



Hadoop I – Big Data

ANDRES TRIANA

Microsoft Certified Trainer (MCT)



CONTACTO

andres.triana@qualitycode.com.co

319 506 0662

www.qualitycode.com.co



Temario del Curso

Modulo 1

Fundamentos

Modulo 2

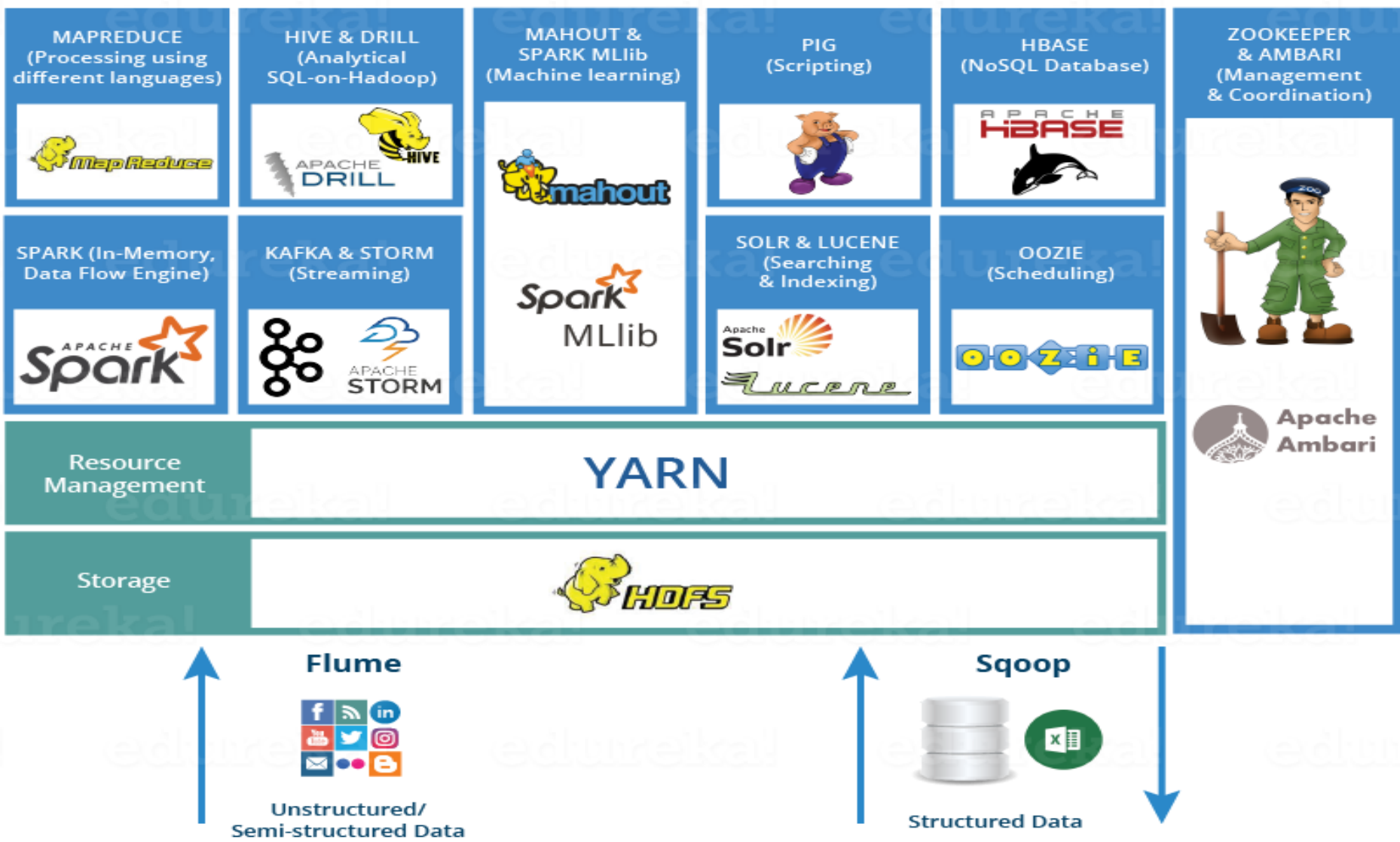
Hadoop

Modulo 3

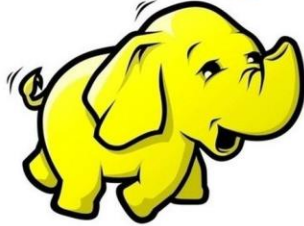
Data mining y Machine Learning

Modulo 4

NoSql - MongoDB



hadoop



- ¿Qué sucede cuando las técnicas de análisis tradicionales se encuentran con sus límites?
- ¿Cuándo llega el momento en que la minería de datos no aporta las soluciones esperadas?
- ¿Cómo se enfrentan los clusters de Hadoop al desafío de los grandes datos y su expresión más desestructurada?
- ¿Es Hadoop una buena opción para mi negocio?

Big Data y Data Science

Big Data

Hace referencia a los sistemas que manipulan grandes conjuntos de datos, también conocidos como data sets.

- Heterogeneidad
- Volatilidad

Data Science

Este concepto es algo más genérico y hace referencia a las técnicas necesarias para manipular y tratar la información desde un punto de vista estadístico/matemático

La solución que propone Hadoop

La mayoría de las empresas estiman que sólo analizan el **12%** de los datos que tienen, dejando **88%** de ellos en la sala de espera

Entre sus puntos clave se encuentran su capacidad de almacenamiento y procesamiento local.

Características





- Consigue escalar desde unos pocos servidores hasta miles de máquinas, todas ellas ofreciendo idéntica calidad de servicio.
- Permite el procesamiento distribuido de grandes conjuntos de datos en clusters de computadoras utilizando modelos sencillos de programación.

Complemento perfecto de BigData

- **Simplifica la interacción** con su aportación informativa.
- **Economiza los procesos.**
- **Reduce las carencias** que big data puede presentar de cara al usuario.

	TRADITIONAL RDBMS	MAPREDUCE
Data Size	Gigabytes (<i>Terabytes</i>)	Petabytes (<i>Hexabytes</i>)
Access	Interactive and Batch	Batch
Updates	Read / Write many times	Write once, Read many times
Structure	Static Schema	Dynamic Schema
Integrity	High (ACID)	Low
Scaling	Nonlinear	Linear
DBA Ratio	1:40	1:3000



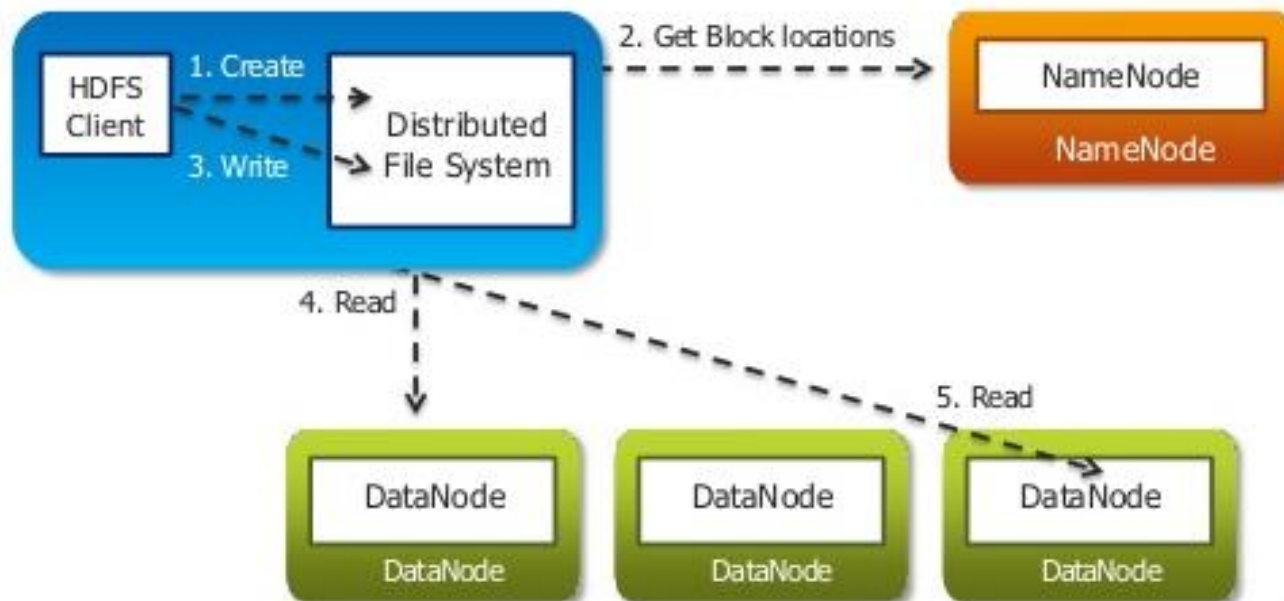
Pero, ¿cómo lo hace? ¿Cómo consigue dejar atrás los límites y superar las dificultades?

- **HDFS (Hadoop Distributed File System):** es un sistema de archivos distribuido, escalable y portátil típicamente escrito en JAVA.
- **MapReduce:** es el modelo de programación utilizado por Google para dar soporte a la computación paralela. Trabaja sobre grandes colecciones de datos en grupos de computadoras y sobre commodity hardware.

Arquitectura básica de HDFS

Este sistema de archivos sobre el que se estructura Hadoop cuenta con tres pilares básicos:

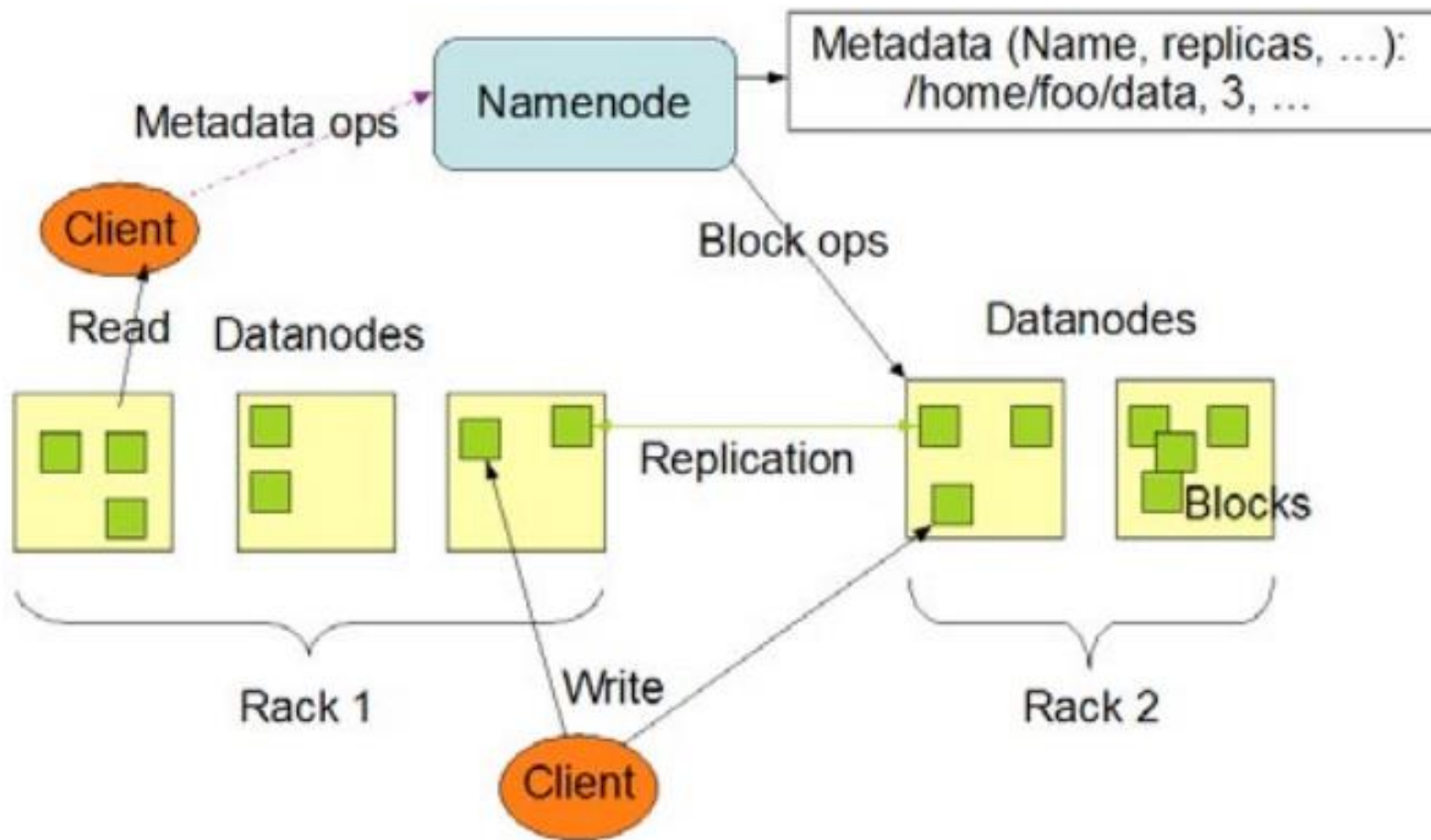
- Namenode: se ocupa del control de acceso y tiene la información sobre la distribución de datos en el resto de nodos.
- Datanodes: son los encargados de ejecutar el cómputo, es decir, las funciones Map y Reduce, sobre los datos almacenados de manera local en cada uno de dichos nodos.
- Jobtracker: este nodo se encarga de las tareas y ejerce el control sobre la ejecución del proceso de MapReduce.



Principales características de HDFS

- *Tolerancia a fallos*: que consigue que no se pierda la información ni se generen retrasos. De hecho, incluso aunque se produzca la caída de algunos datanodes, el cluster sigue funcionando.
- *Acceso a datos en streaming*: los datos son facilitados a medida que se consumen, por lo que no hace falta descargarlos.
- *Facilidad para el trabajo con grandes volúmenes de datos*: los clusters de Hadoop están preparados para almacenar grandes ficheros de todo tipo.
- *Modelo sencillo de coherencia*: ya que no implementa la regla POSIX en un 100% para poder aumentar los ratios de transferencia de datos.
- *Portabilidad de convivencia entre hardware heterogéneo*: Hadoop puede correr en máquinas de distintos fabricantes.

HDFS Architecture



Fases de MapReduce

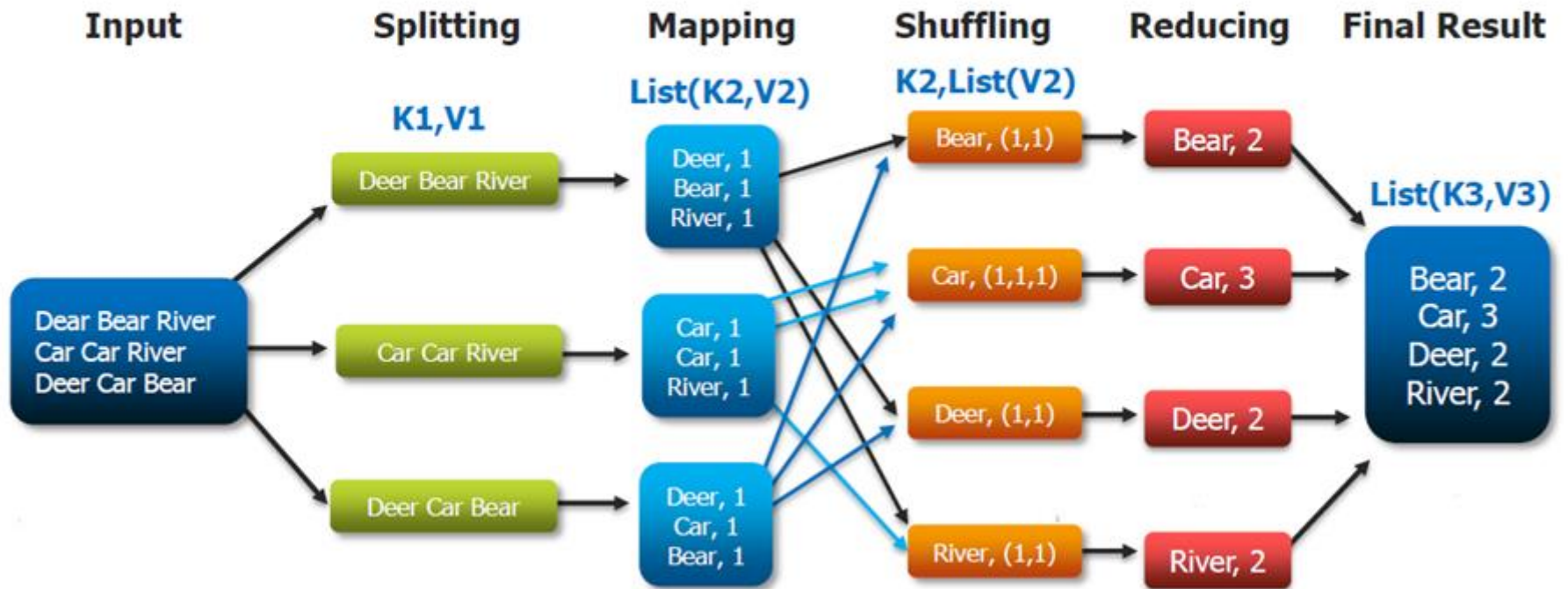
- Su ejecución consta de dos fases principales, **Map** y **Reduce**, ambas programadas por el desarrollador. Se completan con una etapa interna, llamada "**Shuffle and sort**", que permite vincular las dos fases anteriores.

Map: se aplica en paralelo para cada ítem en la entrada de datos.

Shuffle and sort: tiene dos misiones. Por una parte, se encarga de ordenar por clave (key) todos los resultados emitidos por el mapper; y, por otra, de recoger todos los valores intermedios pertenecientes a una clave para combinarlos en una lista asociada a ella.

Reduce: esta función se aplica en paralelo para cada grupo asociado a una clave. El resultado es la obtención de una lista de valores.

The Overall MapReduce Word Count Process



Principales características de MapReduce

- Distribución y paralelización (automáticas).
- Tolerancia a fallos y a redundancias.
- Transparencia: su funcionamiento interno y su mantenimiento son transparentes para los desarrolladores. Es decir, que sólo tienen que programar la lógica de negocio del algoritmo, en vez de necesitar invertir tiempo gestionando errores o parámetros de la computación distribuida.
- Escalabilidad horizontal: permite que, si se necesita más potencia de computación, baste con añadir más nodos en el clúster.
- Localización de los datos: se desplaza el algoritmo a los datos y no al contrario, como suele suceder en sistemas distribuidos tradicionales.
- Dispone de herramientas de monitorización.

Fases de Big Data y sus soluciones dentro del ecosistema Hadoop

1. Descubrimiento de grandes datos

El procedimiento a seguir se resume en cuatro pasos:

- **Definir cuáles son los datos de interés.**
- **Encontrar sus fuentes (históricos o Social Media, entre otros)**
- **Grabar los datos en el sistema.**
- **Determinar cómo serán procesados.**

Dentro de Hadoop, pueden emplearse:

- **Flume y Chukwa** framework para datos no estructurados: se ocupan de los ficheros de logs.
- **Sqoop**: si los datos provienen de una base de datos relacional.

Fases de Big Data y sus soluciones dentro del ecosistema Hadoop

2. Extracción y limpieza de los grandes volúmenes de datos

Para llevar a cabo la extracción y pre-procesamiento de los datos es necesario:

- **Extraer los datos de la fuente de origen datos.**
- **Perfilar y limpiar los datos.**
- **Adecuarlos a las necesidades de la empresa, de acuerdo a las reglas de negocio.**
- **Aplicar los estándares de **calidad de datos**.**

Las dos aplicaciones mencionadas en la fase anterior también son de utilidad en esta etapa, ya que suelen contar con opciones de filtrado previo que proporcionan, a su vez, la estructura conveniente para servir de entrada a Hadoop.

Fases de Big Data y sus soluciones dentro del ecosistema Hadoop

2. Extracción y limpieza de los grandes volúmenes de datos

Para llevar a cabo la extracción y pre-procesamiento de los datos es necesario:

- **Extraer los datos de la fuente de origen datos.**
- **Perfilar y limpiar los datos.**
- **Adecuarlos a las necesidades de la empresa, de acuerdo a las reglas de negocio.**
- **Aplicar los estándares de **calidad de datos**.**

Las dos aplicaciones mencionadas en la fase anterior también son de utilidad en esta etapa, ya que suelen contar con opciones de filtrado previo que proporcionan, a su vez, la estructura conveniente para servir de entrada a Hadoop.

Fases de Big Data y sus soluciones dentro del ecosistema Hadoop

3. Estructuración y análisis de big data

La integración es crucial y aquí es donde intervienen las tres siguientes acciones:

- Dotar de **estructura lógica** a los conjuntos de datos tratados.
- Almacenar los datos en el repositorio elegido (puede ser una base de datos o u sistema)
- Analizar los datos disponibles para hallar relaciones.

En el ecosistema de Hadoop pueden encontrarse dos alternativas para solucionar los problemas de estructuración:

- **HDFS**: antes de que Hadoop pueda proceder al tratamiento de la información, los datos pre-procesados han de almacenarse en un sistema distribuido de ficheros. Este es el rol que cumple HDFS como componente core dentro de Hadoop.
- **Avro**: se trata de un sistema que hace posible serializar datos para codificar los que va a manejar Hadoop. Permite también definir interfaces a la hora de “parsear” información.

Fases de Big Data y sus soluciones dentro del ecosistema Hadoop

4. Modelado de datos

Esta etapa se orienta al procesamiento de datos apoyándose en el modelado y para ello requiere de:

- **Aplicar algoritmos a los datos.**
- **Aplicar procesos estadísticos.**
- **Resolver las peticiones lanzadas mediante el **modelado de datos** en base a técnicas de minería.**

Hay muchas maneras de llevar a cabo estos cometidos. Las más eficientes son:

- **Bases Datos NoSQL:** no tienen esquema de datos fijo, por lo que no es necesario preocuparse de comprobar el esquema cada vez que se realiza la inscripción de un registro en la base de datos. Ello supone una ventaja considerable cuando se trabaja con cifras que alcanzan los millones de registros. Algunas de las más conocidas son MongoDB o Impala, aunque existen muchas más opciones.
- **Frameworks de consultas:**
 - * **HIVE:** es un framework que permite crear tablas, insertar datos y realizar consultas con un lenguaje similar al que podría llevarse a cabo utilizando queries SQL (el lenguaje no es SQL, sino HQL).
 - * **PIG:** permite manejar datos mediante un lenguaje textual conocido como Pig Latin.

Fases de Big Data y sus soluciones dentro del ecosistema Hadoop

5. Interpretación de grandes datos

El fin de todo trabajo con big data pasa por:

- **Interpretar las distintas soluciones.**
- **Aportar un resultado final.**

Las mejores opciones para llevar a cabo esta fase de big data son:

- **Mahout y R:** librería de minería de datos que permite realizar clustering, algoritmos de regresión e implementación de modelos estadísticos sobre los datos de salida ya procesados.





Servicios de Análisis

- En esta sección se enumeran las funcionalidades de Azure HDInsight.

¿Qué es Azure HDInsight?

- Azure **HDInsight** es un servicio de análisis, de código abierto, espectro completo y totalmente administrado en la nube para empresas. Puede usar plataformas de código abierto como Hadoop, Apache Spark, Apache Hive, LLAP, Apache Kafka, Apache Storm, R, etc.

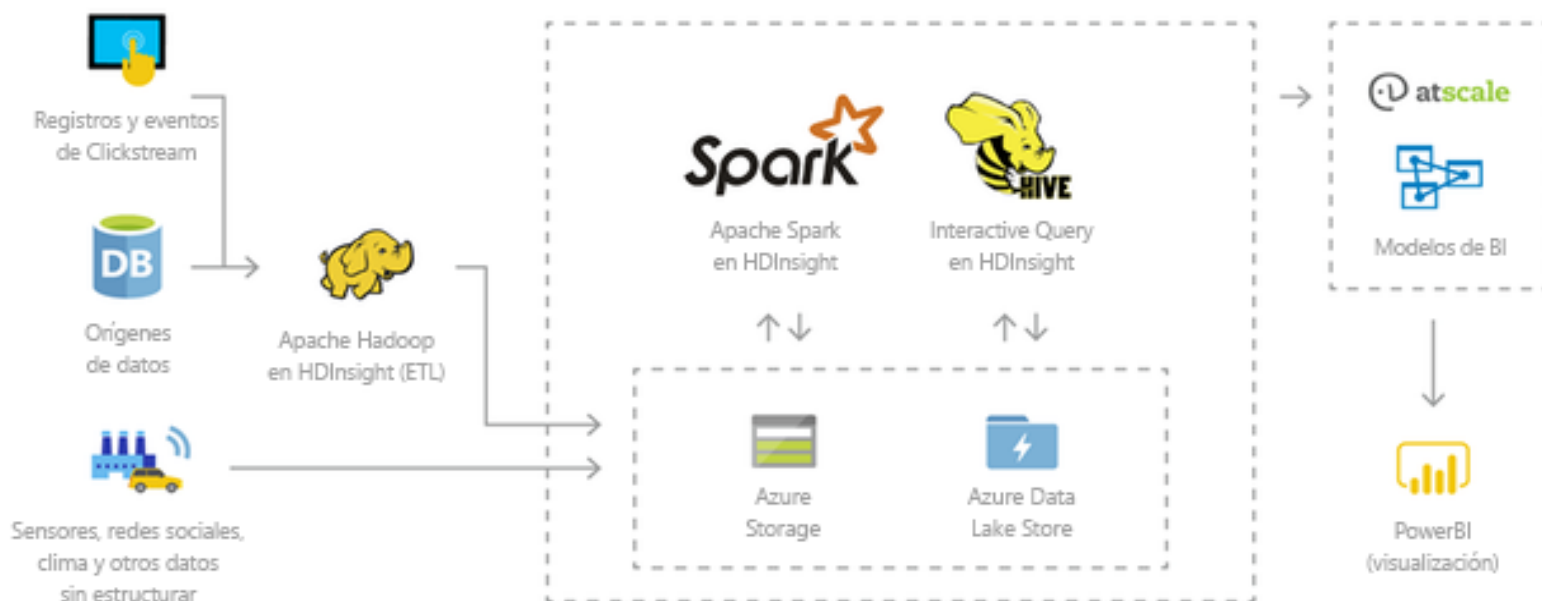
Procesamiento por lotes (ETL)

El de extracción, transformación y carga (ETL) es un proceso en el que se extraen datos estructurados o no estructurados de orígenes de datos heterogéneos. Estos datos se transforman a un formato estructurado y se cargan en un almacén de datos. Los datos transformados se pueden usar para ciencia de datos o almacenamiento de datos..

Escenarios de uso de HDInsight

Almacenamiento de datos

Puede usar HDInsight para realizar consultas interactivas a escalas de **petabytes** sobre datos estructurados o no estructurados en cualquier formato. También puede generar modelos conectándolos a herramientas de BI.



Escenarios de uso de HDInsight

Internet de las cosas (IoT)

Puede usar HDInsight para procesar los datos de streaming recibidos en tiempo real desde varios tipos de dispositivos.



Escenarios de uso de HDInsight

Ciencia de Datos

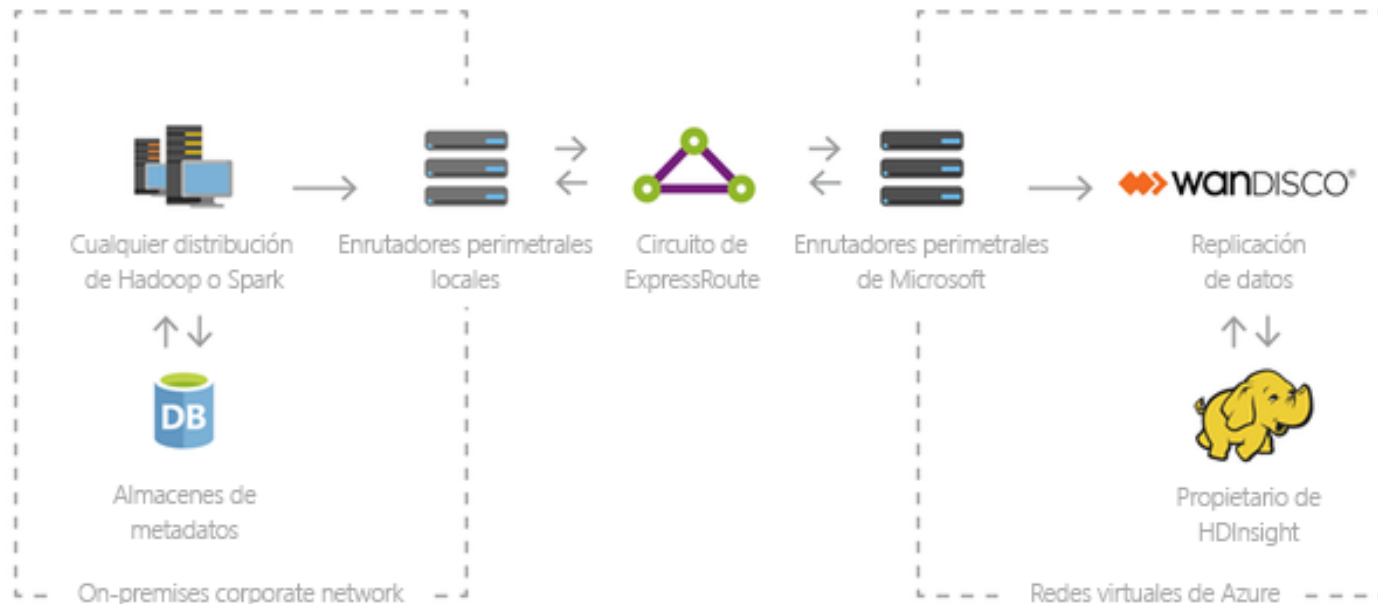
Puede usar HDInsight para compilar aplicaciones que extraigan información crítica de los datos. También puede usar Azure Machine Learning para predecir tendencias futuras de la empresa..



Escenarios de uso de HDInsight

Hibrido

Puede usar HDInsight para ampliar la infraestructura local de macrodatos existente en Azure para aprovechar las avanzadas funcionalidades de análisis en la nube.



Tipos de clúster de HDInsight

Tipo de clúster	Descripción
Apache Hadoop	Una plataforma que utiliza HDFS, administración de recursos YARN y un modelo de programación de MapReduce simple para procesar y analizar datos por lotes en paralelo.
Spark de Apache	Plataforma de procesamiento paralelo de código abierto que admite el procesamiento en memoria para mejorar el rendimiento de las aplicaciones de análisis de macrodatos.
Apache Hbase	Base de datos NoSQL en Hadoop que proporciona acceso aleatorio y coherencia fuerte para grandes cantidades de datos no estructurados y semiestructurados; potencialmente miles de millones de filas multiplicadas por millones de columnas.
ML Services	Un servidor para hospedar y administrar procesos de R distribuidos en paralelo. Proporciona a los científicos de datos, estadísticos y programadores de R acceso a petición a métodos escalables y distribuidos para realizar análisis en HDInsight.
Apache Storm	Sistema distribuido de cálculo en tiempo real para el procesamiento rápido de grandes transmisiones de datos.
Apache Kafka	una plataforma de código abierto que se usa para crear canalizaciones y aplicaciones de datos de streaming. Kafka también proporciona funcionalidad de cola de mensajes que le permite publicar flujos de datos y suscribirse a ellos.

Herramientas de desarrollo para HDInsight

Puede usar herramientas de desarrollo de HDInsight, como IntelliJ, Eclipse, Visual Studio Code y Visual Studio, para crear y enviar trabajos y consultas de datos de HDInsight con una integración perfecta con Azure.

- [Kit de herramientas de Azure para IntelliJ](#)
- [kit de herramientas de Azure para Eclipse](#)
- [Herramientas de Azure HDInsight para VS Code](#)
- [Herramientas de Azure Data Lake para Visual Studio](#)

HANDS-ON LABS

Hadoop

¡GRACIAS!

