UCAB UNIVERSIDAD CATÓLICA ANDRÉS BELLO

HADOOP, SUS COMPONENTES, ECOSISTEMA Y DISTRIBUCIONES

OBJETIVO

Conocer el origen de *Hadoop*, cómo es su ecosistema y qué son sus distribuciones

TABLA DE CONTENIDO

INTRODUCCIÓN

Hadoop es un marco de trabajo para *big data* que surgió gracias a dos desarrollos importantes de Google, como son Google File System y Map Reduce. A partir de este surgieron un conjunto de aplicaciones que cubren diferentes necesidades en el área de manejo y procesamiento de grandes volúmenes de datos, a las cuales se les ha llamado el ecosistema de Hadoop y a través de los años se han convertido en el conjunto de herramientas para *big data* más usado.

El informe de *Market Hadoop Big Data Analytics* indica que el tamaño del mercado de Hadoop se valoró en $ 26,74 mil millones en 2019 y se proyecta que alcance los $ 340,35 mil millones en 2027, creciendo a una tasa compuesta anual del 37,5 % de 2020 a 2027 (Yogendra, Khan, Borasi y Kumar, 2022). Por eso es importante conocer qué es Hadoop, su ecosistema y las distribuciones más importantes en el mercado.

A medida que la web creció a finales del siglo XX y principios del 2000, se crearon **motores de búsqueda** e índices para ayudar a localizar información relevante en medio del contenido basado en texto. En los primeros años, los resultados de las búsquedas los devolvían realmente los humanos, pero a medida que la web creció de docenas a millones de páginas se necesitó automatización.

Uno de esos proyectos fue un motor de búsqueda web de código abierto llamado **Nutch**, una creación de los ingenieros Doug Cutting y Mike Cafarella. En el año 2002 estaban tratando de resolver un problema que no era nada sencillo: indexar todas las páginas web de internet, y para poder alimentar su buscador, los creadores necesitaban descargar el mayor número posible de páginas, ya que luego esto les permitiría procesarlas e indexar las palabras de cada una. ¿Cómo abordarías ese problema?

Doug Cutting

Mike Cafarella

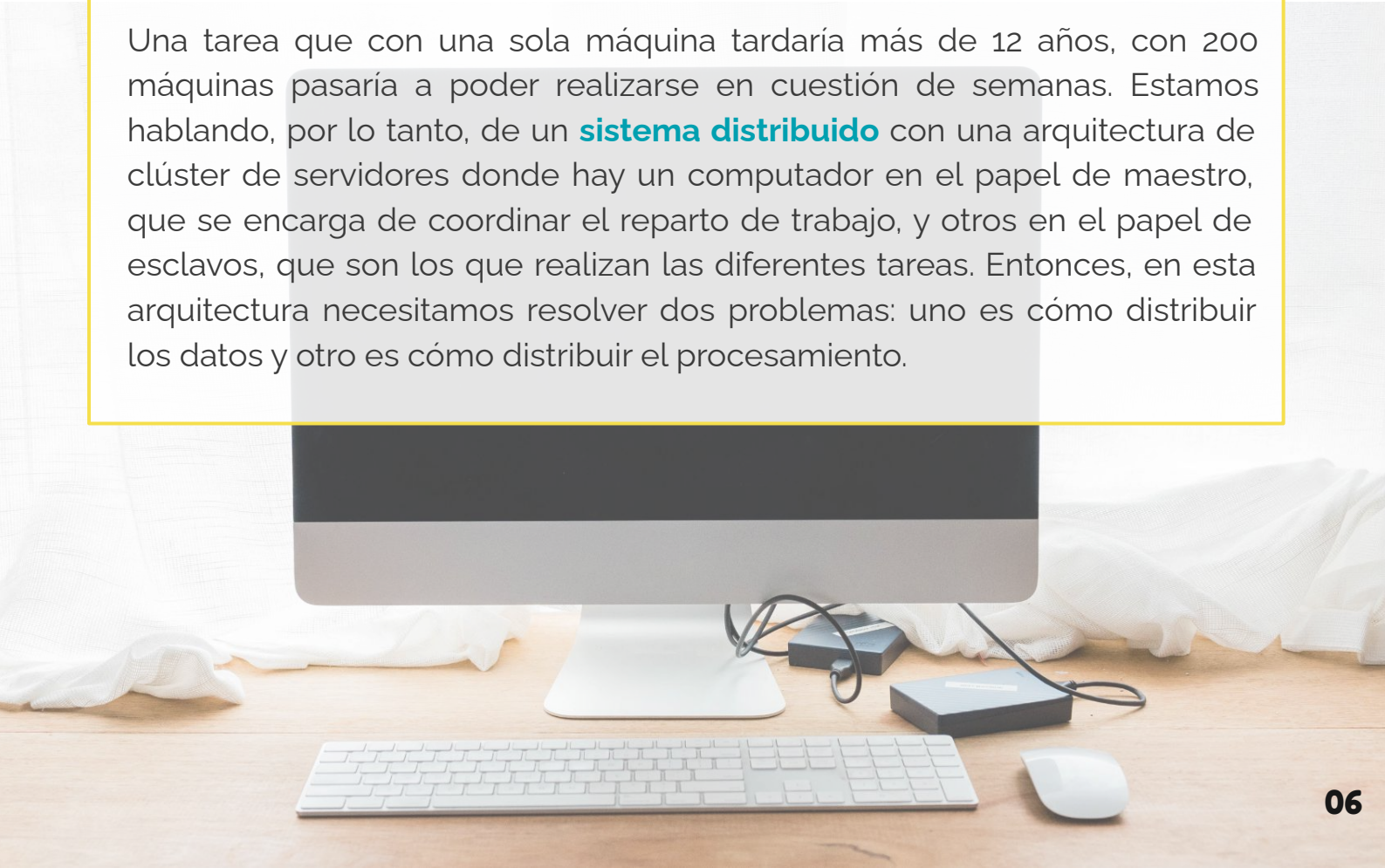
La aproximación más simple al problema que proponemos sería descargar todas las páginas y guardarlas antes de pasar a procesarlas. Para ver si esto podría ser viable necesitaríamos estimar cuántos discos duros harían falta. Hay fuentes que indican que en internet existen alrededor de unos dos mil millones de sitios web; vamos a considerar que cada uno de estos sitios tiene una media de unas 10 páginas cada uno y que cada página ocupa en torno a medio *megabyte* aproximadamente. Si multiplicamos todo esto vemos que necesitaríamos unos **10.000 discos duros de 1Tb cada uno**, para guardar toda esta información, lo que equivaldría aproximadamente a unas 100 salas llenas de esta herramienta.

Ahora, vamos a suponer que hemos descargado toda esa información y ya la tenemos disponible. A continuación, necesitaríamos **procesarla** siguiendo con esta primera aproximación lo más simple posible: imaginemos que tratamos de hacer todo el procesamiento con un único computador, ¿cuánto tiempo nos llevaría procesar esos 10.000 *terabytes*? Precisamente esos 10.000 discos duros que mencionamos.

A la hora de calcular el tiempo de procesamiento total, vamos a partir de que necesitamos dos accesos: uno a un disco duro para leer y a otro para escribir, y vamos a asumir también que cada acceso tarda unos 9000 segundos. En este caso, podemos deducir que necesitaríamos unos 20 milisegundos en total para leer, procesar y escribir los resultados. Pues bien, procesar toda esa información en un único computador **nos llevaría cerca de unos 12 años**. Queda claro, entonces, que esta primera aproximación, basada en utilizar una sola máquina, resulta totalmente inviable. Necesitamos buscar otro tipo de solución.

La opción de usar un solo computador para esta tarea no es posible, pero si conseguimos que **múltiples computadores** hagan el procesamiento en paralelo, podemos reducir considerablemente el tiempo necesario.

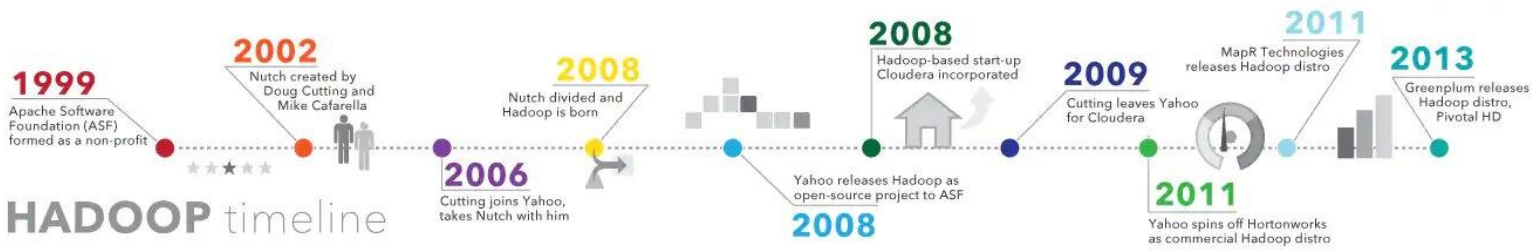
Una tarea que con una sola máquina tardaría más de 12 años, con 200 máquinas pasaría a poder realizarse en cuestión de semanas. Estamos hablando, por lo tanto, de un **sistema distribuido** con una arquitectura de clúster de servidores donde hay un computador en el papel de maestro, que se encarga de coordinar el reparto de trabajo, y otros en el papel de esclavos, que son los que realizan las diferentes tareas. Entonces, en esta arquitectura necesitamos resolver dos problemas: uno es cómo distribuir los datos y otro es cómo distribuir el procesamiento.



Durante este tiempo estaba en marcha otro proyecto de motor de búsqueda llamado **Google.** Se basaba en el mismo concepto: almacenar y procesar datos de forma distribuida y automatizada para que los resultados de búsqueda web relevantes pudieran obtenerse más rápido. Es así como en el 2003 Google publica un artículo sobre su sistema de archivos distribuidos, donde soluciona el tema de distribuir los datos, y en 2004 publica el artículo "Map Reduce", en el cual se trata el tema del procesamiento distribuido.

En 2006, Cutting se unió a Yahoo y se llevó consigo el proyecto **Nutch**, así como ideas basadas en estos trabajos de Google acerca de la automatización del almacenamiento y procesamiento de datos distribuidos. En ese momento el proyecto Nutch se dividió: la parte del rastreador web permaneció como Nutch y la porción de computación y procesamiento distribuidos se convirtió en **Hadoop,** el cual fue llamado así por el elefante de juguete del hijo de Cutting.

En 2008, Yahoo lanzó Hadoop como un **proyecto de código abierto** y puede decirse que alcanza la madurez entrando en la Fundación Apache como proyecto de código abierto. Hoy en día Hadoop sigue en la fundación y desde su página web podríamos descargar el código e incluso hacer nuestras propias contribuciones para mejorarlo, y es que una de las grandes ventajas de este *software* es precisamente su carácter de código abierto. A continuación, se muestra su línea de tiempo:



Fuente: Hadoop Timeline. Extraída de SAS (s.f.)

Estos son algunos de los **hitos** más importantes en el desarrollo de Hadoop antes de ser formalmente un proyecto de código abierto:

Fuente: Hitos de Hadoop. Adaptado (s.f.)

Por consiguiente, Hadoop es un marco de trabajo **(*framework*)** que permite el procesamiento distribuido de grandes conjuntos de datos a través de clústers de computadoras, utilizando modelos de programación sencillos. Está diseñado para escalar desde simples servidores a miles de máquinas, cada una ofreciendo computación local y almacenamiento. En lugar de depender del *hardware* para entregar alta disponibilidad, la biblioteca por sí misma está diseñada para detectar y manipular fallos en la capa de aplicación, de modo que entrega un servicio de alta disponibilidad sobre la parte superior de un clúster de computadoras, cada una de las cuales puede ser propensa a fallos (Aguilar, 2019).

1. Características

Dentro de las **características** más importantes de Hadoop, se tienen:

- **Código abierto:**

El desarrollo de Hadoop está a cargo de la comunidad gobernada por la licencia de la Apache Software Foundation. Se puede hacer más efectivo a Hadoop mediante la adición de características, corrección de fallos de *software*, mejora del rendimiento o de la escalabilidad.

- **Almacenamiento y procesamiento distribuido:**

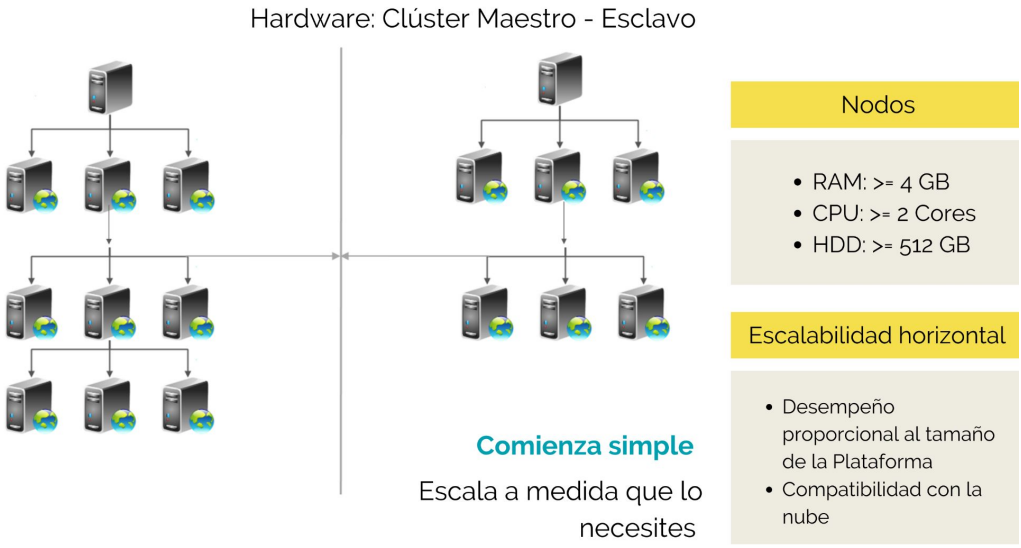
A medida que los datos se almacenan de forma distribuida a través del clúster, estos se procesan de manera distribuida y en paralelo.

- **Amplia variedad de herramientas:**

Por ser de código abierto se han desarrollado muchas herramientas que se ejecutan en Hadoop, orientadas a una funcionalidad o problema específico dentro de un sistema de *big data.*

- **Escalabilidad:**

Es altamente escalable en la forma en que se puede agregar nuevo *hardware* a los nodos (vertical), pero principalmente proporciona escalabilidad horizontal, agregando nuevos nodos al clúster.



Fuente: Clúster Hadoop. Adaptado (s.f.)

Hadoop se ejecuta en *hardware* de **bajo costo comercial** y reduce considerablemente el costo frente a otras alternativas comerciales de procesamiento y almacenamiento de datos. Es una herramienta básica de los gigantes de internet, incluyendo Facebook, Twitter, eBay, eHarmony, Netflix, AOL, Apple, FourSquare, Hulu, LinkedIn, Tuenti, entre otros. También es utilizada por grandes empresas tradicionales del mundo de las finanzas o del comercio como JPMorgan, Chase o Walmart (Aguilar, 2013)

El proyecto está compuesto por los siguientes **módulos**:

HDFS Commom	HDFS
MapReduce	YARN

Fuente: Componentes de Hadoop. Adaptado (s.f.)

1. Hadoop Common:

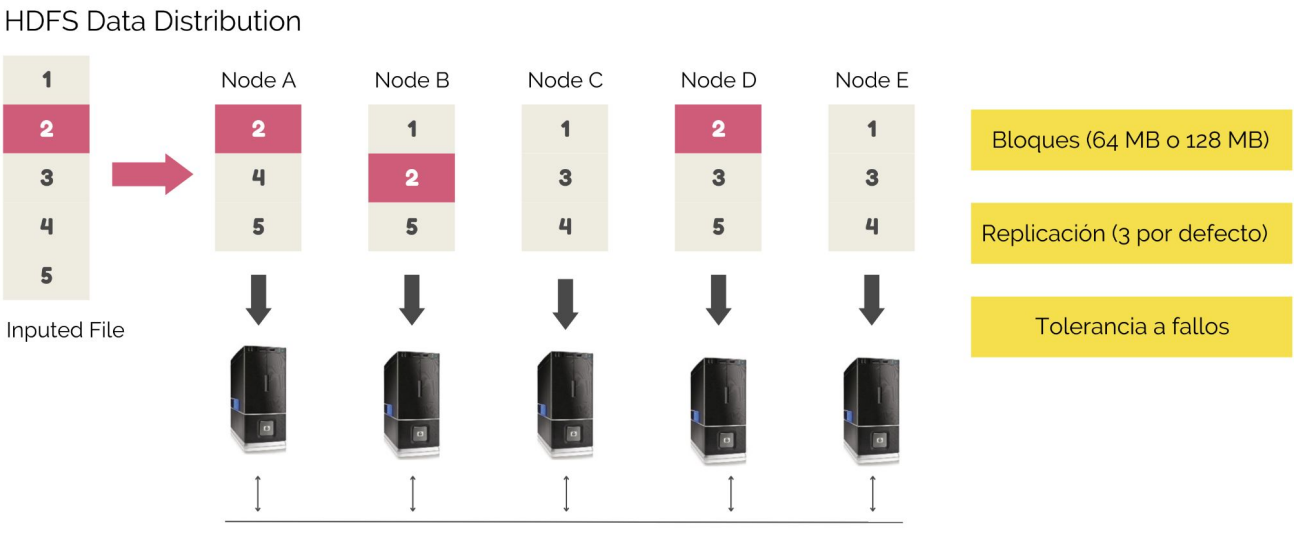
Consiste en la **colección** de bibliotecas y utilidades comunes que soportan los otros módulos para la ejecución del *framework* (Apache Hadoop, 2017). También facilita código fuente y documentación, así como una sección de contribuciones que incluye diferentes proyectos de la comunidad de Hadoop.

2. Hadoop Distributed File System (HDFS):

Este componente es el sistema de archivos distribuidos que está diseñado para **almacenar** de forma fiable grandes conjuntos de datos y proporciona acceso de alto rendimiento a los datos de aplicación de usuario. HDFS es altamente tolerante a fallos y permite la ejecución de tareas de aplicación de usuario en un clúster grande (de 10, 100, 1000 o más nodos) que está conformado por *hardware* de bajo costo (Apache Hadoop, 2017).

HDFS trabaja de la siguiente manera:

- Al recibir un archivo, lo divide en bloques de 64 o 120 MB, de acuerdo a la configuración previa
- Esos bloques son almacenados de manera distribuida en los nodos del clúster con un factor de replicación de 3
- Cuando se lee el archivo, se recupera leyendo cada bloque de alguna de sus réplicas.



Fuente: HDFS Data Distribution. Adaptación (s.f.)

Dentro de sus **características** tenemos:

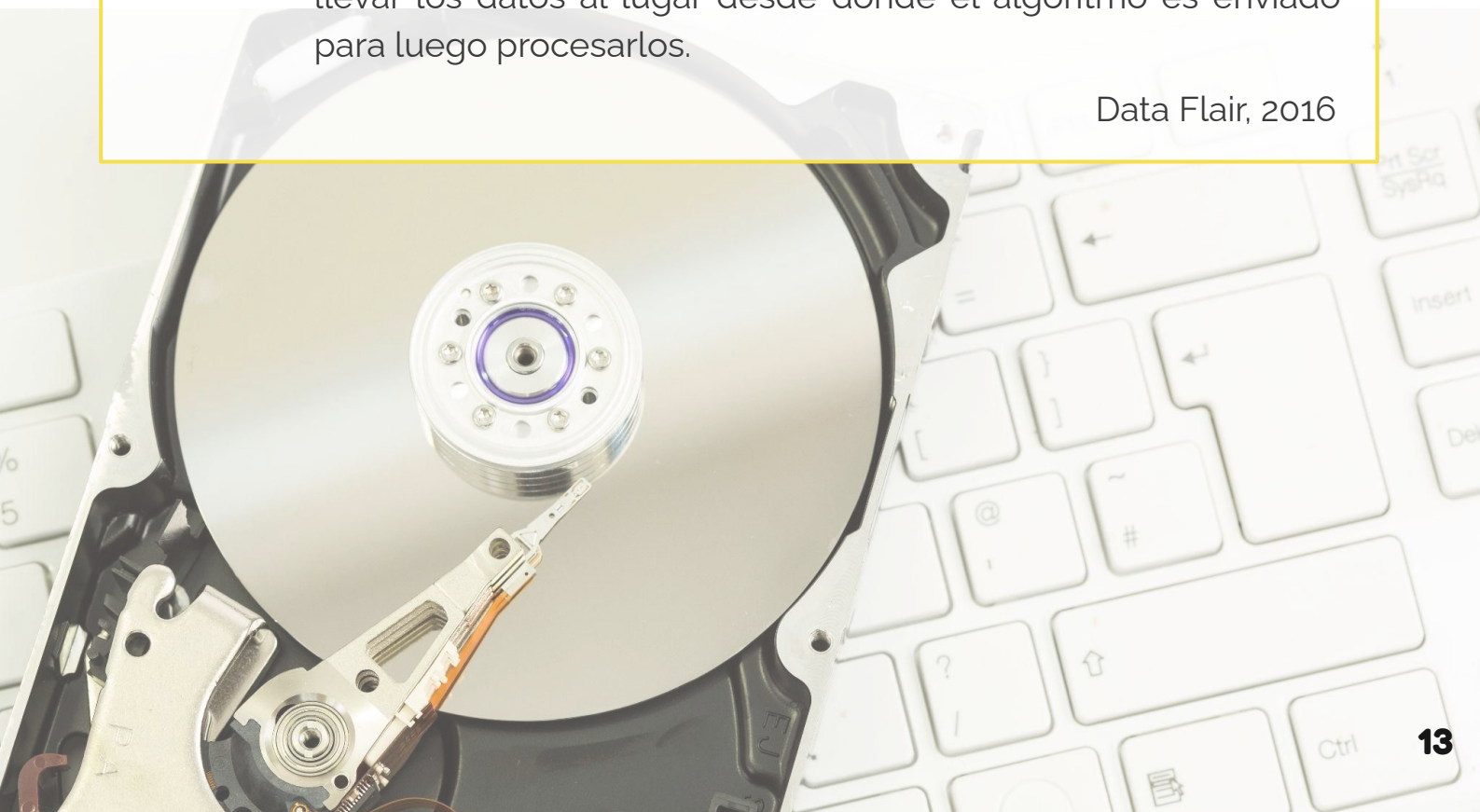
1. **Almacenamiento:** se pueden almacenar archivos muy grandes que pueden ser incluso más grandes que el tamaño de un solo disco, ya que el archivo a almacenar se divide en bloques y se distribuye a través de varios nodos.

2. **Procesamiento distribuido:** a medida que los datos se almacenan de forma distribuida en HDFS, a través del clúster, estos se procesan en paralelo.

3. **Tolerancia a fallos y disponibilidad:** de forma predeterminada se realizan 3 réplicas de cada bloque que son almacenadas en el clúster y que pueden ser modificadas según sea necesario. Así que, si cualquier nodo se cae, los datos de ese nodo se pueden recuperar fácilmente desde otros nodos.

4. **Localidad de datos:** HDFS trabaja en el principio de localidad de datos que establece mover la computación a los datos en lugar de los datos a la computación. Por ejemplo, cuando un cliente envía el algoritmo MapReduce, este algoritmo es movido a los datos que se encuentran en el clúster en lugar de llevar los datos al lugar desde donde el algoritmo es enviado para luego procesarlos.

Data Flair, 2016



3. MapReduce:

Es un **modelo de programación** (originalmente creado por Google) que permite escribir con facilidad, aplicaciones que procesan en paralelo grandes cantidades de datos, tanto estructurados como no estructurados, en muchos nodos (Apache Hadoop, 2017; Hortonworks, s.f.; White, 2015).

MapReduce consiste en varias **funciones** que están definidas con respecto a duplas del tipo clave/valor:

Mapper:

Recibe una lista de valores o pares clave/valor y devuelve una lista de pares clave/valor. Esta función realiza el mapeo que se aplica a cada elemento de la entrada de datos, obteniendo una lista de pares clave/valor por cada vez que se ejecuta la función Map. Luego se agrupan todos los pares con la misma clave de todas las listas, creando un grupo por cada una de las diferentes claves generadas. No hay requisito de que el tipo de datos para la entrada coincida con la salida y no es necesario que las claves de salida sean únicas.

3. MapReduce:

Reducer:

Se encarga de reducir un conjunto de valores intermedios que comparten una clave con un conjunto de valores más pequeños. El número de tareas que reduce son establecidas por el usuario. Esta función tiene tres **fases primarias** (Apache Hadoop, 2017):

- ***Shuffle*:** la entrada que recibe la función *Reducer* es la salida ordenada de la función *Mapper*. En esta fase el *framework* busca la partición relevante de la salida de todos los mappers, vía HTTP.

- ***Sort*:** en esta etapa el *framework* reúne las entradas de la función *Reducer* por claves.

- ***Secondary Sort*:** si se requiere que las reglas de equivalencia que agrupan las claves intermedias sean diferentes de aquellas que agrupan claves antes de realizar la reducción, entonces se puede especificar un comparador que también puede ser usado para controlar cómo deben ser agrupadas las claves intermedias, las cuales pueden ser utilizadas en conjunto para simular un ordenamiento secundario en los valores.

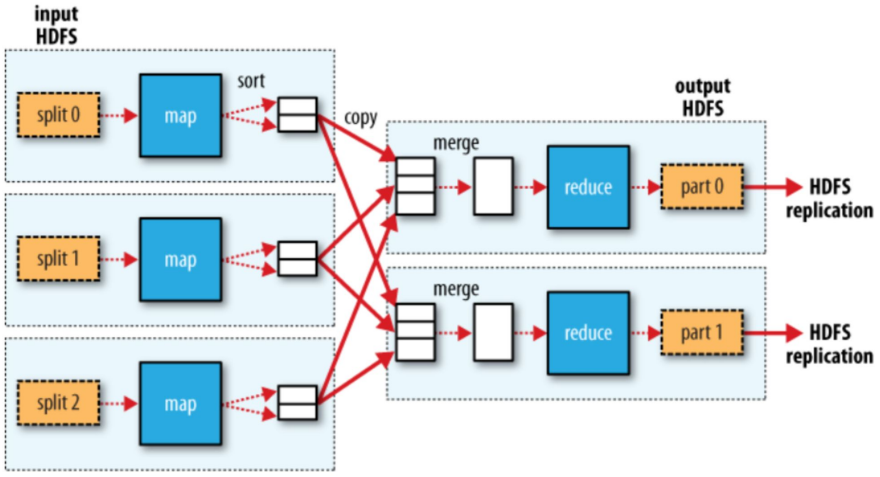
- ***Reduce*:** en esta fase se llama al método *Reduce* para cada par clave/lista de valores en las entradas agrupadas. La salida de la tarea *Reduce* está escrita en los sistemas de archivo y no es ordenada.

- ***Partitioner*:** controla el particionamiento de las claves en las salidas intermedias de la función *Map*. La clave (o un subconjunto de la clave) se utiliza para derivar la partición que por lo general se realiza con una función hash. El número total de particiones es el mismo que el número de tareas *Reduce* para el *job* (es la interfaz principal para que un usuario describa un trabajo de MapReduce para su ejecución en el *framework* Hadoop).

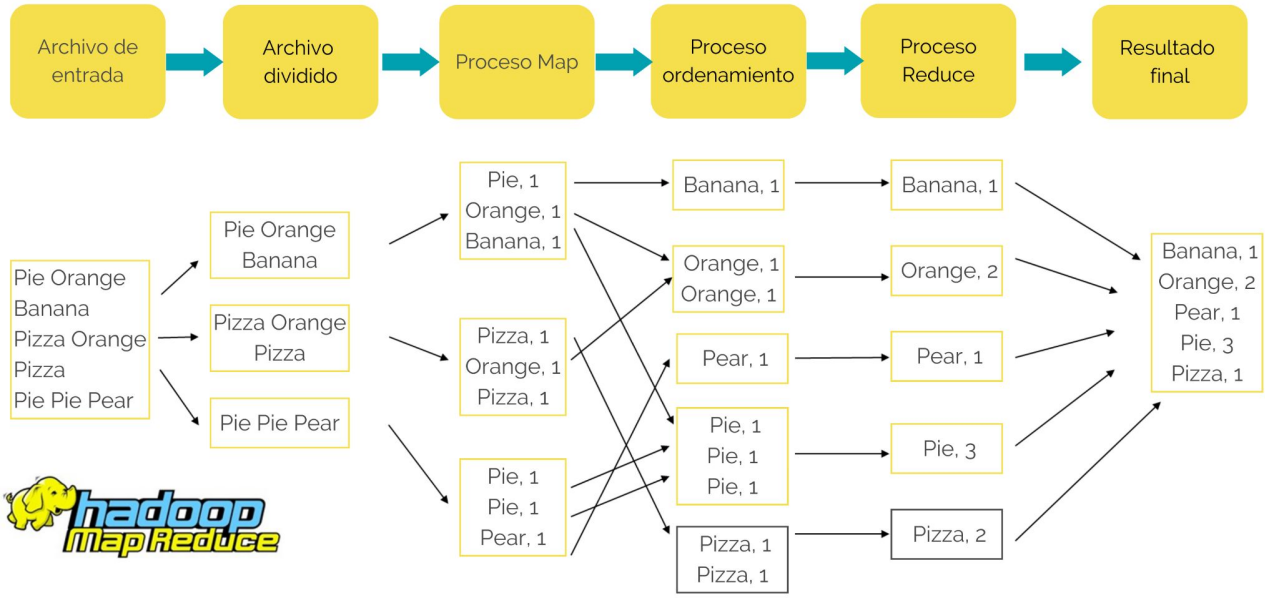
3. MapReduce:

Counter:

Permite que las aplicaciones que están escritas en MapReduce puedan generar **estadísticas.**



Fuente: Funcionamiento de MapReduce. Extraída de White (2015)

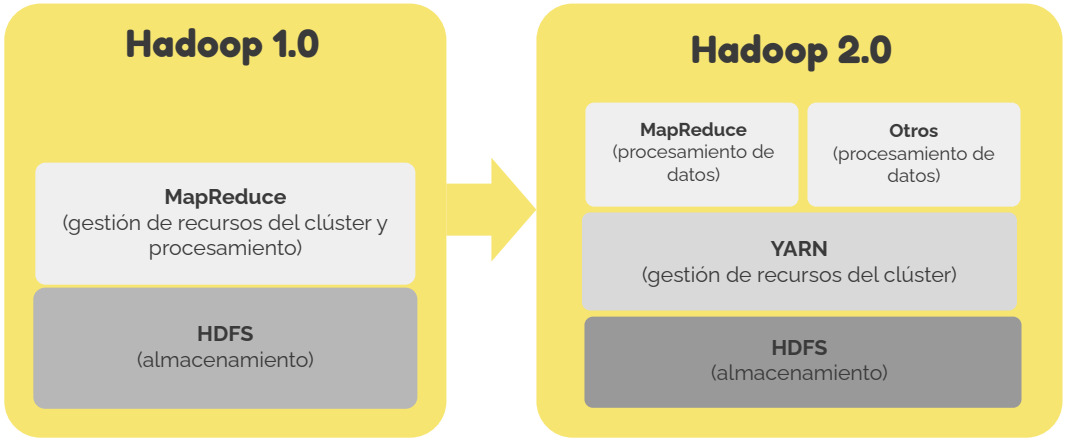


Fuente: Ejemplo: Conteo de palabras. Elaboración propia (s.f.)

3. MapReduce:

Counter:

Estos eran los **componentes** hasta la versión 1.0, donde MapReduce, además del procesamiento paralelo, se encargaba de la gestión de recursos, lo que sobrecargaba su función. En la versión 2.0 se agregó un componente adicional que se encargaba de esto, llamado Yarn.



Fuente: Hadoop 1.0 vs Hadoop 2.0. Adaptado (s.f.)

4. Yet Another Resource Negotiator (YARN):

Apache YARN (por sus siglas en inglés, *Yet Another Resource Negotiator*) es un **sistema de gestión de clústeres.** Se introdujo en Hadoop 2 para mejorar la implementación de MapReduce (White, 2015). Se puede describir como un gestor de recursos rediseñado y caracterizado como un sistema operativo distribuido a gran escala para aplicaciones de *big data* (Vaughan, 2021)

YARN **desacopla** las capacidades de gestión de recursos y planificación de MapReduce del componente de procesamiento de datos, permitiendo a Hadoop soportar enfoques más variados de procesamiento y una gama más amplia de aplicaciones.

Las aplicaciones de YARN **facilitan** solicitar y trabajar con recursos del clúster, pero estas funciones no suelen usarse directamente por código de usuario. En su lugar, los usuarios escriben en aplicaciones de nivel superior que son proporcionadas por *frameworks* de computación distribuida, los cuales son desarrollados en YARN y ocultan los detalles de gestión de recursos al usuario.

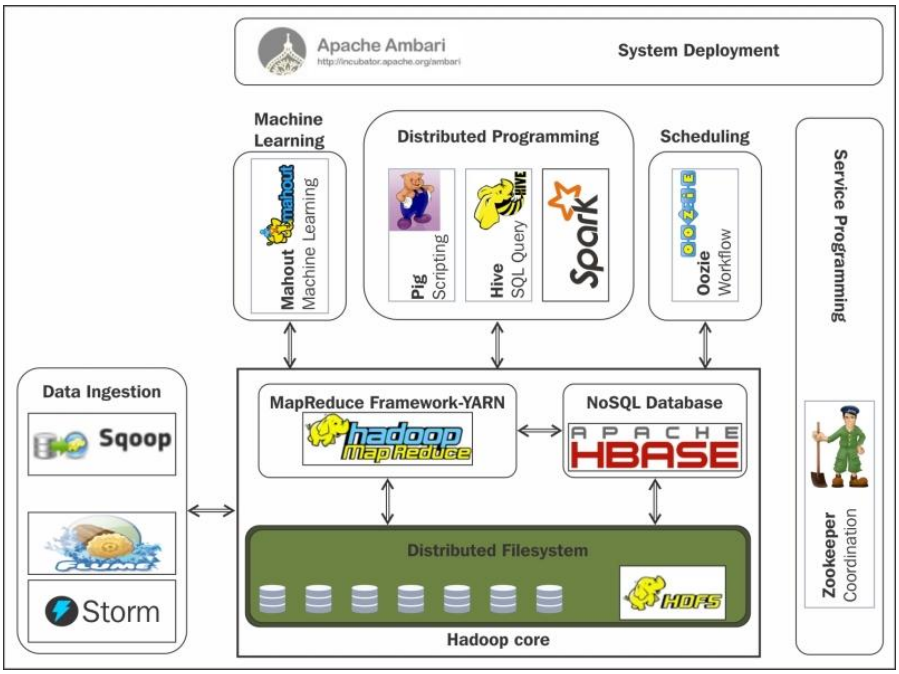
Se puede afirmar que Hadoop no es un solo proyecto, sino más bien un complejo **ecosistema de proyectos** muy diversos que trabajan a la par. Su objetivo es crear un conjunto común de servicios capaces de transformar lo que llamamos *commodity hardware* (*hardware* de bajo coste, sin capacidad de redundancia) en un servicio coherente que permita almacenar de forma redundante *petabytes* de datos y procesarlos eficientemente.

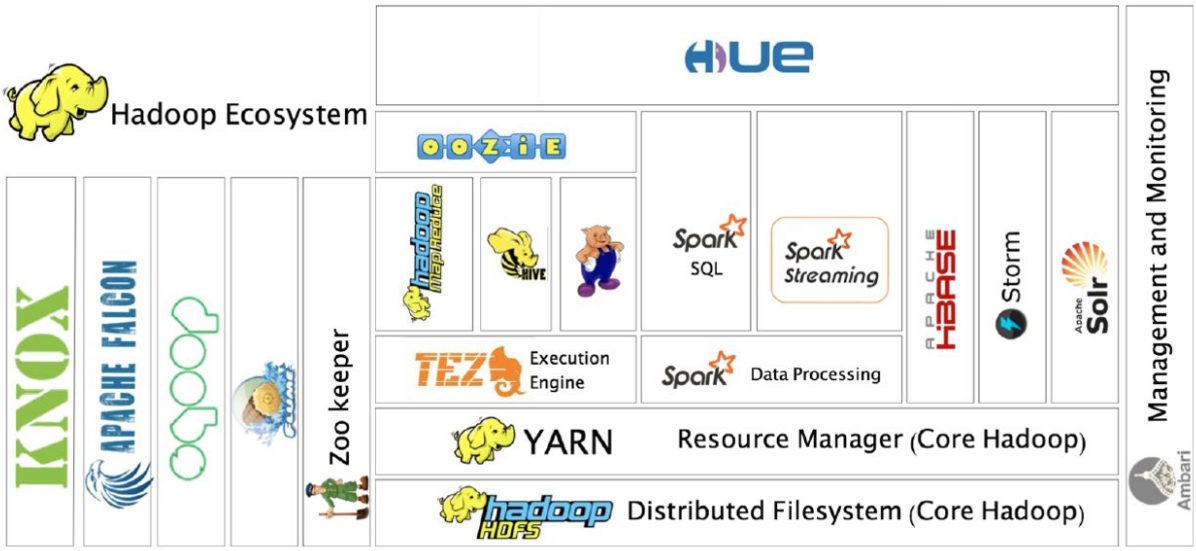
Aunque comenzó como proyecto individual, poco a poco se fueron sumando distintos proyectos, abarcando **áreas** como:

- Plataforma de almacenaje y procesamiento de datos
- Lenguajes de *scripting*
- Bases de datos
- Herramientas analíticas
- Lenguaje *query*
- Gestión de *workflow*
- Entre otros.

Muchos de los **componentes** de la pila Hadoop son proyectos *open source* de la Fundación Apache. Otros han sido creados de forma propietaria por empresas que han comercializado diferentes versiones "empaquetadas" de Hadoop (como Cloudera, MapR, Hortonworks, etc.).

En la siguiente figura puedes ver algunos de los **proyectos** más conocidos del ecosistema Hadoop:





Fuente: Proyectos del ecosistema Hadoop. Extraída de Recuero de los Santos (2017)

Como se puede observar en las figuras anteriores, las herramientas se agrupan de acuerdo a la **funcionalidad** asociada a la cadena de valor del dato, tales como adquisición o ingesta de datos, almacenamiento y tratamiento de los datos, análisis y visualización de datos, además de herramientas para el flujo de trabajo y administración del clúster. A continuación describiremos algunas de ellas:

Para ingestión de datos, algunas de las herramientas más usadas son:

1. **Scoop:** permite ejecutar aplicaciones MapReduce que introducen o extraen información de bases de datos SQL (por tanto, estructuradas).

2. **Flume:** sirve para introducir datos en *streaming* en Hadoop. Si tenemos servidores que generan datos de forma continua, se puede usar Flume para almacenarlos en HDFS (pueden ser datos semiestructurados o no estructurados).

3. **Kafka:** tiene como objetivo proporcionar una plataforma unificada, de alto rendimiento y de baja latencia, para la manipulación en tiempo real de fuentes de datos. Puede verse como una cola de mensajes, bajo el patrón publicación-suscripción, masivamente escalable y concebida como un registro de transacciones distribuidas.

4. **Storm:** componente encargado de procesar flujos de datos en tiempo real. Su uso suele ir acompañado de Apache Kafka.

Para la coordinación se tiene:

Zookeper: es un servicio de coordinación para aplicaciones distribuidas, responsable de mantener la información de configuración y ofrecer coordinación de manera distribuida, simplificando el desarrollo de aplicaciones distribuidas. ZooKeeper está siendo utilizado por algunos de los proyectos de Apache como HBase para ofrecer alta disponibilidad y alto grado de coordinación en un entorno distribuido.

Para el almacenamiento se tienen las bases de datos:

1. **HBASE:** tiene un motor de procesamiento en memoria que le agiliza enormemente las operaciones de lectura-escritura sobre Hadoop, permitiendo así trabajar con datos en *streaming*. También permite trabajar con bases de datos noSQL. Se puede acceder a HBase desde Hive, Pig y MapReduce y usa HDFS para almacenar la información. Por tanto, es completamente tolerante a fallos. Se usa, por ejemplo, en los mensajes de Facebook. Almacena parte de sus metadatos en Zookeeper.

2. **HCatalog:** es un proyecto que sacó los metadatos de Hive para que también se pudiera acceder a ellos desde Pig y MapReduce. Es un servidor de metadatos con algunas mejoras. HCatalog puede acceder a los datos en el estándar HDFS o bien en HBase.

3. **Impala:** es una base de datos SQL sobre Hadoop. Proporciona la capacidad de realizar consultas concurrentes y de baja latencia para analítica y *Business Intelligence* (BI).

Para el procesamiento se tiene:

1. **Hive:** proporciona la interfaz SQL a Hadoop necesaria para convertirse en un *Data Warehouse*. Transforma las consultas SQL en trabajos MapReduce sobre datos estructurados. No es apropiado para realizar consultas de baja latencia.

2. **Pig:** es un lenguaje de alto nivel que traduce a MapReduce. Convierte una descripción de alto nivel de cómo deben ser procesados los datos en *jobs* de MapReduce, sin necesidad de tener que escribir largas cadenas de *jobs* cada vez, mejorando notablemente la productividad de los desarrolladores.

3. **Spark:** motor de procesamiento en memoria compatible con HDFS. Aumenta la velocidad de MapReduce en 100 veces. Soporta aplicaciones ETL, *Machine learning* y *streaming* de datos, así como consultas SQL.

4. **Tez:** *framework* de programación de flujos de datos. Es la evolución de MapReduce que ejecuta sobre Yarn, optimizando el código para alcanzar mejoras de hasta 10 veces en el rendimiento. Muchas tecnologías están adoptando Tez como motor de ejecución principal.

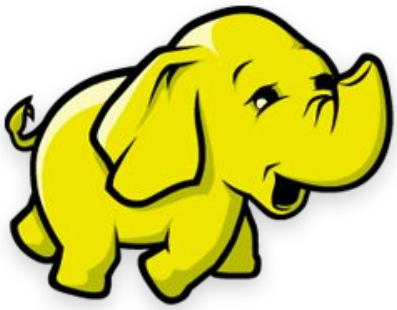
5. ***Hue*** (*Hadoop User Experience*): es la interfaz web para la gestión de Hadoop. Permite crear tablas en Hive, realizar consultas, navegar el sistema de archivos, cambiar permisos y propietarios. También puede diseñar *jobs* de MapReduce y conocer su estado. Además, posibilita llevar a cabo la gestión por parte de los administradores de las cuentas de usuario.

Para la planificación se cuenta con:

Oozie: es un gestor de *workflow*. Te permite definir cuándo quieres que tus *jobs* MapReduce se ejecuten, ya sea de forma programada o cuando haya disponibles nuevos datos.

Para la administración tenemos:

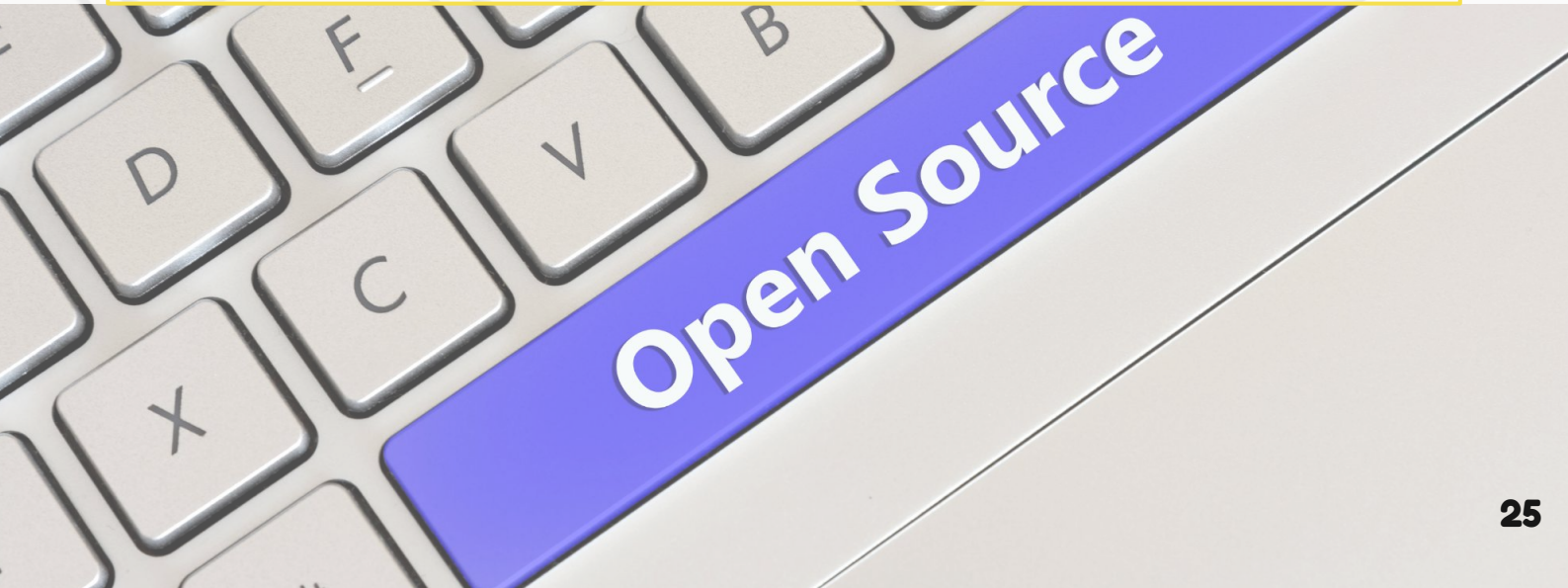
Apache Ambari: es un *framework* de código abierto para el aprovisionamiento, la gestión y la supervisión de clústeres Hadoop. Es útil para instalar servicios Hadoop a través de diferentes nodos del clúster y gestionar la configuración de Hadoop Services en el clúster. Ofrece un panel de control para supervisar la salud general del clúster y, además, posee alertas y mecanismo de correo electrónico para obtener la atención necesaria cuando se requiera. Aunado a esto, Ambari brinda las API REST a los desarrolladores para la integración de aplicaciones.



Desde el lanzamiento de Hadoop en **2011**, ha crecido rápidamente en popularidad y ha emergido un gran ecosistema de distribuidores, vendedores y consultores, con el objetivo de dar soporte a la industria.

Hadoop es un sistema *open source* **(gratuito)**, disponible para cualquiera que lo desee utilizar. Sin embargo, las empresas necesitan alinear las soluciones de Hadoop con sus necesidades para el desarrollo de las soluciones específicas que mejor se adapten a ellas. Por estas razones, las distribuciones comerciales vienen empaquetadas para resolver las necesidades de gestión de datos y soluciones de analítica.

Apache Hadoop es la plataforma de *software* de código abierto de mayor impacto en *big data* pero, como sucede con otras soluciones de *software* abierto, no suele ofertarse con soporte de productos. Por esta razón, han surgido un gran número de vendedores que han lanzado sus propias distribuciones de Apache Hadoop. La mayoría de las empresas que han desplegado Hadoop para uso comercial han seleccionado alguna de las distribuciones comerciales de Hadoop.

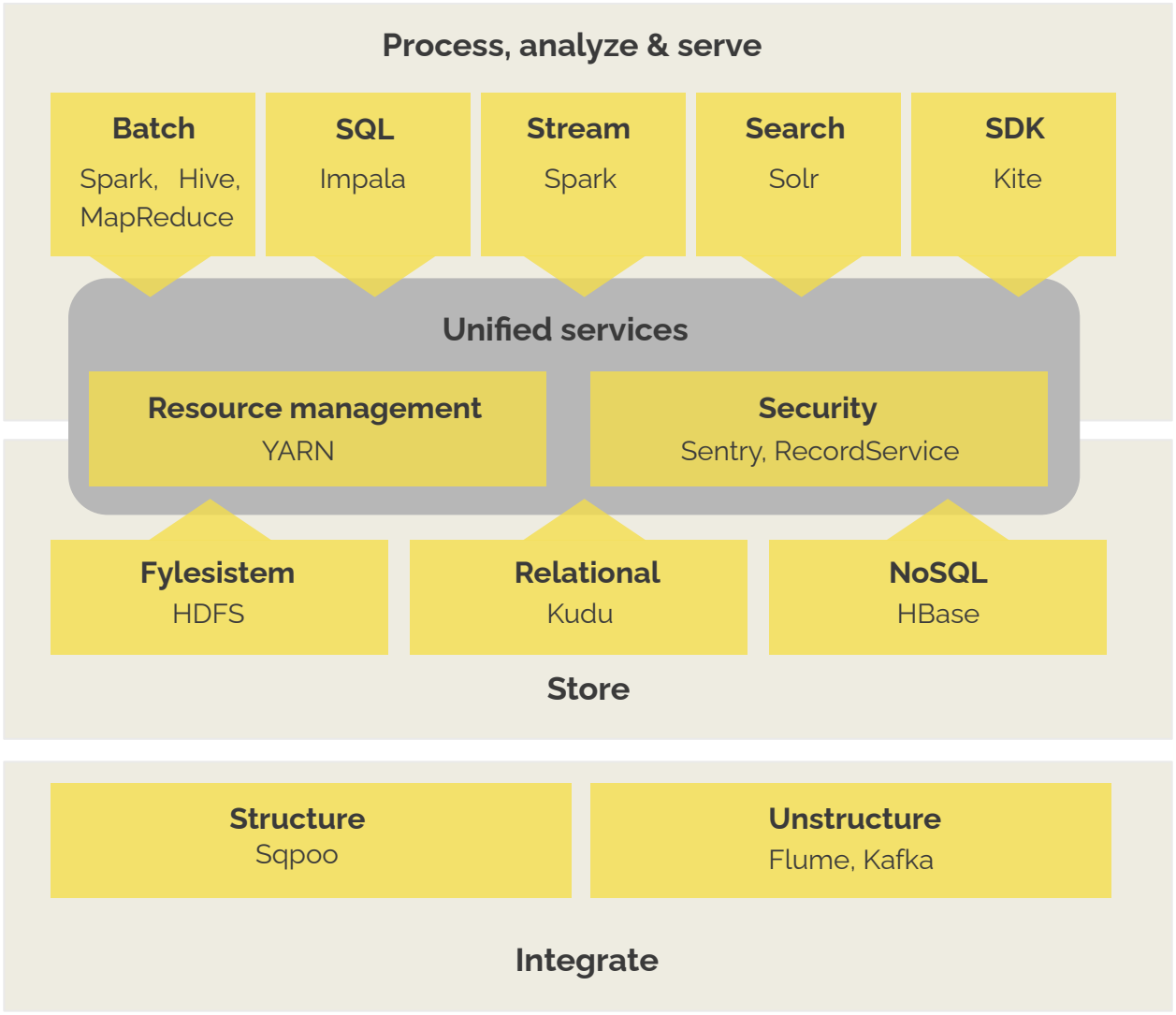


La consultora **Forrester** publicó en enero de 2016 el estudio *The Forrester Wave™: Distribuciones de Big Data Hadoop, primer trimestre de 2016*, donde evaluó las distribuciones comerciales más populares. Incluye Cloudera, Hortonworks, MapReduce, IBM y Pivotal, los cinco principales proveedores de distribuciones de *software* Hadoop. Todos estos proveedores enfocan su *software* en características empresariales clave como seguridad, escala, integración, gobierno y rendimiento, dicen Gualtieri y Yuhanna (2016).

Se pueden implementar en las instalaciones de los clientes, en una nube privada o en una nube pública, pero los clientes administran el *software*. El **informe *Wave*** de Forrester (2016) no evaluó las distribuciones de Hadoop basadas en la nube, como Elastic MapReduce de Amazon Web Service o HDInsight de Microsoft Azure, porque son productos basados únicamente en la nube pública que los clientes no pueden ejecutar en su propio *hardware*.

La **distribución Cloudera**, fundada en 2008, figura como líder en el informe, obteniendo la puntuación más alta por su oferta actual y presencia en el mercado, según una evaluación de 30 criterios que Forrester (2016) utilizó para comparar a los proveedores.

Cloudera fue la **primera** distribución Hadoop del mercado. Su *chief architect* es el propio Dough Cutting, uno de los creadores de Hadoop. Por ello, su ritmo de innovación es vertiginoso. Su producto, "Cloudera Enterprise", está formado por su propia distribución de Hadoop (CDH), un "Cloudera Manager" propietario y soporte de usuario para los componentes *core* de CDH.



Fuente: Ecosistema Hadoop Cloudera. Adaptado de Recuero de los Santos (2017)

En **2013** anunció su estrategia diferenciadora basada en crear componentes propietarios de valor añadido sobre el Hadoop *open source* como Impala, que es muy estimado por los clientes junto con otras herramientas *add-on* como Cloudera Manager y Cloudera Navigator.

Del mismo modo, se caracteriza por **acelerar** la introducción del código *open source* de nivel alfa o beta de las versiones más recientes de Hadoop, y por su política de adquisiciones o colaboraciones con otras empresas que permitan cubrir las carencias en seguridad, gestión de datos y analíticas. Cloudera también ofrece un completo (aunque caro) programa de formación y certificación de profesionales (Recuero de los Santos, 2017).

Hortonworks era quizás el mayor **competidor** de Cloudera y ocupó por mucho tiempo el segundo lugar en cuanto a presencia en el mercado, comprometido con una distribución 100 % de código abierto de Hadoop. Efectivamente, estaba tan involucrado con este modelo que incluso cuando adquiría alguna empresa para rellenar algún "gap" no cubierto por el *core* de Hadoop, aportaba el código al proyecto Apache en beneficio de la comunidad, como hizo cuando adquirió XA Secure, que pasó a convertirse en Apache Ranger.

Fue el **primer vendedor** en usar la funcionalidad HCatalog para los servicios de metadatos y logró optimizar el proyecto Hive, el estándar de facto para las *queries* SQL interactivas, con la iniciativa Stinger.

HORTONWORKS®

Logo de la empresa Hortonworks

Otra de las **ventajas** de esta distribución es una sencilla y práctica *sand-box* para aprender a manejar el entorno, así como numerosos tutoriales online.

Pero la mayor **diferencia** entre Hortonworks y sus competidores fue el desarrollo de importantes mejoras en el *core trunk* de Hadoop que permiten que se ejecute de forma nativa en plataformas Windows, tanto en servidores *on premise*, como en la nube (Windows Azure), mientras que sus competidores trabajan únicamente sobre Linux (Recuero de los Santos, 2017).

Pero en octubre de 2018, Hortonworks y Cloudera anunciaron que se combinarían en una fusión de iguales de todas las acciones. Después de la fusión, los productos Apache de Hortonworks se convirtieron en Cloudera Data Platform. Sin embargo, aún se encuentra disponible en:

https://www.cloudera.com/content/dam/www/marketing/resources/datasheets/hdp-datasheet.pdf.landing.html

Haz clic

Otra distribución líder, **MapR**, también fue adquirida por Hewlett Packard Enterprises HPE en agosto de 2019. MapR, al igual que antes, sigue dedicado a mantener el mejor equilibrio entre alto rendimiento y escalabilidad, al mismo tiempo que maximiza la facilidad de uso (CHAWLA, 2019).

HPE dice que la transacción incluye la tecnología, la propiedad intelectual y la experiencia de dominio de MapR en inteligencia artificial, aprendizaje automático y gestión de datos analíticos, y que la adquisición ayudará a los clientes de HPE a crear canalizaciones de datos en plataformas locales y en la nube y a ejecutar todas esas cargas de trabajo en el mismo entorno propiedad de HPE.

Para MapR, Hadoop ha sido una gran herramienta que ayudó a las empresas a **almacenar y analizar** enormes cantidades de datos no estructurados de numerosas fuentes. Aunque Hadoop se volvió aún más sofisticado con el tiempo para incluir capacidades como el aprendizaje automático y el procesamiento en memoria de Spark, recibió una gran competencia de Amazon Web Services, Microsoft Azure y Google Cloud, todos los cuales satisfacían las necesidades de análisis de *big data* de los usuarios sin que ellos tuvieran que hacerlo.

El auge de las **nubes** híbridas y múltiples ha socavado tanto el valor del uso empresarial de Hadoop que los proveedores de Hadoop como MapR, Cloudera y Hortonworks se vieron afectados negativamente. Cabe señalar que estas tres empresas juntas obtuvieron una financiación de más de 1.500 millones de dólares durante el movimiento de *big data*.

A pesar de que ya se habían hecho públicos, Cloudera y Hortonworks decidieron fusionarse a finales de 2018 por un valor combinado de **5.200 millones** de dólares.

1 Hadoop en la nube:

Las distribuciones que se describieron anteriormente se denominan *on premise*, ya que pueden ser instaladas en centros de datos internos en las empresas o instituciones. Sin embargo, actualmente las empresas pueden desplegar Hadoop a través de **proveedores de nube** como Microsoft Azure, Google App Engine y Amazon S3 (Bigelow, 2015).

Los principales **beneficios** de unir las tecnologías del Cloud Computing y el *big data* se pueden resumir en:

- **Abaratamiento de costos:** en Cloud Computing habitualmente trabajamos con la modalidad de pago por uso. Esto permite huir de cuantiosas cuotas mensuales y pagar únicamente por el uso que le demos a las soluciones en la nube que tengamos contratadas como herramientas para el *big data*. Este hecho es especialmente importante para PYMEs, que tienen a su alcance servicios anteriormente inalcanzables por su elevado costo.

- **Inmediatez:** la contratación y puesta en marcha de una solución de Cloud Computing es un proceso ágil. Dado que no tenemos que realizar complicadas configuraciones de servidores, cuando necesitemos crear un nuevo recurso en un entorno Cloud lo tendremos disponible en cuestión de minutos.

- **Capacidad de proceso:** con un servidor tradicional nuestros recursos estaban limitados. El Cloud Computing permite manejar grandes volúmenes de información, escalando los recursos destinados a cada proceso de forma fácil.

- **Concurrencia:** será habitual que varios usuarios o proyectos requieran acceder al *big data* en paralelo. Las capacidades del Cloud Computing permiten la concurrencia de accesos sin afectar al rendimiento.

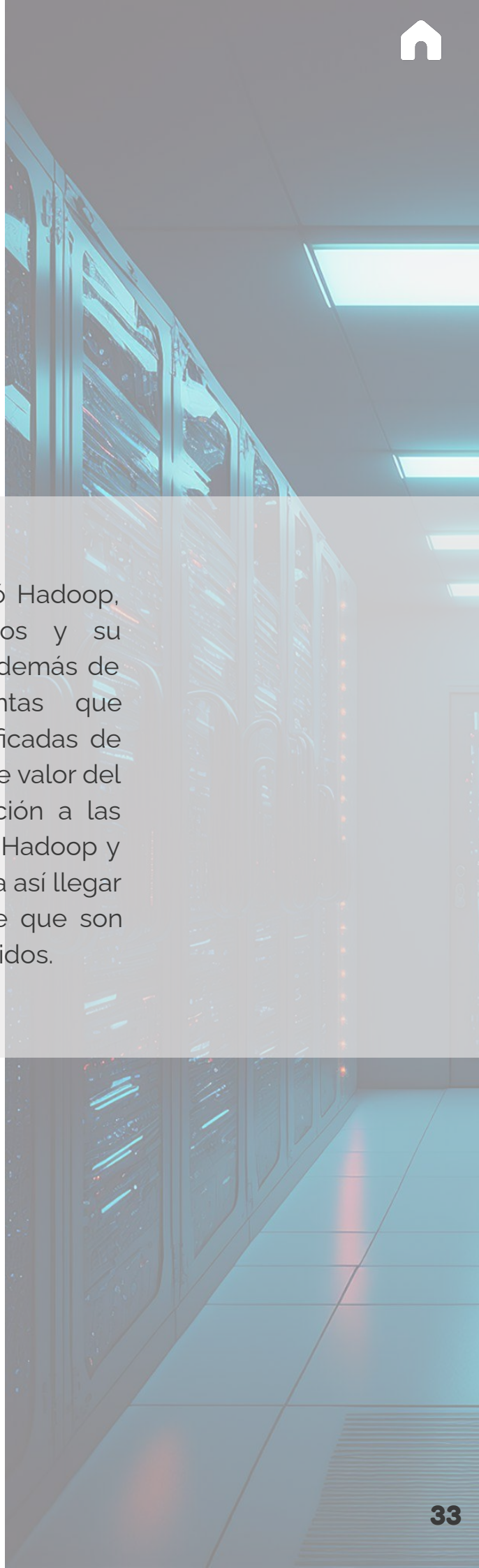
2 ¿Por qué llevar Hadoop a la nube?

Al tratarse de un servicio donde es tan importante la alta capacidad de almacenamiento, Hadoop es perfecto para una **arquitectura Cloud**, capaz de estirarse y encogerse dinámicamente y en caliente, según la demanda. Además, la distribución del sistema de archivos a través de sus diferentes clústers también puede resultar determinante, ya que Cloud nos permite levantar muy fácilmente máquinas virtuales en diferentes centros de datos distribuidos geográficamente o configurar diversos nodos dentro de un clúster (Arsys, 2017).

La nube es ideal para proporcionar la **potencia de cálculo** de *big data* requerida para el procesamiento de estos grandes conjuntos de datos paralelos. La nube tiene la capacidad de proporcionar la plataforma de computación flexible y ágil que se necesita para *big data*, así como la capacidad de recurrir a cantidades masivas de poder de cómputo (para poder escalar según sea necesario), y por todas estas características sería una plataforma ideal para el análisis sobre demanda de cargas de trabajo estructuradas y no estructuradas (Menegaz, 2014).

Conviene recordar que no existen las **"soluciones ideales"** a la hora de desplegar una arquitectura IT para un proyecto basado en algo tan crítico para el negocio como puede resultar Hadoop. Aunque podamos optar por una solución 100 % basada en la nube pública, puede no ser la opción óptima en todos los casos. Con los volúmenes de almacenamiento más exigentes, las infraestructuras Cloud híbridas nos garantizarán el mejor rendimiento para estos aplicativos de *big data* basados en Hadoop. De este modo, contaremos con lo mejor de dos mundos (Arsys, 2017).

CIERRE



En este tema se **revisó** cómo se originó Hadoop, cuáles son sus componentes básicos y su importancia en el mundo de *big data*. Además de conocer algunas de las herramientas que conforman el ecosistema Hadoop, clasificadas de acuerdo a algunas etapas de la cadena de valor del dato, finalmente se hizo una aproximación a las principales distribuciones *on premise* de Hadoop y a cómo estas han ido evolucionando, para así llegar a las soluciones de Hadoop en la nube que son ofrecidas por los proveedores más conocidos.

REFERENCIAS

Aguilar, L. (2013). *Big Data. Análisis de grandes volúmenes de datos en organizaciones.* Alfaomega.

Aguilar, L. (2019). *Inteligencia de negocios y analítica de datos.* Alfaomega.

Apache Hadoop (2017). *Apache Hadoop YARN*. https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html

Arsys (2017). *Cloud Computing, la mejor opción para alojar Hadoop*. Arsys. https://www.arsys.es/blog/hadoop-cloud

Bigelow, S. (2015). *¿Se impulsará big data con Hadoop en la nube?* ComputerWeekly. https://www.computerweekly.com/es/respuesta/Se-impulsara-big-data-con-Hadoop-en-la-nube

Chawla, V. (2019). *It's The End Of The Big Data Era, HPE Acquires A Struggling MapR*. Analytics India Magazine. https://analyticsindiamag.com/hpe-acquisition-mapr/

Cloudera (s.f.). *Hortonworks Data Platform Datasheet*. https://www.cloudera.com/content/dam/www/marketing/resources/datasheets/hdp-datasheet.pdf.landing.html

Data Flair (2016). *Apache Flink vs Apache Spark - Una guía de comparación*. Data Flair. https://data-flair.training/blogs/comparison-apache-flink-vs-apache-spark/

Google Inc. (2004). *MapReduce: Simplified Data Processing on Large Clusters.* https://static.googleusercontent.com/media/research.google.com/es//archive/mapreduce-osdi04.pdf

Gualtieri, M. y Yuhanna, N. (2016). *The Forrester Wave™: Big Data Hadoop Distributions, Q1 2016*. Forrester. https://www.forrester.com/report/The-Forrester-Wave-Big-Data-Hadoop-Distributions-Q1-2016/RES121574

REFERENCIAS

Hortonworks (s.f.). *Apache Hadoop Ecosystem and Open Source Big Data Projects*. https://es.hortonworks.com/ecosystems/

Hortonworks (s.f.). *Apache Hadoop MapReduce*. https://es.hortonworks.com/apache/mapreduce/

Hortonworks (s.f.). *Apache Hadoop Yarn*. https://es.hortonworks.com/apache/yarn/

Menegaz, G. (2014). *What is Hadoop, and how does it relate to cloud?* IBM. https://www.ibm.com/blogs/cloud-computing/2014/05/07/hadoop-relate-cloud/

Recuero de los Santos, P. (2017). *El Ecosistema Hadoop (III): Una gran diversidad "biológica»*. Telefónica Tech. https://empresas.blogthinkbig.com/el-ecosistema-hadoop-iii-una-gran/

Recuero de los Santos, P. (2017). *Cloudera, MapR, Hortonworks... ¿Qué distribución Hadoop necesitas?* Telefónica Tech. https://empresas.blogthinkbig.com/cloudera-mapr-hortonworksque/

Vaughan, J. (2021). *Apache Hadoop YARN*. Computer Weekly. https://www.computerweekly.com/es/definicion/Apache-Hadoop-YARN-Yet-Another-Resource-Negotiator?_gl=1*1g8bah0*_ga*MTc2MTQ0MDI2OC4xNjc0NzY0Mzg4*_ga_TQKE4GS5P9*MTY3NDc2NDM4OC4xLjAuMTY3NDc2NDM4OC4wLjAuMA..&_ga=2.231661272.673066571.1674764388-1761440268.1674764388

White, T. (2015). *Hadoop: The Definitive Guide.* O'Reilly.

Yogendra, B., Khan, S., Borasi, P. y Kumar, V. (2022). *Hadoop Market by Component (Hardware, Software, and Services), Deployment Model (On-premise, Cloud, and Hybrid), Enterprise Size (Large Enterprises and SMEs), and Industry Vertical (Manufacturing, BFSI, Retail & Consumer Goods, IT & Telecommunication, Healthcare, Government & Defense, Media & Entertainment, Energy & Utility, Trade & Transportation, and Others): Global Opportunity Analysis and Industry Forecast, 2021-2030*. Allied Market Research. https://www.alliedmarketresearch.com/world-hadoop-market

Referencias de las imágenes

Recuero de los Santos, P. (2017). Ecosistema Hadoop Cloudera [Imagen]. Disponible en: Telefónica Tech. https://empresas.blogthinkbig.com/cloudera-mapr-hortonworksque/

SAS (s.f.). Hadoop Timeline [Imagen]. Disponible en: https://www.sas.com/es_pe/insights/big-data/hadoop.html

White, T. (2015). Funcionamiento de MapReduce [Imagen]. Disponible en: *Hadoop: The Definitive Guide*.

UCAB UNIVERSIDAD CATÓLICA ANDRÉS BELLO

Has culminado la revisión del tema