# Introducción a Apache Hadoop.

## Inicios

Estamos en el año 2002, Doug y Mike trabajan en el proyecto Apache Nutch, un motor de búsqueda que pretende indexar mil millones de páginas web.

Después de analizar la arquitectura que han diseñado, llegan a la conclusión de que para poder cumplir su objetivo, y con la tecnología del momento, costaría más de medio millón de euros en hardware, y otro medio millón al año de costes de mantenimiento o ejecución.

¡No puede ser! ¡Esto es inviable! ¡Tenemos que darle una vuelta a la arquitectura!

Doug y Mike empiezan a buscar soluciones en el mercado que les permitan, por un lado, almacenar un volumen de datos enorme, y por otro, poder procesar todos los datos para generar los índices de búsqueda. Por si fuera poco, la solución debe ser económica, así que no es tarea sencilla.

Por suerte, descubren algo que solucionará su problema. Con el tiempo, lo que acaban de descubrir se llamará Apache Hadoop

A lo largo de esta unidad vas a aprender por qué surge Apache Hadoop, cómo ha evolucionado y en qué estado se encuentra actualmente. Además, vas a poder entender cómo funciona, qué arquitectura tiene y qué funcionalidades ofrece, y sobre todo, entenderás qué beneficios aporta, así como las desventajas que tiene frente a otras tecnologías o las dificultades que tiene Hadoop para ser utilizado por las organizaciones.

Los contenidos de la unidad serán los siguientes:

- Motivación, origen e historia de Apache Hadoop: te mostrará por qué se originó Hadoop, en qué se basó y cómo ha ido evolucionando desde su origen.
- ¿Qué es Apache Hadoop?: detallará qué es Hadoop y qué características principales tiene.
- Ecosistema y distribuciones: Hadoop es una plataforma con un ecosistema de herramientas asociado. Este apartado te mostrará las herramientas de este ecosistema y cómo se han agrupado dichas herramientas para facilitar su uso.
- Arquitectura de Hadoop: en este apartado conocerás dónde se instala, qué tipo de hardware necesita y cómo funciona.
- Beneficios, desventajas y dificultades: Hadoop tiene unos beneficios importantes, pero como toda nueva tecnología, tiene una serie de dificultades que se detallarán en este apartado.

## 1.- Motivación y origen.

### Inicios

Doug y Mike están buscando una solución que permita almacenar mil millones de páginas y procesarlas para el proyecto Apache Nutch un motor de búsqueda, como los que ya tienen Google o Yahoo. Aunque en el año 2002 no era habitual enfrentarse a problemas que requieran tratar con un volumen de datos tan grande, ya había compañías, como las que acabamos de mencionar, que habían resuelto ese problema.


¿Por qué no investigar entonces cómo Google o Yahoo han solucionado este reto?

En octubre del año 2003, Google publicó un *paper* sobre su Sistema de almacenamiento escalable denominado The **Google File System**. En este *paper* Google explicó cómo resolvían la problemática de almacenar un gran volumen de datos (petabytes) a un bajo coste utilizando un modelo de almacenamiento distribuido.

Este *paper* sirvió de inspiración a Doug Cutting y Mike Cafarella para diseñar la

solución de almacenamiento de Apache Nutch. ¡Sólo quedaba por resolver cómo procesar los datos para generar los índices de búsqueda!

En diciembre de 2004, Google publicó otro *paper*, cuyo título era **Map Reduce: Simplified Data Processing on Large Clusters**, donde se describía cómo había resuelto Google el procesamiento sobre conjuntos de datos voluminosos utilizando un paradigma de computación ya existente, MapReduce.

Este segundo *paper* resolvía, por lo tanto, el procesamiento de los datos para generar los índices de búsqueda de Apache Nutch

Los *papers* de Google describían la solución de Google a alto nivel, pero no daban detalles sobre cómo se había implementado a nivel de código. Doug Cutting y Mike Cafarella empezaron a trabajar en la implementación dentro del proyecto Apache Nutch Sin embargo, se encontraron con la dificultad de que el desarrollo era complejo, y que sólo dos personas necesitarían mucho tiempo para implementarlo. Además, se dieron cuenta de que Apache Nutch no podría sacar toda la potencia de lo que estaban desarrollando por limitaciones técnicas.

Por estos motivos, en 2006, Doug Cutting se incorpora a Yahoo! con el objetivo de aprovechar la capacidad y el equipo que esta compañía tenía, para poder implementar GFS y MapReduce y ofrecerlo al mundo como código libre. Doug Cutting estaba convencido de que un sistema de almacenamiento masivo y económico, junto con un sistema de procesamiento de datos para grandes volúmenes de datos a bajo coste, iba a revolucionar el estado de la tecnología del momento, y por ese motivo, sacó GFS y MapReduce del proyecto original Apache Nutch y les dio entidad propia. Este nuevo proyecto se llamaría **Hadoop**

## Para saber más


Apache Software Foundation ((Apache License, Version 2.0)

Doug Cutting eligió el nombre Hadoop porque era el nombre de un peluche de elefante que tenía su hijo. El nombre le parecía fácil de pronunciar y de recordar, además de tener un componente emocional para él.

En 2007, Hadoop ya se había implementado en Yahoo! en una primera versión beta, y fue probado en una infraestructura de más de 1000 nodos.

En enero de 2008, Yahoo! donó Hadoop a Apache Software Foundation, pasando a ser un proyecto Top-Level de la fundación. A partir de este momento, una gran cantidad de compañías y de desarrolladores mostraron interés en Hadoop y comenzaron a contribuir en el proyecto, haciendo que éste creciera en gran medida en los siguientes años.

En el año 2009, Doug Cutting abandonó Yahoo! y se incorporó a Cloudera, que pretendía hacer de Hadoop un producto de uso empresarial, ya que hasta ahora había tenido una vocación más de investigación o innovación.

La primera versión estable de Hadoop, la 1.0, fue liberada en diciembre de 2011.

### Para saber más

Si quieres echar un vistazo a los *papers* que Google publicó sobre Google File System y Map Reduce, aquí tienes los enlaces a las publicaciones:

- Google File System: https://pdos.csail.mit.edu/6.824/papers/gfs.pdf
- MapReduce: Simplified Data Processing on Large Clusters: https://static.googleusercontent.com/media/research.google.com/es//archive/mapreduce-osdi04.pdf

## 2.- Apache Hadoop a alto nivel.

Hadoop es una plataforma que permite almacenar y procesar grandes volúmenes de datos. No es una plataforma sencilla porque tiene muchos componentes, y además, se instala en muchos servidores.

Vamos a descubrir Hadoop empezando por un alto nivel, entendiendo qué es, para qué sirve y cómo funciona, y en los siguientes módulos iremos desgranando todos los componentes que pertenecen a la plataforma.

### Para saber más

Apache Hadoop tiene su propia documentación oficial, donde puedes encontrar todos los detalles sobre la plataforma. Es una documentación extensa, pero si quieres acceder y ver su contenido, o para profundizar en algún punto, aquí tienes un enlace a la página oficial: https://hadoop.apache.org/

## 2.1.- ¿Qué es Apache Hadoop?

Apache Software Foundation (Apache License, Version 2.0)

Apache Hadoop es una **plataforma *opensource*** que ofrece la capacidad de **almacenar** y **procesar**, a "bajo" **coste**, grandes **volúmenes** de datos, sin importar su **estructura** en un entorno **distribuido escalable** y **tolerante a fallos**, basado en la utilización de **hardware commodity** y en un paradigma acercamiento del **procesamiento a los datos**

Vamos a desgranar la definición, extrayendo toda la información que contiene:

# Plataforma

Hadoop es una plataforma, lo que significa que es la base sobre la que construir aplicaciones. Se podría hacer el símil a que Hadoop es una caja de herramientas que proporciona un conjunto de herramientas con las que construir una gran variedad de aplicaciones que requieran almacenar y procesar grandes volúmenes de datos. La selección de qué herramienta utilizar para cada aplicación la realizaremos en función de las necesidades de cada caso de uso.

Otras soluciones, como MongoDB u otras bases de datos NoSQL no se consideran plataformas, ya que tienen un único propósito y ofrecen un tipo de funcionalidad.


Íñigo Sanz (Dominio público)

# *Opensource*

Hadoop no es una solución comercial, sino que todo su código es libre y por lo tanto, no

hay que pagar licencias o costes de adquisición del software de la plataforma.

## Almacenar

Hadoop ofrece la capacidad de almacenar y recuperar datos mediante un sistema de ficheros que se llama HDFS, que veremos más adelante.

## Procesar

Hadoop ofrece la capacidad de crear aplicaciones que procesen los datos almacenados en el sistema de ficheros, tanto de forma *batch* como *real-time*

## Coste

El coste de implantar una plataforma Hadoop es órdenes de magnitud más bajo que otras soluciones tradicionales de almacenamiento y procesamiento de datos, como podrían ser las bases de datos relacionales o los sistemas mainframe.

## Volumen

Permite almacenar prácticamente cualquier volumen de datos, desde volumetrías pequeñas (megabytes) a volumetrías muy altas (petabytes).

## Estructura

Los datos que pueden almacenarse y procesarse en Hadoop pueden tener cualquier tipo: estructurados, semiestructurados o datos no estructurados.


Íñigo Sanz (Dominio público)

## Distribuido

Hadoop se basa en una infraestructura que tiene muchos servidores (también llamados

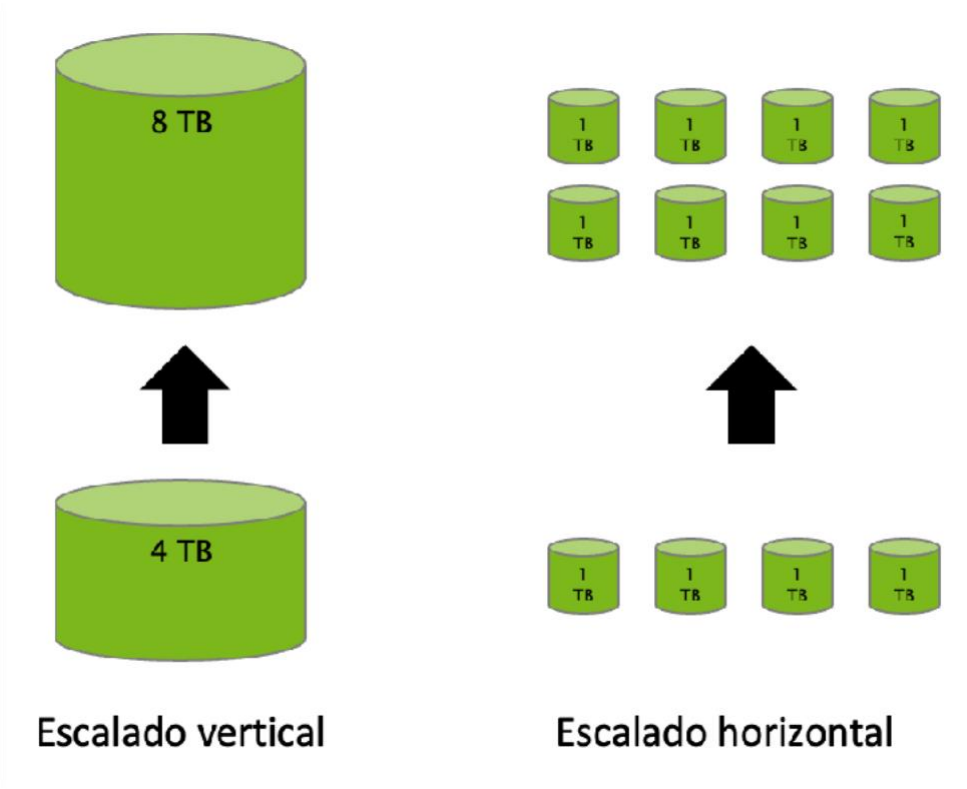
nodos) que trabajan conjuntamente para almacenar o para procesar los datos, a diferencia de los sistemas centralizados, donde todo se realiza en un único servidor.

# Escalable

Hadoop permite crecer en infraestructura (servidores) hasta adecuarse a las necesidades de almacenamiento o procesamiento del caso de uso. El modelo de escalado es horizontal, es decir, si nuestra plataforma necesita crecer, en lugar de cambiar el servidor por uno de mayor capacidad (escalado vertical), se añaden más servidores del mismo tipo. Este tipo de escalabilidad tiene dos ventajas principales:

El coste de incrementar la capacidad es **lineal**, es decir, si mi plataforma tiene una capacidad de 1 petabyte, y necesitamos incrementarla a 2 petabytes, el coste será el doble del coste inicial que hubo. En los sistemas de escalado vertical, el coste suele ser exponencial, es decir, incrementar de 1 terabyte a 2 terabytes puede suponer un coste 3 veces superior.

2 La capacidad máxima vendrá determinada por el número máximo de servidores que se puede añadir. En el caso de Hadoop, el límite está en torno a unos 10.000 nodos (un nodo puede almacenar muchos terabytes). En los sistemas que tienen escalado vertical, el límite de crecimiento lo determina el tamaño máximo de un servidor que se puede comprar.




Íñigo Sanz (Dominio público)

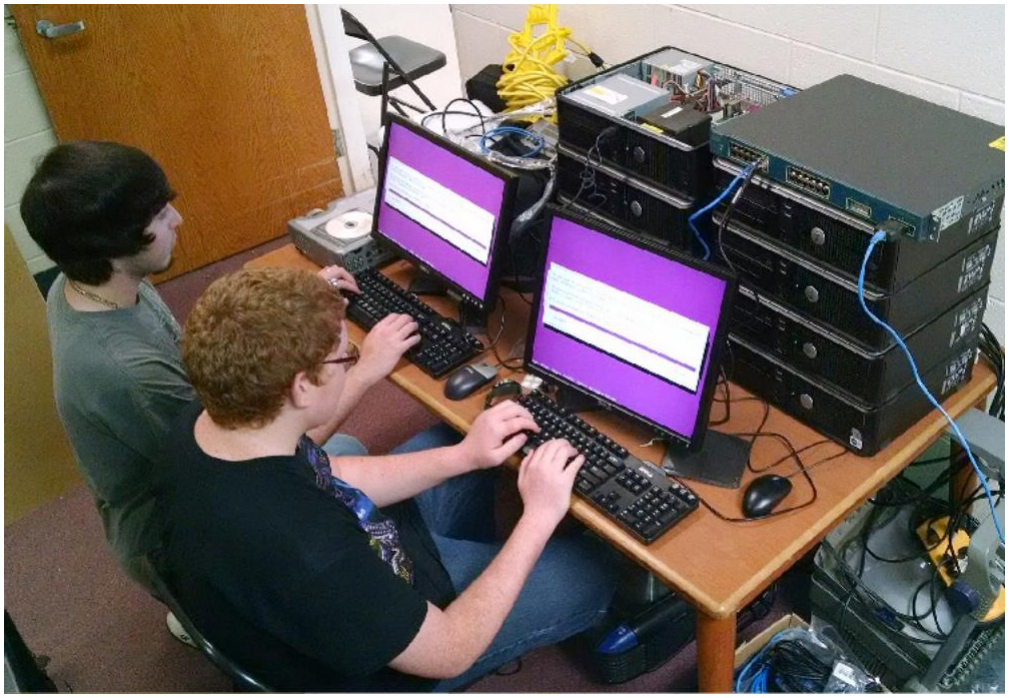
# Tolerante a fallos

Hadoop es una plataforma que garantiza que ante la caída de uno de los servidores, el sistema sigue funcionando y no se pierden datos. Esto es fundamental por varios motivos:

- Un sistema empresarial, es decir, un sistema que se va a utilizar para dar soporte a las operaciones de una empresa, debe funcionar correctamente 24x7, ¿a quién le gusta que cuando quiere hacer una llamada de teléfono o una transferencia bancaria le digan que en ese momento no se puede realizar la operación? O peor, ¿a quién le gustaría que el banco te comunique que ha perdido tus datos y que no sabe cuánto dinero tienes en la cuenta?
- Al tener un modelo distribuido, y estar formado por múltiples servidores, la probabilidad de que un servidor se rompa cada día es muy elevada. Piensa en una plataforma Hadoop de 1.000 servidores, con 12 discos duros por servidor, es decir, un sistema con 12.000 discos. Si un disco tiene una duración estimada de 3 años, significaría que todos los días se romperían 11 discos. Hadoop garantiza que aunque se rompan 11 discos cada día, no se van a perder datos y las aplicaciones seguirán funcionando correctamente.

## *Hardware commodity*

Hadoop no requiere servidores específicos con unas exigencias muy concretas. El concepto de *hardware commodity* ya se ha tratado con anterioridad. Ojo, ¡no significa que Hadoop se puede desplegar con los portátiles reciclados de una oficina!



Virginia Department of Education *Ejemplo de un despliegue de Hadoop en una universidad* (CC BY)

# Acercamiento del procesamiento a los datos

Los sistemas de procesamiento masivo de datos tradicionales se basaban en tener un sistema de almacenamiento separado del sistema de procesamiento. Este modelo

requiere que antes de hacer cualquier proceso, hay que leer todos los datos y transportarlos al sistema de computación. Este transporte se realiza por la red de comunicaciones, que no tiene un ancho de banda comparable con el ancho de banda de lectura en disco y procesamiento en la CPU de la misma máquina. En los sistemas tradicionales, el cuello de botella siempre es la red de comunicaciones.

## Debes conocer

Hadoop no fue la primera plataforma capaz de almacenar y procesar un volumen de datos grande.

Haz la siguiente reflexión: ¿los bancos, en los años 90, necesitaban gestionar un volumen de datos grande? Piensa en la cantidad de transferencias, pagos con tarjeta, movimiento de la valoración de las acciones cada día, etc. que un banco tiene que almacenar y tratar.

Efectivamente, ya había casos de uso "Big Data" antes de la aparición de Hadoop u otras tecnologías Big Data, y por ejemplo, los bancos utilizaron sistemas *mainframe* para almacenar todos los datos de su operativa así como su procesamiento. La principal diferencia entre Hadoop y otros sistemas tradicionales es que el coste es órdenes de magnitud inferior (10 veces, 1.000 veces, ...).

## 2.2.- Ecosistema Hadoop y distribuciones.

Los componentes *core* principales de Hadoop son HDFS, MapReduce y YARN

- **HDFS:** un sistema de ficheros (capa de almacenamiento) que almacena los datos en una estructura basada en espacios de nombres (directorios, subdirectorios, etc.).
- **YARN:** un gestor de recursos (capa de procesamiento) que permite ejecutar aplicaciones sobre los datos almacenados en HDFS.
- **MapReduce**: un sistema de procesamiento masivo de datos que se puede utilizar directamente, programando sobre su API, o indirectamente, con aplicaciones que lo utilizan de forma transparente.

Sin embargo, normalmente se identifica el nombre Hadoop con todo el ecosistema de **componentes independientes** que suelen incluirse para dotar a Hadoop de funcionalidades necesarias en proyectos Big Data empresariales, como puede ser la ingesta de información, el acceso a datos con lenguajes estándar, o las capacidades de administración y monitorización.

Estos componentes suelen ser proyectos *opensource* de Apache.

Los principales componentes o proyectos asociados al ecosistema Hadoop son los siguientes:

| Nombre | Descripción | Logotipo |
|---|---|---|
| **Apache Hive** | Permite acceder a ficheros de datos estructurados o semiestructurados que están en HDFS como si fueran una tabla de una base de datos relacional, utilizando un lenguaje similar a SQL |  Apache Software Foundation (Apache License) |
| **Apache Pig** | Utilidad para definir flujos de datos de transformación o consulta mediante un lenguaje de *scripting*. |  Apache Software Foundation (Apache License) |
| **Apache HBase** | Base de datos NoSQL de tipo columnar que permite el acceso aleatorio, atómico y con operaciones de edición de datos. |  Apache Software Foundation (Apache License) |
| **Apache Flume** | Componente para ingestar *streams* de datos procedentes de sistemas real-time en Hadoop. |  Apache Software Foundation (Apache License) |
| **Apache Sqoop** | Componente para importar o exportar datos estructurados desde bases de datos relacionalesa Hadoop y viceversa. |  Apache Software Foundation (Apache License) |
| **Apache Oozie** | Herramienta que permite definir flujos de trabajo en Hadoop así como su orquestación y planificación. |  Apache Software Foundation (Apache License) |
| **Apache ZooKeeper** | Herramienta técnica que permite sincronizar el estado de los diferentes servicios distribuidos de Hadoop. |  Apache Software Foundation (Apache License) |
| **Apache Storm** | Sistema de procesamiento real-time de eventos con baja latencia. |  Apache Software Foundation |

| | | |
|---|---|---|
| | | (Apache License) |
| **Apache Spark** | Aunque habitualmente no se asocia al ecosistema Hadoop, Apache Spark ha sido el mejor complemento de Hadoop en los últimos años. Apache Spark es un motor de procesamiento masivo de datos muy eficiente que ofrece funcionalidades para ingeniería de datos, *machine learning*, grafos, etc. | Apache Software Foundation (Apache License) |
| **Apache Kafka** | Sistema de mensajería que permite recoger eventos en tiempo real así como su procesamiento. | Apache Software Foundation (Apache License) |
| **Apache Atlas** | Herramienta de gobierno de datos de Hadoop. | Apache Software Foundation (Apache License) |
| **Apache Accumulo** | Base de datos NoSQL que ofrece funcionalidades de acceso aleatorio y atómico. | Apache Software Foundation (Apache License) |
| **Apache Mahout** | Conjunto de librerías para desarrollo y ejecución de modelos de *machine learning* utilizando las capacidades de computación de Hadoop. | Apache Software Foundation (Apache License) |
| **Apache Phoenix** | Capa que permite acceder a los datos de HBase mediante interfaz SQL. | Apache Software Foundation (Apache License) |
| **Apache Zeppelin** | Aplicación web de notebooks que permite alos Data Scientists realizar análisis y evaluar código de forma sencilla, así como la colaboración entre equipos. | Apache Software Foundation (Apache License) |
| **Apache Impala** | Herramienta con funcionalidad similar a Hive (tratamiento de los datos de HDFS mediante SQL) pero con un rendimiento elevado (tiempos de respuesta menores). | Apache Software Foundation (Apache License) |

## Recomendación

No te preocupes si ves muchos componentes y piensas que es imposible dominar todos. En la realidad, los proyectos suelen utilizar sólo una pequeña parte de los componentes dependiendo de las necesidades.

Los más utilizados son: Apache Spark, Apache Hive y Apache Kafka, además de los componentes *core*: HDFS y YARN.

Cada componente es un proyecto Apache independiente, lo que impacta, entre otros a:

- **Política de versionado (periodicidad, identificación, ...)**: cada componente tiene sigue su propio camino en cuanto a cuándo se publican las nuevas versiones, qué mejoras o evoluciones incluyen, etc.
- **Dependencias del proyecto con otras versiones de componentes del ecosistema y librerías externas**: los componentes suelen tener dependencias entre ellos. Por ejemplo, Hive tiene dependencia de HDFS, o Phoenix de HBase. Las dependencias suelen ser difíciles de gestionar, por ejemplo, porque una versión de Phoenix requiere una versión específica de HBase.
- ***Roadmap* y estrategia del proyecto**: al tener grupos de trabajo diferentes, cada proyecto tiene su propia estrategia en cuanto a cómo evolucionar la solución, cuándo adaptarse a cambios externos, etc. y no siempre están alineados.
- ***Commiters* / desarrolladores**: los desarrolladores de cada proyecto son diferentes.

Por este motivo, realizar una **instalación** de toda una plataforma Hadoop con sus componentes asociados de forma independiente (lo que se denomina Hadoop Vanila) resulta muy complicado. Por ejemplo, al instalar la versión X de Phoenix necesitas la versión Y de HBase, pero otro componente (Hive, por ejemplo), requiere la versión Z de HBase.

La misma dificultad ocurre para la **resolución de incidencia**s que puedan ocurrir en la plataforma cuando se ejecuta en producción. Es decir, si has conseguido poder instalar todos los componentes y que no haya fallos de configuración o dependencias entre ellos (¡enhorabuena!), es bastante normal que puedan ocurrir errores cuando la plataforma está usándose en producción (algún servidor se queda sin espacio en disco, por ejemplo, o simplemente hay un fallo al ejecutar un determinado trabajo). Si la plataforma se está ejecutando en producción y está dando soporte a las operaciones de una empresa, el

fallo debe corregirse lo antes posible, y no es tarea fácil porque hay que buscar en muchos componentes para averiguar dónde puede estar el fallo.

Para solventar las dos dificultades mencionadas, surgen las **distribuciones comerciales de Hadoop**, que contienen en un único paquete la mayor parte de componentes del ecosistema, resolviendo dependencias, añadiendo incluso utilidades, e incorporando la posibilidad de contratar soporte empresarial 24x7. Es decir, una distribución comercial ofrece:

- Un "instalador" de toda la plataforma, simplificando enormemente el proceso de instalación y despliegue de la plataforma.
- Un servicio de soporte 24x7 para resolver todas las incidencias que puedan aparecer en la plataforma en producción.
- Documentación más completa que la que se puede encontrar en los proyectos Apache.

Las principales distribuciones que aparecieron son:

- **Cloudera**: fue la primera distribución en salir al mercado (2009) y la que ha tenido un mayor número de clientes. Utiliza la mayor parte de componentes de Apache, en algún caso realizando algunas modificaciones, y añade algún componente propietario (Cloudera Manager, Cloudera Navigator, etc.).
- **Hortonworks**: surgió en 2012 y es una distribución que contiene, sin ninguna modificación, los componentes originales de Apache.
- **MAPR**: rehizo la mayor parte de componentes utilizando los mismos interfaces pero reimplementando el *core* para ofrecer un mayor rendimiento.

## Debes conocer

En octubre de 2018 se anunció la fusión de las dos principales distribuciones, Cloudera y Hortonworks, que se hizo efectiva durante 2019,

lo que generó un movimiento sin precedentes en el mercado de las tecnologías Big Data.

Asimismo, en 2019, MAPR, ante la imposibilidad para obtener fondos que financiaran su actividad, cesó su actividad, vendiendo todo su portfolio a HPE.

Con estos cambios, la única distribución libre preinstalada de Hadoop que hubo durante los siguientes años fue la de Cloudera, que integra las funcionalidades relevantes de Hortonworks (versión HDP).

Recientemente ha dejado de ofrecer su versión gratuita de Hadoop (HPD), añadiéndole nuevas funcionalidades y pasando a ser de pago (versión CPD).

## Reflexiona

¿Por qué crees que en un mercado que se supone que tiene suficiente volumen con el auge de Big Data o la Inteligencia Artificial, sólo hay una distribución Hadoop disponible?

La respuesta está en el auge del *cloud*.

Los principales proveedores de *cloud*, es decir, Amazon (AWS), Microsoft (Azure) y Google (Google Cloud), han lanzado y potenciado servicios de Hadoop gestionados, es decir, ofrecen la posibilidad de desplegar plataformas Hadoop en modalidad pago por uso.

La mayor parte de las empresas están inmersas en procesos de migración a entornos *cloud*, por lo que cada vez más utilizan servicios de esos proveedores, desmantelando infraestructura locales (*on-premise*, en su propio CPD). Las plataformas Hadoop no son una excepción. Por este motivo, la principal competencia de Cloudera no es otro fabricante de distribuciones Hadoop, sino los principales proveedores de cloud.

Además de las distribuciones mencionadas, es necesario añadir las soluciones Hadoop-as-a-Service de los proveedores de *cloud*:

- Amazon Elastic Map Reduce (EMR).
- Microsoft Azure HDInsight (y evoluciones).
- Google Dataproc.

Estas soluciones permiten levantar infraestructuras elásticas en pocos minutos en modalidad pago por uso, con un coste aproximado es de 0,25 - 2 € por nodo y hora.

Estas soluciones aportan algunas ventajas muy interesantes:

- **Reducen considerablemente el tiempo de aprovisionamiento** (instalación, configuración y despliegue) de infraestructuras Hadoop, de meses en el caso de instalaciones en la propia infraestructura de las empresas, a minutos en un

proveedor *cloud*. Las empresas se encuentran inmersas en procesos de transformación digital donde prima lo que se conoce como el *time-to-market*, es decir, la rapidez para lanzar nuevas soluciones.

- Ofrecen **elasticidad**, es decir, cuando lanzas una plataforma Hadoop en la nube, si necesitas más capacidad o potencia, el proceso de escalar o incrementar el tamaño de la infraestructura es muy sencilla, y lo mismo ocurre si deseas reducir el tamaño de la plataforma.
- Ofrecen **pago por uso**: el coste suele ser en número de servidores por las horas que están levantados, por lo que por un lado no requiere una inversión inicial importante (comprar las máquinas, contratar el soporte por un año como mínimo, contratar a una empresa especialista para la instalación, etc.), y por otro, se paga sólo por el tamaño de la plataforma, que como hemos visto, puede adecuarse a la necesidad real en cada momento (elasticidad).

En resumen, las principales ventajas son una reducción del riesgo (no hay inversión inicial) y un incremento de la agilidad.

Sin embargo, estas soluciones *cloud* presentan algunas desventajas:

- Se produce un efecto que se denomina *vendor lock-in*, es decir, la barrera para salir de una solución *cloud* a otra de otro fabricante *cloud* o a un Hadoop propio, es elevada. Por ejemplo, los proveedores *cloud* aplican un cargo por sacar los datos fuera de su entorno
- Las soluciones que ofrecen no suelen ser estándar, sino adaptaciones de Hadoop que han realizado los proveedores.
- El coste puede ser mucho más elevado y, de hecho, difícilmente se conoce a priori al utilizar fórmulas de cálculo de los costes que añaden a veces variables que no se pueden estimar (por ejemplo, el consumo de CPU que vamos a tener).

## Debes conocer

No te preocupes por la diferencia que pueda haber entre las distribuciones Hadoop, la versión de Hadoop "estándar" o los servicios Hadoop en la nube.

En la mayor parte de las actividades, es suficiente conocer Hadoop "estándar" ya que el resto de alternativas apenas difieren y lo aprendido en una sirve perfectamente para otra.