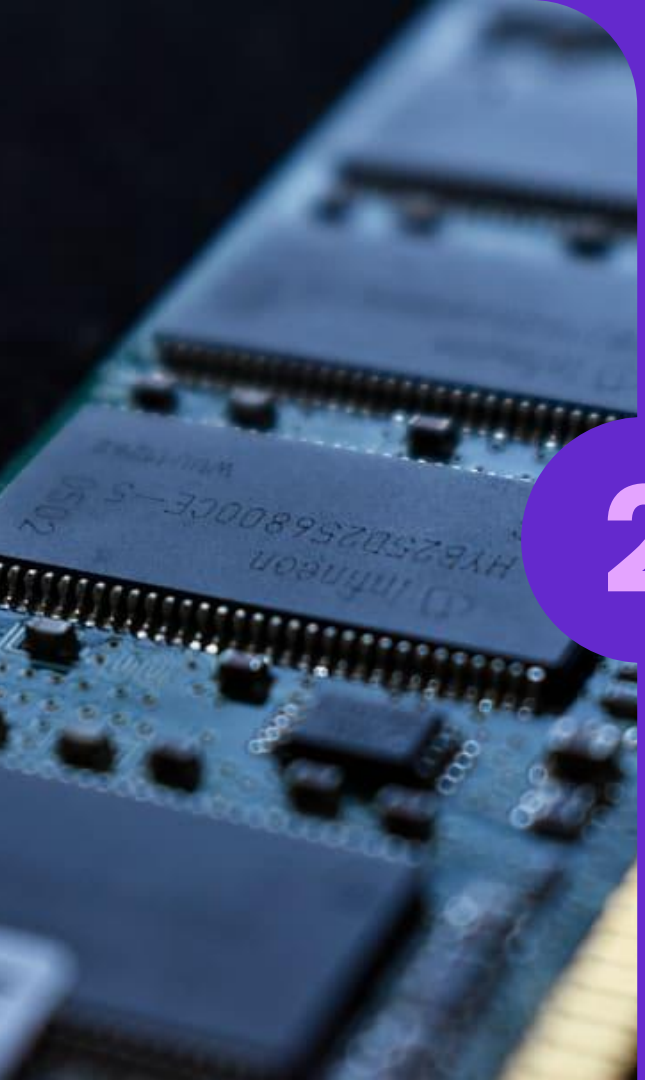

- La fase Shuffle reordena los pares clave-valor generados por la fase Map.
- Los pares clave-valor se agrupan por clave.


- La fase Reduce toma como entrada los pares clave-valor agrupados de la fase Shuffle.
- Una función Reduce se aplica a cada grupo de valores.
- La función Reduce genera un resultado final para cada clave.

# HADOOP CLUSTER ARCHITECTURE




Hadoop Architecture - https://www.altexsoft.com/blog/hadoop-vs-spark/



# 2 Frameworks para el procesamiento de **Big Data**

## Spark

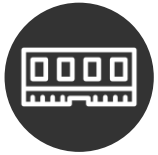
# ¿Qué es Apache Spark?


Es un motor unificado de analíticas para procesar datos a gran escala que integra módulos para SQL, streaming, aprendizaje automático y procesamiento de grafos. Spark se puede ejecutar de forma independiente o en Apache Hadoop, Apache Mesos, Kubernetes, la nube y distintas fuentes de datos.

# Características
# Apache Spark

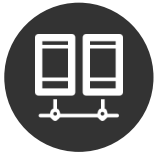
## Modelo de programación

- **RDD** (Resilient Distributed Datasets): Abstracción de datos que permite el procesamiento distribuido de grandes conjuntos de datos.
- **API de alto nivel**: Ofrece interfaces en Scala, Java, Python y R para facilitar el desarrollo de aplicaciones.



## Procesamiento en memoria

- Almacena datos en **memoria caché** para un acceso rápido y eficiente.
- Optimización de tareas para aprovechar la **memoria** disponible.



## Escalabilidad

- Ejecución en clústeres de **miles de nodos.**
- **Escalabilidad horizontal** para agregar o quitar nodos según sea necesario.

## Resiliencia

- Detección y **recuperación** automática de fallos.
- **Replicación** de datos para garantizar la disponibilidad.

# Características
# Apache Spark

## RDD
## (Resilient Distributed Datasets)

Los RDD son colecciones de elementos **tolerantes a fallas** que se pueden distribuir entre varios nodos en un clúster y trabajar en paralelo. Los RDD son una estructura **fundamental** en Apache Spark.

- **Inmutabilidad**
- **Particionamiento**
- **Resiliencia**
- **Lazy Evaluation**
- **Persistencia**
- **Procesamiento en memoria**

## DAG
## (Directed Acyclic Graph)

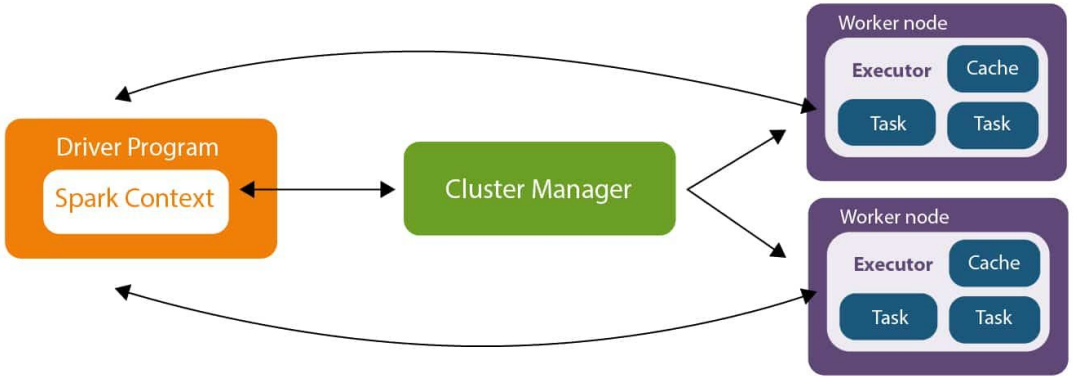
Spark utiliza un Gráfico Acíclico Dirigido (DAG) para **programar tareas** y orquestar los nodos de trabajador en el clúster. A diferencia de MapReduce, que tiene un proceso de ejecución de dos etapas, Spark puede **ejecutar tareas de forma más eficiente** gracias al DAG.

- **Eficiencia**
- **Tolerancia a fallos**
- **Visibilidad**
- **Escalabilidad**
- **Flexibilidad**

# Arquitectura
## Apache Spark

Apache Spark tiene una arquitectura jerárquica **maestro/esclavo**. Spark Driver es el nodo maestro que controla el administrador del clúster, los nodos de trabajador (esclavos) y entrega los resultados de los datos al cliente de la aplicación.

Basado en el código de la aplicación, Spark Driver genera el **SparkContext**, que trabaja con el administrador de clústeres independiente de Spark u otros administradores de clústeres como Hadoop YARN, Kubernetes o Mesos, para distribuir y supervisar la ejecución en los nodos.




2

# Taller práctico

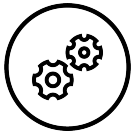
## AWS Glue + PySpark

# Taller aplicado
## Objetivos

**Comprender las principales herramientas** disponibles en AWS Glue para la extracción, catalogado y transformación de datos.

Aprender algunas de las **principales operaciones** que se pueden realizar en AWS Glue usando PySpark en un entorno de desarrollo con Jupyter Notebooks.

**Realizar un pipeline de datos en AWS Glue** que incluya los procesos de extracción, transformación y almacenamiento de datos para explotación usando las herramientas de analítica de AWS.



Comprender el **concepto de DynamicFrame** en Glue y su relación con los DataFrames de pandas y PySpark.

Identificar los principales **tipos de datos en Spark,** cómo se consulta el esquema de los datos y de qué manera se realizan transformaciones entre tipos.

Aprender los conceptos básicos sobre el **catálogo de Glue** y la consulta de información a partir de archivos parquet, usando **Amazon Athena.**

# Taller aplicado
## Recursos

- Configuring AWS Glue interactive sessions for Jupyter and AWS Glue Studio notebooks
- aws-glue-samples/examples/join_and_relationalize.md at master
- Tutorial: Adding an AWS Glue crawler
- AWS Glue type systems
- PySpark-Reference-Notebook/PySpark Tutorial.ipynb

# Mejoramos
# la vida de la gente
# transformando empresas

pragma

www.pragma.co

Carrera 42 # 5 Sur 47
Edificio SELF - Piso 16
Medellín, Colombia
t. (323) 563 9223

Keep moving