

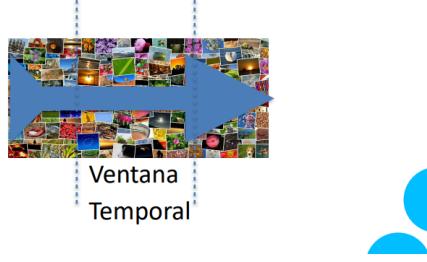
## 1 - QUÉ ES BD

Sistemas Big Data	Volumen	BD	UNIDADES	ESCALA
			<ul style="list-style-type: none"><li>Gigabytes <math>10^9</math> bytes</li><li>Terabytes <math>10^{12}</math> bytes</li><li>Petabytes <math>10^{15}</math> bytes</li><li>Exabytes <math>10^{18}</math> bytes</li><li>Zettabytes <math>10^{21}</math> bytes</li><li>Yottabytes <math>10^{24}</math> bytes</li><li>Universo <math>4 \times 10^{79}</math> átomos H</li></ul>	  100 PB 2012  44 ZB 2020 

Con ZB se puede medir la cantidad de información que se genera en el **mundo**. Incluye datos generados por redes sociales, dispositivos de IoT, plataformas de transmisión en línea, empresas, gobiernos, entre otros.

**Universo:** si comparamos esta explosión de datos con la cantidad máxima de información posible en el universo, entendemos que estamos manejando solo una fracción minúscula de la información teóricamente posible. Aún así, la cantidad de datos generados está creciendo rápidamente y podría parecer "astronómica" en comparación con lo que solíamos gestionar.

**Los SBD están hablando de cantidades que van desde los Gigabytes hasta los PetaBytes.**

Sistemas Big Data	Velocidad
	 900 millones fotos/día 

Los datos se están generando en cada momento. Debemos tener una **capacidad de cómputo elevada**, de manera que podamos procesarlos en un tiempo dado.

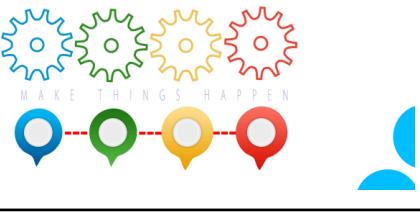
Fotos en Facebook: requiere procesar esa cantidad de datos en un día.

Debemos ser capaces de procesar una **Cantidad de datos/Unidad de tiempo**

<p><b>Sistemas Big Data</b></p> <p><b>Variedad</b></p> <ul style="list-style-type: none"> <li>■ Ficheros XML</li> <li>■ Ficheros log</li> <li>■ Bases de datos</li> <li>■ Imágenes</li> <li>■ Audios</li> <li>■ Videos</li> <li>■ Webs</li> <li>■ Textos</li> </ul> 	<p>Hay una gran variedad de fuentes y de formatos. Es decir, de dónde recibimos la información y de cómo la recibimos.</p> <p>La gran variedad impone un reto que consiste en extraer de cualquier fuente lo que nos están transmitiendo.</p>
---	---

### Resto de V's del Big Data

- **Valor:** "lo que importa, son los datos importantes", los que aportan valor, transformando datos en información y, a su vez, en conocimiento que facilita la toma de decisiones.
- **Veracidad:** debemos asegurar que los datos que tenemos son reales y no contienen datos erróneos, es decir, que son fiables. Se dedica un esfuerzo importante en explorar y validar los datos para que la analítica realizada sea veraz.
- **Viabilidad:** es necesario saber la capacidad que tiene una empresa para realizar un uso eficaz de los datos, cuestionarse qué y cuántos datos se necesitan para predecir los resultados más interesantes para la empresa.
- **Visualización:** necesitamos poder representar los datos, ya sea de manera visual mediante gráficos o codificados para hacer que sean legibles y accesibles.
- **Vulnerabilidad:** se refiere a la seguridad y privacidad de los datos en grandes volúmenes. A medida que las organizaciones manejan más datos, aumenta el riesgo de filtraciones, ciberataques y violaciones de privacidad. La vulnerabilidad enfatiza la necesidad de implementar medidas robustas para proteger los datos sensibles de accesos no autorizados o del mal uso.

<p><b>Sistemas Big Data</b></p> <p><b>Modelo de negocio BD</b></p> <ul style="list-style-type: none"> <li>▪ Usuarios de datos</li> <li>▪ Proveedores de datos</li> <li>▪ Facilitadores de sistemas BD</li> </ul> 	<p><b>Modelo de negocio Big Data</b></p> <p>Las empresas están <b>explorando</b> diferentes formas de <b>crear valor</b> a partir del <b>análisis y gestión de grandes volúmenes</b> de datos.</p> <p><b>Clasificación de empresas según su modelo de negocio basado en Big Data (MNBD)</b></p>
--	---

<p><b>Sistemas Big Data</b></p> <p><b>MNBD: Usuarios de datos</b></p> <ul style="list-style-type: none"> <li>▪ Toma de decisiones estratégicas</li> <li>▪ Mejorar los procesos internos</li> <li>▪ Enriquecer productos, servicios y experiencia de clientes</li> <li>▪ Desarrollo de <b>nuevos</b> productos y servicios</li> </ul> 	<p><b>Usuarios de datos</b></p> <p>Son empresas que utilizan los datos para <b>mejorar sus procesos o productos</b>, capturando valor a través de la aplicación de Big Data en <b>su propia organización</b>.</p> <p>Ejemplo: Una tienda online que utiliza Big Data para analizar el comportamiento de compra de los clientes y personalizar las recomendaciones.</p>
---	--

## Sistemas Big Data

MNBD:  
Proveedores de datos

- **Recopilando** datos primarios y vendiéndolos a terceros
- **Agregando** datos y **empaquetando** datos internos para la venta



### Proveedoras de datos

Estas empresas se especializan en recopilar, empaquetar y vender datos a terceros. Crean valor generando datos propios o agregando datos de múltiples fuentes y luego comercializándolos.

Ejemplo: Empresas que venden información de salud de dispositivos portátiles (como relojes inteligentes) a empresas aseguradoras o de investigación.

## Sistemas Big Data

MNBD:  
Facilitadores

- Ofreciendo la **infraestructura**
- **Consultoría**
- Subcontratación de **técnicas analíticas**

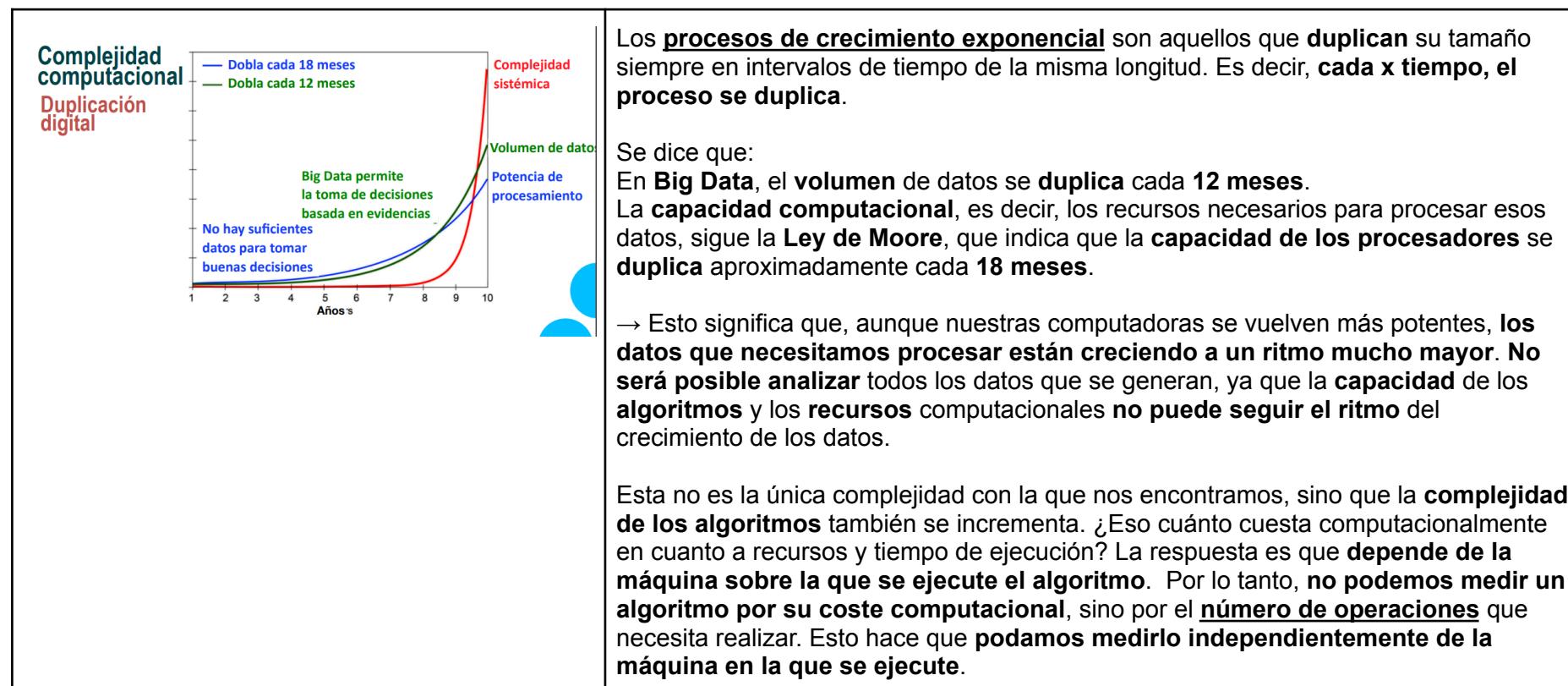


### Facilitadores de Big Data

Empresas que proveen la **infraestructura tecnológica** y **servicios** necesarios para que otras empresas implementen soluciones de Big Data. Ellos no necesariamente trabajan directamente con los datos, sino que **facilitan el procesamiento, almacenamiento y análisis** de los mismos.

Ejemplo: Proveedores de infraestructura en la nube como Amazon Web Services (AWS) o Microsoft Azure, que alquilan servidores y servicios para el procesamiento de datos a gran escala.

## 2 - COMPLEJIDAD COMPUTACIONAL



## Complejidad computacional

Tiempo de ejecución

### Ordenación de un vector

(método burbuja)

32    3    17    ...

n tamaño del vector

$f_{\mathcal{A}}(n)$  número de operaciones de  $\mathcal{A}$

$$f_{\mathcal{A}}(n) \leq an^2 + bn + c$$

$$\mathcal{A} \equiv O(n^2)$$



$O(n^2)$  = donde O significa **tiempo de ejecución del algoritmo**.

## Notación Big O

## Complejidad computacional

Crecimiento polinomial y exponencial

$$f_{\mathcal{A}}(n) \rightarrow +\infty$$
$$n \rightarrow +\infty$$

$$\lim_{n \rightarrow +\infty} \frac{P(n)}{a^n} = 0, \quad a > 1$$



Por tanto cualquier algoritmo tiene la propiedad que si **n (tamaño) crece** hacia infinito, su **coste computacional también crece** hacia infinito

La cuestión esencial es que **existen varias velocidades de crecer a infinito**, unas significativamente más rápidas que otras. La relación matemática nos indica que **la velocidad de crecimiento hacia infinito de un polinomio es mucho menor que la velocidad de una función exponencial**, que es mucho mayor. podemos verlo en la expresión del límite cuando tiende a infinito.

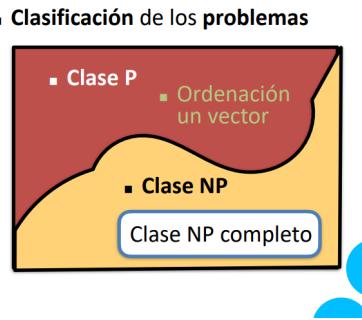
La expresión está comparando el **crecimiento de un polinomio P(n)** con el **crecimiento de una función exponencial a<sup>n</sup>** conforme n se acerca al infinito.

- $P(n)$ : representa un polinomio de grado n cualquiera. Por ejemplo:  $P(n) = 3n^2 + 2n + 1$
- $a^n$ : representa una función exponencial. En este caso, a es una constante mayor que 1, y n es la variable que se eleva a la potencia a. Si a = 2, entonces  $a^n=2^n$ , que crece muy rápidamente conforme n aumenta.

Conclusión: el denominador crece muchísimo más rápido que el numerador. Por lo tanto, el límite cuando n tiende a infinito, es 0. Por ejemplo, si divides 1 entre 900.000.000, estará cerca de 0.

## Complejidad computacional

Clase P/NP



Esto nos va a determinar 2 tipos de problemas:

- **Tipo P (tiempo polinomial).**
- **Tipo NP (tiempo polinomial no determinista).** Son problemas de los que **no conocemos algoritmos cuyo tiempo de ejecución sea polinomial**. Sin embargo, si nos dan la solución, se puede verificar en tiempo polinomial.
  - **Tipo NP-completo:** se refiere a problemas que son los más difíciles dentro de NP, y **no se ha encontrado un algoritmo eficiente para resolverlos**, pero no significa que sean irresolvibles. Si uno de estos problemas tuviera un algoritmo de resolución polinomial entonces todos los problemas de la clase NP también tendrían un algoritmo polinomial. En dicho caso **P = NP**.

## Complejidad computacional

Clase P/NP

¿ $P = NP$ ?



Clay Mathematics Institute

### Relación entre P y NP

Todos los problemas en P están también en NP, porque si puedes resolver un problema en tiempo polinomial, también puedes verificar la solución en tiempo polinomial (simplemente ejecutando el algoritmo de resolución).

¿Todos los problemas en NP están en P? Esta es la gran pregunta no resuelta en la teoría de la computación:

Se cree que  $P \neq NP$ , lo que significa que **no todos los problemas en NP se pueden resolver en tiempo polinomial**. Sin embargo, esto no ha sido probado formalmente.

Si se pudiera encontrar un algoritmo polinomial para un problema NP-completo, eso implicaría que  $P = NP$ , lo que significa que todos los problemas en NP también se podrían resolver en tiempo polinomial.

## Complejidad computacional



Actividad

Considerar ordenación de un vector de  $n=1000$

Método	Operaciones
Método burbuja $O(n^2)$	1 millón
Ordenación por mezcla $O(n \log_{10} n)$	3000
Ordenación Radix (recursiva) $O(n^3)$	1000 millones

## Actividad

Según el algoritmo que usemos para resolver un problema, podemos pasar drásticamente de una complejidad computacional a otra totalmente distinta, como podemos ver en este ejemplo, donde al resolver un mismo problema, el de ordenar un vector de tamaño 1000, la ordenación por mezcla se resuelve en un máximo de 3000 operaciones mientras que la ordenación Radix lo hace en 1000 millones de operaciones.

## 3 - COMPLEJIDAD COMPUTACIONAL Y ESCALABILIDAD

### Complejidad Escalabilidad



Actividad

Sistema criptográfico de clave pública RSA (Rivest, Shamir y Adleman)

$$n = p \cdot q$$

$p, q$  primos

$$33 = 3 \cdot 11$$

La seguridad del algoritmo RSA se basa en la dificultad de **factorizar números grandes**. Para romper la seguridad de RSA, un atacante necesitaría factorizar  $n$  (producto de los dos primos grandes  $p$  y  $q$ ), lo cual es extremadamente difícil y requiere mucho tiempo computacional para números suficientemente grandes.

## Complejidad Escalabilidad

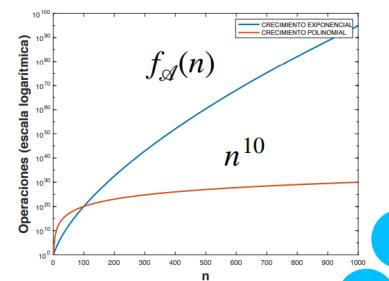


### Algoritmo factorizar en primos estado-del-arte

$$f_A(n) = 10^{\sqrt{n \cdot \log_{10}(n)}}$$

## Complejidad Escalabilidad

### Crecimiento polinomial/exponencial



## Complejidad Escalabilidad



### Algoritmo factorizar en primos estado-del-arte

X=123456789012345678901234567890

X tiene n=30 dígitos

$$\begin{aligned}f_A(30) &= 10^{\sqrt{30 \cdot \log_{10}(30)}} \\&= 1.2318 * 10^{08}\end{aligned}$$

## Actividad (Complejidad)

El algoritmo para resolver la factorización de un número en primos (estado del arte, es decir, la mejor técnica que existe hoy en día), tal y como podemos ver en la diapositiva, este número de operaciones.  
 $f_A(n) = 10 \text{ RAIZ } (n \cdot \log^{10}(n))$

Esto tiene apariencia de ser **exponencial** (ya que la base es 10 y las variables están en el exponente), por lo que su crecimiento es muy rápido. Como consecuencia, como puede verse en esta gráfica de escala logarítmica, resolverlo a través de algoritmos sería **computacionalmente muy complejo**, obteniendo un problema que se sitúa en el tipo **NP-completo**.

Imaginemos un problema que consiste en factorizar un número de 30 dígitos. El **tamaño del problema** es la **cantidad de números** que tiene, por lo que en este caso sería n=30. Según el mejor algoritmo, que hemos visto antes, daría un total de más de 123 millones de operaciones, que son muchísimas.

¿Esto sería descifrable por un ordenador? ¿existe esta capacidad computacional?

Nos planteamos hasta qué tamaño de clave RSA seríamos capaces de romper con un ordenador muy potente.

ACTIVIDAD: Investiga qué es (o qué fue) el Roadrunner de IBM

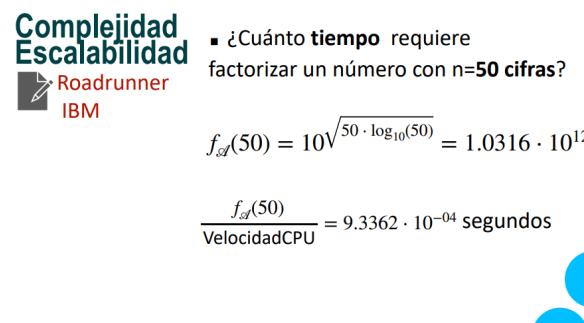


El Roadrunner de IBM (correcaminos) alcanzó realizar 1000 billones de operaciones por segundo. En el año 2008 era el nº 22 en el mundo.

En menos de 1 milésima de segundo (concretamente en 0.93362 milésimas de segundo) sería capaz de factorizar un número con 50 dígitos.



Como sabes, la velocidad de un ordenador se mide en cuántas operaciones de coma flotante puede realizar por segundo. La velocidad del Correcaminos es de 1.1025 Petaflops.



Vamos a calcular cuánto tiempo tarda el Correcaminos en resolver el factorial de un número de 50 cifras.

## Complejidad Escalabilidad



- ¿Qué tamaño máximo  $n$  se puede factorizar en un año?

$nOp = \text{VelocidadCPU} * \text{segundos en un año}$

$$f_{\mathcal{A}}(n) \leq nOp$$



## Complejidad Escalabilidad

costeComputacional.py

```
from math import *
VelocidadCPU = 1.105 * 10 ** 15
n=1
fA=1
nOp=VelocidadCPU *24 * 60 * 60 *365
while fA <= nOp:
    n = n+1
    fA =10** ( pow(log(n,10),0.5))
print(n-1)
```



## Complejidad Escalabilidad



- ¿Qué tamaño máximo  $n$  se puede factorizar en un año?

▪  $n = 2^{17}$

▪  $2 * \text{VelocidadCPU}$

▪  $n = 2^{17} + 5$



Ahora vamos a calcular cuántas operaciones puede realizar Correcaminos en 1 año y descubrir cuántos dígitos puede tener un número como máximo que pueda factorizar en un año por este superordenador. Para ello se ha creado el código en python (siguiente diapo).

Como resultado obtenemos el decepcionante resultado de **217 dígitos**. Un año entero trabajando. En comparación con el poco tiempo que tardó en calcular el factorial el número de 50 cifras.

Planteamos ahora tener **2 Correcaminos**. Modificamos el código en python y da como resultado **222 dígitos**... menuda decepción.

## Complejidad Escalabilidad



75 millones Euros

- 1234567890123456789012345678901  
2345678901234567890123456789012  
3456789012345678901234567890123  
456789012345678901234567



150 millones Euros

- 1234567890123456789012345678901  
2345678901234567890123456789012  
3456789012345678901234567890123  
45678901234567890123456712345

Viendo las infraestructuras y su precio... ¿merece la pena la inversión? La primera es para 217 dígitos y la segunda para 222.

Podemos resumir que en la escalabilidad intervienen 2 aspectos:

- la máquina
- la complejidad de los algoritmos

## 4 - DIVIDE Y VENCERÁS

La **computación distribuida** es un modelo de procesamiento en el que múltiples computadoras, ubicadas en diferentes lugares, **trabajan juntas** para resolver un problema o ejecutar tareas de forma coordinada. En este enfoque, los recursos de hardware y software están distribuidos entre varios sistemas, pero actúan como un único sistema para el usuario final.

### Sistemas Big Data



- Coste exponencial es inabordable
- A efectos prácticos  $O(n^3)$  no es escalable
  - n=1000 números float

#### Memoria

- 1000 datos\*64 Bytes
- 1MB=1.048.576 Bytes
- 0.06 MB



La **computación distribuida** es buena porque aparte de tener **más recursos computacionales**, estos **operan sobre cantidades más pequeñas de datos**.

Cuando el crecimiento es exponencial, no son aplicables a problemas de un tamaño moderado.

Sin embargo, cuando tenemos un problema polinomial, tampoco es escalable. Veamos un ejemplo para la complejidad de 1000 números.

En cuanto al **almacenamiento** de los números en **memoria**, es posible. Su tamaño es casi insignificante.

## Sistemas Big Data



Motivación

Coste exponencial es inabordable

- A efectos prácticos  $O(n^3)$  no es escalable
- $n=1000$  números float

### Número de operaciones

- $1000^3 = 1000 * 10^6 =$
- 1000 millones de operaciones

En cuanto al **coste computacional**, si el número de operaciones a realizar es  $n^3$ , es decir,  $1000 \times 1000 \times 1000$ , hay que realizar un total de 1000 millones de operaciones, lo que ya es número bastante considerable.

## Sistemas Big Data



Motivación

Coste exponencial es inabordable

- A efectos prácticos  $O(n^3)$  no es escalable
- $n=10000$  números float

### Número de operaciones

$$10000^3 = (10 \cdot 1000)^3 = 10^3 \cdot 1000^3 \\ = 1 \text{ billón de operaciones}$$

Si aumentamos el número de operaciones, multiplicándolo por 10, ¿también se multiplica por 10 el número de operaciones? En este caso **no**, se multiplica por 1000. **Tenemos 1 billón de operaciones**.

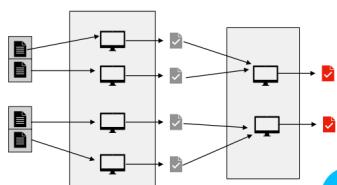
Por lo tanto, este procedimiento no sería escalable.

## Sistemas Big Data



Divides y vencerás

Partitionamiento datos      Resultados intermedios      Output



Nos planteamos ahora ejecutar este algoritmo en un **sistema distribuido** como el que aparece en la figura. Es decir, partir los 1000 datos y ejecutar cada paquete en distintos ordenadores. Cada ordenador produce un resultado intermedio y estos se unen en el resultado final.

Esta es la base de Sistemas Big Data basados en MapReduce.

**MapReduce** es un modelo de programación distribuido que divide el procesamiento de grandes volúmenes de datos en 2 fases principales:

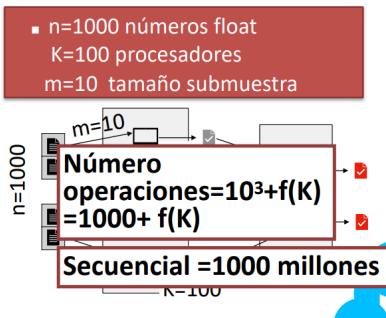
1. **Map (Mapeo)**: la tarea se divide en partes más pequeñas y se distribuye a diferentes nodos (ordenadores) para su procesamiento. Cada nodo trabaja en su propio

- subconjunto de datos y produce un **resultado intermedio**.
2. **Reduce (Reducción)**: Los resultados intermedios producidos por los nodos se combinan (reducen) para obtener el **resultado final**.

Este enfoque es muy utilizado en sistemas Big Data, como **Hadoop**, para **procesar grandes conjuntos de datos en un entorno distribuido**.

## Sistemas Big Data

Computación distribuida



Planteamos de nuevo el problema con 1000 números pero, en este caso, tenemos 100 máquinas, por lo que la cantidad de datos que va a cada máquina es de 10. ([ver imagen de moodle “100 procesadores”](#))

Aunque estén trabajando 100 procesadores, como lo hacen a la vez, el tiempo computacional cuenta como 1 sólo. Por ello, el número de operaciones sería  $10^3$  más  $f(K)$ , que es el coste de fusionar los resultados intermedios (en este caso  $f(k)$  valdría 100 porque son 100 resultados intermedios).

De este modo, **gracias al modelo distribuido**, hemos pasado de 1 billón de operaciones a 1000 operaciones más lo que cueste fusionar los datos.

Hay que tener en cuenta que **no todos los algoritmos son paralelizables**, o lo que es lo mismo, **no pueden resolverse en el modelo distribuido**.

## Sistemas Big Data

Cálculo Media

$$\bar{x} = \frac{x_1 + \dots + x_n}{n}$$

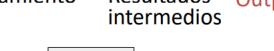
41 | 9 | 32 | 6 | 12

$$\bar{x} = \frac{41 + 9 + 32 + 6 + 12}{5} = \frac{100}{5} = 20$$

Veamos otro ejemplo: calcular la media de elementos.

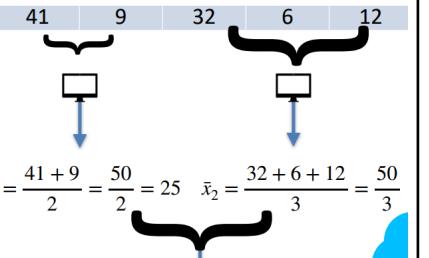
Aquí lo vemos resuelto de **manera secuencial**.

Vamos a plantearlo en un **sistema distribuido**.

Sistemas Big Data	Particionamiento datos	Resultados intermedios	Output	<p>Tenemos 2 ordenadores, donde distribuimos los datos. En primer lugar, calculamos las medias de los dos grupos de datos y, con los resultados obtenidos intermedios, calculamos las medias de las medias, obteniendo así el resultado final.</p> 
----------------------	---------------------------	---------------------------	--------	--

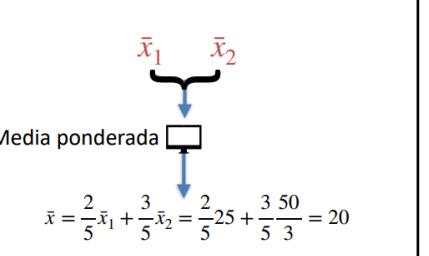
**Sistemas Big Data**

 Cálculo Media


$$\bar{x}_1 = \frac{41 + 9}{2} = \frac{50}{2} = 25 \quad \bar{x}_2 = \frac{32 + 6 + 12}{3} = \frac{50}{3}$$

**Sistemas Big Data**

 Cálculo Media

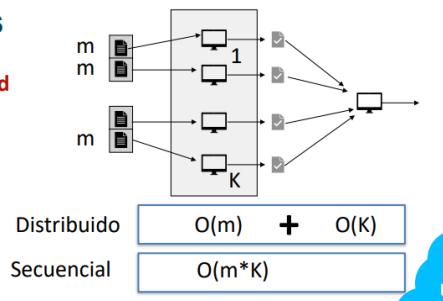

$$\bar{x} = \frac{2}{5}\bar{x}_1 + \frac{3}{5}\bar{x}_2 = \frac{2}{5}25 + \frac{3}{5}\frac{50}{3} = 20$$

Aquí vemos como en el primer ordenador hacemos la media de 2 datos, y en el segundo hacemos la media de 3 datos.

Por tanto, en el ordenador final tenemos que hacer la media ponderada. Por ello, el primer resultado lo multiplicamos por  $\frac{2}{5}$  y el segundo por  $\frac{3}{5}$ .

$$\bar{x} = \frac{x_1 + \dots + x_n}{n}$$

$O(n)$  secuencial  
 $n=m \cdot K$ ; particionamos en grupos de  $m$  datos



¿En qué hemos mejorado gracias a la distribución?

**Secuencial:** el número de operaciones realizadas en este caso son:  $n-1$  sumas y 1 división → por lo que hacemos  $n$  operaciones en total

Nos planteamos qué ocurre si  $n$  lo dividimos en grupitos de  $m$  datos en un total de  $k$  máquinas, ¿cuántas operaciones saldrían?

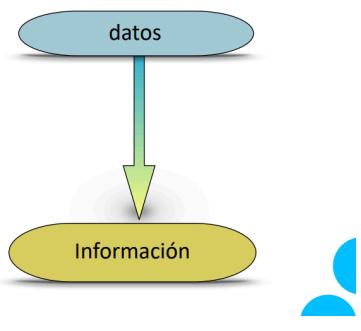
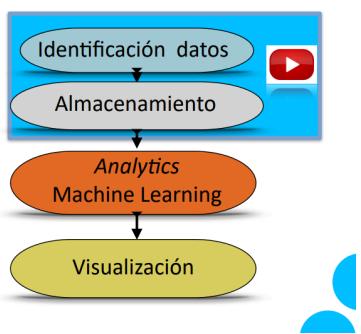
El número de operaciones que haríamos, dado que las operaciones se hacen de manera simultánea en cada ordenador, serían:

- en primer lugar tendríamos el coste de hacer la media de esos grupitos de  $m$  datos,
- y en segundo lugar el coste de hacer la segunda media con esos resultados intermedios, que sería el coste de  $f(k)$ .
- → Por lo tanto, el **coste total** sería la **suma del coste de  $m$  más el coste de  $k$** .

Hemos conseguido, a través de la computación distribuida, **convertir una multiplicación en una suma**. Esto significa que haremos un total de menos operaciones, ya que la suma produce menos operaciones que la multiplicación.

→ **Usar un sistema distribuido nos va a permitir escalar el procedimiento.**

## 5 - PROCESO DE EXTRACCIÓN DE LA INFORMACIÓN

<p><b>Proceso extracción información</b></p> 	<p>Hoy en día lo que queremos es <b>transformar los datos en información útil</b> para la empresa. Y esto requiere una serie de etapas.</p>
<p><b>Proceso extracción información Etapas</b></p> 	<p>Recordando el esquema que vimos de convertir los datos en información, vamos a ver en este apartado las 2 primeras etapas: la de <b>adquisición de datos</b> y su <b>almacenamiento</b>.</p> <p>En primer lugar, hay que identificar las <b>fuentes</b> útiles para los <b>objetivos</b> que se están persiguiendo dentro de un determinado proyecto.</p> <p>Luego tenemos una fase que consiste en <b>almacenar</b> los datos, en la que se plantea un <b>modelo de datos</b> para <b>almacenerlos físicamente</b>.</p> <p>En la etapa de <b>análisis</b> vamos a <b>analizar</b> los datos, identificando <b>patrones</b>, determinando <b>tendencias</b>, detectando <b>anomalías</b>.</p> <p>Finalmente, tendremos que <b>mostrar</b> los resultados para, a partir de ahí, <b>tomar decisiones</b>.</p>

## Proceso extracción información

### 1.- Identificación datos



#### Primera etapa: identificación de datos.

Tenemos 2 tipos de datos que se pueden generar:

- **Internos:** se generan **dentro del entorno empresarial**. Con sus aplicaciones, CRMs (*Customer Relationship Management*). Software que permite a las empresas gestionar las relaciones y las interacciones con clientes potenciales y actuales de manera eficiente. Ej: *Salesforce*, etc.
- **Externos:** generados en rrss, mediante móviles, bases de datos de otras entidades que sean relevantes para el propósito de la empresa. Es necesario **identificar** esas **fuentes** que sean **de interés para los objetivos** que se persiguen en la **empresa**.

## Proceso extracción información

### Tipología datos

- Datos estructurados
- Datos **NO** estructurados
- Datos **SEMI** estructurados

Una vez identificados los datos, podremos clasificarlos en 3 categorías:

- **Estructurados**
- **No estructurados**
- **Semi-estructurados**

## Proceso extracción información

### Tipología datos

#### Datos estructurados

LU	360	DANZIG
LH	012	HAMBURG
LH	416	WASHINGTON
AA	071	DALLAS FORT WOR
DE	6164	MONTEGO BAY
BA	1759	BIRMINGHAM
BA	1707	MANCHESTER
KL	1766	AMSTERDAM
LH	6810	KOELN HBF
PS	408	SIMFEROPOL
LH	564	ACCRA-LAGOS

**Estructurados:** están bien definidos en cuanto a formato, longitud y significado.

Ejemplo: magnitudes, fechas, cadenas, etc.

Se almacenan en **tablas**. Si hubiera muchos datos, quizás habría que requerir **data warehouse** (como AWS Redshift o Azure Synapse Analytics), para poder analizar mejor los datos y realizar consultas más complejas.

Si no, normalmente con bases de datos relacionales, sería suficiente. En estas bases de datos especificaremos los campos necesarios y cómo se relacionan las tablas entre ellas.

<p><b>Proceso extracción información</b></p> <p><b>Tipología datos</b></p> <ul style="list-style-type: none"> <li>Datos <b>NO</b> estructurados</li> </ul> 	<p><b>No estructurados:</b> no tienen un formato fijo ni están predefinidos.</p> <p>Ejemplos: vídeos, audios, imágenes. Están generados por múltiples fuentes, como son las rss, teléfonos móviles, sensores, etc.</p> <p>Requieren tipos de almacenamiento más sofisticados como bases de datos NoSQL (MongoDB, Cassandra, Neo4j, etc.).</p>
<p><b>Proceso extracción información</b></p> <p><b>Tipología datos</b></p> <ul style="list-style-type: none"> <li>Datos <b>semi-estructurados</b></li> </ul> 	<p><b>Semi-estructurados:</b> combinan datos estructurados con datos no estructurados.</p> <p>Existe una <b>estructura</b> en forma de <b>árbol</b> que relaciona los distintos textos con los objetos que existen. Estos esqueletos pueden evolucionar conforme se añaden más datos, formando cada vez un esquema más definido.</p> <p>Ejemplos: correo electrónico, contiene texto junto con archivos adjuntos (ficheros). La combinación de estos elementos permite almacenar tanto información estructurada (como campos de dirección) como no estructurada (el cuerpo del mensaje). Otro ejemplo sería un fichero XML (aunque XML permite organizar los datos en una estructura jerárquica y definida mediante etiquetas, no impone un esquema rígido como las bases de datos relacionales).</p>

**Proceso extracción información**  
**2.-Almacenamiento**



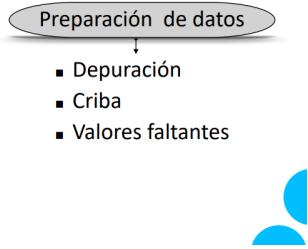
Una vez identificadas todas las fuentes de datos, hay que **almacenar** los datos.

Esta fase de almacenamiento consta de 3 etapas:

- Preparación de los datos
- Definir un modelo de datos
- Almacenar los datos

Esto da un perfil laboral de **ingeniero de datos**, que son los encargados de hacer estas tareas.

**Proceso extracción información**  
**2.-Almacenamiento**



### Preparación de los datos.

- **Depuración:** consiste en eliminar el mayor número de datos erróneos o inconsistentes, es decir, limpiarlos. Si se eliminan o corrigen datos incorrectos e inconsistentes al comienzo, todo el análisis posterior se basa en datos fiables. Evita que valores problemáticos influyan en los siguientes pasos, como la criba o el tratamiento de valores faltantes.

Durante la depuración, se verifica la **coherencia y consistencia** de los datos, lo cual incluye revisar si las unidades son correctas y están presentes donde deben estar.

Algunas de las acciones que se llevan a cabo en la depuración son: corrección de errores de formato, corrección de la inconsistencia en las unidades (recoger un peso en otra unidad distinta a los gramos/kg, eliminación de duplicados, corrección de errores tipográficos, corrección de la inconsistencia en valores categóricos (por ejemplo, “femenino” o “fem” para el mismo valor), errores de codificación o caracteres especiales (tildes incorrectas, comillas...), valores incompletos o incorrectos en campos clave (identificadores o claves primarias ausentes...).

- **Criba:** tiene como objetivo identificar y **eliminar datos irrelevantes o atípicos**, de manera que el conjunto de datos final refleje solo **datos útiles y precisos**. A través de **métodos estadísticos**, se detectan valores que se consideran **outliers** (valores atípicos) o que son poco relevantes para el análisis.

Eliminación de datos irrelevantes: se revisa el dataset para identificar valores que no aportan información significativa para el análisis o que no son necesarios. Esto puede incluir columnas o atributos con poca variabilidad, como datos redundantes o que no se relacionan directamente con el objetivo del estudio..

Detección de outliers: a través de **métodos estadísticos** se identifican valores que se alejan significativamente de la mayoría de los datos. Estos valores atípicos pueden deberse a errores de entrada o a características inusuales pero válidas.

Existen diferentes técnicas para manejar outliers:

- **Eliminación**: si el valor atípico se determina erróneo o innecesario para el análisis, se elimina.
- **Modificación**: en algunos casos, los *outliers* pueden corregirse o ajustarse, especialmente si son errores evidentes.
- **Conservación con etiquetas**: si se decide conservar los *outliers* (como en estudios donde los datos extremos pueden ser valiosos), se pueden etiquetar para distinguirlos en el análisis.

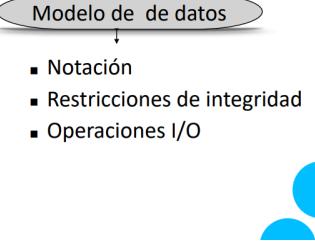
Con los datos ya depurados, los métodos estadísticos para detectar valores atípicos son más precisos, pues se eliminan datos incorrectos que podrían generar falsos positivos en la detección de outliers. Esto asegura que los outliers detectados sean significativos y no se deban a inconsistencias previas.

Tras la criba, el dataset queda optimizado, con los valores atípicos y datos irrelevantes gestionados, lo cual permite un análisis más claro y preciso.

- **Valores faltantes**: los valores faltantes, o **missing values**, son datos que no han sido registrados o que están ausentes en un conjunto de datos. Su presencia puede afectar los resultados del análisis, ya que pueden llevar a conclusiones sesgadas o reducir la precisión del modelo. Es crucial **decidir cómo tratarlos** para obtener un dataset lo más completo y representativo posible.

La detección de valores ausentes puede hacerse mediante el **análisis de los campos vacíos** o a través de **funciones de librerías de análisis de datos** que identifican estos valores. Si los valores faltantes son pocos o no afectan significativamente el análisis, se puede optar por **eliminar** las filas o columnas con datos incompletos. También se pueden aplicar **métodos estadísticos**, como rellenar

	<p>los valores faltantes con la media, mediana o moda de la columna, lo cual es útil cuando los datos faltantes son numéricos y la variabilidad de los datos es baja. En casos avanzados, se puede emplear un <b>modelo predictivo</b> para predecir los valores faltantes en función de otros datos.</p> <p>Es recomendable registrar qué se ha decidido hacer con los valores faltantes y por qué. Esto facilita la interpretación del análisis y permite replicar el tratamiento en futuros datasets.</p> <p>Esta fase nos da una mejor <b>calidad</b> de los datos, ya que el conjunto de datos se vuelve más <b>limpio, consistente y representativo</b>, eliminando errores y asegurando que las conclusiones y modelos generados sean <b>precisos y confiables</b>.</p>
--	--

<p><b>Proceso extracción información</b> <b>2.-Almacenamiento</b></p> 	<p><b>Modelo de datos.</b></p> <p>Es una herramienta teórica para describir las propiedades más relevantes de los datos. Contiene 3 elementos fundamentales:</p> <ul style="list-style-type: none"> <li>• Notación: permite describir la estructura de los datos, los tipos, cómo se relacionan.</li> <li>• Restricciones de integridad: qué cosas debe cumplir un registro para considerar que es válido.</li> <li>• Operaciones para la actualización y recuperación de los datos. Por ejemplo, operaciones para insertar, eliminar, consultar, modificar.</li> </ul> <p>Tienen distintas abstracciones.</p> <ul style="list-style-type: none"> <li>• <b>Modelo Conceptual:</b> es el <b>nivel más alto de abstracción</b> y en este nivel se define lo que el sistema contiene, estableciendo su organización.</li> <li>• <b>Modelo Lógico:</b> situado en el <b>nivel intermedio de abstracción</b> y define cómo el sistema debería estar implementado, independiente del tipo de base de datos que se empleará. El objetivo es desarrollar un mapa técnico para las reglas y la estructura de los datos.</li> <li>• <b>Modelo Físico:</b> es el <b>nivel más bajo de abstracción</b> y describe cómo el sistema será implementado usando una base de datos específica.</li> </ul>
--	---

**Proceso extracción información**  
2.-Almacenamiento

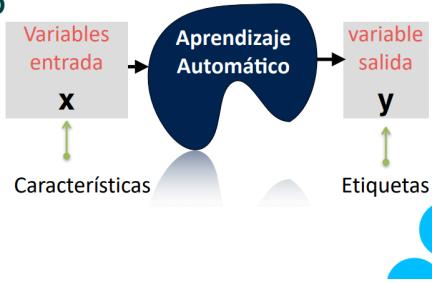


La tercera etapa es la **implementación** en la base de datos, que la veremos en el siguiente apartado.

## 6 - APRENDIZAJE AUTOMÁTICO

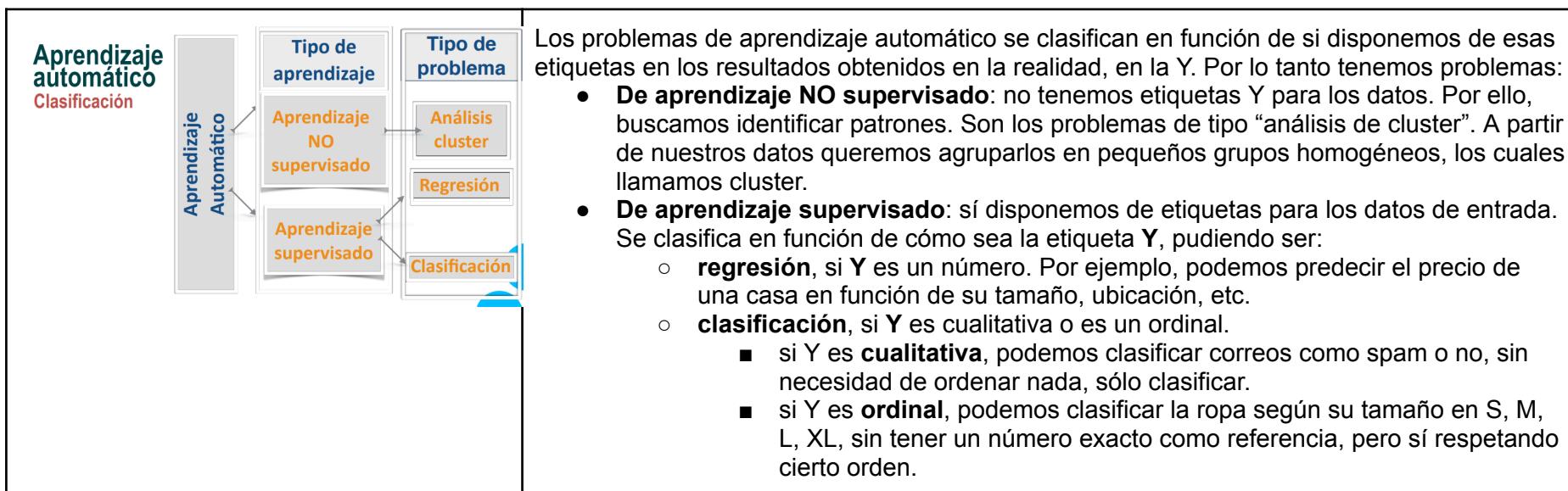
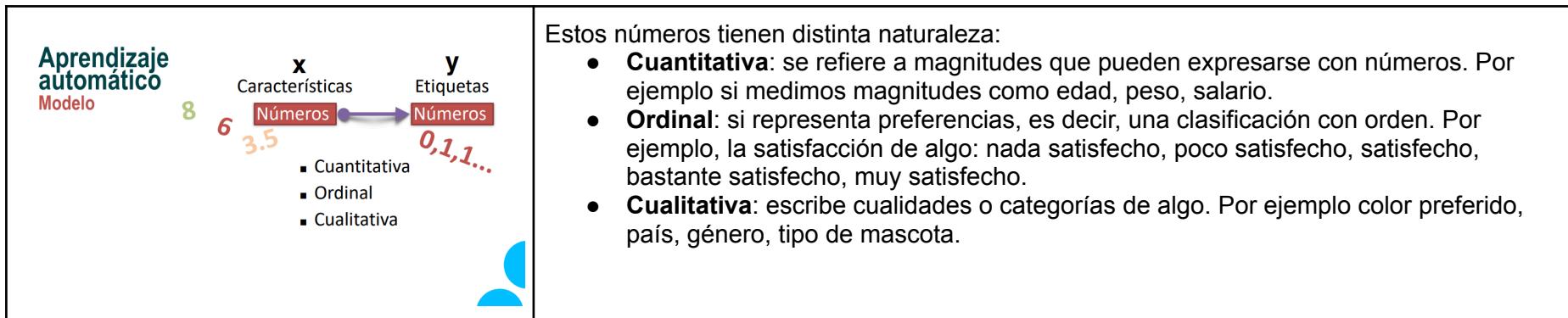
Vamos a introducir la **ANALÍTICA** del proceso de transformación de datos en información. En esta capa podemos utilizar técnicas **estadísticas**, aunque nosotros nos vamos a centrar en el aprendizaje automático. En concreto nos vamos a centrar en tipos de problemas de aprendizaje automático aplicado, obviamente, al Big Data, sobre todo orientado a la segunda V, a la **VARIEDAD**, a datos no estructurados.

**Aprendizaje automático**  
Modelo



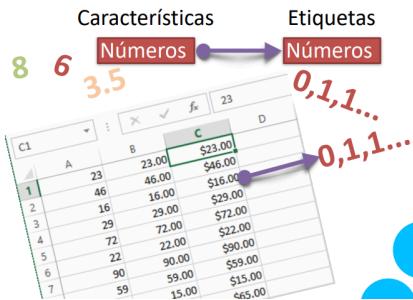
Un modelo de aprendizaje automático actúa como una caja negra e intenta emular lo que ocurre en la realidad, en la cual hay unas variables de entrada de las cuales obtienen unas variables respuesta.

Los modelos de aprendizaje automático intentan emular esa realidad y obtener las variables respuesta que producen esos valores de entrada.  
Las variables de entrada son las características, y las de salida son las etiquetas. Tanto las variables como las etiquetas se deben de codificar en números.



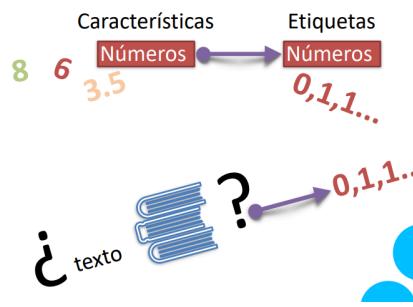
## Aprendizaje automático

Datos estructurados



## Aprendizaje automático

Datos NO estructurados



Para poder aplicar estos problemas, lo que tenemos que hacer es **medir estas características y etiquetas**.

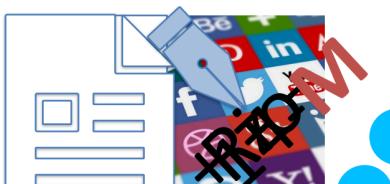
Cuando se trata de **datos estructurados**, es **sencillo**. Lo mismo ocurre cuando son etiquetas cualitativas, ya que podemos codificarlas numéricamente (por ejemplo: "rojo", "azul" o "amarillo", sería 1, 2 o 3). Así, podemos convertir tanto las características como las etiquetas en valores numéricos que los algoritmos de aprendizaje automático pueden procesar.

El **problema** ocurre cuando tenemos **datos no estructurados**, como texto, imágenes o sonido. En el caso del texto, no es tan simple convertir palabras en números. Entonces ¿cómo clasificamos esos conjuntos de palabras? Para ello, el aprendizaje automático utiliza técnicas específicas, como la **minería de textos** y métodos como **Bag of Words (Bolsa de Palabras)**, para representar el texto de manera numérica.

## Aprendizaje automático

Texto

- Análisis de sentimientos
- Entidades con nombre
- Extracción eventos



Veamos ahora cómo resolver los distintos tipos de problemas: análisis cluster, regresión y clasificación.

Un ejemplo típico de clasificación es el análisis de sentimientos en texto, como en redes sociales. Consiste en asignar etiquetas como 0, 1 o 2 según el mensaje sea negativo, neutro o positivo. El modelo aprende a clasificar el sentimiento de un texto en una de estas categorías.

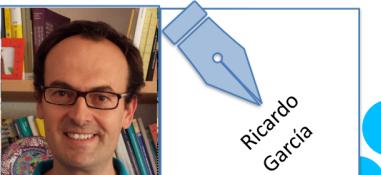
Identificar entidades con nombre en un texto, como "el Ebro es un río", también es un problema de **clasificación**. Aquí, el sistema clasifica las palabras o frases según su tipo de entidad (río, persona, organización, etc.), un proceso que se conoce como **Reconocimiento**

	<b>de Entidades Nombradas (NER).</b>  El problema de <u>extracción de eventos</u> también es de <b>clasificación</b> . Consiste en identificar eventos específicos mencionados en un texto, como "el huracán Milton", y clasificar el evento según su naturaleza (desastre natural, evento político, etc.).
--	---

<p><b>Aprendizaje automático</b> Imagenes</p> <ul style="list-style-type: none"> <li>▪ Extracción de textos y objetos</li> </ul> 	<p>Si los datos con los que trabajamos son imágenes, los problemas a resolver pueden ser variados.</p> <ul style="list-style-type: none"> <li>• Un ejemplo es la detección de objetos, como reconocer y localizar la matrícula de un coche en una imagen. Aquí, el modelo no solo identifica que hay una matrícula, sino que también detecta su posición exacta dentro de la imagen.</li> </ul>
<p><b>Aprendizaje automático</b> Imagenes</p> <ul style="list-style-type: none"> <li>▪ Extracción de textos y objetos</li> <li>▪ Entendimiento de imágenes</li> </ul> 	<ul style="list-style-type: none"> <li>• Otro caso sería clasificar una imagen en un conjunto de categorías predefinidas. Por ejemplo, una imagen de un autobús escolar puede ser etiquetada con la categoría de "transporte escolar". El modelo asigna la imagen a una de las etiquetas predefinidas basándose en su contenido.</li> </ul>

## Aprendizaje automático Imagenes

- Extracción de textos y objetos
- Entendimiento de imágenes
- Reconocimiento facial

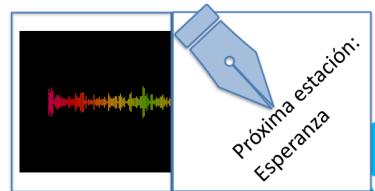


- También podemos plantear el reconocimiento facial. Aquí, el objetivo es identificar a una persona a partir de una imagen. Dada una imagen de un rostro, el modelo debe reconocer a quién pertenece ese rostro, comparándolo con una base de datos de rostros conocidos.

- En problemas de imágenes geoespaciales, como imágenes satelitales, un ejemplo de **regresión** podría ser predecir un valor numérico a partir de la imagen. Por ejemplo, dado un área geográfica capturada en una imagen, se podría estimar la **contaminación lumínica** asociada a esa área, asignándole un número que representa el nivel de contaminación.

## Aprendizaje automático Audio

- Reconocimiento de audio



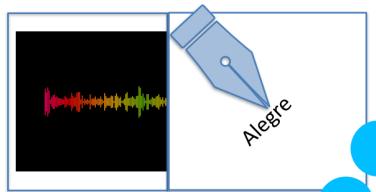
En el caso del audio, uno de los problemas sería el reconocimiento de audio.

- Un ejemplo sería el reconocimiento de voz, como cuando se escucha este audio (escuchar [AUDIO METRO](#)). El objetivo que tiene el sistema es **entender lo que se está diciendo** y transformarlo en texto. Este tipo de tecnología es utilizada por asistentes virtuales como Google Assistant, que transforma el audio en texto para procesarlo.

## Aprendizaje automático

Audio

- Reconocimiento de audio
- Extracción de características



- Otro problema consiste en la extracción de características. (escuchar [AUDIO ESTADO DE ÁNIMO](#)) Aquí no se trata de entender lo que se está diciendo, sino de **analizar características del audio**. Por ejemplo, se podría analizar el tono de voz para detectar el **estado de ánimo** de la persona que habla (si está feliz, triste o enojada), sin necesidad de transcribir el contenido del discurso.

## Aprendizaje automático

Vídeo

- Creación de resúmenes automáticos

Si trabajamos con **vídeo**, se pueden generar **resúmenes visuales** que contengan imágenes de los momentos clave (*frames*), como cuando se mencionan palabras importantes o hay reacciones como aplausos. También es posible crear **resúmenes de texto** que describan los eventos, basado en el contenido hablado (transcripción de palabras clave).

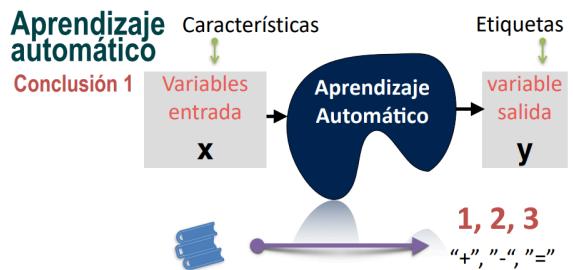
## Aprendizaje automático

Vídeo

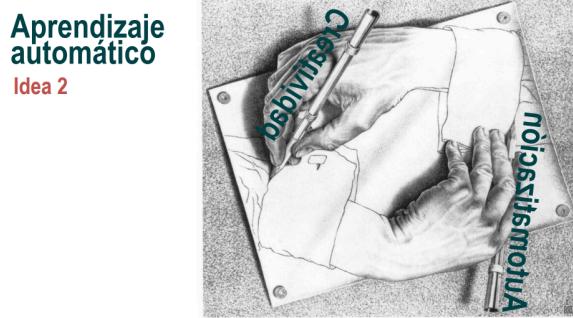
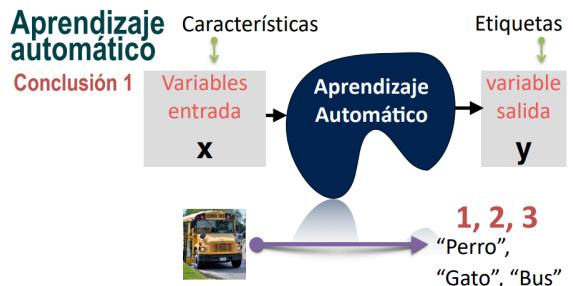
- Creación de resúmenes automáticos
- Reconocimiento de lugares, objetos o acciones

Otro caso es el **reconocimiento de lugares** que aparecen en el vídeo o la **detección de objetos o de acciones específicas** que permiten identificar lugares (como una calle o un edificio), o identificar acciones particulares (si se ha producido una infracción de tráfico o un allanamiento de una propiedad). Estos problemas se resuelven con técnicas de **visión por computadora** y, en algunos casos, análisis del contexto utilizando **redes neuronales** entrenadas para reconocer situaciones específicas.

La **detección de acciones** requiere algoritmos más avanzados que no solo analicen imágenes estáticas, sino también la secuencia de imágenes en el tiempo para entender el contexto.

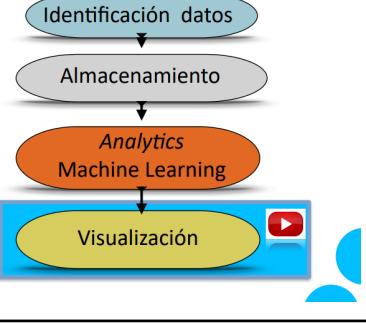
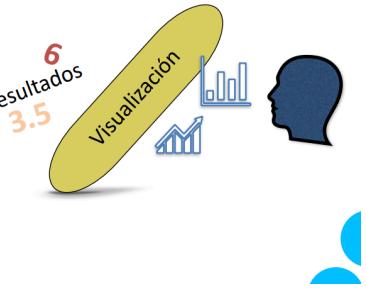


Como resumen, podemos decir que tenemos **diferentes fuentes de datos** de las que necesitamos **extraer características relevantes**. La manera en que se realiza esta extracción depende del tipo de problema (como texto, imágenes o audio). Sin embargo, una vez que las características se han extraído, muchos de los **algoritmos de aprendizaje automático** pueden aplicarse de manera similar, independientemente del tipo de datos original.



Los métodos de aprendizaje automático permiten **automatizar** gran parte del proceso, incluyendo la **extracción y clasificación** de características. Esto facilita la creación de sistemas que **aprenden automáticamente** de los datos.

Sin embargo, la parte más fundamental sigue siendo el **diseño creativo** de cómo **estructurar y representar** esos datos, ya que la **calidad de las características extraídas** afecta en gran medida el **rendimiento del modelo**.

<p><b>Procesos extracción información</b></p>  <pre> graph TD     A[Identificación datos] --&gt; B[Almacenamiento]     B --&gt; C["Analytics Machine Learning"]     C --&gt; D[Visualización]     </pre>	<p>En este apartado vamos a centrarnos en el último paso del proceso de extracción de la información: la <b>visualización</b>.</p> <p>Veremos un caso notable que son los <b>cuadros de mando</b>.</p> <p>También vamos a ver qué problemática nos encontramos en este proceso de visualización en Big Data.</p>
<p><b>Visualización</b> ¿Qué es?</p> 	<p>La <b>visualización</b> es una capa <b>intermedia</b> que conecta los resultados obtenidos en la capa de <b>análisis</b> con el <b>decisor</b>.</p> <p>Su principal <u>objetivo</u> es transmitir la información de manera que el decisor pueda <b>comprender fácilmente</b> qué está ocurriendo, facilitando la <u>toma de decisiones</u>.</p>

### Visualización Finalidad

#### Datos estructurados

- explicar los datos existentes
- realizar predicciones



En el caso de los **datos estructurados**, la visualización tiene una doble finalidad:

1. Explicar los datos existentes: permite **analizar** los datos y **detectar patrones o diferencias**. Por ejemplo, un histograma puede mostrar cómo un fenómeno varía entre hombres y mujeres.
2. Realizar predicciones: ayuda a visualizar **tendencias y posibles evoluciones futuras**.

### Visualización Finalidad

#### Datos NO estructurados

- comunicar
- evaluación
- interpretar



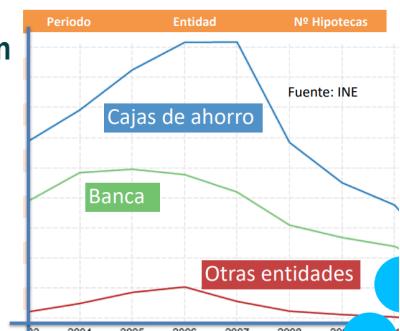
Para los **datos no estructurados**, la finalidad de la visualización puede no ser tan clara como en los datos estructurados, ya que surgen otros tipos de desafíos.

Uno de los objetivos principales es **comunicar la información de manera rápida y objetiva**, pero lo más importante es que la visualización permita **evaluar e interpretar** adecuadamente lo que está ocurriendo. Para lograrlo, es crucial que en esta capa se **identifiquen claramente los patrones** presentes en los datos que estamos analizando.

## Visualización

### Finalidad

- interpretación
- evaluación
- comunicar



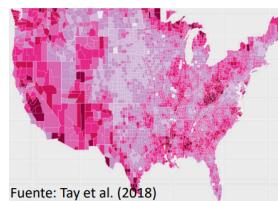
Veamos cómo se aplican las tres finalidades de **interpretación, evaluación y comunicación** en la visualización de datos. Si disponemos de los datos sobre la cantidad de préstamos hipotecarios en una tabla, al ver los datos sueltos (VER TABLA HIPOTECAS MOODLE), esta información puede ser difícil de interpretar, evaluar y comunicar.

Sin embargo, al representarla gráficamente como se ve en la **imagen** (ver gráfico), es mucho más fácil observar una dinámica clara de los préstamos: primero una tendencia ascendente y expansiva, seguida de una contracción. Al realizar la **evaluación**, se puede identificar claramente un colapso en el número de préstamos alrededor de 2011, reflejando el impacto de la crisis inmobiliaria en España.

## Visualización

### Volumen

- Resumir bases de datos masivas
- Identificación patrones
- Identificación de datos relevantes



Nos vamos a plantear ahora cuáles son los retos que pretende la visualización cuando hablamos de Big Data. En cuanto al **VOLUMEN**.

En la imagen podemos ver la tasa de paro en el año 2015 en EE. UU., incluyendo pequeñas localidades y pueblos. Un objetivo evidente es resumir bases de datos masivas. Otro aspecto importante es la identificación de patrones. Se observa, por ejemplo, que en la costa **oeste** hay una mayor tasa de paro en comparación con la costa **este**.

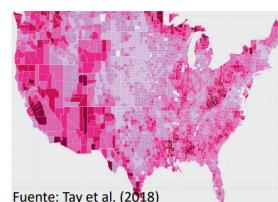
Finalmente, podemos identificar datos relevantes, como zonas donde no hay paro y otras con índices de paro altísimos.

→ En cuanto al volumen, esta visualización ilustra la **dicotomía** entre, por un lado, **resumir la información** y, por otro, **identificar detalles específicos**. Por lo tanto, el desafío consiste en elegir el nivel adecuado de agregación de los datos. Esto significa decidir qué nivel de detalle es más útil: ¿mostramos la tasa de paro por pueblo, por ciudad o por estado? Este es un punto clave en la visualización de Big Data, porque si los datos están muy agrupados (agrupados, poca información detallada), perdemos la capacidad de ver patrones finos, pero si no agregamos nada (sin grupos, mucha información), la visualización se vuelve compleja y difícil de interpretar.

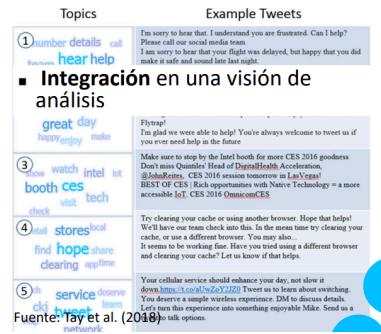
## Visualización

### Volumen

- Resumen / identificación
- Nivel de agregación



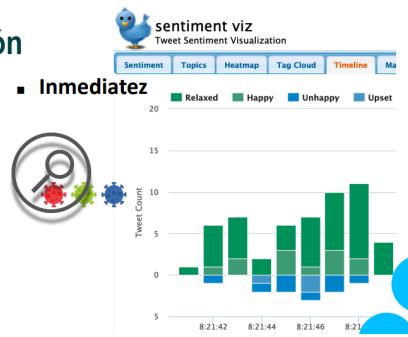
## Visualización Variedad



Es aspecto de la **VARIEDAD** en Big Data es un problema.

En este ejemplo, vemos un análisis de **tweets**, donde el reto consiste en **convertir el texto no estructurado** (como los tweets) en algo que pueda ser **interpretado y analizado** de manera coherente. Esto requiere **integrar** esta información en una forma clara y comprensible, es decir, transformar el texto en datos organizados que se puedan analizar dentro de una visión más amplia.

## Visualización Velocidad



Otro reto es la **VELOCIDAD**.

Los datos se generan rápidamente y, para obtener valor de ellos, deben ser analizados casi en **tiempo real**. Un buen ejemplo es el análisis de redes sociales, donde las reacciones de las personas cambian constantemente, como sucedió durante la pandemia de COVID-19. El **aspecto temporal** es crucial y debe reflejarse en nuestros gráficos, permitiéndonos ver **cómo evolucionan las emociones o los sentimientos en respuesta a eventos importantes de manera inmediata**.

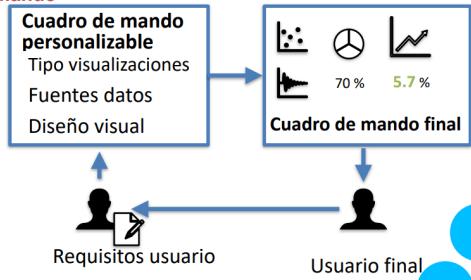
## Visualización Cuadro de mando



Los **cuadros de mando integrales** no son simples visualizaciones, sino **herramientas** que permiten **representar los aspectos generales y críticos de la empresa**, como los KPI, además de lo que está ocurriendo en el entorno externo. Si bien forman parte de la etapa de visualización, son mucho más que eso: son **instrumentos de gestión estratégica** que nos permiten **tanto definir la estrategia como evaluar su desempeño**.

## Visualización

### Cuadro de mando



Un cuadro de mando es mucho más que visualizar los datos. En su **elaboración y construcción**, debe recorrer todo el proceso, desde la definición de la estrategia de la empresa, su traducción a objetivos concretos, y la identificación de los KPI esenciales (*Key Performance Indicators*, Indicadores Clave de Desempeño). Son métricas que sirven para evaluar si una organización está cumpliendo con sus metas).

Además, debe incorporar en la **fase de análisis** aspectos sobre lo que está ocurriendo dentro de la empresa y su **impacto** en el negocio, junto con **recomendaciones**. Todo esto debe visualizarse adecuadamente, pero **la clave es que exista una fase de análisis antes de la visualización**.

En la **etapa de visualización**, es fundamental que se **interpreten los datos correctamente**, y por ello, los cuadros de mando deben hablar el mismo lenguaje que el **decisor**, es decir, presentar la información de forma **clara, relevante y alineada** con los objetivos estratégicos.