



Sistemas Distribuidos

Grupo 01



Hadoop, MapReduce, Lustre P2P, BitTorrent

Aguilar Maya Daniel

Diego López Ortega

Mancera Hernández Eduardo Abdiel

Laylet Rojas Terrazas



Marzo, 2024



Preliminares de Hadoop

Para una comprensión del contexto en el cual se desenvuelve Hadoop, será necesario comprender el significado del Big Data. Big data es un término de origen inglés cuya traducción equivale a "datos masivos". Muchas son las definiciones que entidades y organizaciones han dado para el término big data, pero todas ellas se pueden resumir en el conjunto de datos cuyo tamaño supera considerablemente la capacidad de captura, almacenado, gestión y análisis del software convencional de bases de datos.

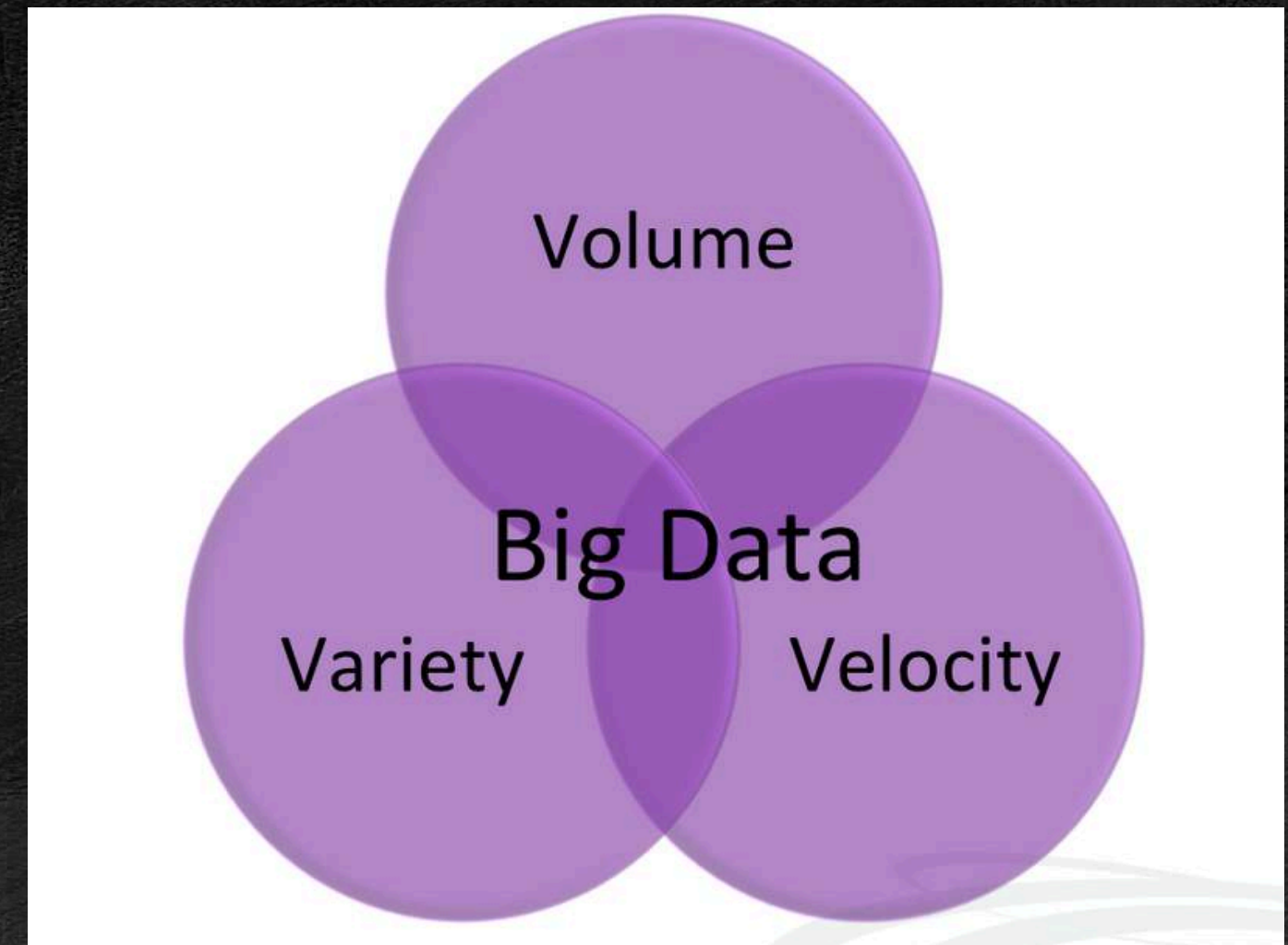


Las 3 V's

Volumen. Big data corresponde al gran volumen de datos que se generan diariamente en las empresas y organizaciones de todo el mundo.

Velocidad. Cómo de rápido se están produciendo los datos, así como la rapidez en la que se trata de satisfacer la demanda de éstos. La tecnología big data ha de ser capaz de almacenar y trabajar en tiempo real con las fuentes generadoras de información tales como sensores, redes sociales, blogs, páginas webs, etc., fuentes que generan millones de datos al segundo; y, por otro lado, capacidad de analizar dichos datos con la suficiente rapidez.

Variedad. Big data ha de tener la capacidad de combinar una gran variedad de información digital en los diferentes formatos en los que se puedan presentar.



W HADOOP

historia



- La explosión de la Web, de docenas a millones de páginas, impulsó la automatización de los procesos de búsqueda.
- En este contexto, surgió Nutch, un proyecto de código abierto liderado por Doug Cutting y Mike Cafarella.
- El surgimiento de Hadoop se gestó en el contexto de la creciente necesidad de procesar volúmenes cada vez mayores de Big Data.
- Su inspiración provino de MapReduce, un modelo de programación desarrollado por Google que permite dividir una aplicación en componentes más pequeños para ejecutarlos en diferentes nodos de un servidor.
- Actualmente, la estructura y el ecosistema de tecnologías de Hadoop son gestionados y mantenidos por la Apache Software Foundation (ASF), una comunidad global sin fines de lucro integrada por programadores de software y otros colaboradores.



Doug Cutting



HADOOP

definición

Hadoop es un framework de software de código abierto utilizado para el almacenamiento y procesamiento distribuido de conjuntos de datos grandes en clusters de computadoras. Está diseñado para ser escalable, confiable y tolerante a fallos, y se utiliza principalmente para trabajar con datos estructurados y no estructurados en entornos distribuidos. Hadoop consta de varios módulos, incluyendo Hadoop Distributed File System (HDFS) para el almacenamiento distribuido y MapReduce para el procesamiento paralelo de datos. Además, Hadoop cuenta con una amplia gama de herramientas y proyectos complementarios en su ecosistema para diversas tareas de análisis y procesamiento de datos.







HADOOP

funcionamiento

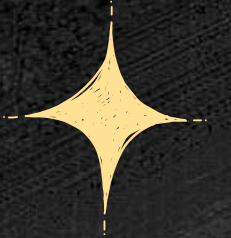
Funciona en un modelo de programación de "divide y conquista", donde los datos se dividen en bloques más pequeños y se distribuyen en diferentes nodos del clúster para su procesamiento paralelo.

Supongamos que tenemos un conjunto de datos muy grande que deseamos analizar para encontrar información específica. Este conjunto de datos se almacena en un clúster de servidores que ejecutan el software de Hadoop.



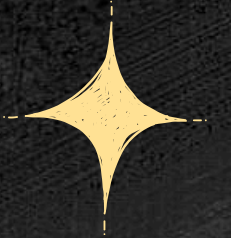


HADOOP **funcionamiento**

- **División de datos:** Primero, Hadoop divide el conjunto de datos en bloques más pequeños. Por ejemplo, si tenemos un archivo de 1 terabyte, Hadoop podría dividirlo en bloques de 128 megabytes cada uno.
 - **Distribución de datos:** Luego, estos bloques de datos se distribuyen en diferentes nodos del clúster. Cada nodo es responsable de almacenar y procesar uno o más bloques de datos.
 - **Procesamiento paralelo:** Una vez que los datos están distribuidos en el clúster, Hadoop ejecuta el proceso de procesamiento paralelo. Esto significa que múltiples nodos del clúster trabajan simultáneamente en el procesamiento de los datos.
- 

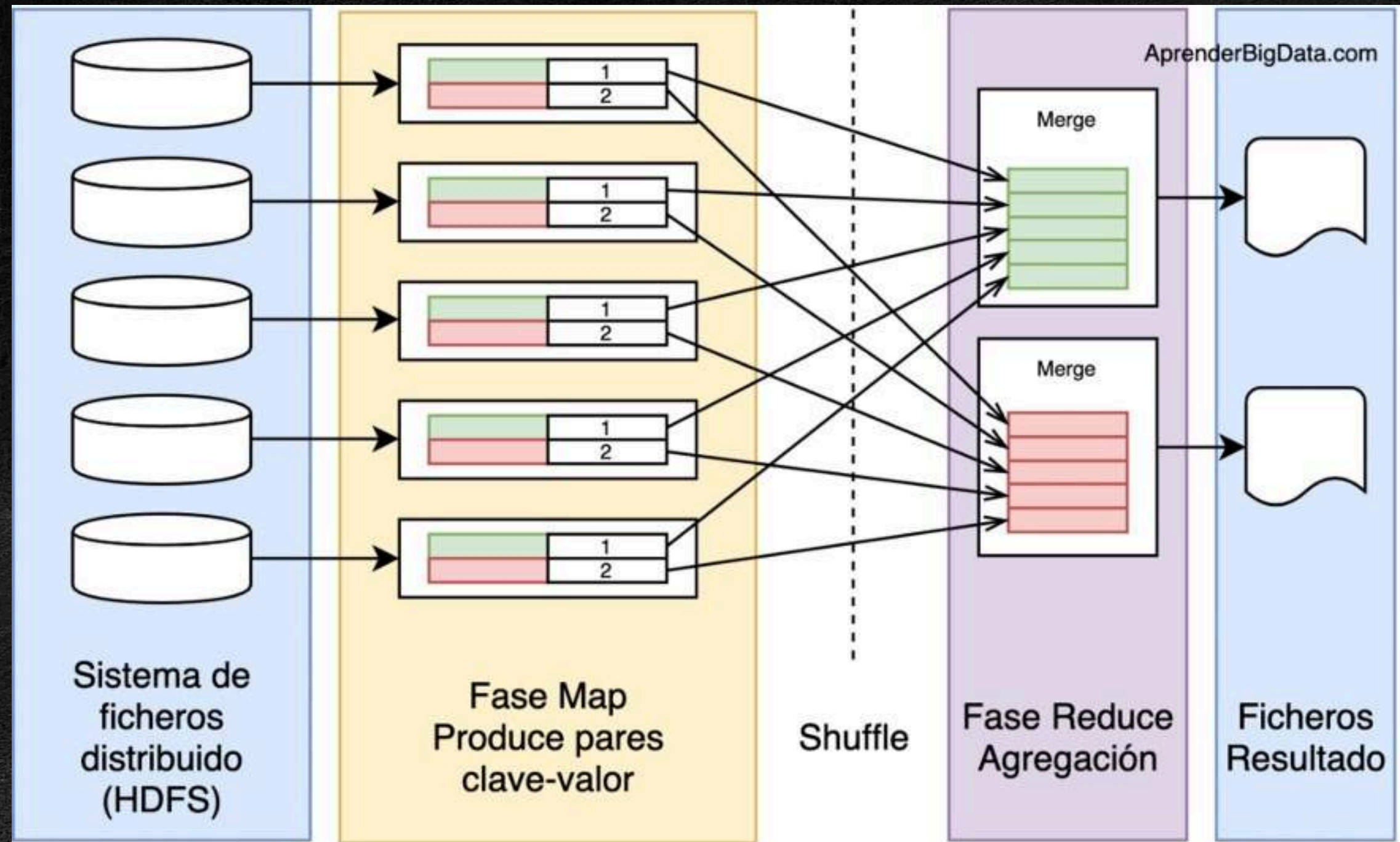


HADOOP **funcionamiento**

- **Recopilación de resultados:** Después de que todos los nodos hayan completado su procesamiento, Hadoop recopila los resultados parciales de cada nodo y los combina para generar el resultado final. Esto se realiza de manera eficiente mediante el uso de un componente llamado "reducción" en el modelo MapReduce de Hadoop.
 - **Almacenamiento de resultados:** Finalmente, los resultados finales se almacenan en el sistema de archivos distribuido de Hadoop, como Hadoop Distributed File System (HDFS), para su posterior análisis o uso.
- 

HADOOP

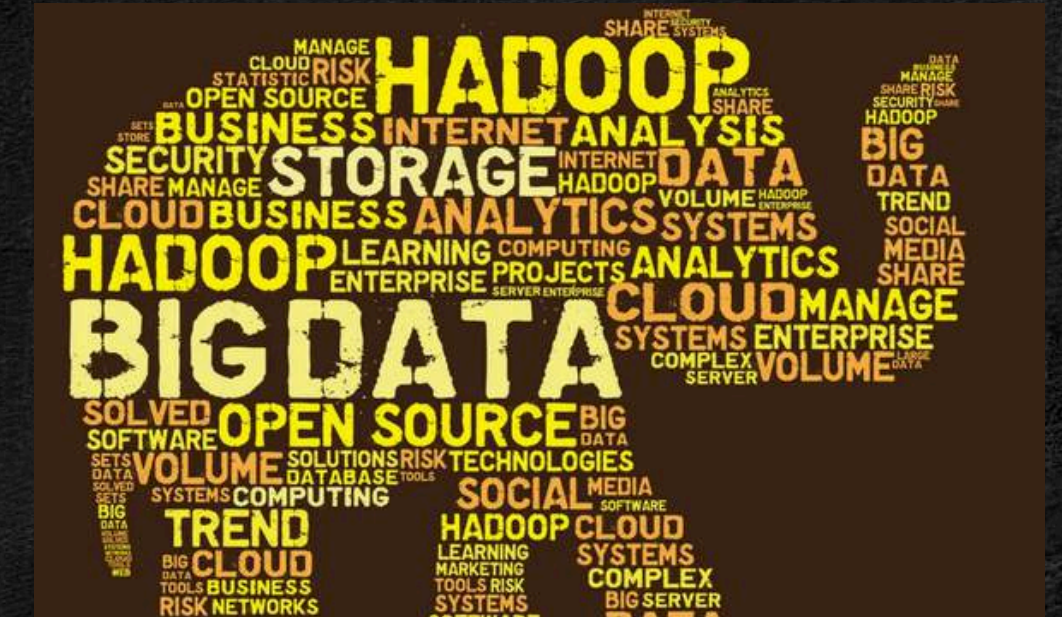
funcionamiento





HADOOP

utilidad



01

Análisis de Datos a gran escala

03

Almacenamiento de Datos a gran escala

02

Procesamiento de Datos en Tiempo Real

04

Aprendizaje automático e Inteligencia Artificial



☆☆ HADOOP industrias

- Tecnología y medios de comunicación
- Finanzas
- Salud
- Retail y Comercio Electrónico
- Telecomunicaciones
- Energía y recursos Naturales



MAPREDUCE historia

- MapReduce se remonta a principios de la década de 2000, cuando Google estaba lidiando con el desafío de procesar grandes volúmenes de datos generados por su motor de búsqueda y otros servicios en línea.
- Para abordar este desafío, Google desarrolló un modelo de programación llamado MapReduce, que se basaba en los principios de programación funcional y procesamiento distribuido.
- La inspiración detrás de este algoritmo se encuentra en la programación funcional y las funciones "map" y "reduce", que son inherentes a lenguajes como Lisp y Scheme.
- Jeff Dean y Sanjay Ghemawat tomaron prestadas estas ideas y las adaptaron al procesamiento de datos distribuidos en clústeres de servidores.
- El trabajo inicial en MapReduce fue publicado por Google en un artículo técnico titulado "MapReduce: Simplified Data Processing on Large Clusters", en 2004.
- Con el tiempo, MapReduce se convirtió en un componente fundamental en el ecosistema de Big Data y fue adoptado por muchas empresas y organizaciones para procesar y analizar grandes volúmenes de datos de manera eficiente.



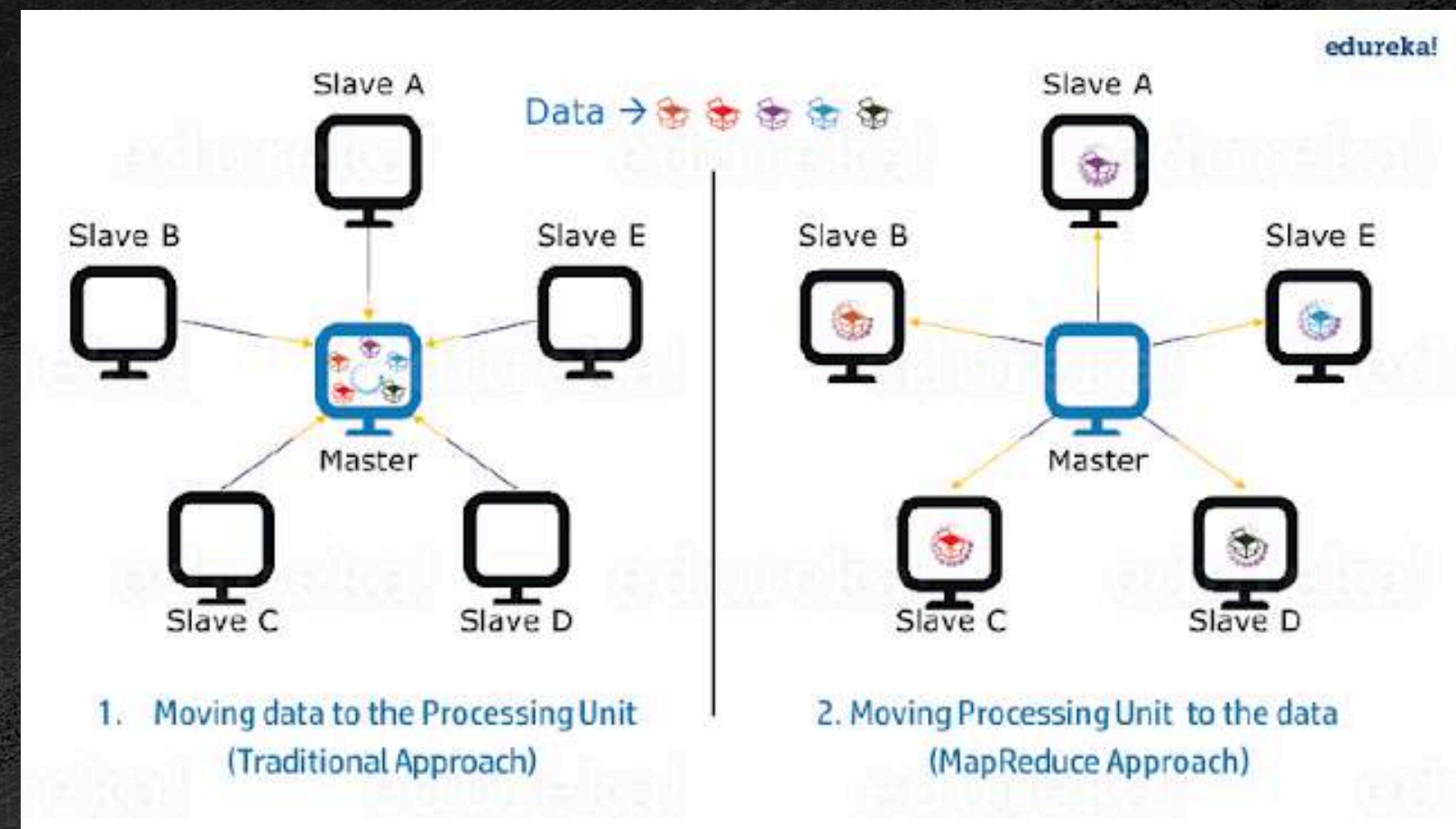
Sanjay Ghemawa y Jeff Deant



MAPREDUCE

definición

MapReduce es un modelo de programación y un algoritmo de procesamiento distribuido diseñado para procesar grandes conjuntos de datos de manera paralela en un clúster de computadoras. Se compone de dos fases principales: la fase de mapeo (Map) y la fase de reducción (Reduce). MapReduce es ampliamente utilizado en el procesamiento de grandes volúmenes de datos en entornos distribuidos, y forma parte del ecosistema de Hadoop, aunque también se ha implementado en otros sistemas y plataformas.







MAPREDUCE **funcionamiento**

1. División de datos: El proceso comienza con la división del conjunto de datos de entrada en fragmentos más pequeños llamados "splits". Cada split consiste en un conjunto de datos independiente que puede ser procesado de forma paralela en los nodos del clúster.

2. Fase de Map:

- Cada split se envía a uno o más nodos del clúster para la fase de map.
 - En esta fase, se aplica una función de map a cada registro del split para producir un conjunto intermedio de pares clave-valor.
 - La función de map toma como entrada un par clave-valor y genera cero o más pares clave-valor como salida.
 - Los pares clave-valor intermedios se almacenan en un área temporal de almacenamiento local en cada nodo.
- 
- 

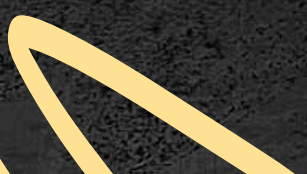



MAPREDUCE **funcionamiento**

3. Shuffle y clasificación

- Después de la fase de map, los pares clave-valor intermedios se envían a los nodos de reducción correspondientes.
- Durante este proceso de shuffle, los pares clave-valor se clasifican y agrupan por clave para que todos los valores asociados con una misma clave se envíen al mismo nodo de reducción.

4. Combinación y ordenamiento opcional

- Opcionalmente, se puede realizar una operación de combinación y ordenamiento antes de la fase de reduce para reducir el tráfico de red y mejorar la eficiencia.
 - Durante esta operación, los pares clave-valor intermedios se combinan y se ordenan localmente en cada nodo de map antes de ser enviados a los nodos de reducción correspondientes.
- 
- 

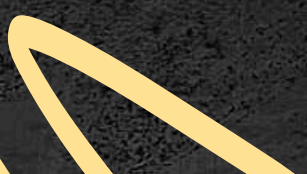



MAPREDUCE **funcionamiento**

5. Fase Reduce

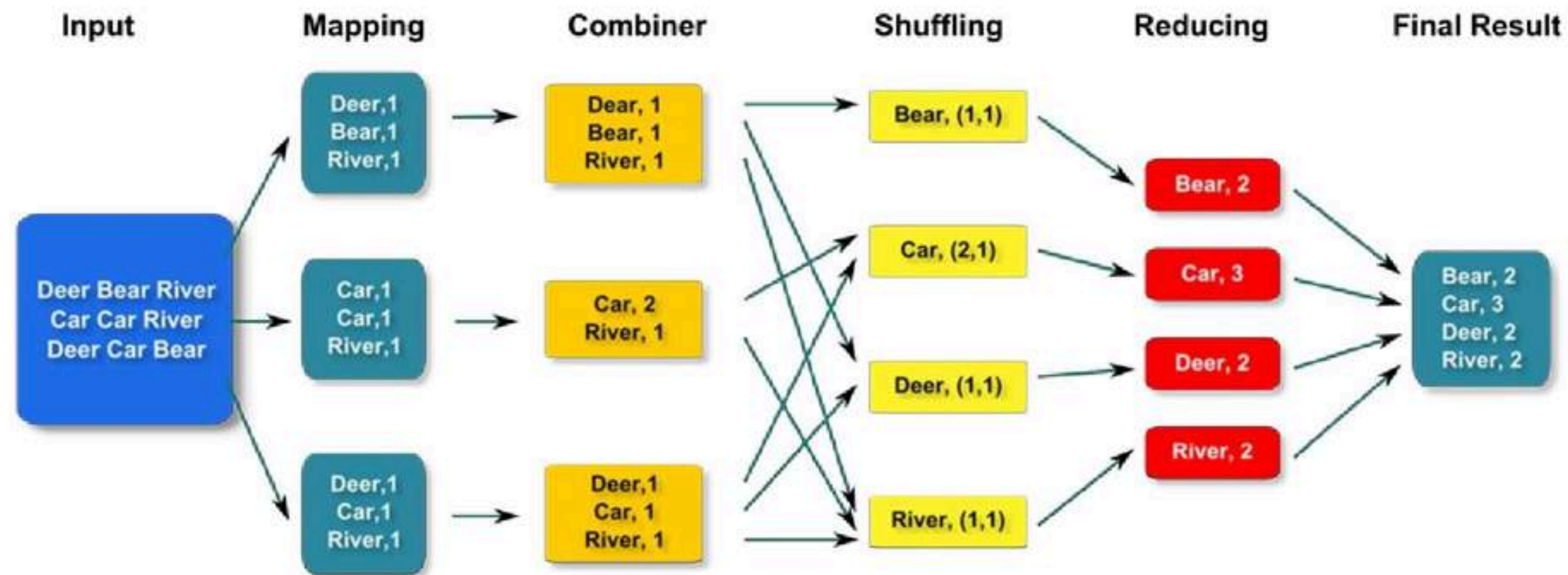
- En esta fase, cada nodo de reducción procesa los pares clave-valor intermedios asociados con una misma clave.
- Se aplica una función de reducción a cada conjunto de valores asociados con una misma clave para producir los resultados finales.
- La función de reducción toma como entrada una clave y una lista de valores asociados con esa clave, y produce cero o más pares clave-valor como salida.
- Los resultados finales de la fase de reducción se escriben en el sistema de archivos de salida final.

6. Almacenamiento de resultados finales

- Los resultados finales del procesamiento se almacenan en el sistema de archivos de salida final, que puede ser utilizado por otras aplicaciones o procesos según sea necesario.
- 
- 

MAPREDUCE funcionamiento

Combiner - Local Reduce





MAPREDUCE

utilidad



01

**Procesamiento de
Grandes Volúmenes de
Datos**

03

**Procesamiento de Datos
en la Nube**

02

**Análisis de Datos
Estructurados y No
Estructurados**

04

**Pensado para Lidar con
Grandes Problemas**



☆☆ MAPREDUCE **industrias**

- Tecnología y medios de comunicación
- Finanzas
- Salud
- Retail y Comercio Electrónico
- Telecomunicaciones
- Energía y recursos Naturales



LUSTRE

historia

- Diseñado originalmente en 1999 por Peter J. Braam en la Universidad de Stanford
- Braam fundó Cluster File Systems en 2001.
- En 2007, Sun Microsystems adquirió los activos de Cluster File Systems, integrando Lustre en su hardware de alto rendimiento.
- Posteriormente, en 2010, Oracle Corporation, tras adquirir Sun, asumió la administración de Lustre.
- Surgieron nuevas organizaciones como Whamcloud, OpenSFS y EUROPEAN Open File Systems (EOFS) para continuar el desarrollo de Lustre en un modelo de desarrollo comunitario abierto.
- El hecho de que la marca comercial Lustre haya sido transferida a OpenSFS y EOFS en noviembre de 2019 refleja el compromiso de mantener Lustre como un proyecto de código abierto y fomentar la colaboración dentro de la comunidad.



LUSTRE

historia

Optimización de los componentes Lustre
Mayor velocidad de procesamiento de datos
Aspectos de seguridad y encriptado
Uniformidad en los procesos de trabajo

The logo for Lustre, featuring the word "lustre" in a blue, lowercase, sans-serif font. Each letter is connected to the next by a small blue dot, and a registered trademark symbol (®) is located at the end of the word.

·l·u·s·t·r·e·®

LUSTRE

Definición

- El sistema de archivos Lustre es un sistema de archivos paralelo de código abierto que admite muchos requisitos de entornos de simulación HPC de clase líder. El sistema de archivos Lustre ha crecido hasta convertirse en un sistema de archivos que soporta algunas de las supercomputadoras más poderosas de la Tierra.
- Los componentes clave del sistema de archivos Lustre son los servidores de metadatos (MDS), los destinos de metadatos (MDT), los servidores de almacenamiento de objetos (OSS), los destinos de servidores de objetos (OST) y los clientes Lustre.



LUSTRE

¿Cómo funciona?

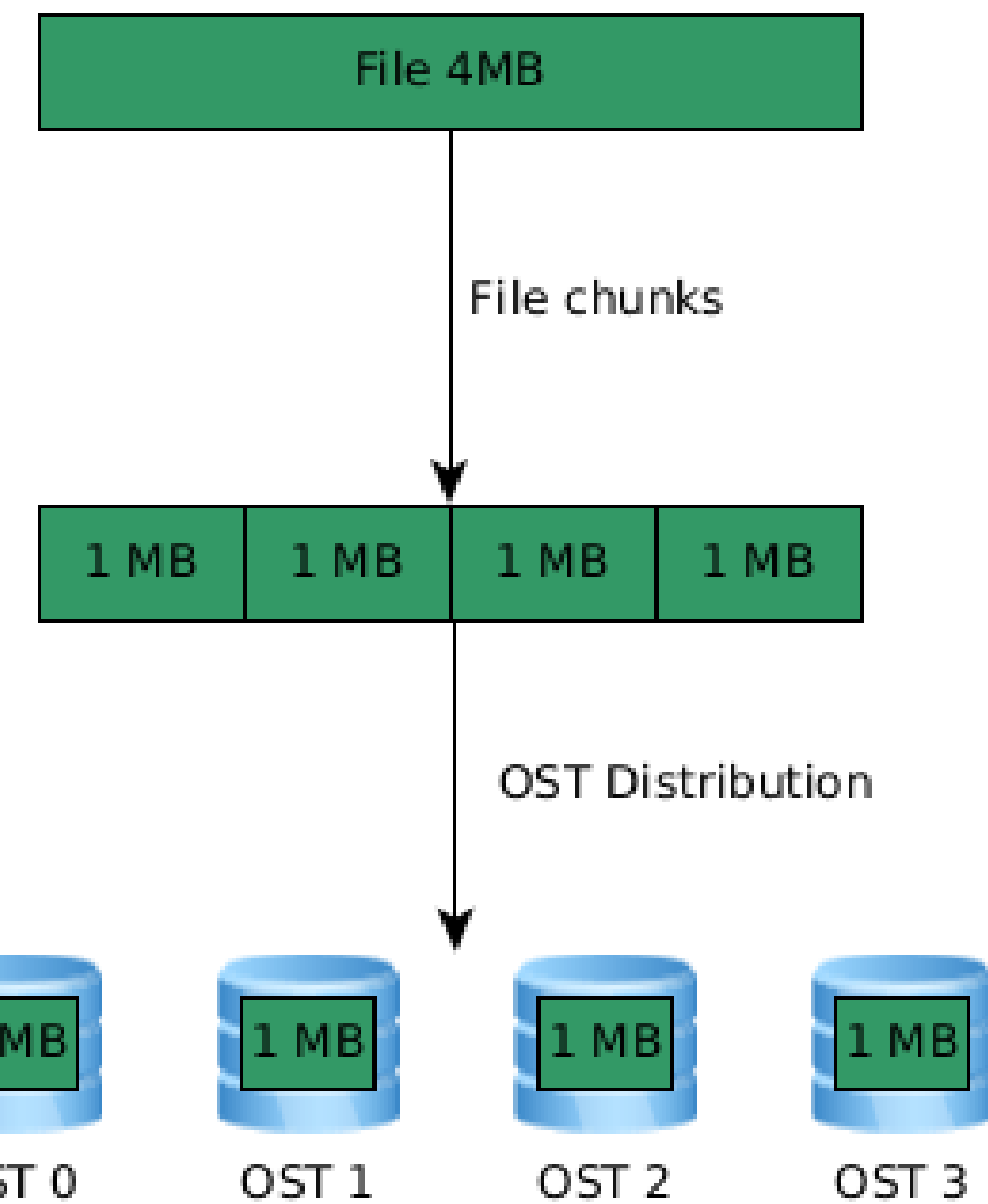
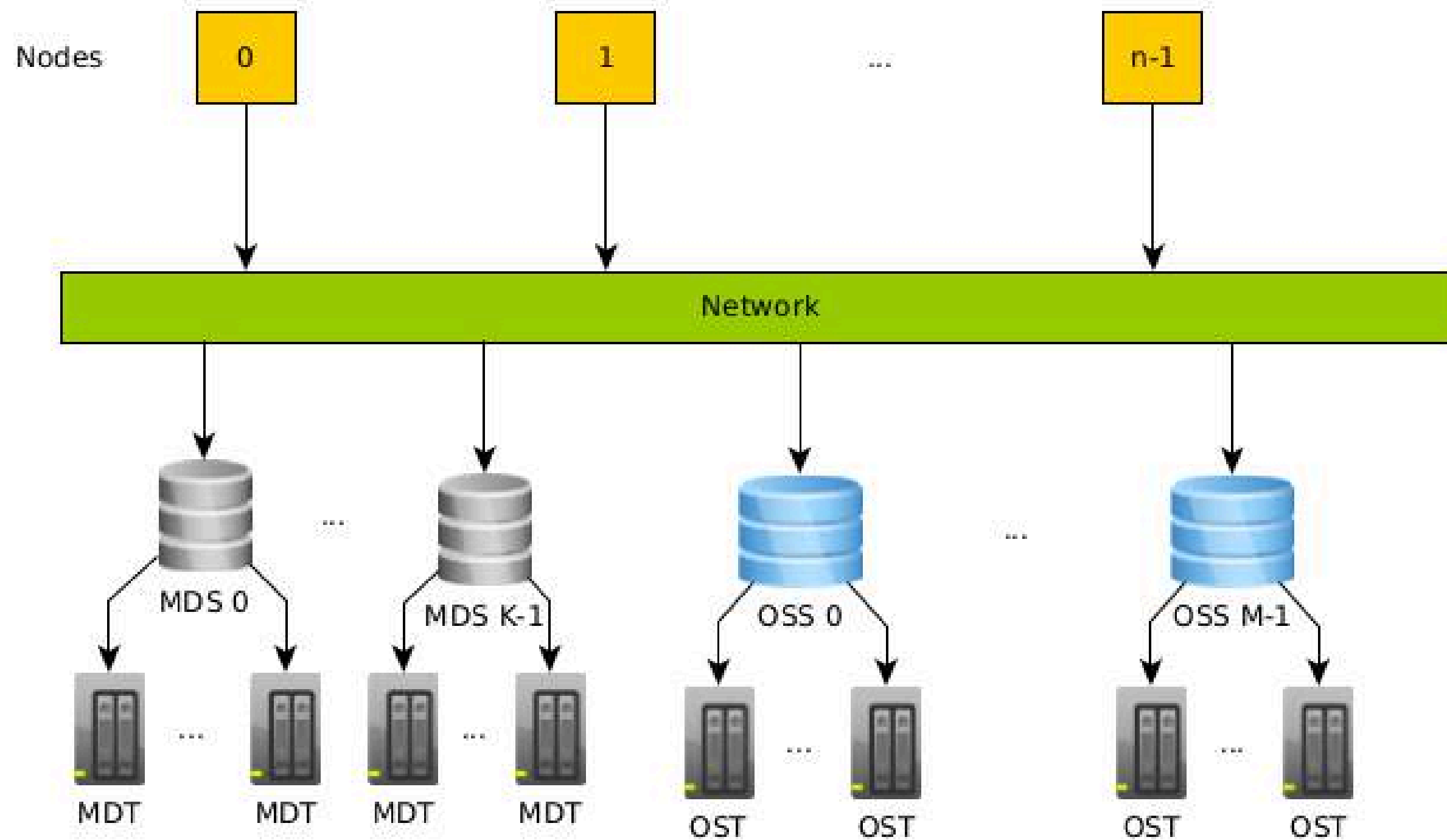
Lustre separa las informaciones de los archivos en dos partes distintas: los datos del archivo y los metadatos. Los datos representan el contenido real de los archivos, mientras que los metadatos contienen información sobre atributos como el tamaño del archivo, permisos de acceso, y fechas relacionadas.



LUSTRE

¿Cómo funciona?

- Servidores de almacenamiento de objetos (OSS): manejan las solicitudes de los clientes para acceder al almacenamiento. Además, gestionan un conjunto de OST; Cada OSS puede tener más de un OST para mejorar el paralelismo de E/S .
- Destinos de almacenamiento de objetos (OST): normalmente, un OST consta de un bloque de dispositivos de almacenamiento en configuración RAID. Los datos se almacenan en uno o más objetos y cada objeto se almacena en un OST independiente .
- Servidor de metadatos (MDS): un servidor que rastrea las ubicaciones de todos los datos para poder decidir que OSS y OST se utilizarán. Por ejemplo, una vez que se abre un archivo, el MDS ya no interviene.
- Destino de metadatos (MDT): el almacenamiento contiene información sobre los archivos y directorios, como tamaños de archivos, permisos y fechas de acceso. Para cada archivo, MDT incluye información sobre el diseño de los datos en los OST, como los números de OST y los identificadores de objetos.



* LUSTRE

Utilización actualidad

- Computación de Alto Rendimiento (HPC)
- Análisis de Big Data
- Industria del Entretenimiento y Medios
- Bioinformática y Genómica
- Investigación Científica



LUSTRE

Principal industria

- Investigación Científica
- Industria de la Energía
- Finanzas
- Industria Aeroespacial y de Defensa
- Investigación en Salud






lfs setstripe -c <número de OST> -S <tamaño de la franja> <nombre del archivo>

lfs setstripe -c 4 -S 1M archivo.txt

lfs getstripe my_file

```
my_file
lmm_stripe_count: 2
lmm_stripe_size: 1048576
lmm_pattern: raid0
lmm_layout_gen: 0
lmm_stripe_offset: 11
  obdidx      objid      objid      group
    11      29921659  0x1c8917b      0
    20      29922615  0x1c89537      0
```

Regla: Una regla general es utilizar como franja, la raíz cuadrada del tamaño del archivo en GB. Si el archivo tiene 90GB, la raíz cuadrada es 9.5, así que utilice al menos 9 OST

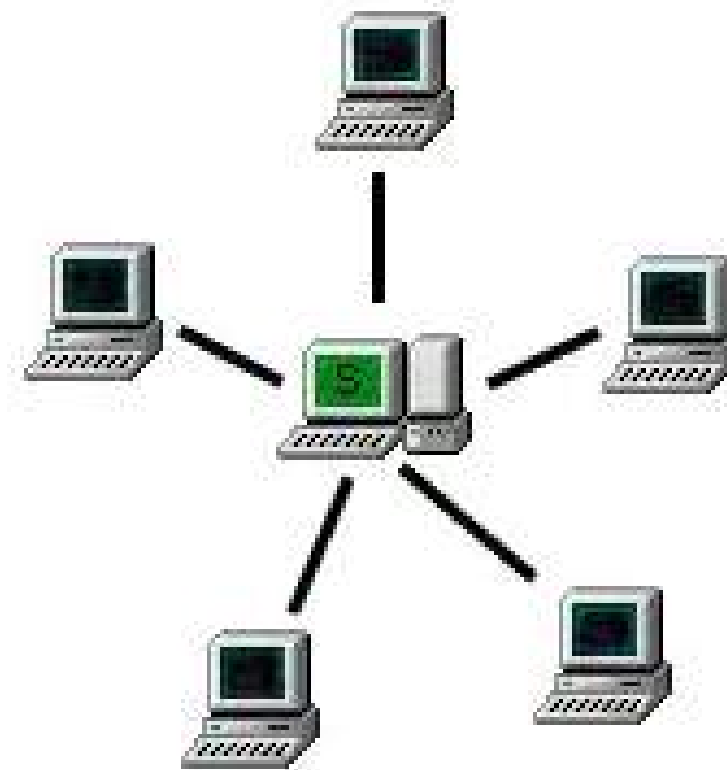


Peer to Peer (P2P)

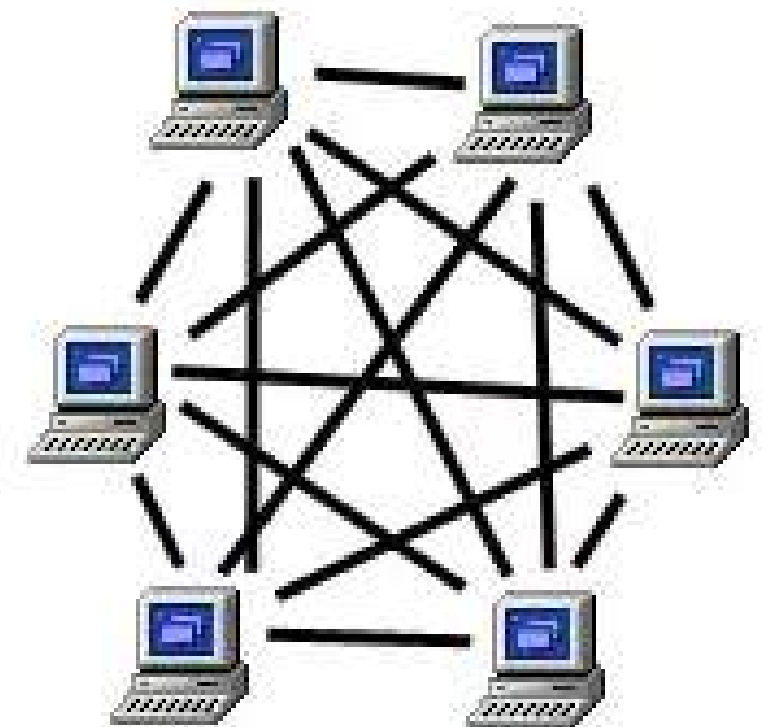
¿Qué es?

una red informática P2P se refiere a una red que no tiene clientes y servidores fijos, sino una serie de nodos que se comportan a la vez como clientes y como servidores de los demás nodos de la red.

Estructura Cliente-Servidor

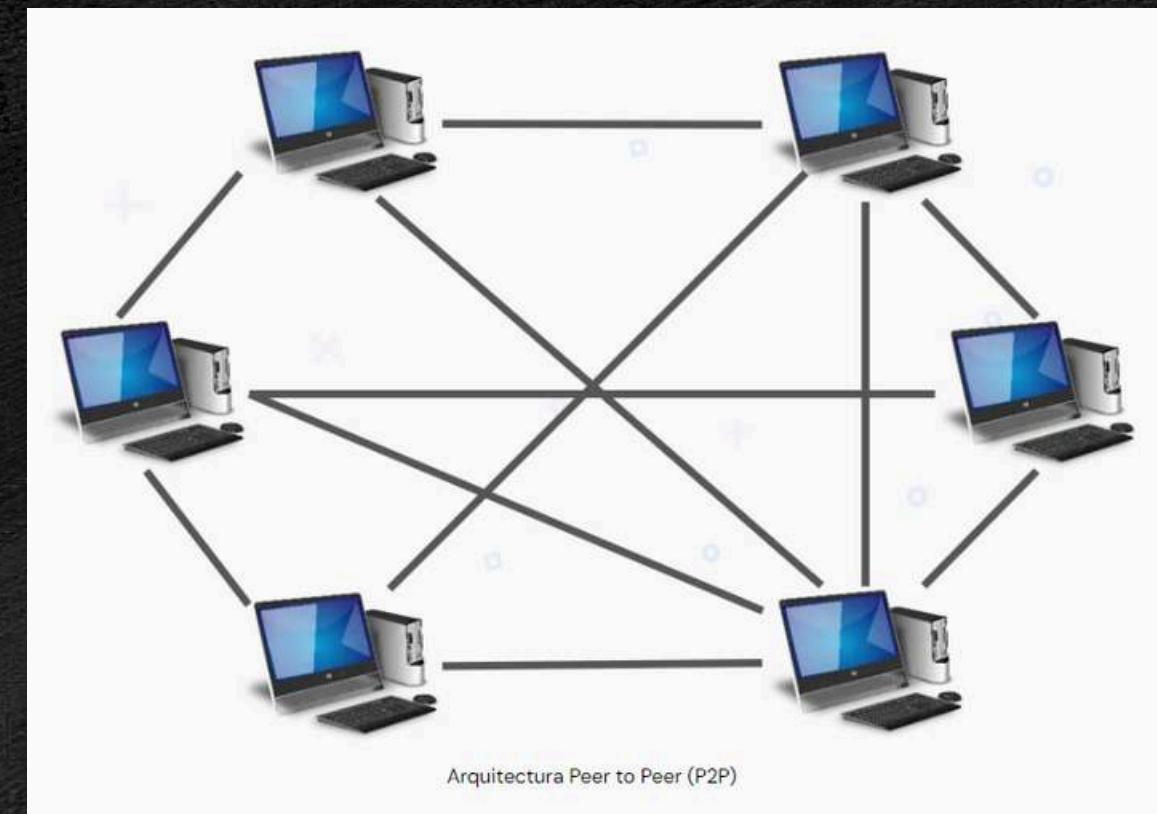
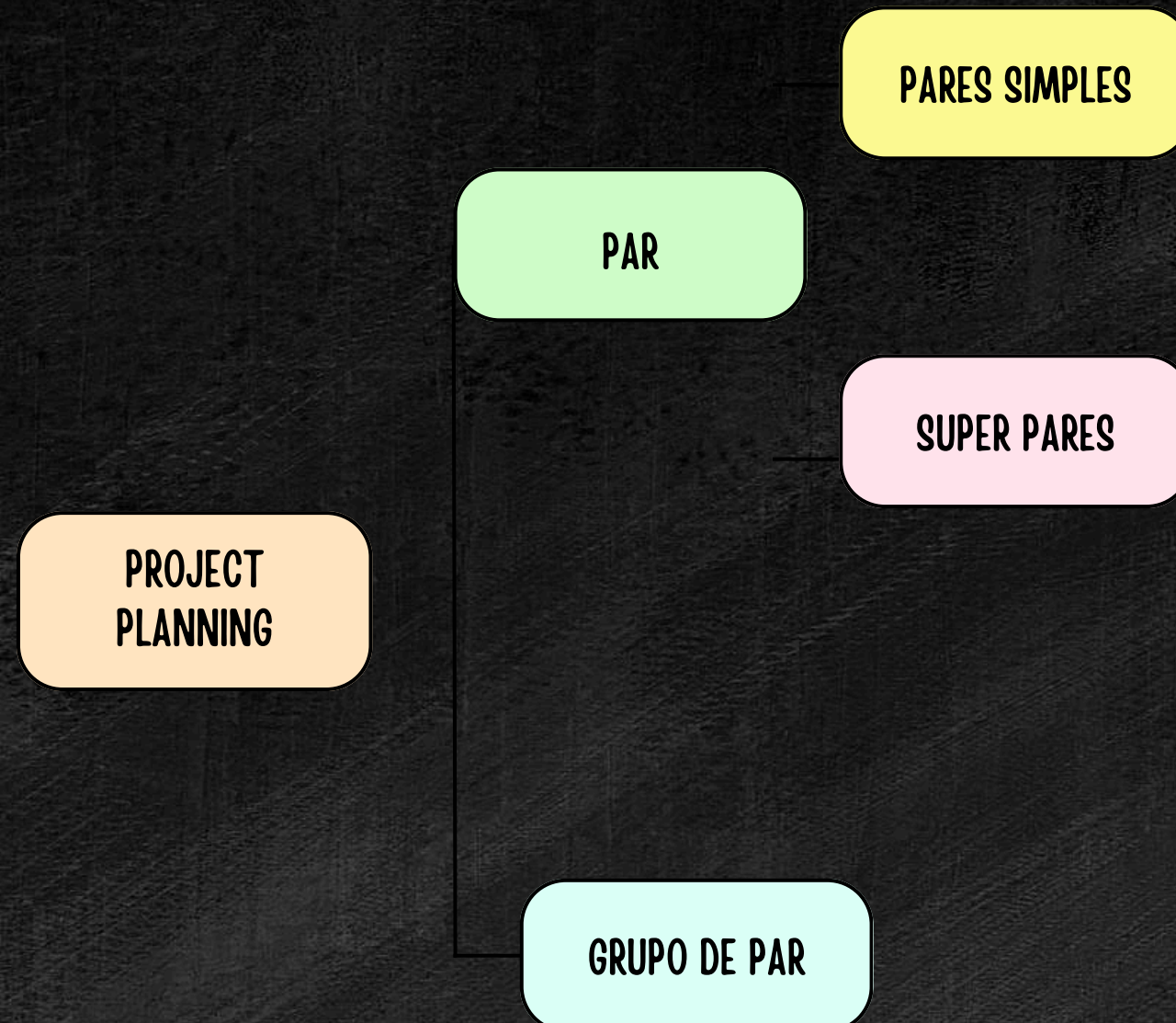


Estructura P2P

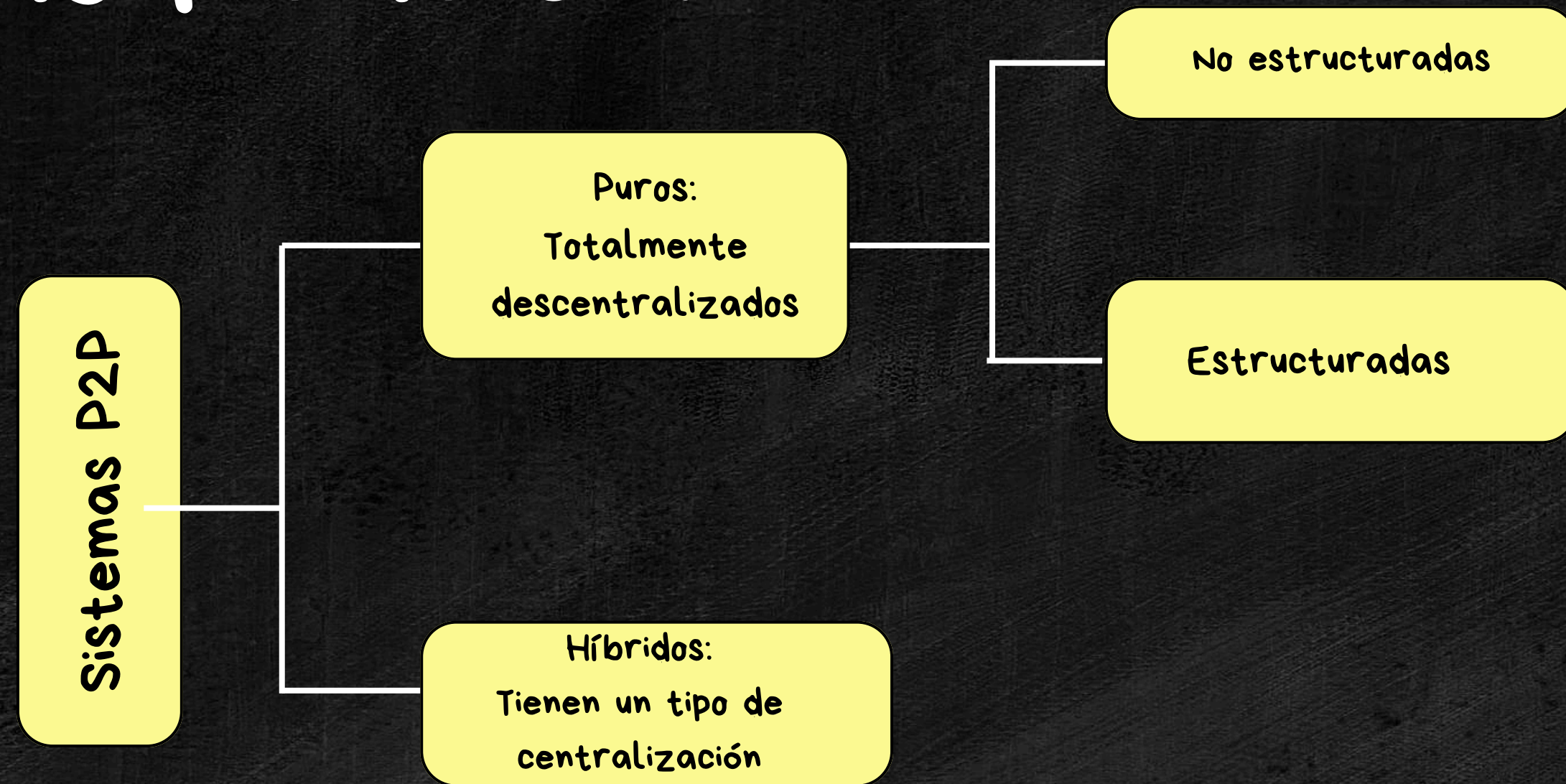


P2P

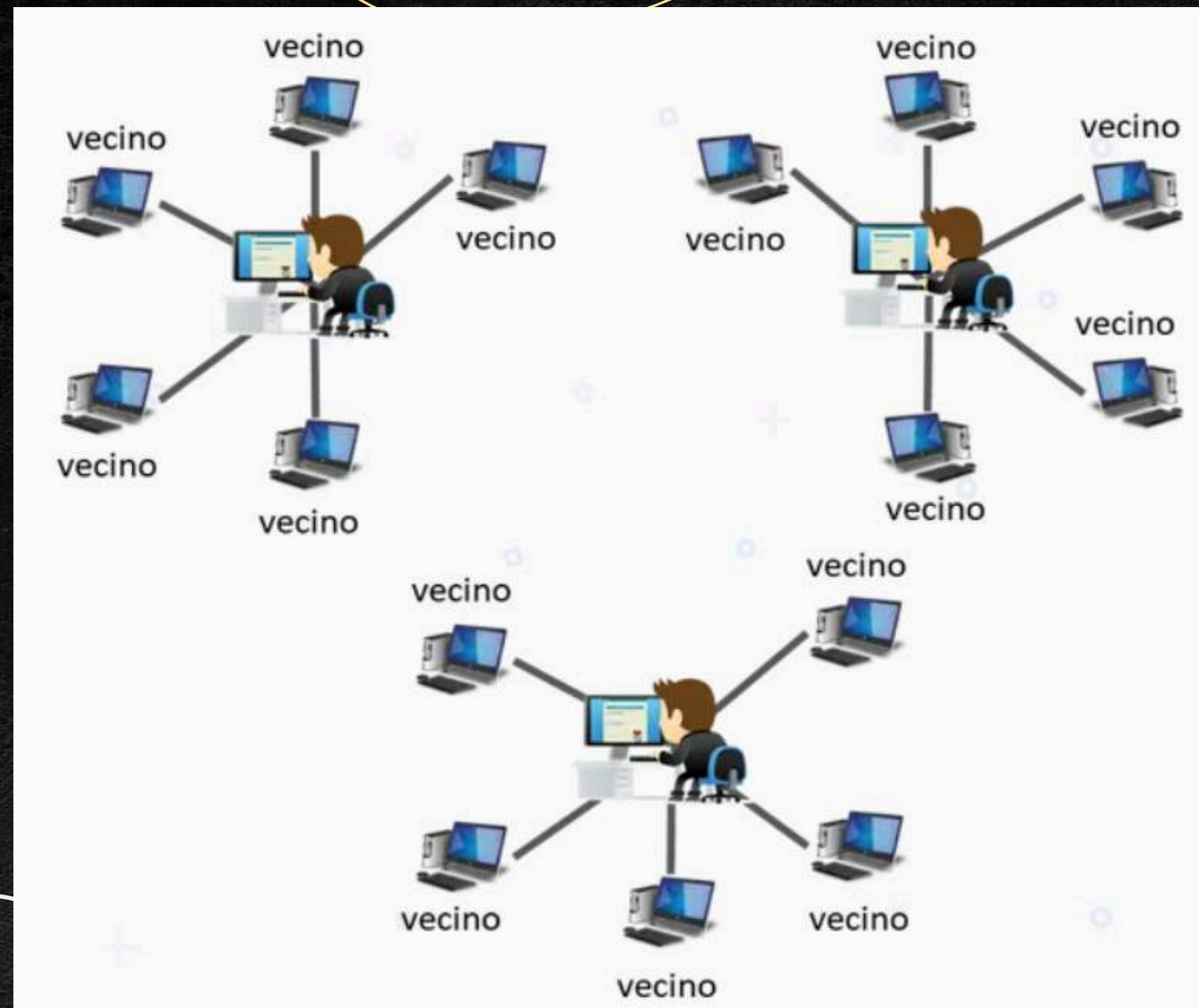
Elementos que la conforman



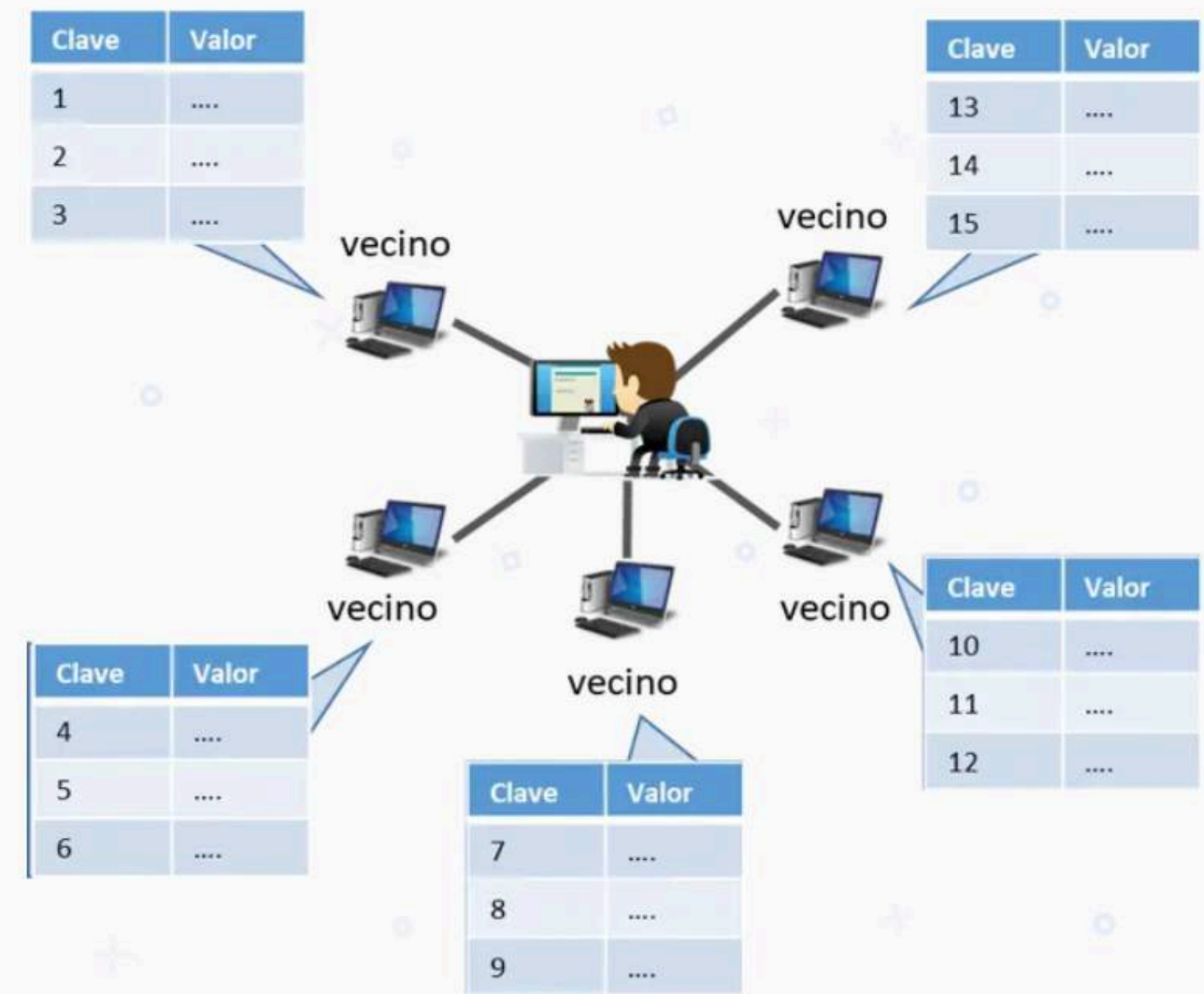
Clasificación:



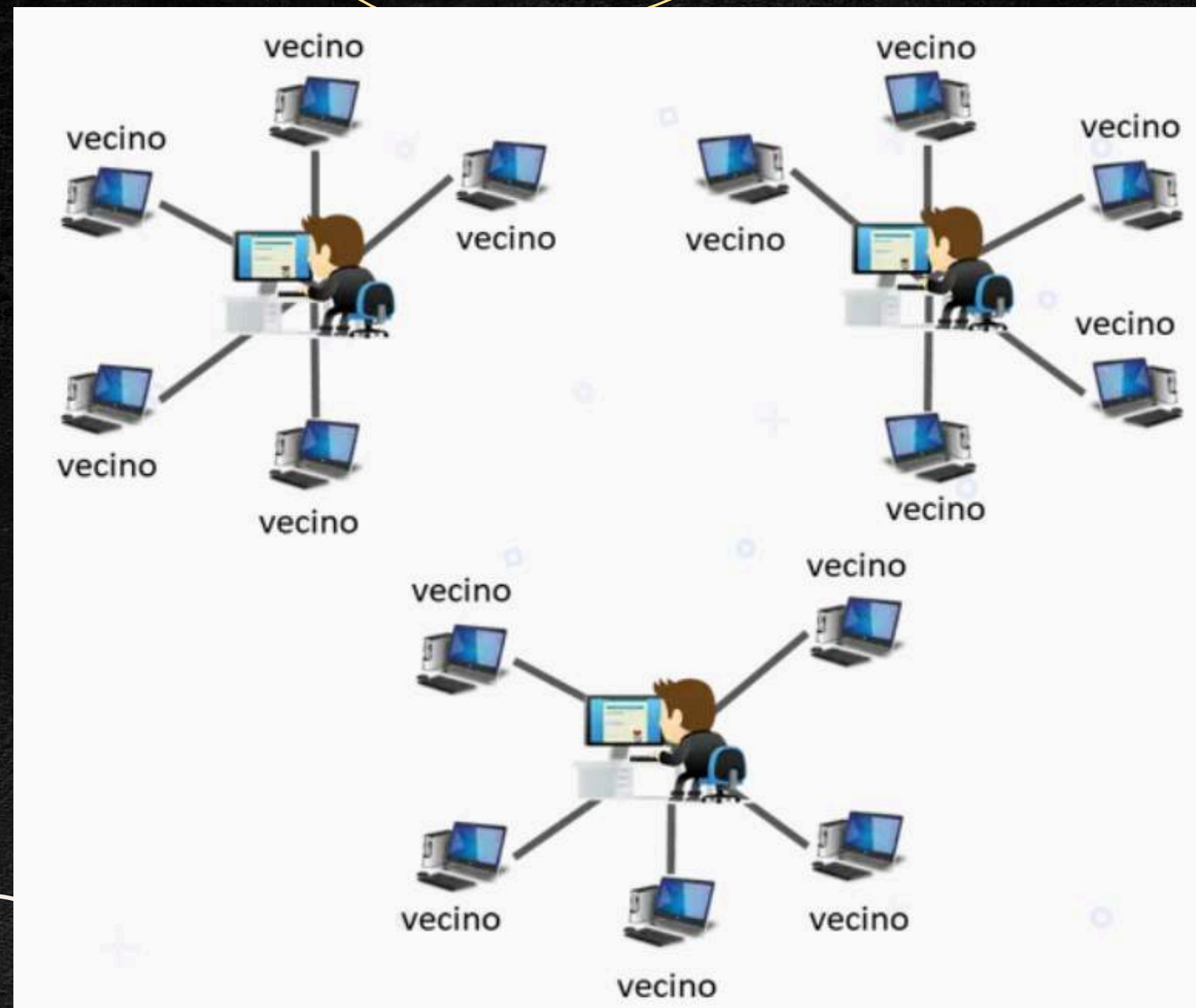
No estructuradas



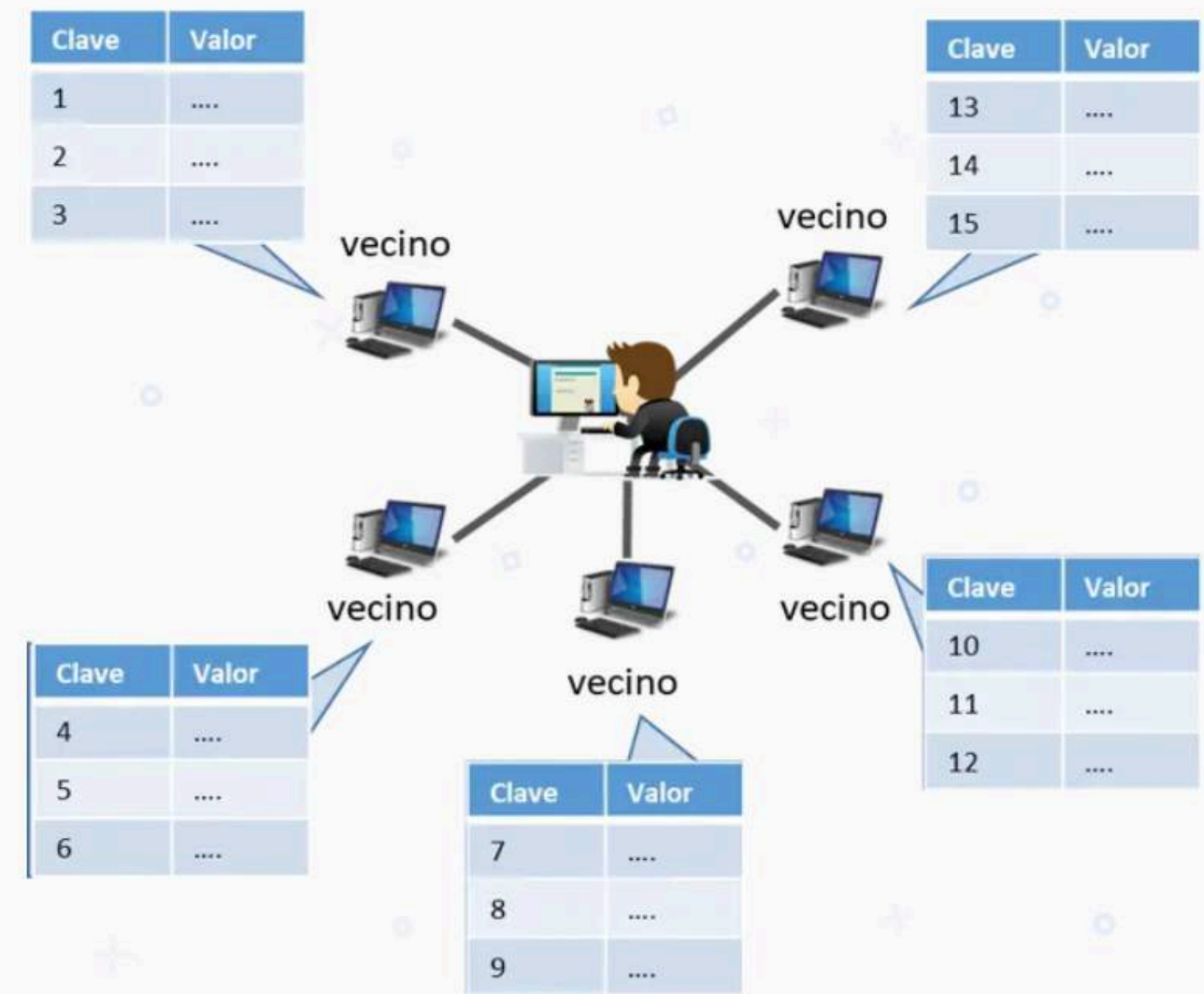
Estructuradas



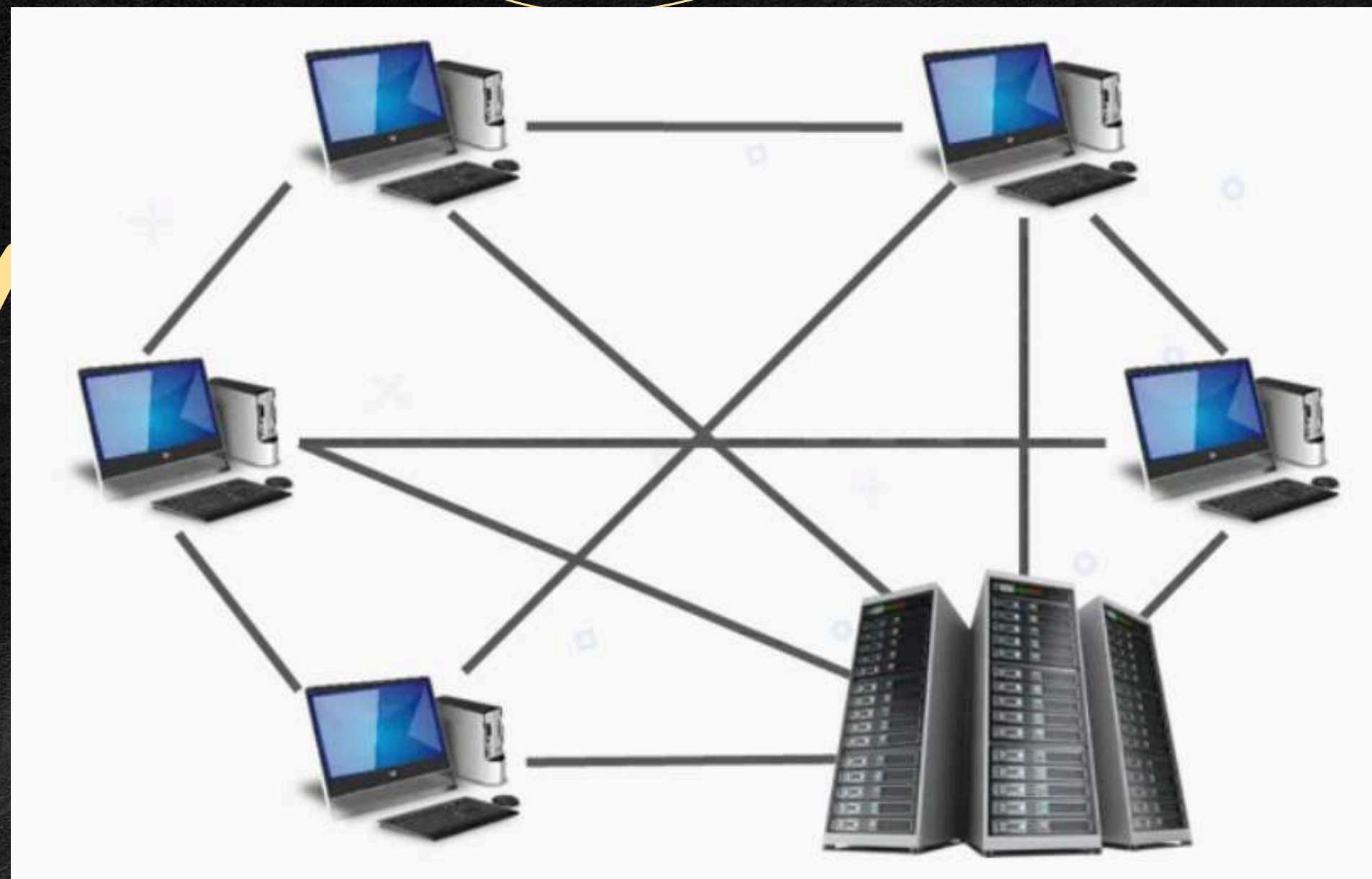
No estructuradas



Estructuradas



Híbrida



P2P

Evolución del sistema P2P






Napster



¿Qué es?



Napster fue un servicio pionero de intercambio de archivos peer-to-peer (P2P) el cual permitía a los usuarios compartir archivos de música de forma gratuita a través de una red descentralizada.



Napster

Historia



Fundado por Shawn Fanning y Sean Parker en 1999, representó una innovación revolucionaria en la distribución de archivos de música en línea. Este servicio transformó la manera en que las personas compartían y accedían a la música digital. Napster se destacó al utilizar servidores centralizados como vastos almacenes de archivos MP3, permitiendo a los usuarios acceder y descargar música de manera rápida y sencilla.

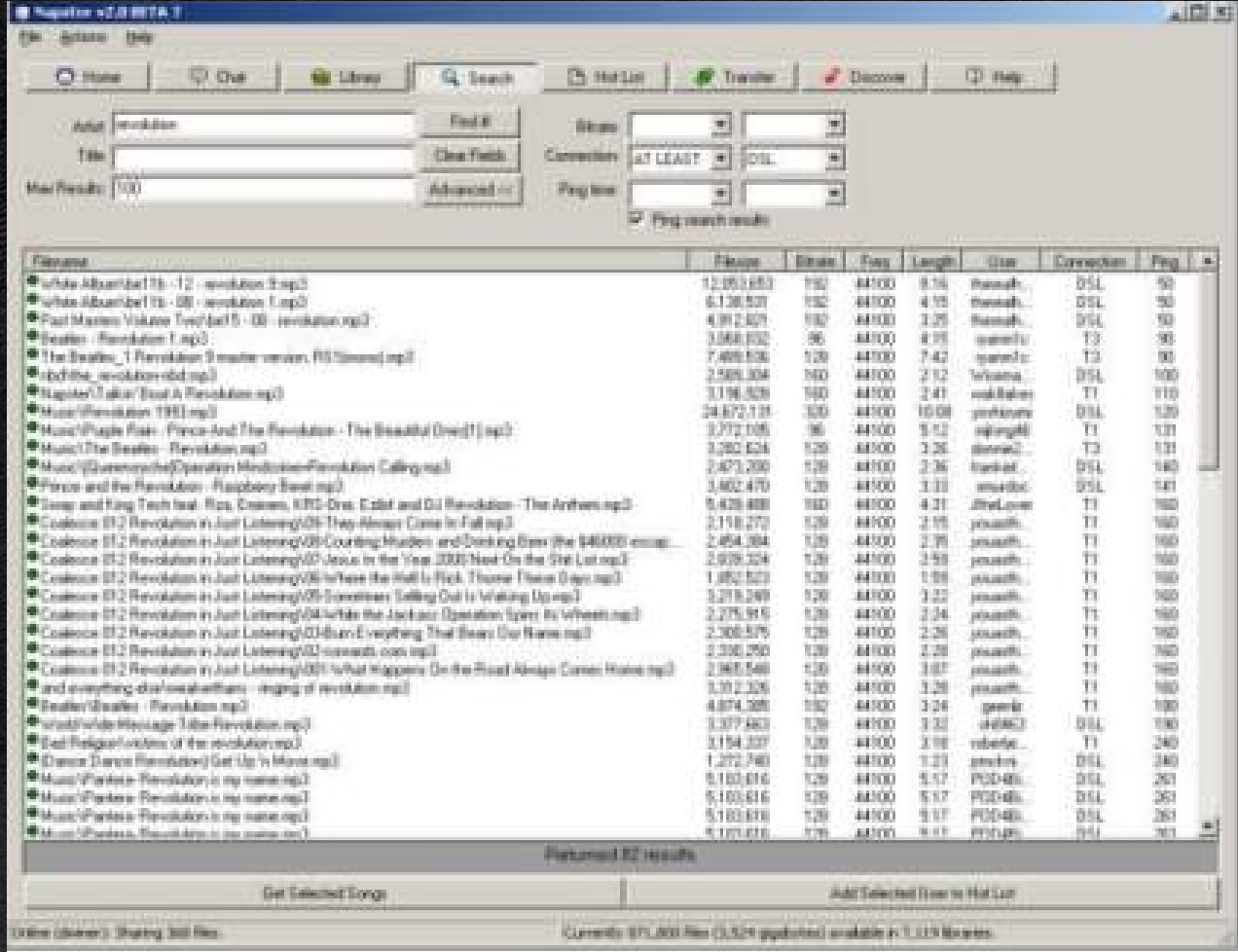
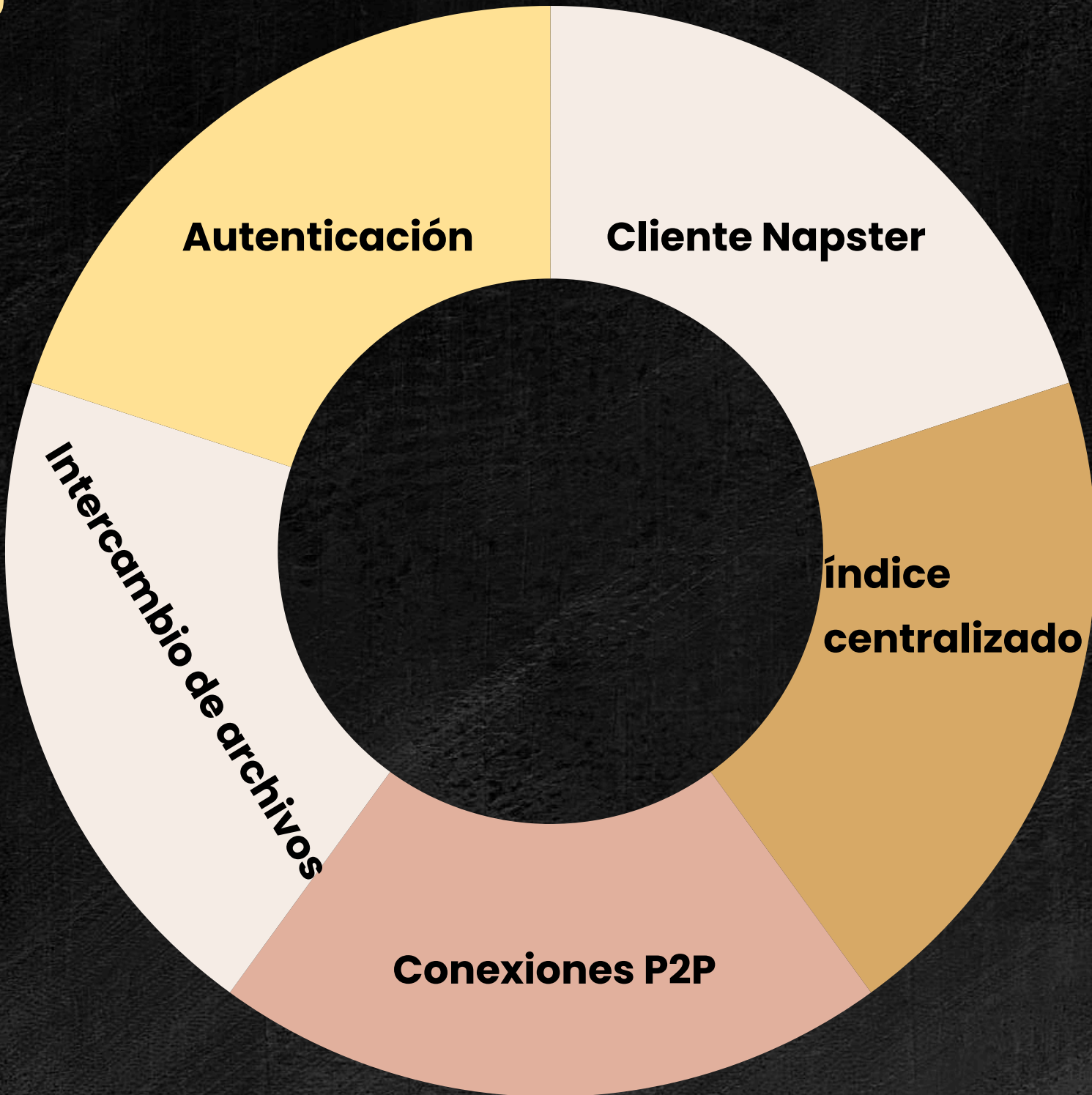
La esencia de Napster radicaba en su enfoque colaborativo: para descargar música, los usuarios debían compartir sus propios archivos MP3 con la comunidad. Este modelo desplazó a métodos anteriores como las aplicaciones FTP, IRC o Usenet, que resultaban engorrosas y lentas en comparación con la eficiencia del P2P.





Napster

Arquitectura






MIDDLEWARE P2P

¿Qué es?

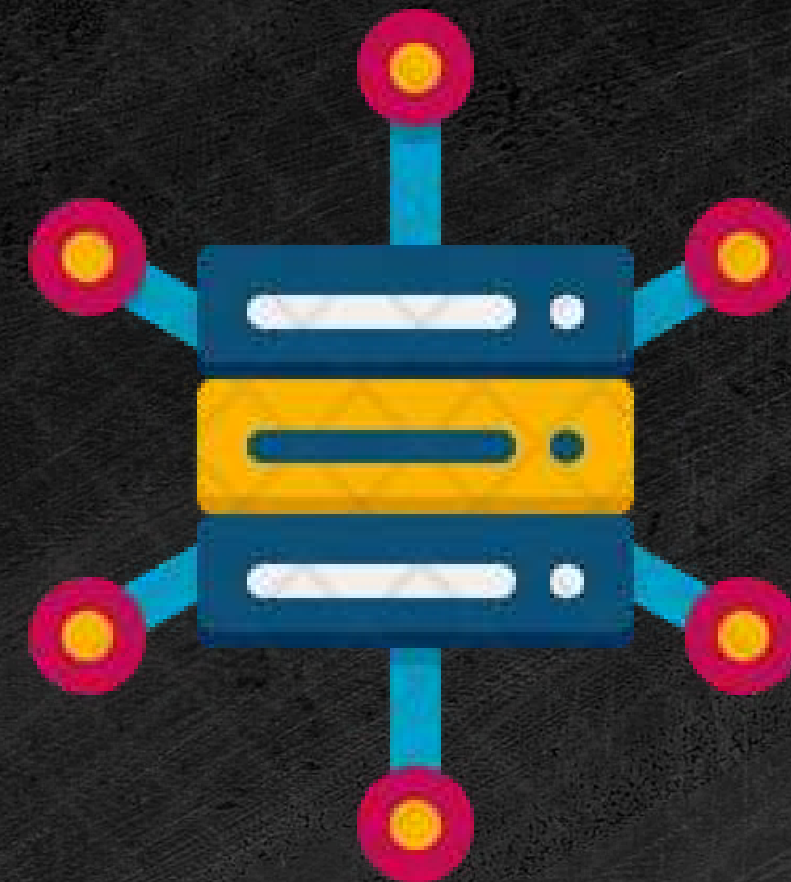
Es una capa de software distribuido que abstrae la complejidad y heterogeneidad de los sistemas P2P subyacentes con su multitud de servicios, protocolos y aplicaciones. Actualmente las aplicaciones P2P más importantes y populares son las de intercambio de datos. La capa de middleware implementa una abstracción fundamental por encima de los sistemas P2P subyacentes y proporciona servicios comunes de intercambio de datos para aplicaciones de alto nivel.



MIDDLEWARE P2P

Características

- Abstracción de la Red
- Gestión de Recursos
- Descubrimiento de Nodos
- Seguridad y Autenticación
- Gestión de Transacciones
- Escalabilidad
- Integración con Aplicaciones



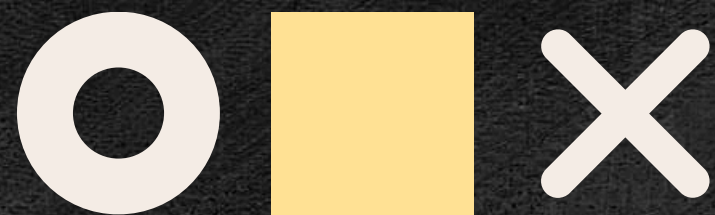


¿Qué es?

Es un protocolo de comunicación utilizado para compartir archivos de forma descentralizada.

Utiliza la capacidad de carga de los pares que están descargando un archivo.

Breve historia



BitTorrent fue creado en 2001 por Bram Cohen, un programador estadounidense frustrado por los largos tiempos de espera que experimentaba con aplicaciones que usaban otros protocolos como FTP. En 2002 lo presentó en una conferencia. Su objetivo era proporcionar a las personas una forma rápida y sencilla de distribuir e intercambiar software de Linux en línea. Sin embargo, pronto se vio el potencial de BitTorrent para la distribución de películas y programas de televisión, lo que llevó a un crecimiento explosivo.

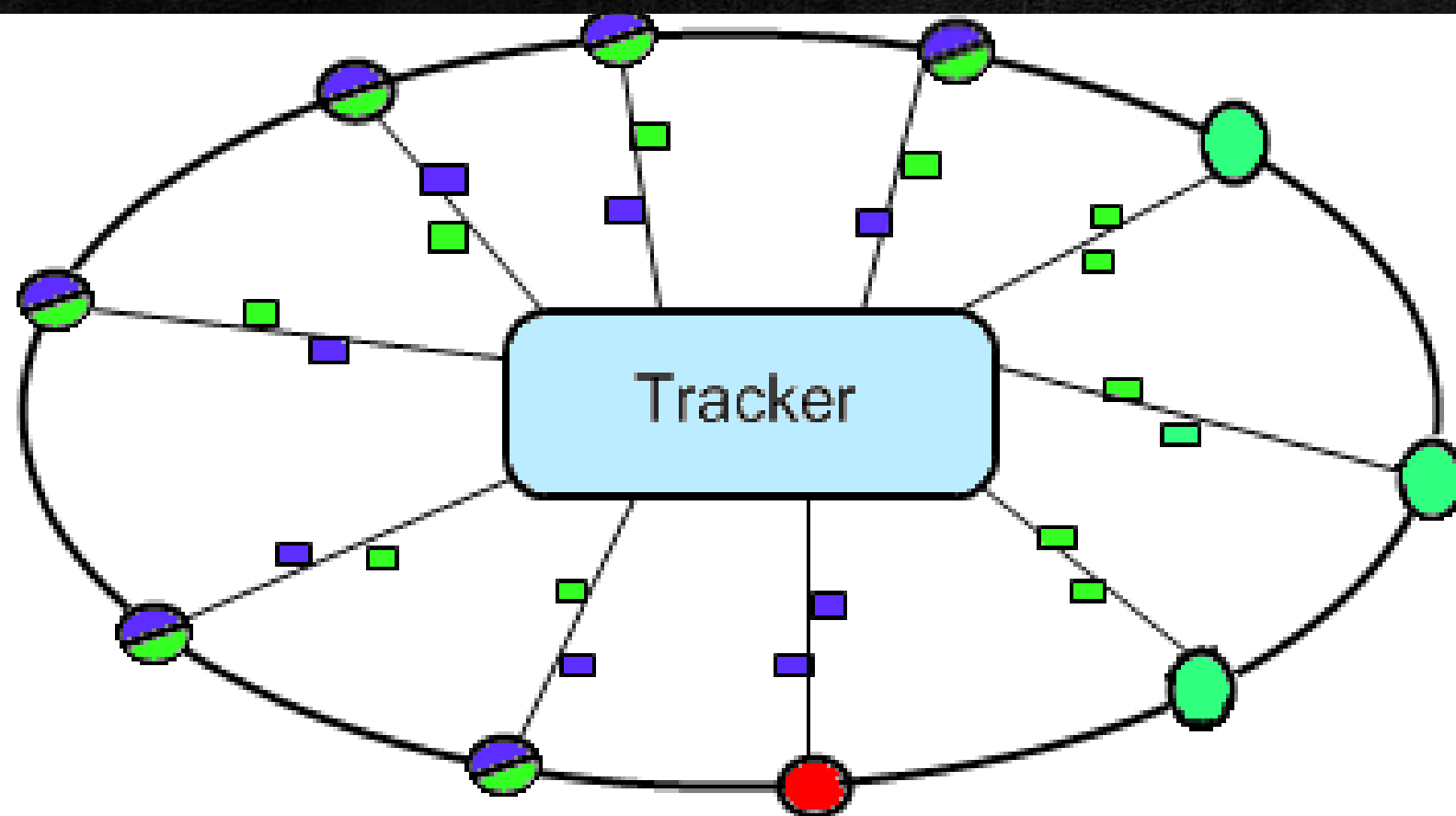


Cientes BitTorrent

Son aplicaciones de software diseñadas para interactuar con el protocolo BitTorrent y permitir a los usuarios descargar y compartir archivos a través del protocolo BitTorrent.

Mediante éstos se crean los archivos de metadatos que tras compartirse permitirán iniciar la descarga del archivo publicado. Alternativamente, una descarga se puede iniciar si se cuenta con el enlace magnético, el cual contiene la información suficiente para localizar el archivo





Legend

- Peer
- Seed
- Leech (BitThief)

Upload

Download

Funcionamiento

- El dueño del archivo original crea un archivo torrent o enlace magnético y lo comparte
- Abrir archivo torrent (o vínculo magnético)
- Tracker ayuda a pares a encontrarse
- A medida que el usuario descarga piezas del archivo, también se convierte en un par
- Este proceso se repite hasta que todos los usuarios hayan descargado el archivo
- Una vez que un usuario ha descargado todas las partes del archivo, puede optar por convertirse en una semilla

BOINC (Berkeley Open Infrastructure for Network Computing)



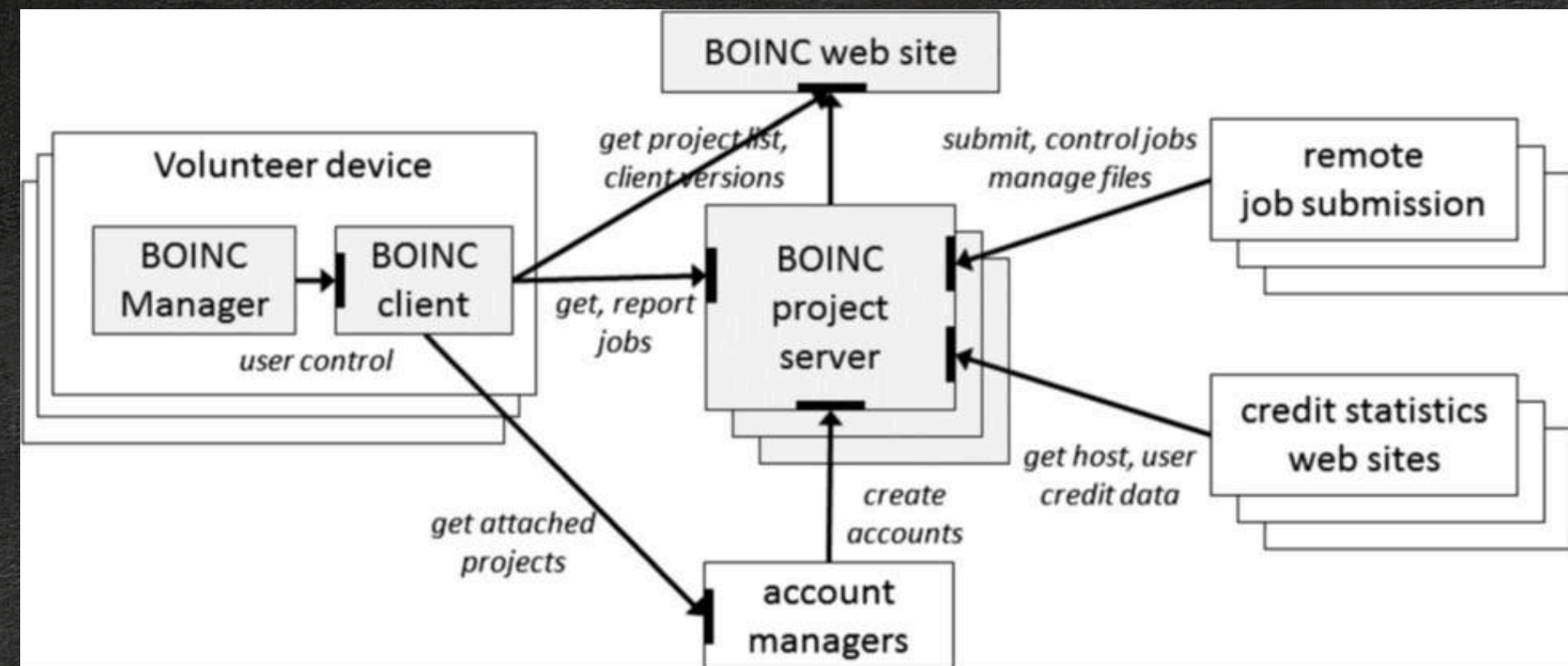
¿Qué es?

Es un sistema de software desarrollado en 2002 por un grupo basado en la Universidad de California, en Berkeley . Es utilizado para cómputo voluntariado: les permite a las personas donar tiempo en sus computadoras domésticas (Windows, Mac, Linux) y teléfonos inteligentes (Android) a proyectos de investigación científica.



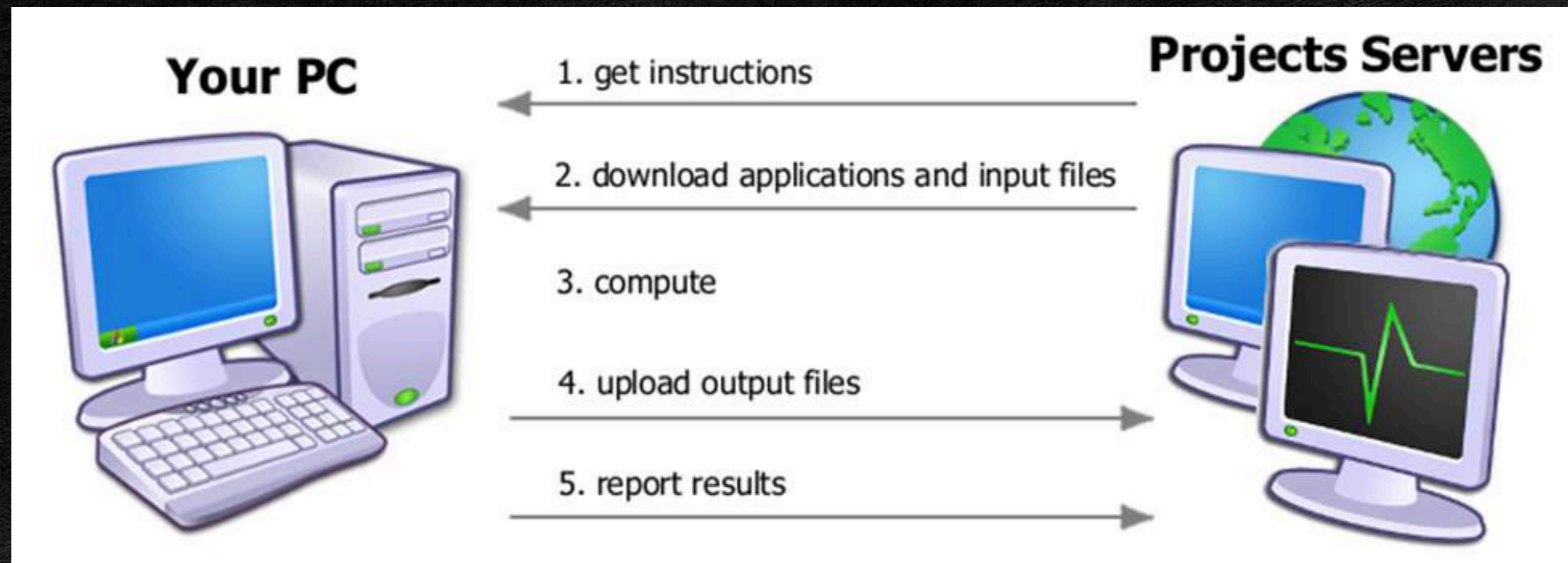
Componentes

- Servidor del BOINC
- Proyecto BOINC
- Dispositivo voluntario
 - Cliente BOINC



Voluntariado

101



1. Recibir instrucciones
2. Descargar aplicaciones y archivos de entrada
3. Computar
4. Cargar archivos de salida
5. Reportar resultados

Folding@Home

¿Qué es?

Es un proyecto de cómputo distribuido que comenzó en octubre de 2000 en la Universidad de Stanford, desde el 2019 hasta la actualidad está siendo administrado por el Dr. Gregory Bowman en la Universidad de Pennsylvania. Para esto hace uso de la computación distribuida, permitiendo a los voluntarios conducir experimentos diseñados y asignados por el equipo de Bowman desde Pennsylvania.

Enfermedades

Entre las enfermedades que el equipo de Bowman investiga se encuentran:

- Cáncer: Cáncer de mama, p53, epigenética, cáncer de riñón.
- Enfermedades infecciosas: Dengue, enfermedad de Chagas, virus del Zika, Hepatitis C, virus del Ébola, COVID-19.
- Enfermedades neurológicas: Enfermedad de Alzheimer, enfermedad de Huntington, enfermedad de Parkinson.



UNAM@Home



¿Qué es?

Fue un proyecto liderado por el M. C. Alejandro Velázquez Mena, actual jefe de la División de Ingeniería Eléctrica en la Facultad de Ingeniería de la UNAM. Haciendo uso de BOINC se implementó la plataforma UNAM@Home

Logros

El proyecto se llevó a cabo en un periodo de 2015 a 2017. En este tiempo se participó en 23 proyectos. También se creó el primer proyecto de la Facultad de Ingeniería, encontrar más dígitos del número de Euler y se apoyó al proyecto Serpent, que buscaba modelar partículas nucleares y analizarlas.

4110781

Es equivalente a casi la mitad de los caracteres empleados por Cervantes al escribir *Don Quijote* (dos millones 59 mil cinco) y poco más que los usados por Víctor Hugo en su novela *Los miserables*

El resultado se obtuvo con una estación de trabajo que corre *Windows 7*, conectado a la plataforma BOINC, que puede ser dos veces más veloz que la supercomputadora china *Tianhe-2*

🇲🇽 MENA_COMP_DIE_FI_UNAM

	Credits:	BSrac:	Rank:	Rank%:
Collatz	6,730,808,915	0	79	95.666
PrimeGrid	960,078,678	0	115	96.518
GPUGRID	505,505,825	0	156	91.545
Moo!	332,165,530	0	83	90.247
Asteroids	182,094,720	0	53	97.480
WCG	174,165,097	522	256	99.091
BitcoinUt	153,380,350	0	138	59.884
Einstein	153,146,829	0	435	96.394
MilkyWay	151,688,262	0	225	95.298
SETI	91,863,367	0	352	99.457
yoyo	24,922,962	0	57	95.605
SAT@home	22,427,365	0	11	97.566
Rosetta	21,311,942	0	357	97.185
POGS TSN	20,197,621	0	60	93.428
SIMAP	15,982,422	0	46	98.166
WUProp	13,406,652	0	25	96.580
Malaria	9,464,463	0	55	97.781
NmbrField	9,090,923	0	87	89.403
DENIS@Hom	6,522,065	0	47	90.656
ATLAS@Hom	3,821,182	0	18	96.498
LHC	3,821,182	0	238	95.748
LHC-dev	3,821,182	0	239	-251.471
EDGeS	1,024,252	0	53	92.251
CPDN	814,117	0	1,354	83.230
Acoustics	93,198	0	126	59.486
	9,587,797,930	63,165	140	99.875



Team Stats
boincstats.com

Hoy en día

Tras su finalización formal en 2017, la plataforma en su modo cliente sigue aportando a diversos proyectos bajo el nombre de equipo MENA_COMP_DIE_FI_UNAM, amasando un total 9,587,797,930 créditos BOINC. Esto lo coloca en segundo lugar a nivel país en cuanto a contribuciones a los proyectos de la infraestructura representando al 48.93% de las aportaciones de parte de México.

¡Muchas
gracias!