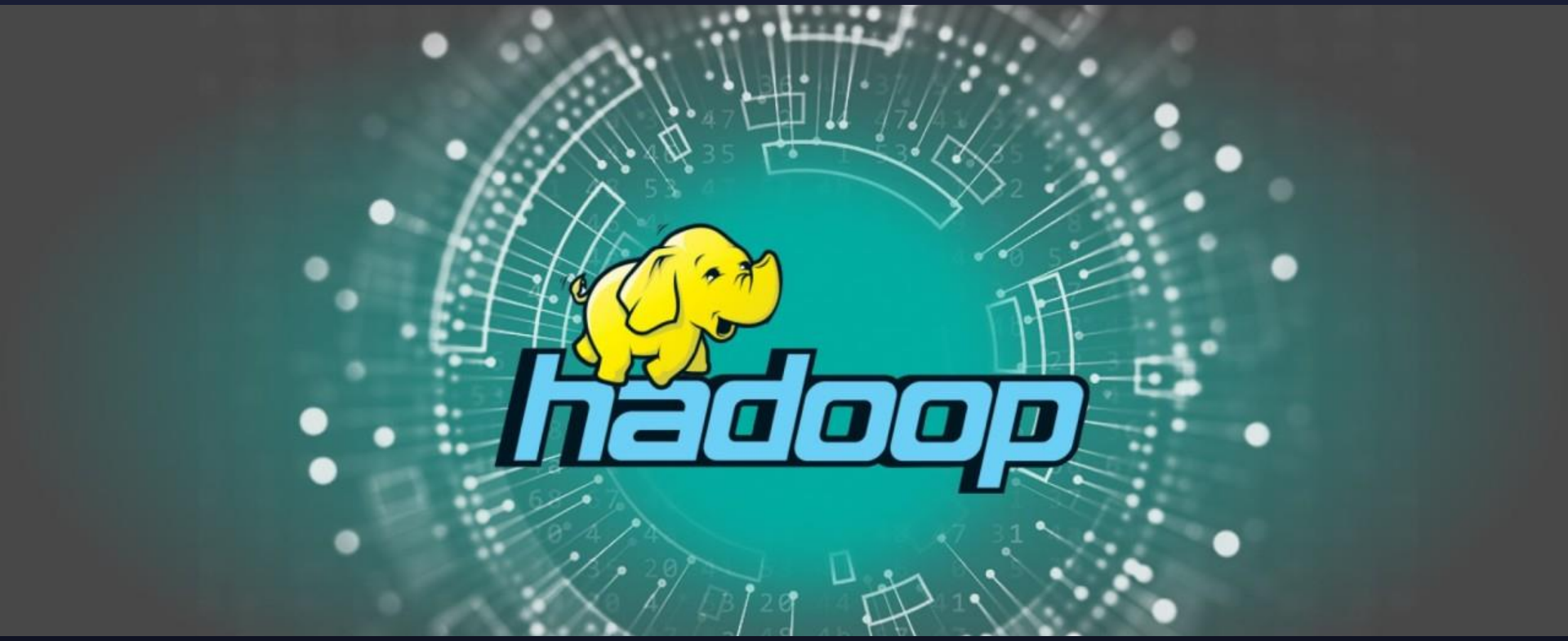
IES SAN ANDRÉS

ESPECIALIZACIÓN EN INTELIGENCIA ARTIFICIAL Y BIG DATA

BIG DATA APLICADO



UT01: INTRODUCCIÓN A APACHE HADOOP

ÍNDICE

1.- Contexto y motivación

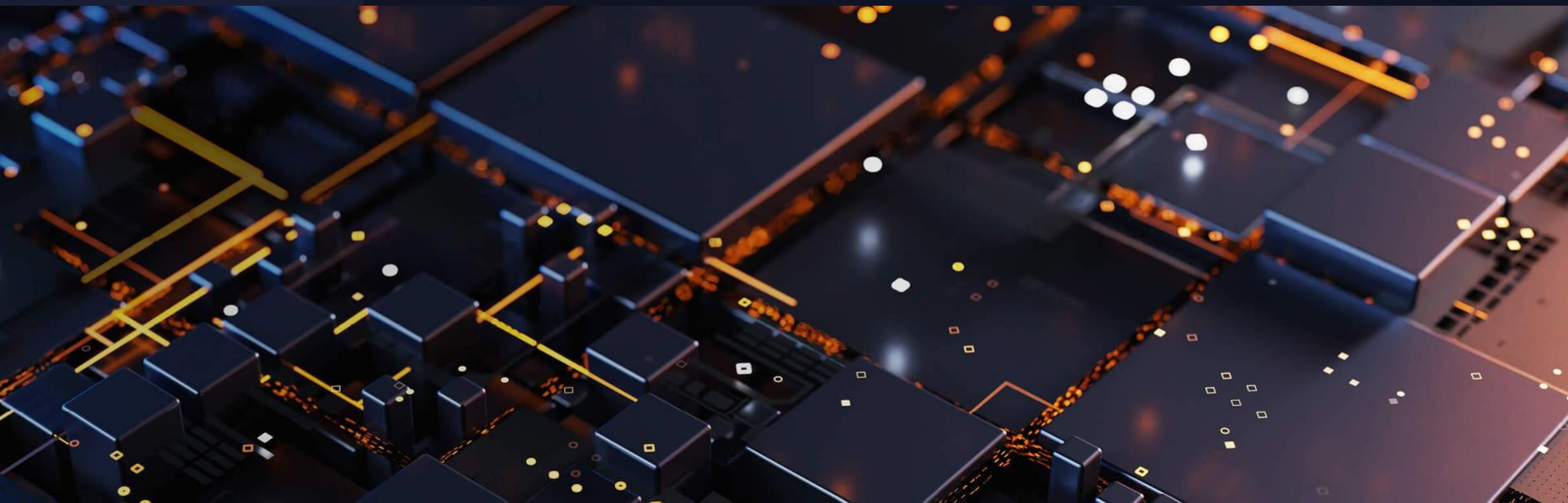
2.- ¿Qué es Apache Hadoop?

3.- Historia de Apache Hadoop

4.- Problemas que resuelve Apache Hadoop

5.- Preparación del entorno

1 CONTEXTO Y MOTIVACIÓN



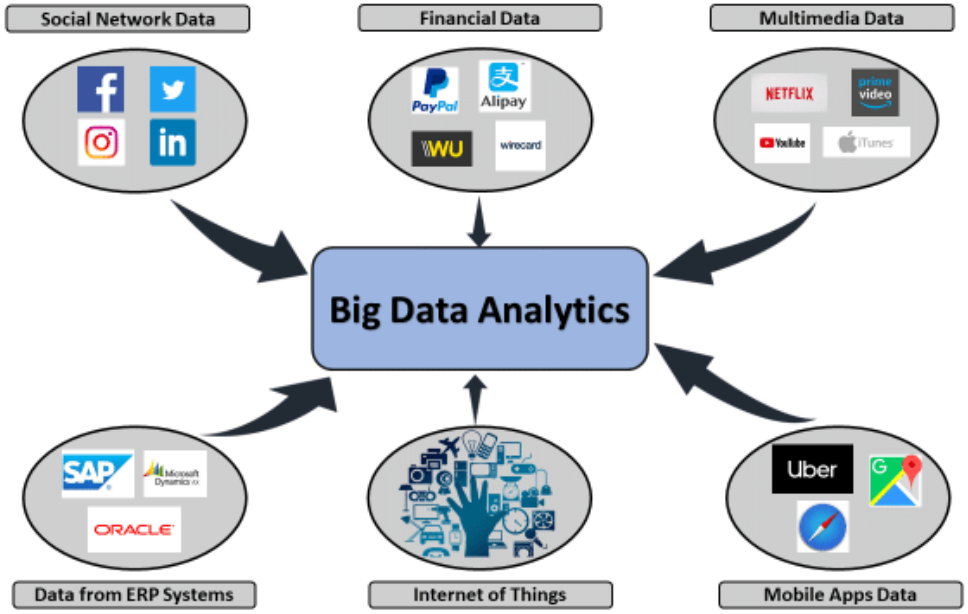
EL CRECIMIENTO DEL BIG DATA

Víctor J. González

En las últimas décadas, el volumen de datos ha crecido exponencialmente.

Múltiples fuentes de datos:

- Redes sociales
- Sensores IoT
- Registros de transacciones
- Sistemas empresariales
- Vídeos
- Imágenes
- Logs de aplicaciones
- …



EL CRECIMIENTO DEL BIG DATA

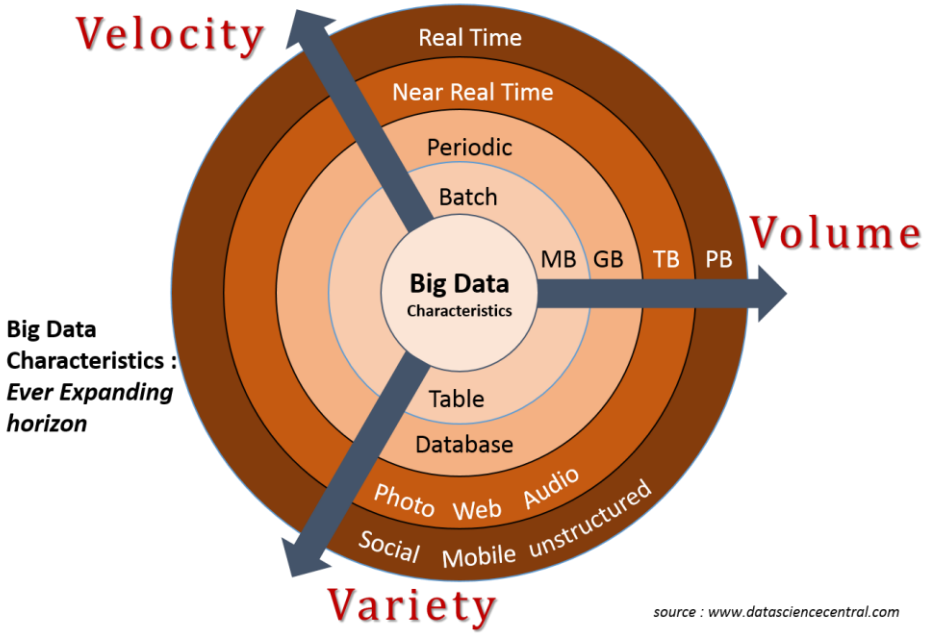
Víctor J. González

Este fenómeno se llama **Big Data** y se caracteriza por las 3V

Volumen: se generan ingentes cantidades de datos de forma constante.

Velocidad: los datos se producen y necesitan procesarse en tiempo real o casi real

Variedad: los datos pueden ser estructurados (tablas), semiestructurados (JSON, XML) o no estructurados (texto libre, imágenes, vídeo)



Las soluciones tradicionales (como BBDD relacionales o los sistemas monolíticos) tienen serias **limitaciones** para manejar eficientemente este nuevo paradigma.

LIMITACIONES SISTEMAS TRADICIONALES

Víctor J. González

Los sistemas tradicionales diseñados para:

Diseñados para

- Entornos controlados
- Volúmenes de datos moderados
- Alta estructura
- Bajo nivel de concurrencia masiva

LIMITACIONES SISTEMAS TRADICIONALES

Víctor J. González

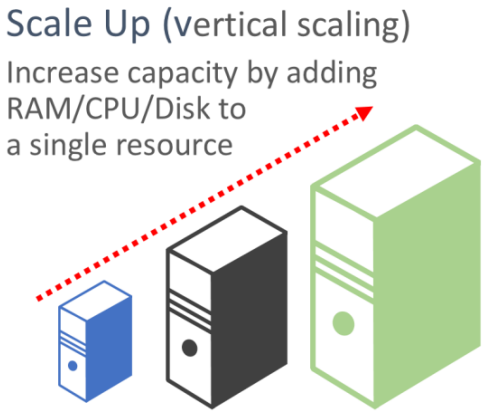
Los sistemas tradicionales diseñados para:

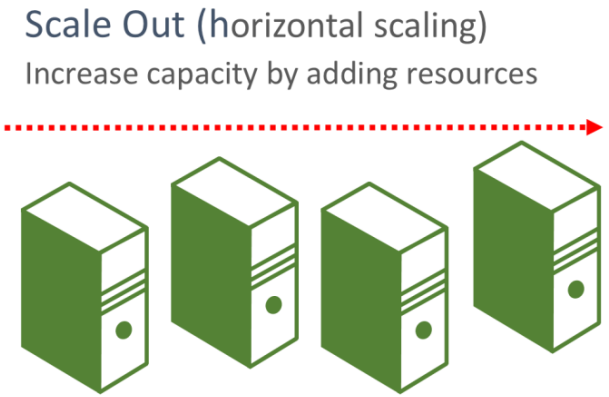
Diseñados para

- Entornos controlados
- Volúmenes de datos moderados
- Alta estructura
- Bajo nivel de concurrencia masiva

Limitaciones

1 **Escalabilidad vertical limitada**: para mejorar el rendimiento requieren hardware más potente. Costoso y no escalable a largo plazo





LIMITACIONES SISTEMAS TRADICIONALES

Víctor J. González

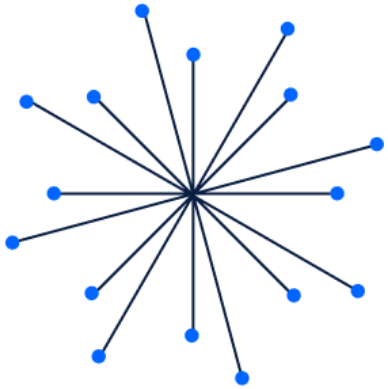
Los sistemas tradicionales diseñados para:

Diseñados para

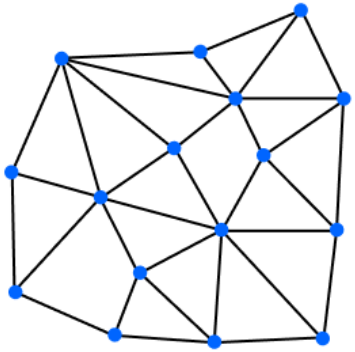
- Entornos controlados
- Volúmenes de datos moderados
- Alta estructura
- Bajo nivel de concurrencia masiva

Limitaciones

2 **Procesamiento centralizado**: los datos se suelen almacenarse y procesarse en un único nodo o servidor, lo que supone un cuello de botella.



Centralized



Distributed

LIMITACIONES SISTEMAS TRADICIONALES

Víctor J. González

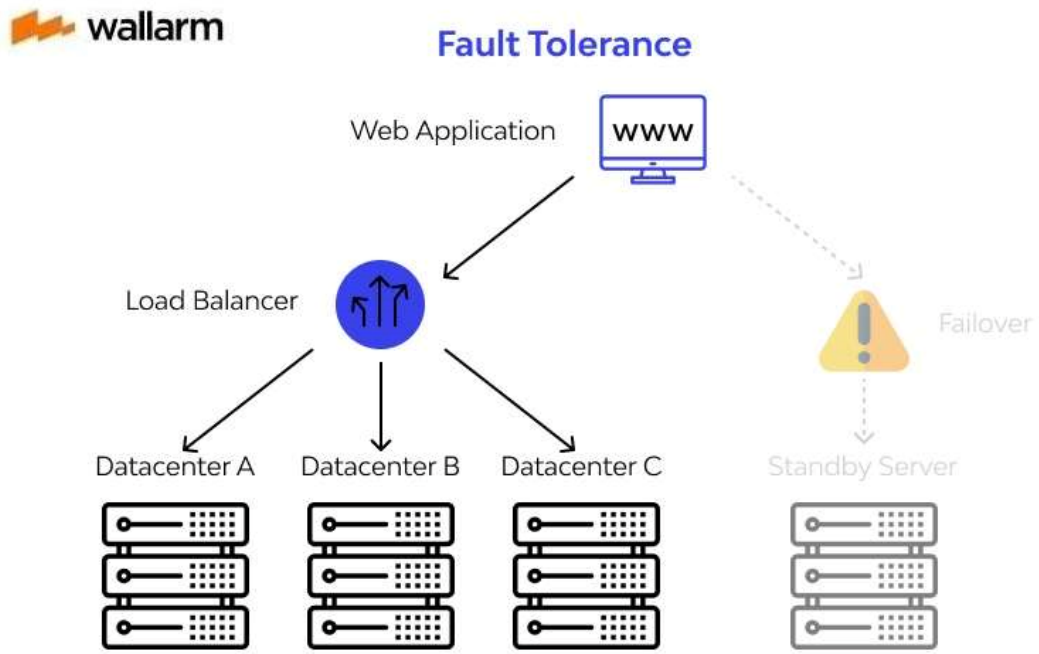
Los sistemas tradicionales diseñados para:

Diseñados para

- Entornos controlados
- Volúmenes de datos moderados
- Alta estructura
- Bajo nivel de concurrencia masiva

Limitaciones

3 **Falta de tolerancia a fallos**: si el nodo central falla, el sistema completo puede dejar de funcionar



LIMITACIONES SISTEMAS TRADICIONALES

Víctor J. González

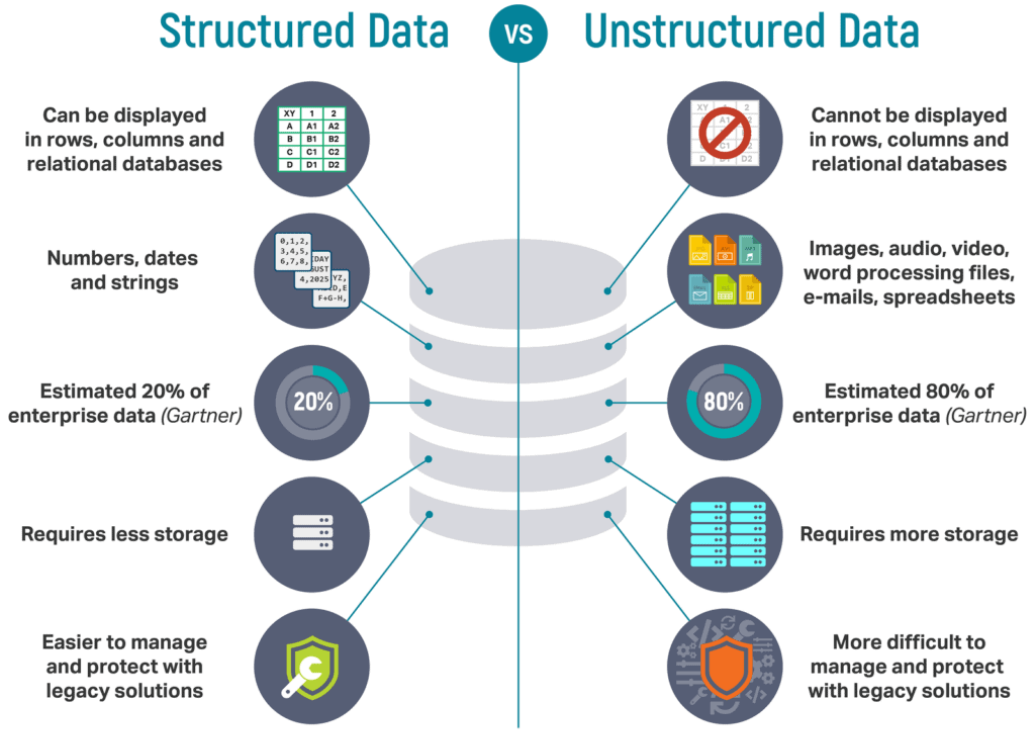
Los sistemas tradicionales diseñados para:

Diseñados para

- Entornos controlados
- Volúmenes de datos moderados
- Alta estructura
- Bajo nivel de concurrencia masiva

Limitaciones

4 **Dificultad para trabajar con datos semiestruturados o no estructurados**: los RDBMS están diseñados para datos altamente estructurados.



LIMITACIONES SISTEMAS TRADICIONALES

Víctor J. González

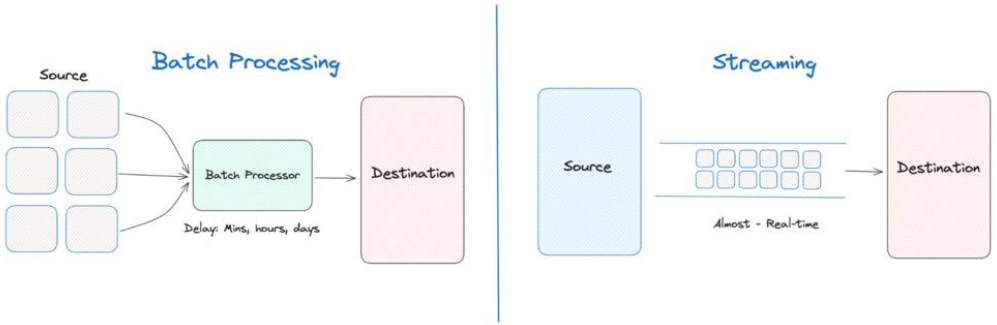
Los sistemas tradicionales diseñados para:

Diseñados para

- Entornos controlados
- Volúmenes de datos moderados
- Alta estructura
- Bajo nivel de concurrencia masiva

Limitaciones

5 **Procesamiento batch lento**: el análisis de grandes volúmenes de datos puede llevar horas o días



Streaming vs Batch Processing

decube

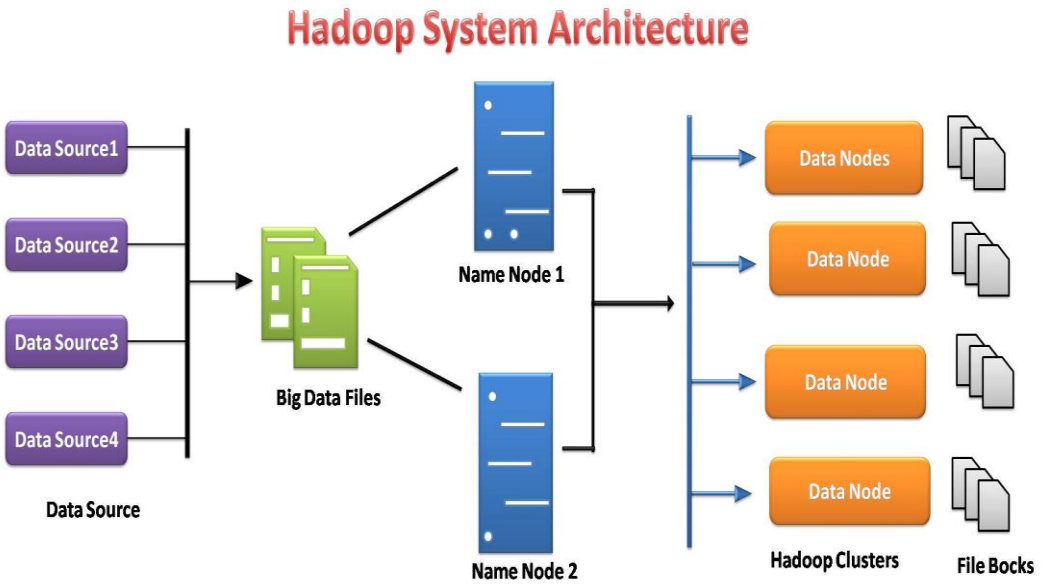
Ante este panorama, surge la necesidad de adoptar un **nuevo enfoque**.

Apache Hadoop surge como respuesta a esta necesidad, basando su modelo en **principios distribuidos**.

Almacenamiento distribuido (HDFS)

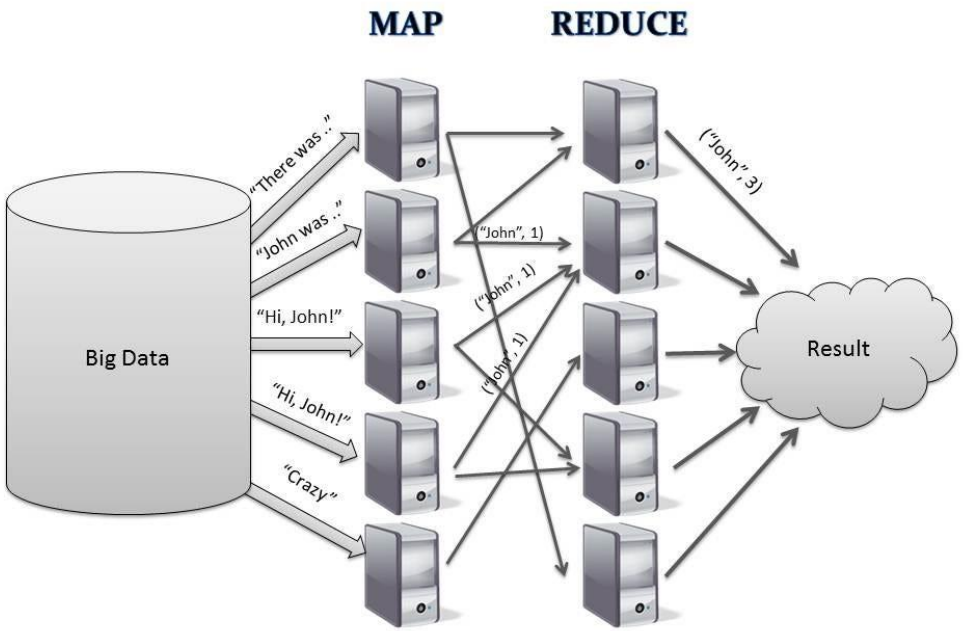
Los datos se dividen en bloques y se almacenan en múltiples nodos del clúster.

Se aprovecha la capacidad de múltiples máquinas en lugar de depender de un único servidor.



Ante este panorama, surge la necesidad de adoptar un **nuevo enfoque**.

Apache Hadoop surge como respuesta a esta necesidad, basando su modelo en **principios distribuidos**.



Procesamiento distribuido (MapReduce, Spark)

Las tareas se dividen y ejecutan en paralelo en los distintos nodos del clúster.

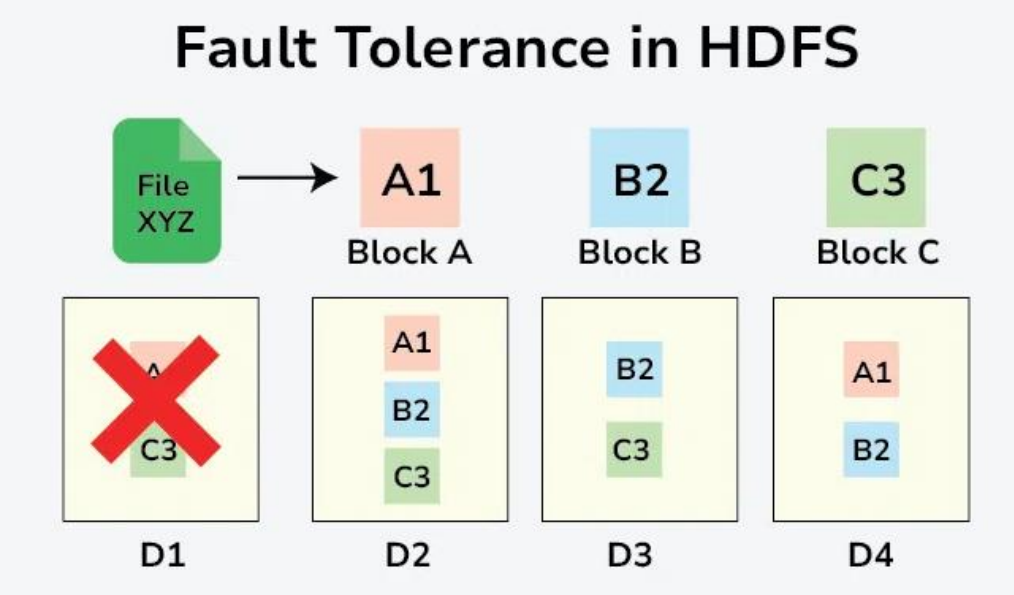
Se reducen enormemente los tiempos de procesamiento

Ante este panorama, surge la necesidad de adoptar un **nuevo enfoque**.

Apache Hadoop surge como respuesta a esta necesidad, basando su modelo en **principios distribuidos**.

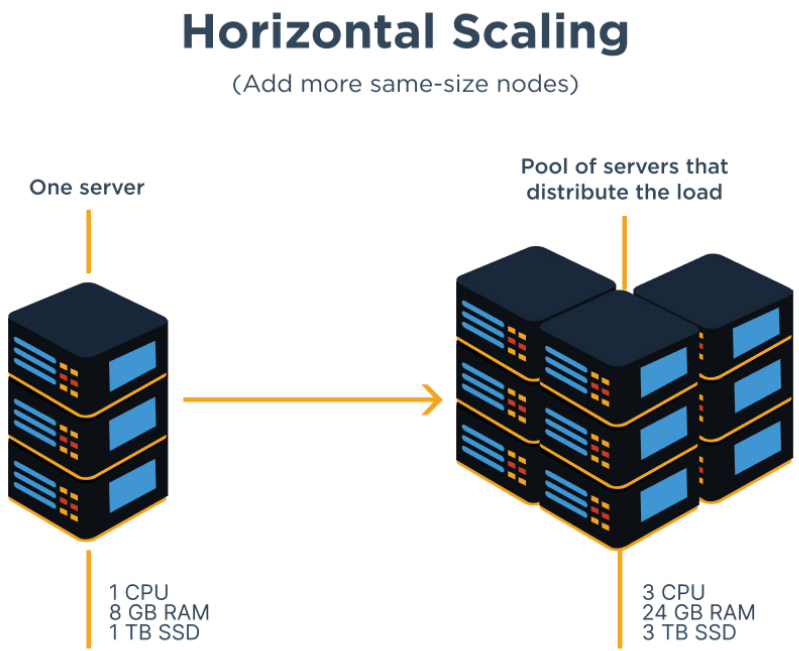
Tolerancia a fallos

Los sistemas distribuidos como Hadoop replican datos y pueden reintentar tareas automáticamente en caso de fallos, garantizando la **alta disponibilidad**



Ante este panorama, surge la necesidad de adoptar un **nuevo enfoque**.

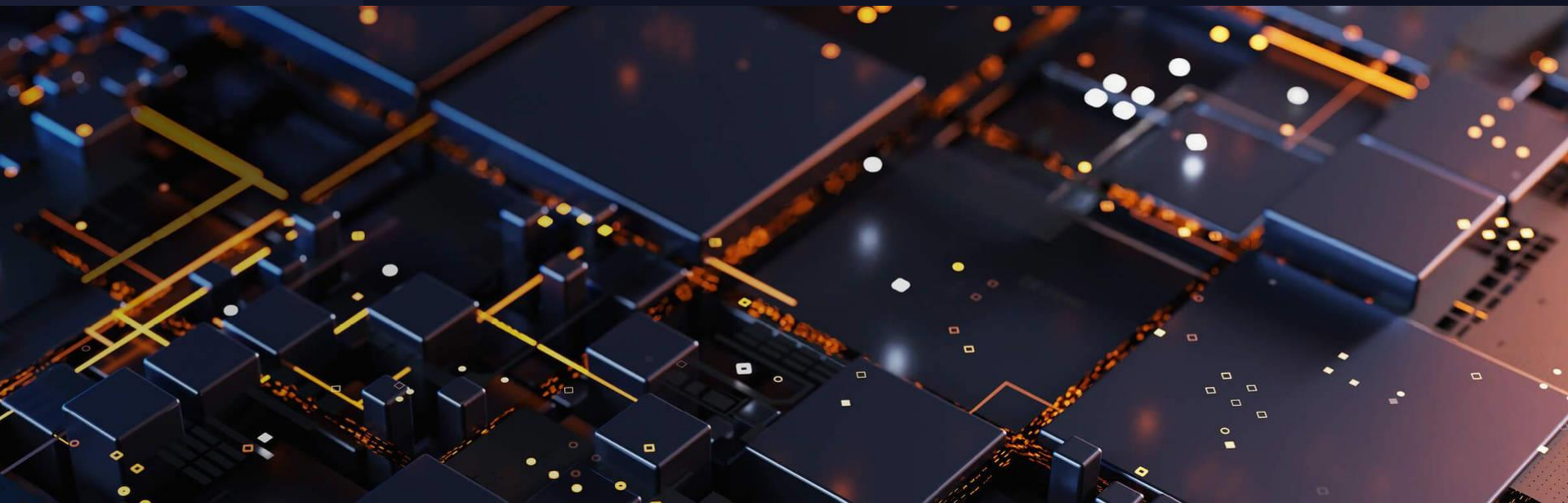
Apache Hadoop surge como respuesta a esta necesidad, basando su modelo en **principios distribuidos**.



Escalabilidad horizontal

Es posible añadir más nodos al clúster para aumentar la capacidad de almacenamiento o procesamiento sin interrumpir el sistema.

2 ¿QUÉ ES APACHE HADOOP?



¿QUÉ ES APACHE HADOOP?

Apache Hadoop es un framework de software de código abierto que permite el almacenamiento distribuido y el procesamiento paralelo de grandes volúmenes de datos utilizando clústeres de ordenadores convencionales.

¿QUÉ ES APACHE HADOOP?

Víctor J. González

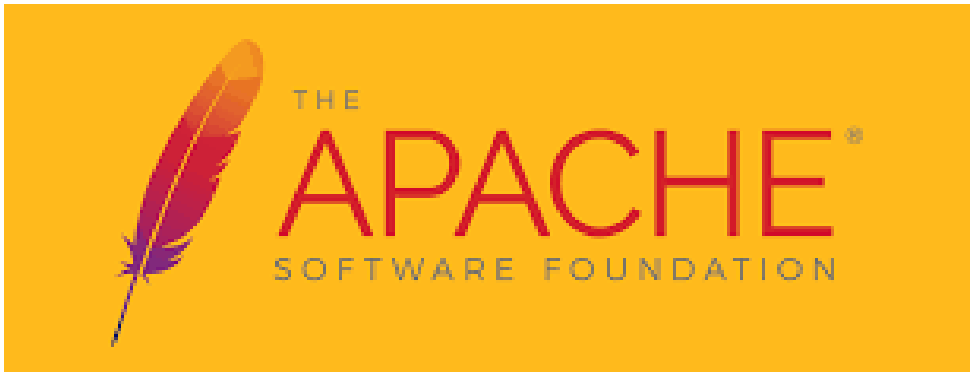
Apache Hadoop es un **framework de software** de código abierto que permite el almacenamiento distribuido y el procesamiento paralelo de grandes volúmenes de datos utilizando clústeres de ordenadores convencionales.

Framework de software: Hadoop es un marco de trabajo, es decir, un conjunto de librerías, servicios y herramientas ya preparados para que los desarrolladores o administradores puedan construir aplicaciones sin tener que hacerlo desde cero.

¿QUÉ ES APACHE HADOOP?

Apache Hadoop es un framework de software **de código abierto** que permite el almacenamiento distribuido y el procesamiento paralelo de grandes volúmenes de datos utilizando clústeres de ordenadores convencionales.

De código abierto: Hadoop pertenece a la Apache Software Foundation y está disponible mediante una licencia Open Source, de forma que cualquiera puede descargar, usar, modificar y redistribuir el código sin coste de licencias.

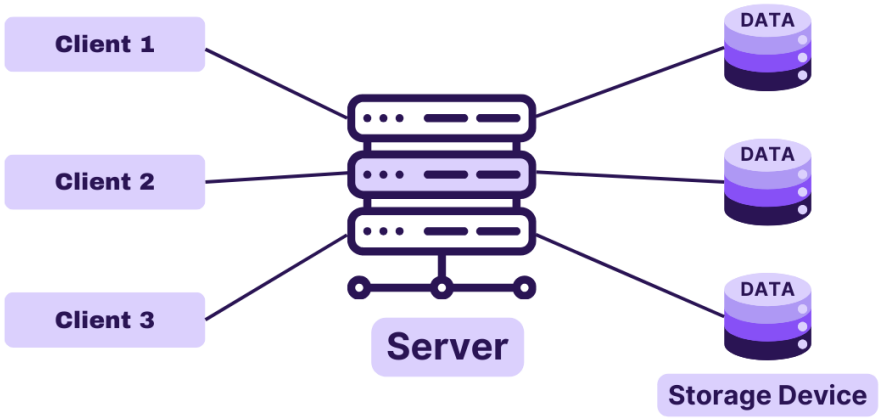


¿QUÉ ES APACHE HADOOP?

Víctor J. González

Apache Hadoop es un framework de software de código abierto que permite el **almacenamiento distribuido** y el procesamiento paralelo de grandes volúmenes de datos utilizando clústeres de ordenadores convencionales.

Almacenamiento distribuido: los datos no se guardan en un solo servidor, sino que se parten en bloques y se guardan en diferentes nodos del clúster.

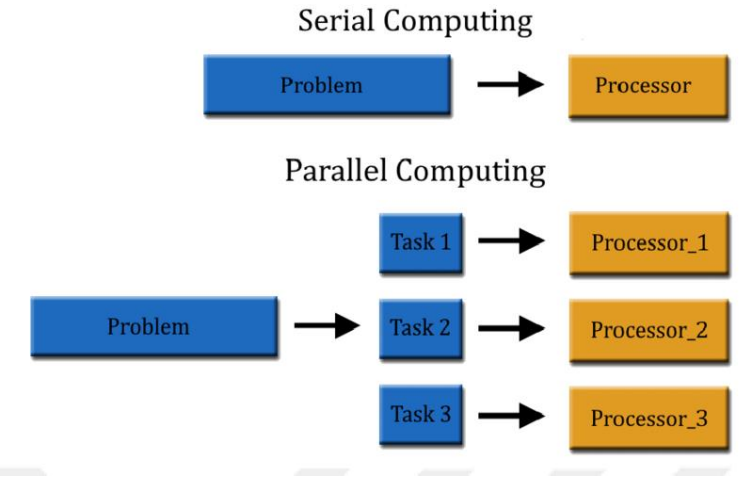


¿QUÉ ES APACHE HADOOP?

Víctor J. González

Apache Hadoop es un framework de software de código abierto que permite el almacenamiento distribuido y el **procesamiento paralelo** de grandes volúmenes de datos utilizando clústeres de ordenadores convencionales.

Procesamiento paralelo: en lugar de que un único ordenador procese todos los datos secuencialmente, se divide el trabajo en muchas tareas que corren en paralelo en diferentes nodos.

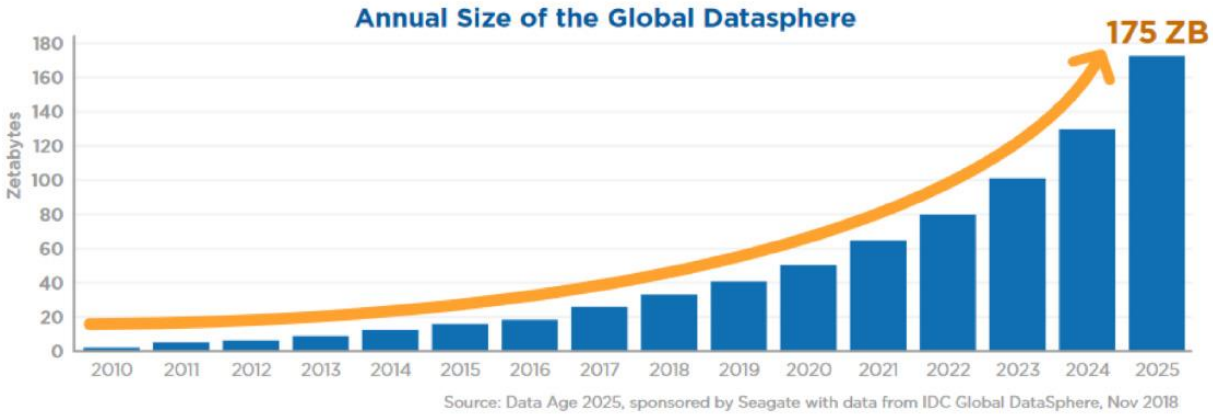


¿QUÉ ES APACHE HADOOP?

Víctor J. González

Apache Hadoop es un framework de software de código abierto que permite el almacenamiento distribuido y el procesamiento paralelo de **grandes volúmenes de datos** utilizando clústeres de ordenadores convencionales.

Grandes volúmenes de datos: Hadoop está diseñado para lo que llamamos Big Data, conjuntos de datos del orden de Terabytes o Exabytes que no pueden ser procesados con métodos tradicionales.

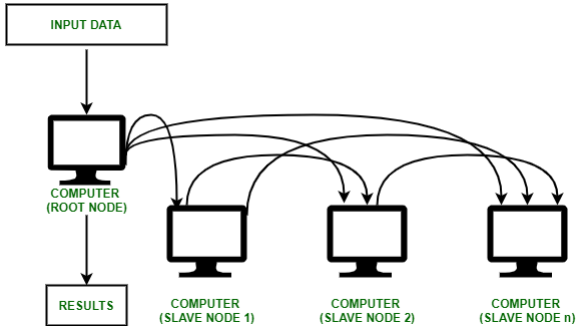


¿QUÉ ES APACHE HADOOP?

Víctor J. González

Apache Hadoop es un framework de software de código abierto que permite el almacenamiento distribuido y el procesamiento paralelo de grandes volúmenes de datos utilizando **clústeres de ordenadores convencionales**.

Clústeres de ordenadores convencionales: Hadoop trabaja sobre conjuntos de ordenadores conectados en red que trabajan como si fueran uno solo. Además, no requiere supercomputadoras, sino que se puede montar sobre hardware barato (commodity hardware) y aprovechar muchos nodos simples en lugar de unos pocos muy potentes.



¿QUÉ ES APACHE HADOOP?

Víctor J. González

Apache Hadoop **resuelve dos problemas clave** del Big Data:

Almacenar datos que no caben en un único servidor.

Procesar grandes cantidades de información de forma eficiente y tolerante a fallos

Para conseguir resolver estos dos problemas, Hadoop tiene **dos tecnologías fundamentales**.

COMPONENTES PRINCIPALES DE HADOOP

Víctor J. González

HDFS (Hadoop Distributed File System)

Sistema de archivos distribuido

Permite almacenar archivos de gran tamaño dividiéndolos en bloques y distribuyéndolos entre los nodos del clúster.

Ofrece alta disponibilidad gracias a la replicación de bloques

Diseñado para trabajar eficientemente con lecturas y escrituras secuenciales de archivos de gran tamaño.

YARN (Yet Another Resource Negotiator)

Es el sistema que gestiona los recursos del clúster y controla la ejecución de aplicaciones.

YARN permite múltiples motores de procesamiento que pueden compartir recursos dentro del mismo entorno.

YARN está disponible a partir de Hadoop 2.x

COMPONENTES PRINCIPALES DE HADOOP

Víctor J. González

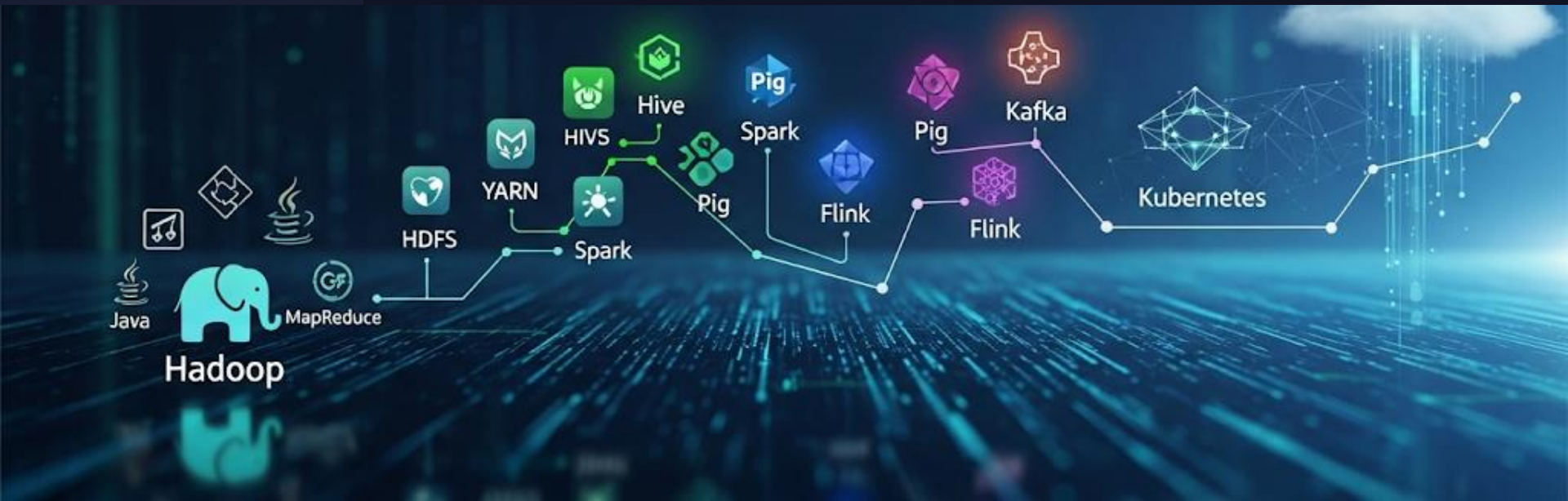
Aunque la base de Hadoop son HDFS y YARN, dispone de un enorme ecosistema con múltiples herramientas como:

- **MapReduce**: el motor de procesamiento por lotes que forma parte del diseño original de Hadoop.
- **Apache Hive**, **Apache Pig**: lenguajes de alto nivel para realizar consultas y transformaciones de datos.
- **Apache Spark**: motor de procesamiento distribuido en memoria que puede ejecutarse sobre Hadoop.
- **HBase, Oozie, Zookeeper**, entre otros.

CARACTERÍSTICAS DE HADOOP

ESCALABILIDAD	Puede crecer fácilmente añadiendo más nodos al clúster
TOLERANCIA A FALLOS	Los datos están replicados en distintos nodos para prevenir pérdidas por fallos de hardware
PORTABILIDAD	Al estar desarrollado íntegramente en Java, puede ejecutarse en múltiples plataformas
EFICIENCIA	Permite el procesamiento paralelo masivo de datos en bloques distribuidos
ECOSISTEMA EN EXPANSIÓN	Hay una gran comunidad y una amplia gama de herramientas complementarias

3 ¿HISTORIA DE APACHE HADOOP?



HISTORIA DE HADOOP

Víctor J. González

La historia de Hadoop está ligada al nacimiento del Big Data.

Antes de Hadoop, procesas conjuntos de datos masivos a bajo coste era prácticamente imposible para la mayoría de las empresas.

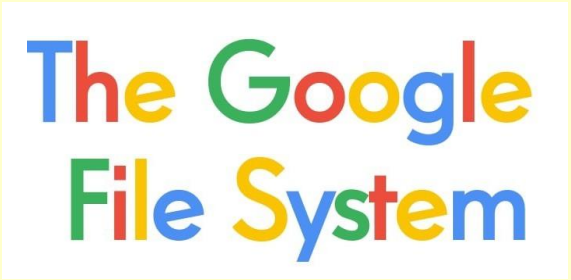
La solución fue Hadoop, que **democratizó el análisis de datos a gran escala**.

Años 2000

A principios del 2000, Google vio la necesidad de buscar alguna tecnología que le permitiera indexar y procesar millones de páginas web

2003

En este año Google publica un artículo titulado "*The Google File System*" que detalla su diseño e implementación como sistema de ficheros distribuidos para el procesamiento de grandes cantidades de datos.



2004

Los ingenieros de Google, Jeffrey Dean y Sanjay Ghemawat publican un artículo titulado "*MapReduce: simplified data processing on large clusters*"

2004

Simultáneamente, dos ingenieros de Yahoo!, Doug Cutting y Mike Cafarella, que estaban trabajando en el proyecto de búsqueda **Nutch**, se encontraron los mismos problemas de escalabilidad, por lo que decidieron implementar una **versión de código abierto** de estas tecnologías.

2005

El proyecto Nutch migró a lo que se llamó **Hadoop**, un nombre que Doug Cutting le puso en honor al elefante de juguete de su hijo

HISTORIA DE HADOOP

Víctor J. González

2006

Hadoop se separa de Nutch para convertirse en un **proyecto independiente** de **Apache Software Foundation**, con Yahoo! como principal patrocinador.

2012

Llega **Hadoop 2**, que supone un punto de inflexión. Se introduce **YARN**, un gestor de recursos que desacopló la gestión de recursos del modelo de procesamiento.

HISTORIA DE HADOOP

Víctor J. González

2012

Con Hadoop 2, MapReduce pasa a ser una aplicación más. Esto permitió que otras tecnologías de procesamiento, como **Apache Spark** puedan coexistir en el mismo clúster de Hadoop.

Actualidad

Actualmente, Hadoop ya no es solo HDFS y MapReduce, sino que es un vasto **ecosistema** de proyectos de código abierto que trabajan en conjunto para resolver diferentes problemas de Big Data

ECOSISTEMA DE HADOOP

Víctor J. González

En la actualidad, el ecosistema asociado a Hadoop es enorme, con todo tipo de aplicaciones orientadas al trabajo con grandes cantidades de datos desde cualquier perspectiva.

Apache Hive

Herramienta de data warehousing que permite **consultar y analizar** grandes volúmenes de datos almacenados en HDFS usando un lenguaje similar a SQL (**HiveQL**)

Apache Pig

Plataforma de alto nivel para crear programas de MapReduce utilizando un lenguaje de scripting llamado **Pig Latin**. Simplifica la escritura de tareas complejas de procesamiento de datos

Apache Spark

Motor de **procesamiento de datos en memoria** que permite realizar análisis de datos en tiempo real, procesamiento por lotes y *machine learning*. Compatible con lenguajes como Scala, Java, Python o R

Apache HBase

Base de datos **NoSQL** distribuida y escalable que se ejecuta sobre HDFS

Apache Kafka

Plataforma de *streaming* distribuida para la **ingesta y procesamiento de flujos de datos en tiempo real**. Ideal para aplicaciones de mensajería y procesamiento de eventos

Apache Flume

Herramienta para la recolección, agregación y movimiento de grandes cantidades de **datos de logs** hacia HDFS

Apache Zookeeper

Servicio centralizado para mantener la **configuración, sincronización y coordinación de servicios distribuidos**. Es esencial para la gestión de clústeres y la alta disponibilidad

Apache Scoop

Herramienta para **transferir datos** entre Hadoop y bases de datos relacionales (MySQL, Oracle)

Apache Oozie

Sistema de programación de **flujos de trabajo** (*workflow*) para gestionar tareas de Hadoop. Permite orquestar tareas complejas que involucran múltiples herramientas del ecosistema

Apache Mahout

Biblioteca de ***machine learning*** escalable para realizar tareas como clasificación, recomendación y clasificación

Apache Storm

Sistema de **procesamiento de flujos de datos** en tiempo real

Apache NiFi

Herramienta para **automatizar el flujo de datos** entre sistemas mediante una interfaz gráfica

Apache Tez

Framework para ejecutar tareas de **procesamiento de datos** de manera más eficiente que MapReduce.

Apache Drill

Motor de consultas SQL distribuido para **datos no estructurados y semiestructurados**. Permite consultar datos en formato JSON, Parquet o CSV sin necesidad de un esquema predefinido

Apache Atlas

Herramienta de gobierno y metadatos para **gestionar y rastrear el linaje de los datos** en Hadoop

Apache Zeppelin

Notebook interactivo para realizar **análisis de datos, visualizaciones y colaboración**. Compatible con múltiples lenguajes (Scala, Python, R) y herramientas (Spark, Hive)

Apache Ranger

Framework de **seguridad** para gestionar políticas de acceso y auditoría en Hadoop

Apache Ambari

Herramienta de **gestión y monitorización de clústeres** Hadoop que simplifica la instalación, configuración y mantenimiento del clúster

Apache Kylin

Motor de OLAP (Procesamiento Analítico en Línea) para Hadoop.

Apache Phoenix

Capa SQL sobre HBase para realizar **consultas en tiempo real**.
Combina la escalabilidad de HBase con la facilidad de uso de SQL

Apache Flink

Framework de **procesamiento de flujos de datos en tiempo real** y por lotes, ideal para aplicaciones de streaming.

Apache Knox

Puerta de enlace (*gateway*) para proporcionar **acceso seguro** a los servicios de Hadoop

Apache Superset

Herramienta de visualización y exploración de datos que permite crear ***dashboards* interactivos y gráficos** a partir de datos almacenados en Hadoop

4 PROBLEMAS QUE RESUELVE

HISTORIA DE HADOOP

Víctor J. González

Los sistemas tradicionales de almacenamiento y procesamiento se han quedado cortos frente al **crecimiento masivo**, la **variedad** y la **velocidad** con la que se generan los datos.

Hadoop es la solución a múltiples desafíos:

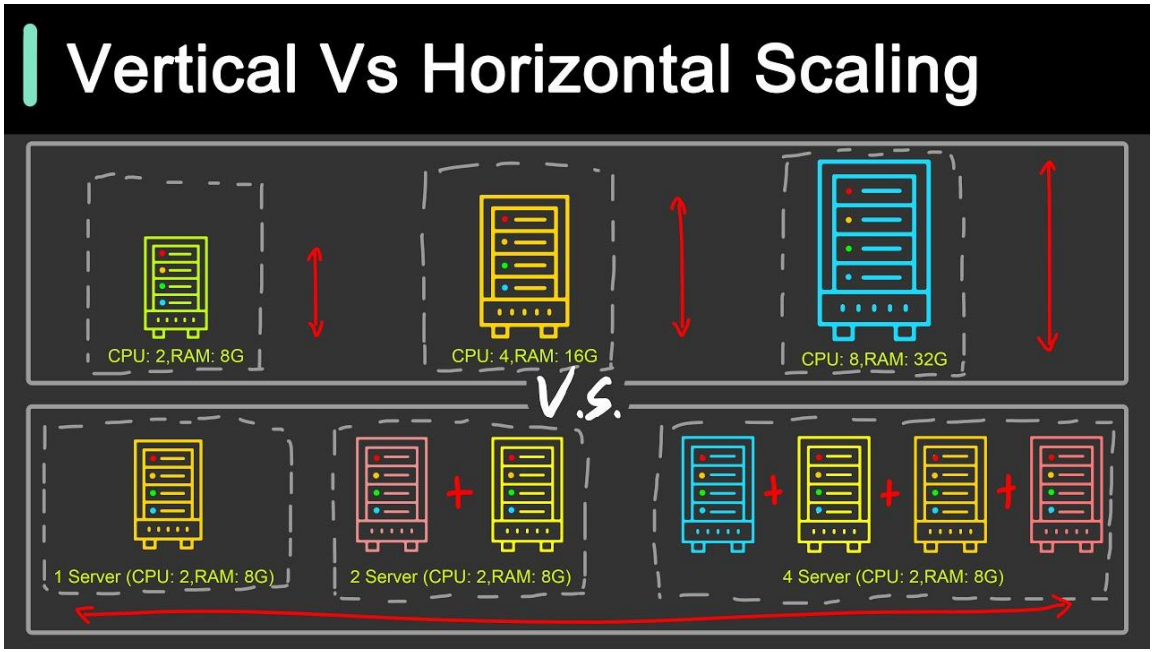
- Escalabilidad horizontal
- Almacenamiento distribuido
- Procesamiento paralelo
- Tolerancia a fallos
- Flexibilidad en el tratamiento de datos
- Coste

ESCALABILIDAD HORIZONTAL

Las bases de datos tradicionales escalan de forma **vertical** (más recursos a una única máquina).

Hadoop escala **horizontalmente**, agregando más nodos económicos al clúster.

Permite cantidades masivas de datos sin invertir en hardware especializado y costoso

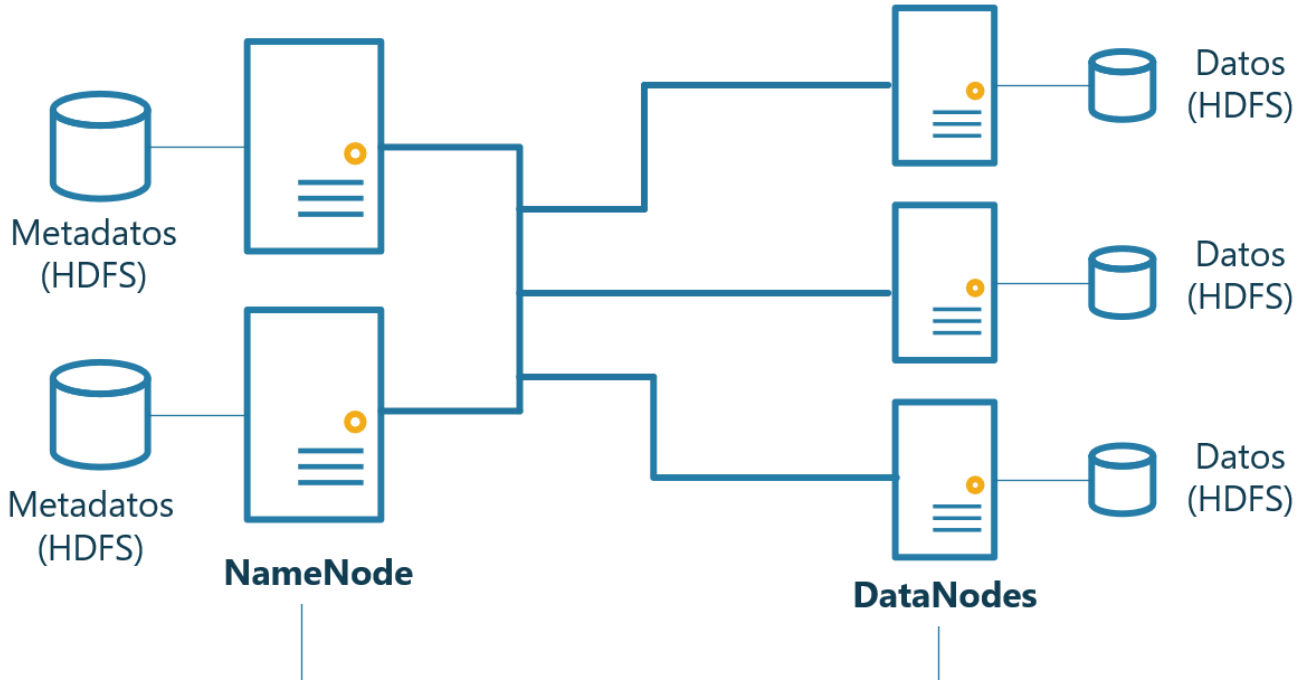


ALMACENAMIENTO DISTRIBUIDO

Otro desafío es el almacenamiento de grandes cantidades de datos.

HDFS distribuye los archivos en bloques y los reparte entre diferentes nodos.

Además, cada bloque se replica automáticamente, garantizando disponibilidad e integridad.

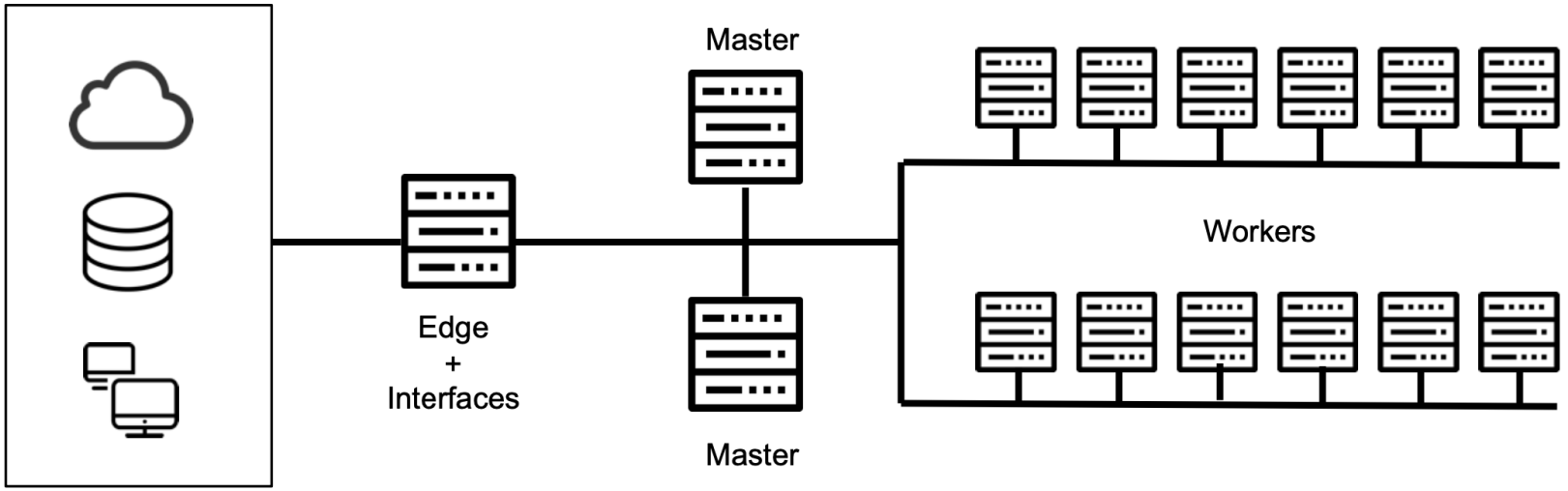


PROCESAMIENTO PARALELO

Procesar grandes cantidades de datos eficientemente es otro desafío.

Hadoop permite procesamiento paralelo masivo mediate modelos como MapReduce o Spark.

Las tareas se distribuyen y ejecutan simultáneamente en múltiples nodos.

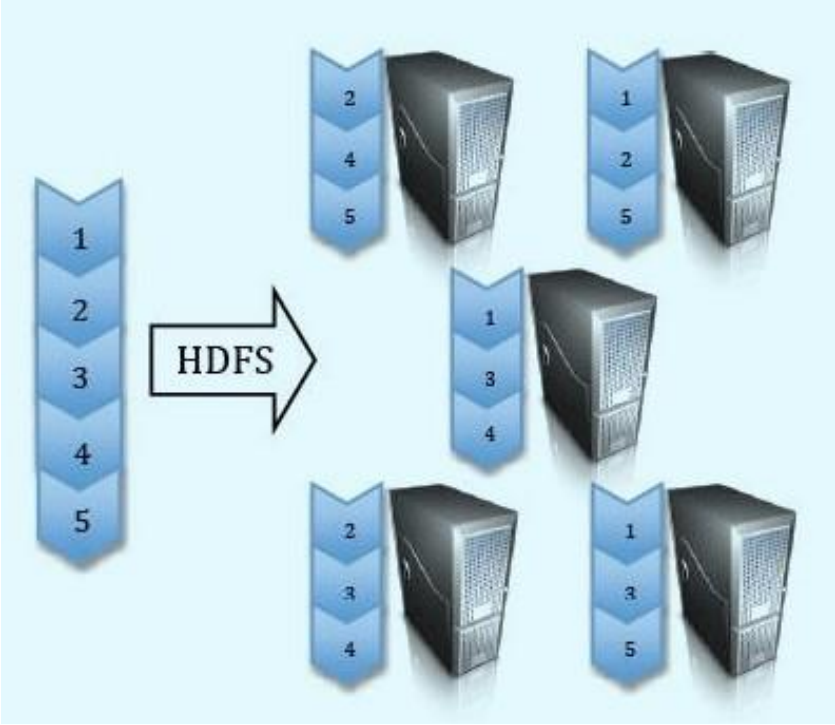


TOLERANCIA A FALLOS

En sistemas distribuidos en habitual que algunos nodos fallen.

Hadoop está diseñado con una fuerte tolerancia a fallos gracias a la **replicación de bloques** y a la **replanificación automática de tareas**.

Esto evita la realización de backups y garantiza alta disponibilidad



FLEXIBILIDAD EN EL TRATAMIENTO DE DATOS

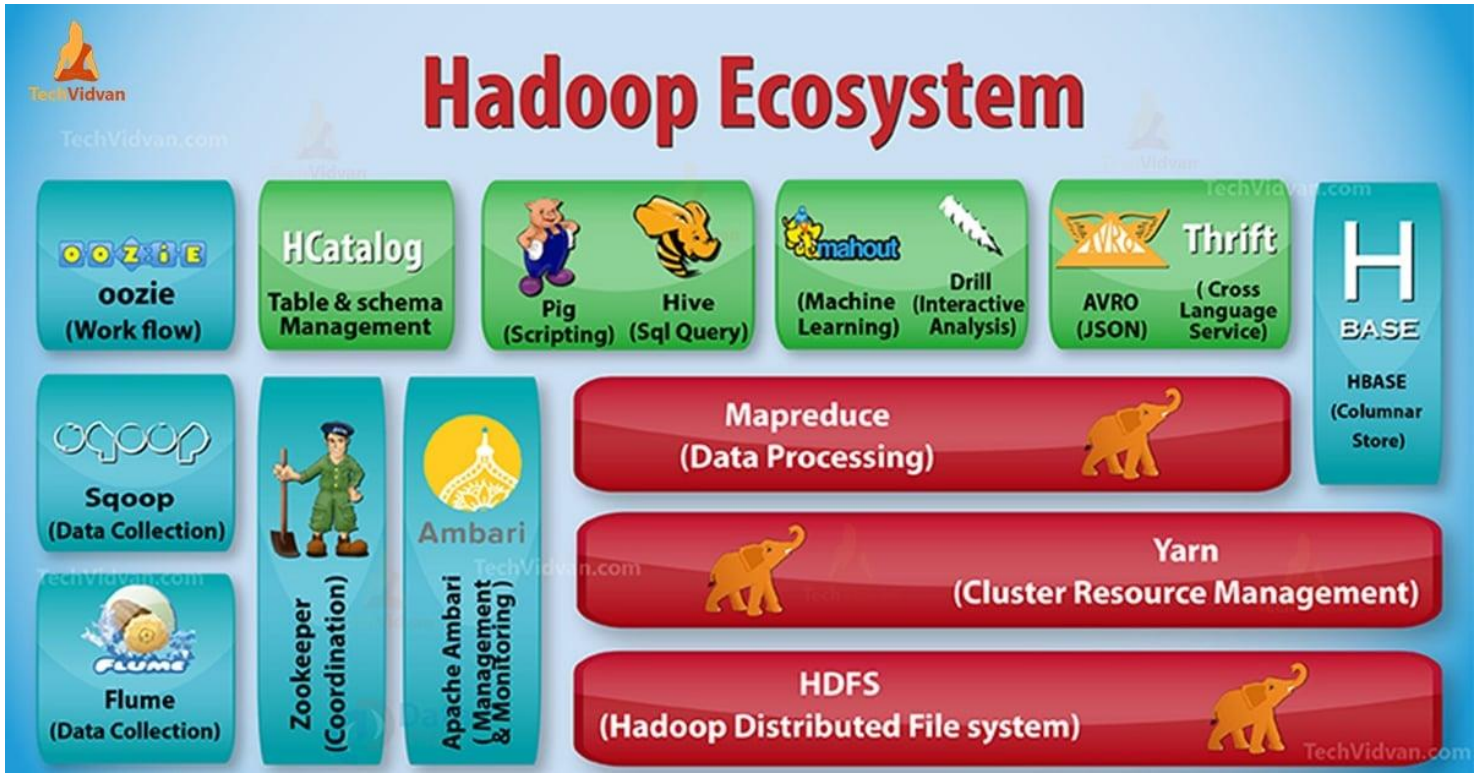
Los sistemas estructurados exigen **datos estructurados**.

Hadoop permite trabajar con datos estructurados, semi-estructurados y no estructurados, como logs de servidores, archivos JSON, XML, imágenes, vídeos o datos en texto plano.

COSTE

Hadoop es software libre y se puede ejecutar sobre hardware convencional, lo que permite reducir costes frente a soluciones propietarias.

Esto democratiza el acceso al procesamiento de grandes cantidades de datos.



5 PREPARACIÓN DEL ENTORNO