DIPLOMADO

Ciencia de Datos

Módulo 12

Datos masivos

Mtro. Omar Mendoza González

DGTIC

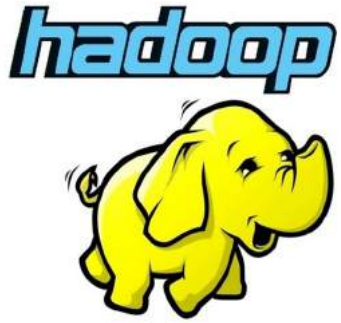
Universidad Nacional Autónoma de México

Dirección General de Cómputo y de Tecnologías de Información y Comunicación

Contenido

2. Procesamiento paralelo
    2.1 Apache Hadoop
        2.1.2 HDFS
        2.2.2 MapReduce

Apache Hadoop

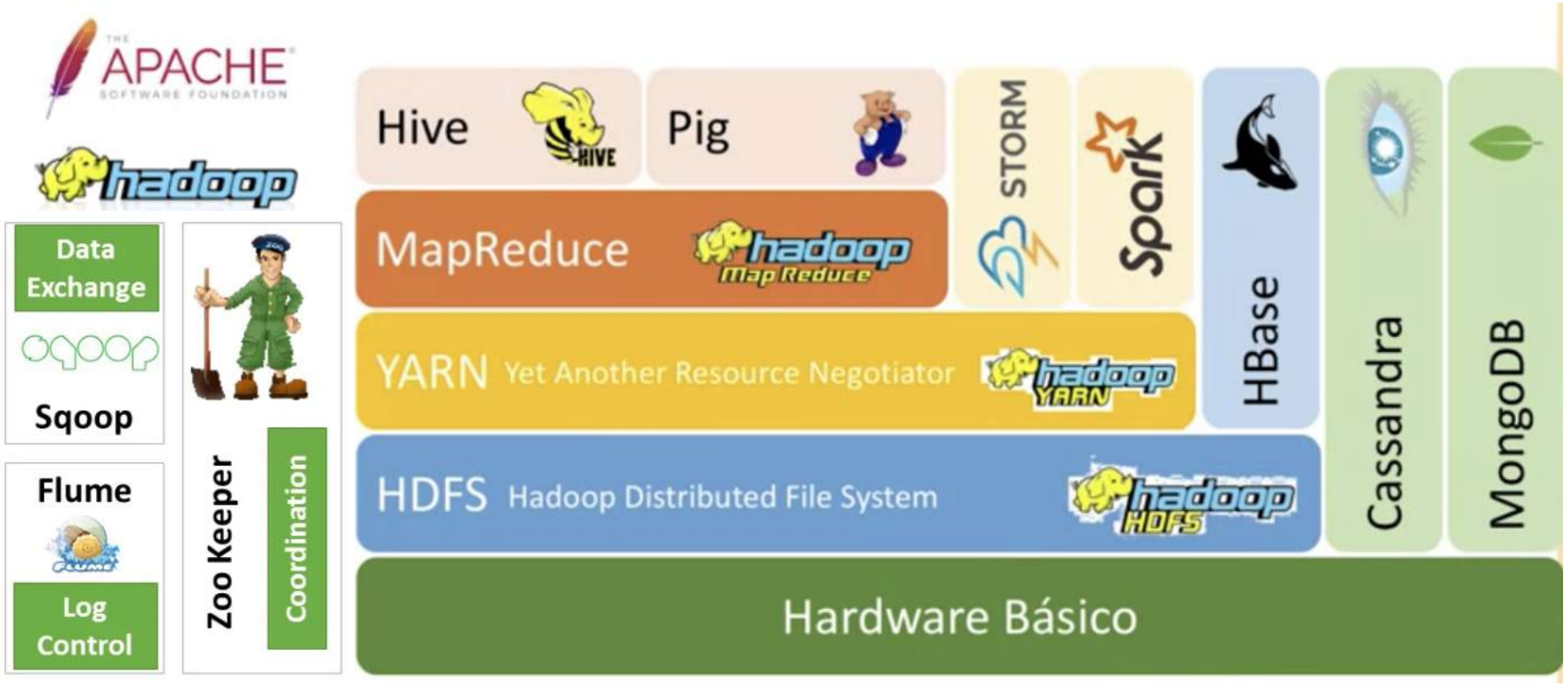


- Framework de software abierto
- Procesamiento distribuido de grandes conjuntos de datos en clusters de servidores básicos.
- Puede extender un sistema de servidor único a miles de máquinas, con un muy alto grado de tolerancia a las fallas.
- Capacidad para detectar y manejar fallas al nivel de las aplicaciones.

Apache Hadoop

- Redimensionable
- Rentable
- Flexible
- Tolerante a fallas
- Entorno compartido
- Procesamiento distribuido
- Almacenamiento distribuido

El ecosistema Hadoop



Componentes principales Hadoop

Map-Reduce

- capa de procesamiento de datos de Hadoop.

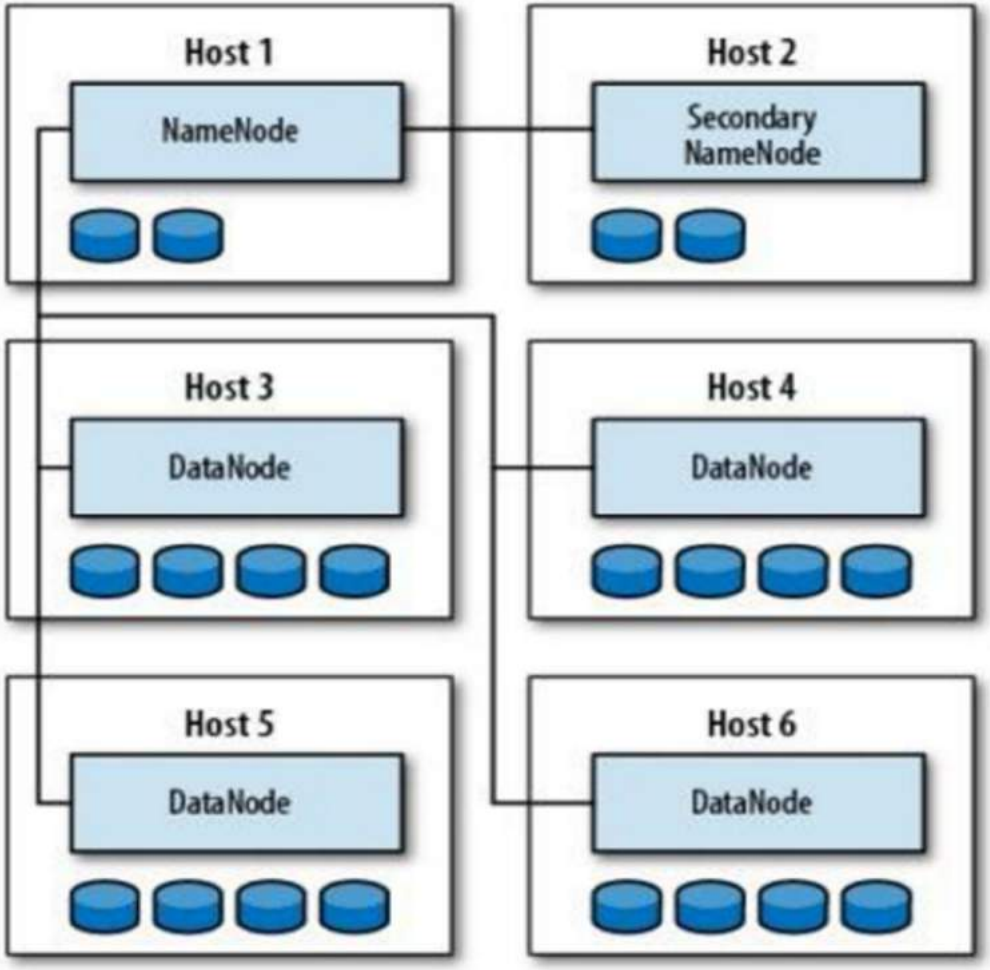
YARN

- capa de administración de recursos de Hadoop.

Hadoop Distributed File System (HDFS)

- capa de almacenamiento de Hadoop.

Arquitectura Hadoop



- **Bloques**

Bloques de gran tamaño replicados

- **NameNodes**

Metadatos

- **DataNodes**

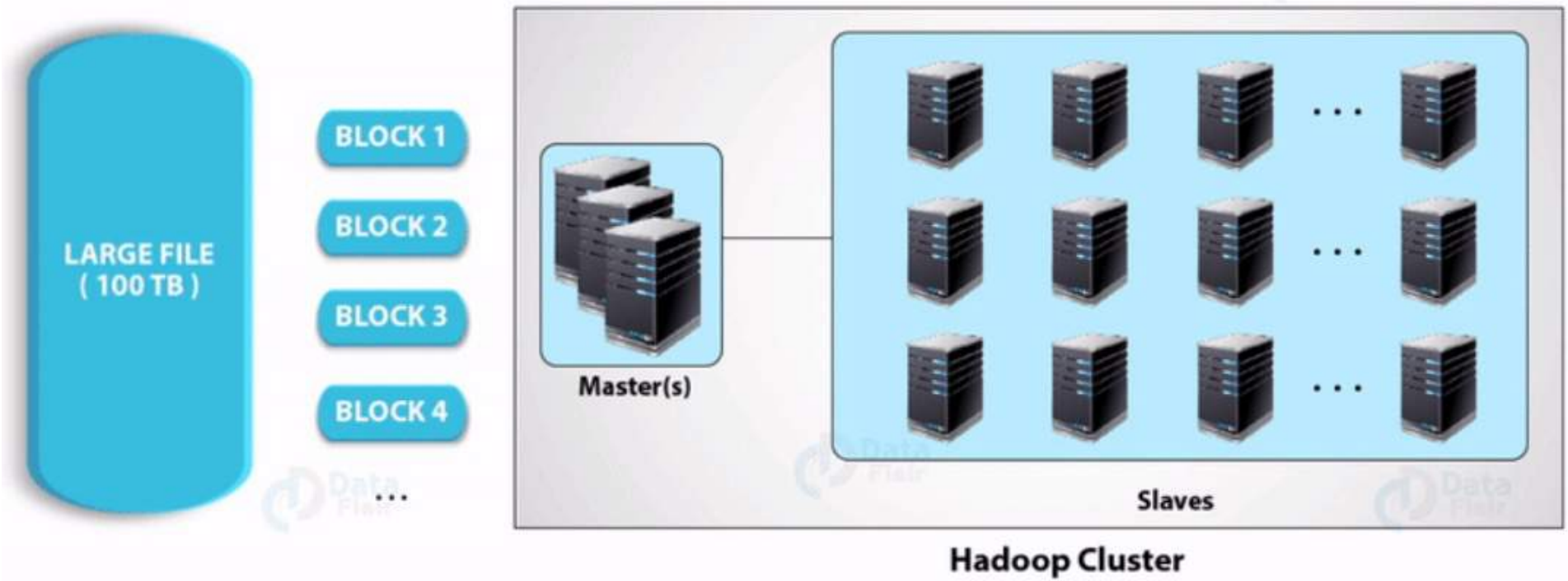
Datos

Arquitectura Hadoop

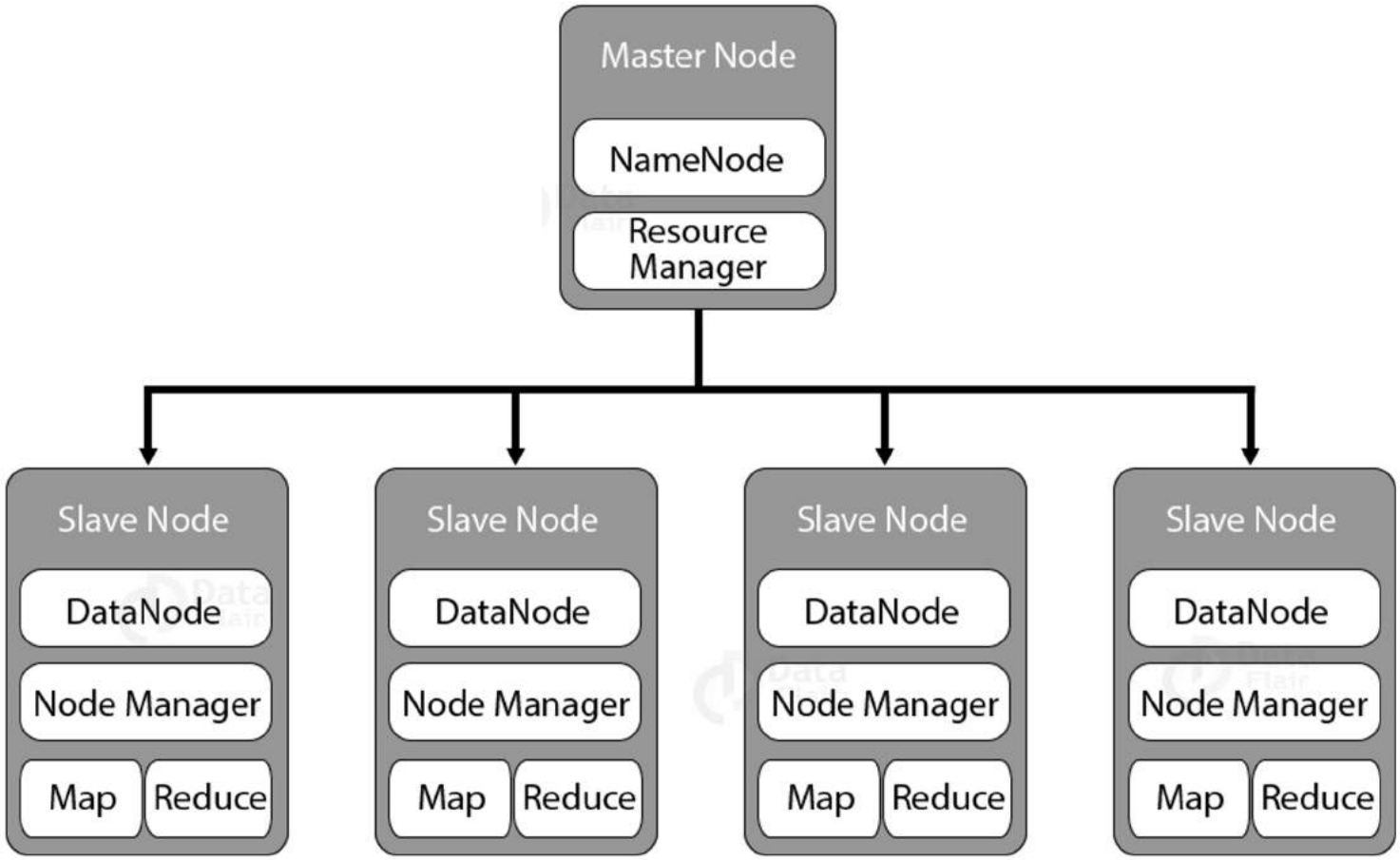
- **NameNode**
  - NameNode Daemon se ejecuta en la máquina maestra.
  - Es responsable de mantener, monitorear y administrar los DataNodes.
  - Registra los metadatos de los archivos

- **DataNode**
  - DataNode se ejecuta en la(s) máquina(s) esclava(s)
  - Almacena los datos
  - Sirve la solicitud de lectura y escritura del usuario

HDFS


LARGE FILE
(100 TB)
BLOCK 1
BLOCK 2
BLOCK 3
BLOCK 4
...
Master(s)
Slaves
Hadoop Cluster

Arquitectura Hadoop

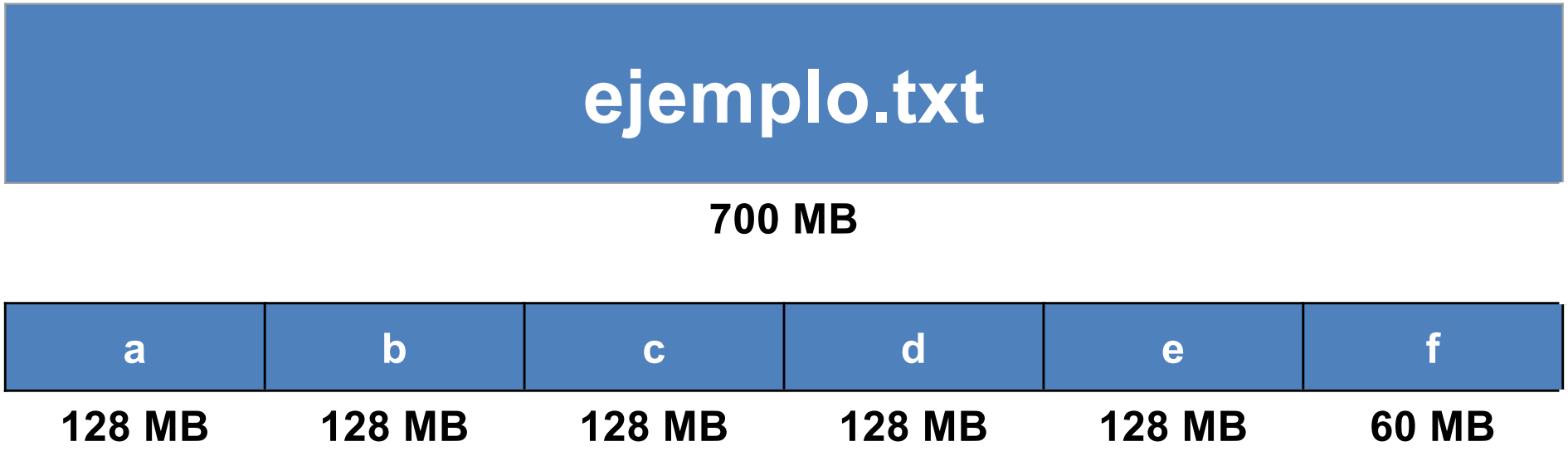


HDFS

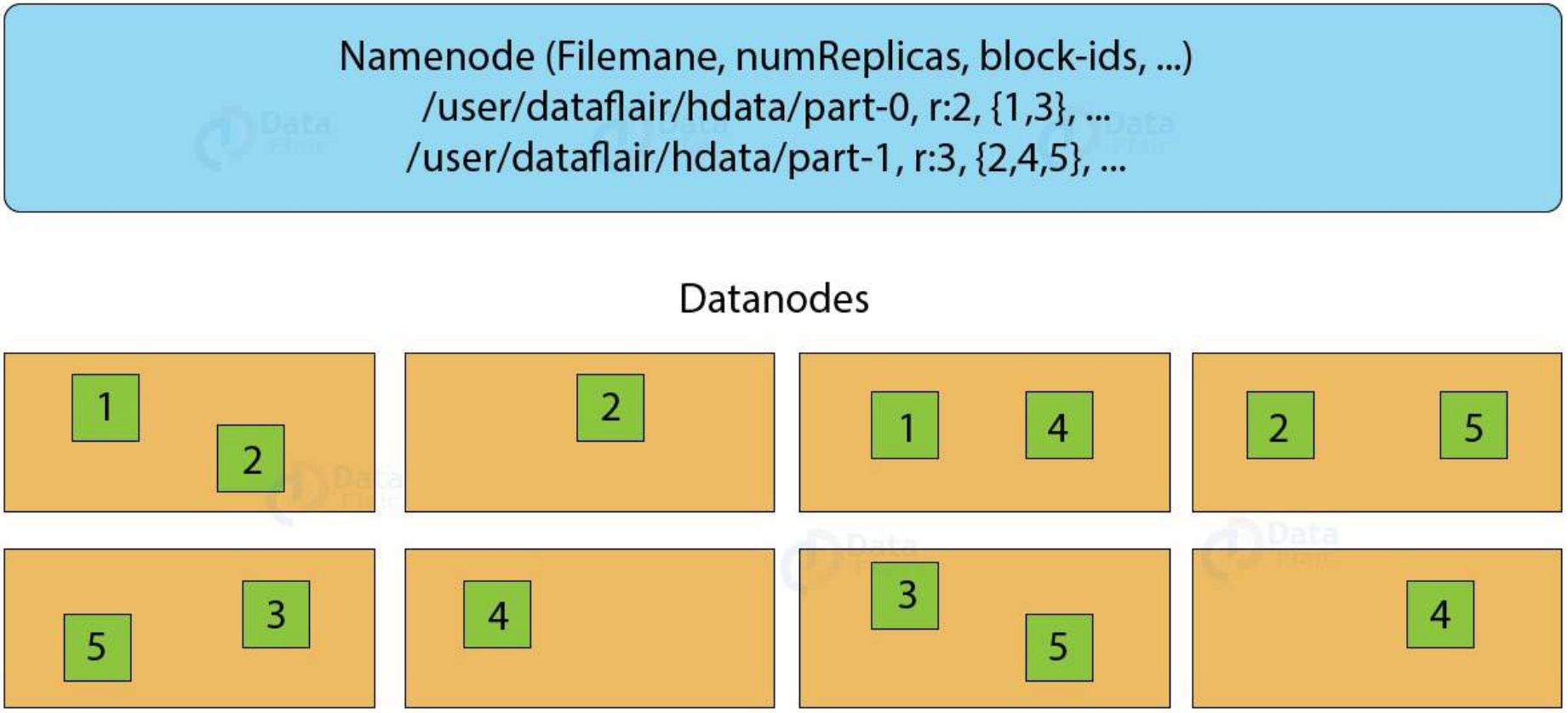
- Sistema de archivos distribuido, escalable y portátil para trabajar con archivos de gran tamaño
  - Tamaño de bloque de 128MB o 256MB

- Procesa en forma paralela un archivo, dividiéndolo en bloques (blocks), y ejecutándolo en varios equipos (nodos).

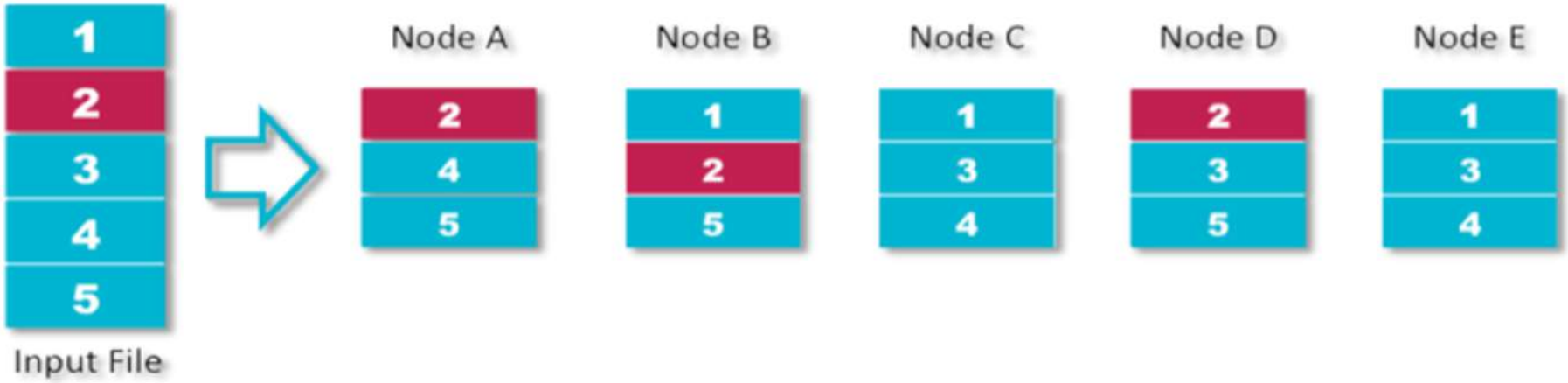
HDFS


ejemplo.txt
700 MB
a
b
c
d
e
f
128 MB
128 MB
128 MB
128 MB
128 MB
60 MB

HDFS replicación de bloques



Arquitectura HDFS



HDFS

- Diseñado para trabajar en sistemas de cómputo de bajo costo
- Adecuado para aplicaciones que manejan grandes volúmenes de datos
- Una instancia HDFS puede estar constituida por cientos de nodos, cada uno almacenando parte de los datos
- Capaz de gestionar millones de archivos en una sola instancia

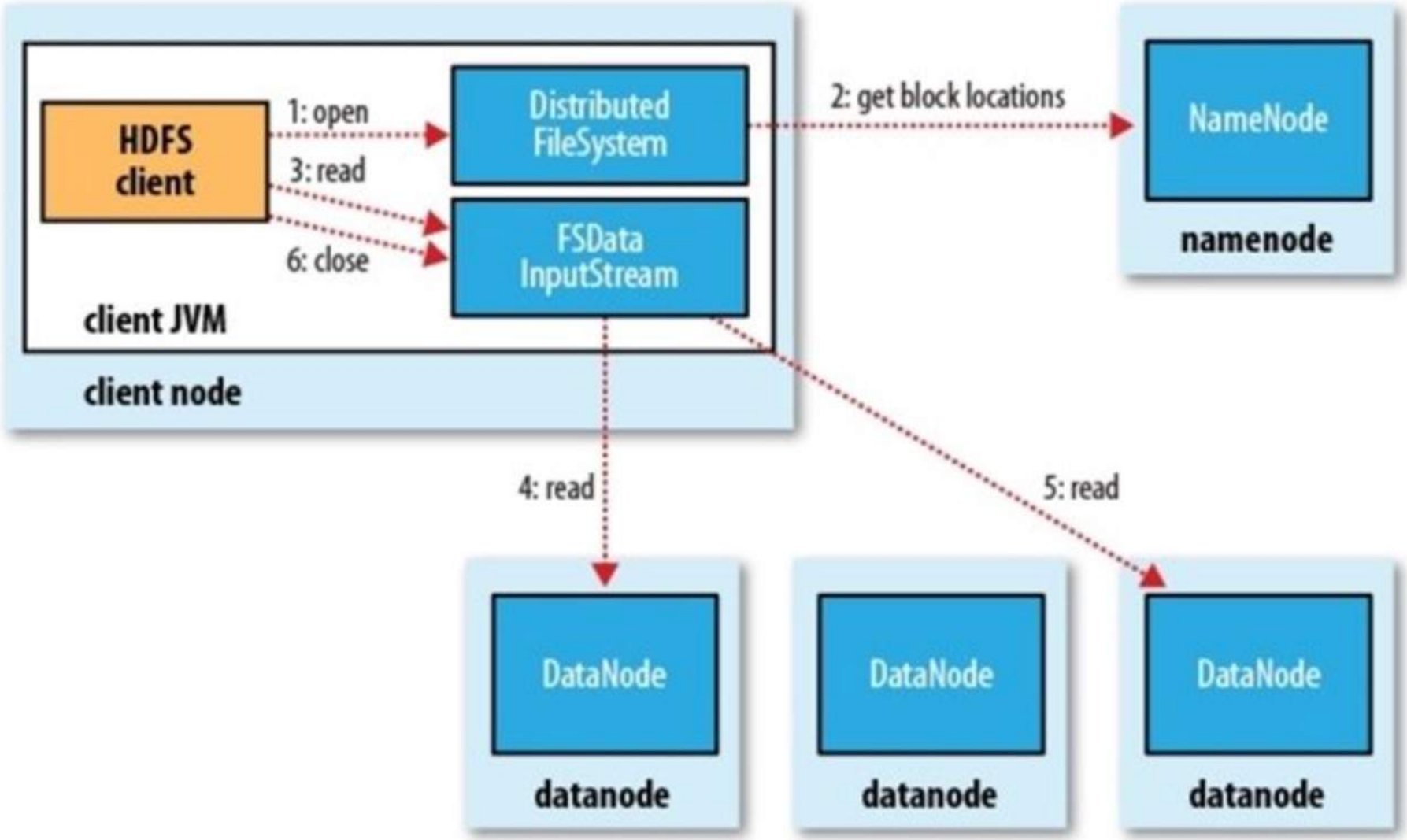
HDFS

- Modelo de coherencia simple
  - Write-once
  - Read-many

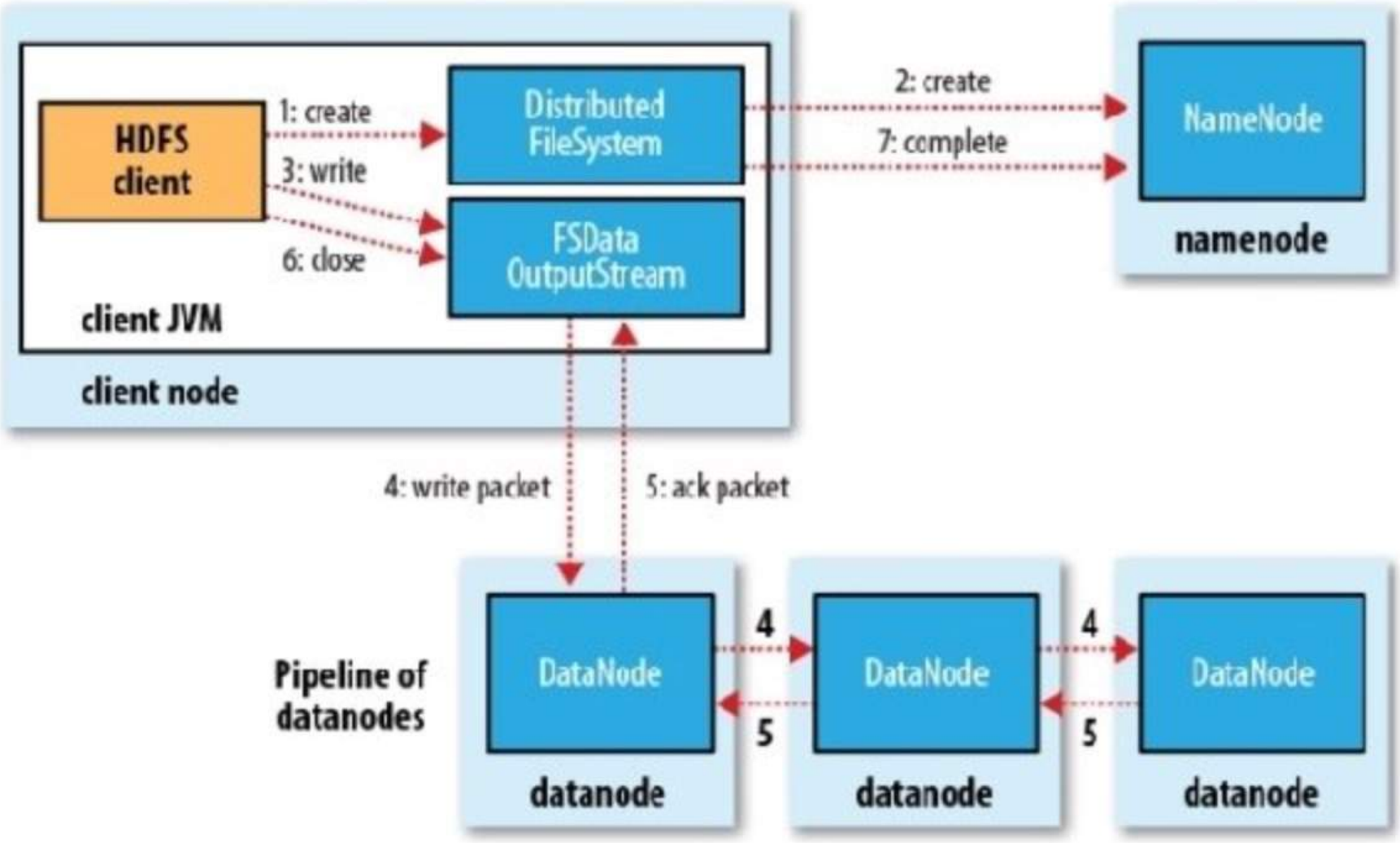
- Portabilidad entre plataformas

- Tres interfaces:
  - 1. Interfaz en línea de comandos
  - 2. Interfaz Web
  - 3. API de programación

Lectura en HDFS



Escritura en HDFS



Accseo a HDFS

```
hdfs dfs -ls /user/hadoop/file1
hdfs dfs -mkdir /user/hadoop/dir1 /user/hadoop/dir2
hdfs dfs -rm hdfs://nn.example.com/file /user/hadoop/emptydir
```

MapReduce

- Problema típico en Big Data
  - Operar sobre un gran número de “registros”
  - Extraer información relevante de cada uno de ellos
  - Combinar y ordenar resultados intermedios
  - Agregar resultados intermedios
  - Generar los resultados de salidas finales

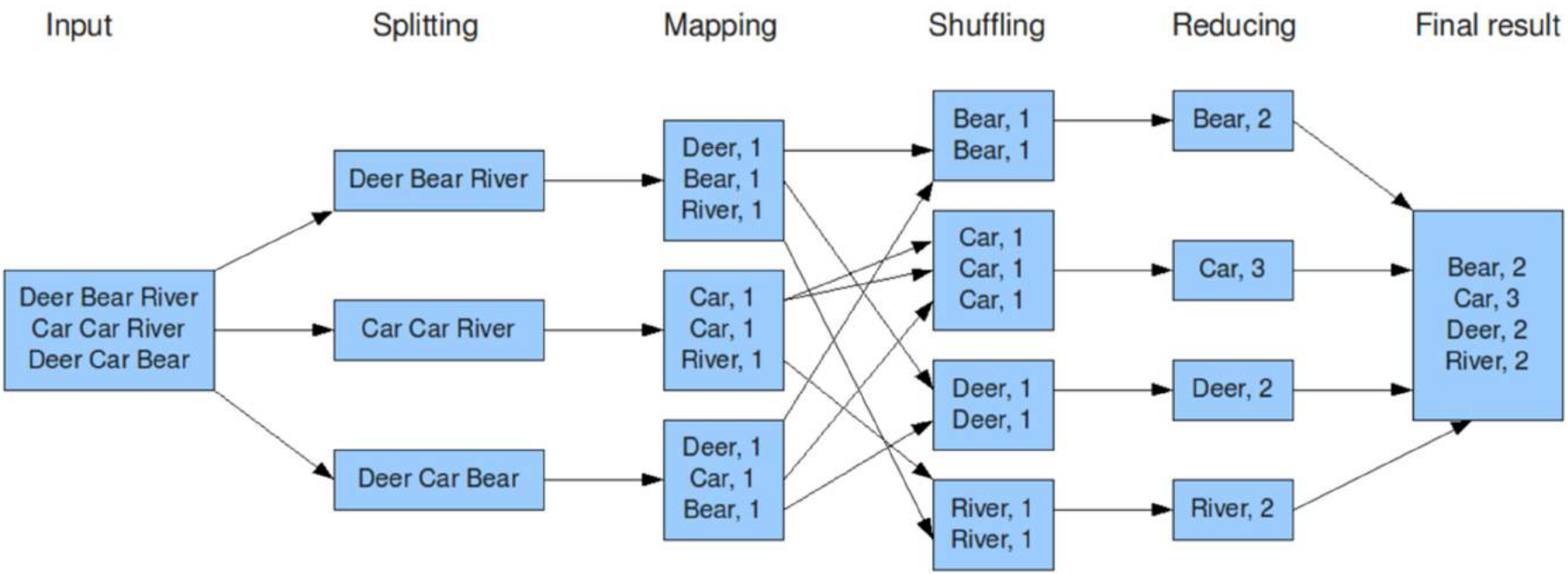
MapReduce

- Modelo de programación para procesamiento distribuido y generación de grandes sets de datos
- Permite explotar el paralelismo para el análisis y procesamiento de datos
- MapReduce permite el procesado a gran escala de conjuntos de datos.

MapReduce

- **Pasos del modelo**
  - **Map**: La función Map convierte el archivo de entrada en una secuencia de pares <clave, valor>
  - **Shuffle**: Los resultados son recolectados y ordenados según el valor de la clave
  - **Reduce**: combinando todos los valores asociados a la clave de forma específica a cada problema

El modelo MapReduce



El modelo MapReduce

```
Esto es una linea
Esto también
```

Map

```
map("Esto es una linea") =
    esto, 1
    es, 1
    una, 1
    linea, 1
map("Esto también") =
    esto, 1
    también, 1
```

Reduce

```
reduce(es, {1}) =
    es, 1
reduce(esto, {1, 1}) =
    esto, 2
reduce(linea, {1}) =
    linea, 1
reduce(también, {1}) =
    también, 1
reduce(una, {1}) =
    una, 1
```

Resultado:

```
es, 1
esto, 2
linea, 1
también, 1
una, 1
```

Contacto

Omar Mendoza González

Profesor de carrera ICO FES Aragón

omarmendoza564@aragon.unam.mx

Referencias

- **Corea, Francesco**, **An Introduction to data : everything you need to know about AI, Big data and data science / Francesco Corea -- Cham, Switzerland :** Springer, [2019].-- xv, 131 páginas : ilustraciones (Studies in Big data, 2197-6503 ; 50)

- **Casas Roma, Jordi**, **Big data : análisis de datos en entornos masivos / Jordi Casas Roma, Jordi Nin Guerrero, Francesc Julbe López -- Barcelona :** Editorial UOC, 2019 287 páginas : ilustraciones (Tecnología ; 623).

- **Caballero, Rafael, Big data con Python recolección, almacenamiento y proceso /** Rafael Caballero Adrián Riesco Enrique Martín: Universidad Complutense de Madrid Editorial AlfaOmega, 2019 282 páginas

- **Rioux, Jonathan, Data Analysis with Python and PySpark** / Jonathan Rioux: Editorial Manning Publications, 2020 259 páginas

- **Singh, Pramod, Machine Learning with PySpark: With Natural Language Processing and Recommender Systems** / Pramod Singh: Editorial Apress, 2019 233 páginas