



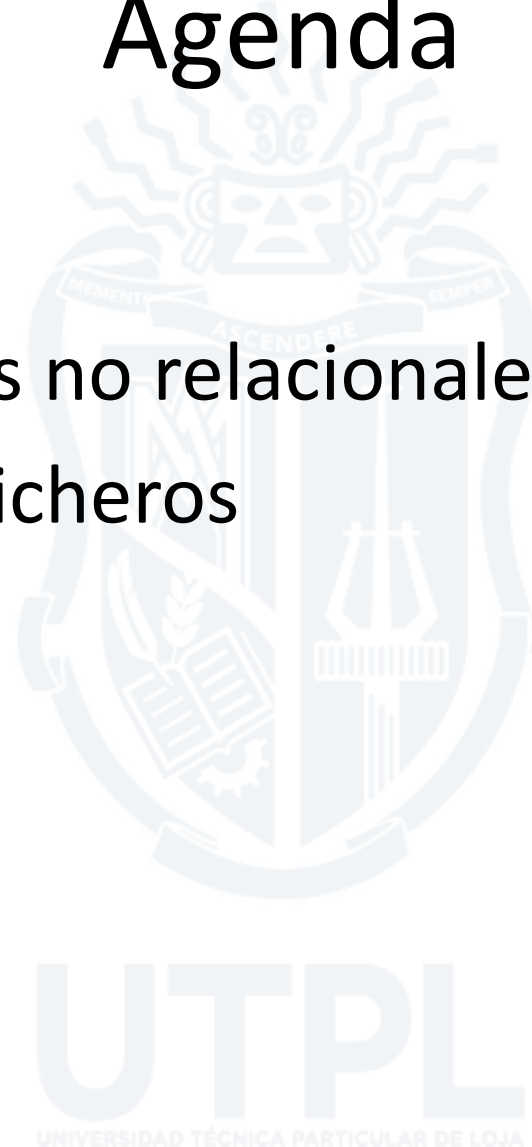
# Unidad 8

## Otros modelos de bases de datos

*Parte del documento está basado en material de autoría de Luis Valencia Cabrera de la Universidad de Sevilla*

# Agenda

- Introducción
- Bases de datos no relacionales
- Formatos de ficheros
- MongoDB



# Introducción

## Mundo relacional

Hemos trabajado con bases de datos relacionales...

- Analizando requisitos de datos.
- Abstrayendo a un diseño conceptual.
- Trasladando y normalizando a un diseño lógico.
- Implementando en un motor relacional (Oracle, MySQL, etc.).
- Usando SQL para realizar consultas sobre la base de datos.

# Fortalezas de las bases de datos relacionales

- Modelo estándar
- Control de restricciones de integridad
- Soporte ACID (manejo de transacciones y control de concurrencia)
- Lenguaje de consulta estandarizado



# Limitaciones de las bases de datos relacionales

- No está optimizada para soportar altos volúmenes de datos más allá de los TeraBytes de información.
- No facilitan un escalamiento horizontal.
- Solo soporta campos simples.
- Esquema rígido.
- Único punto de fallo.

# Nuevos retos

- Nuestro entorno cambia día a día, y más en el aspecto tecnológico.
- Google, Wikipedia, redes sociales, Twitter, etc. han implicado un nuevo **cambio de escala** en el empleo de las TIC, sobre todo en lo que refiere a la gestión de enormes conjuntos de información en red (Big Data).
- Big Data lleva el tamaño de los volúmenes de datos a otra dimensión.
- Surge el mundo **NoSQL** (Not Only SQL)

# Introducción

## Cambio de escala

### Very Large Databases (VLDB)

Las siglas VLDB (*Very Large Databases*) denotan a bases de datos que contienen tablas cuyo número de tuplas resulta **extremadamente alto**

No se puede hablar de cifras concretas, esto va cambiando con el tiempo, evolucionando con la capacidad de almacenamiento de equipos personales, servidores, aplicaciones, etc.

Para hacernos una idea, se puede entender como VLDB una base de datos de cientos de terabytes, con cientos de miles de millones de filas.

Aparece incluso el concepto de XLDB (*Extremely Large Databases*).

# Introducción

## Big Data

Los términos anteriores han ido dejando paso a lo largo de los últimos años a la denominación de **Big Data**.

*Entendemos por Big Data a conjuntos de información enormes, que superan con mucho la capacidad del software de gestión de bases de datos convencionales para procesar datos en tiempo razonable.*

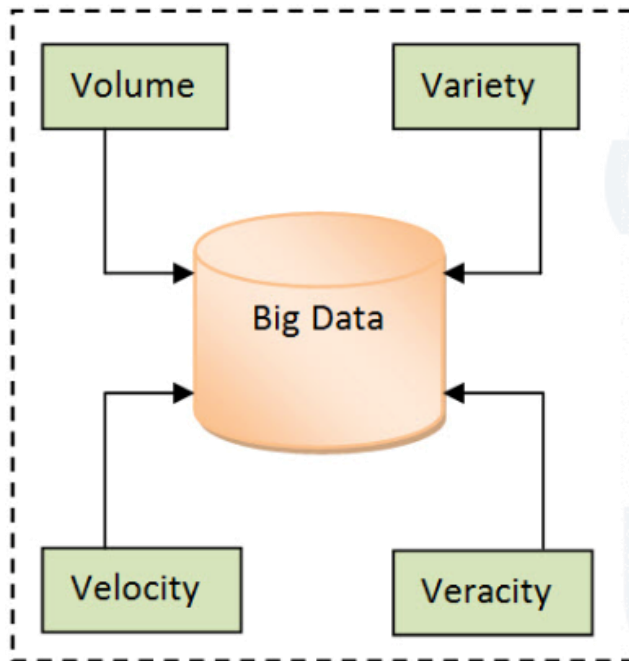
Los retos con este tipo de tecnologías se centran en la captura, almacenamiento, búsqueda, compartición, análisis y visualización de los datos.



# Introducción

## Big Data

En el marco de Big Data se viene imponiendo la terminología de las 4 V's para resumir las características fundamentales:



### 4 V's

- Volume** Grandes volúmenes de datos (petabytes, exabytes, etc.)
- Variety** No limitado a información relacional, puede incluir datos no estructurados
- Velocity** Los datos se deben procesar y analizar rápidamente
- Veracity** Veracidad, adhesión a la verdad, precisión.

# Introducción

## Big Data

### 7 V's

<b>Volume</b>	Grandes volúmenes de datos (petabytes, exabytes, etc.)
<b>Variety</b>	No limitado a información relacional, puede incluir datos no estructurados
<b>Velocity</b>	Los datos se deben procesar y analizar rápidamente
<b>Veracity</b>	Veracidad, adhesión a la verdad, precisión.
<b>Variability</b>	Datos cuyo significado cambia constantemente
<b>Visualisation</b>	Presentar los datos de forma comprensible
<b>Value</b>	Extraer información para la toma de decisiones

# Bases de datos no relacionales

## Modelos de bases de datos

- Jerárquicas
- De red
- Relacionales
- Orientadas a Objetos
- NoSQL

No Relacionales

No only SQL

# Panorámica de las bases de datos

## NoSQL

- NoSQL - es un término utilizado para describir un conjunto de bases de datos que se diferencian de las bases de datos relacionales (*RDBMS*) en los siguientes aspectos:
  - Esquema prescindible (mayor flexibilidad)
  - Escalan horizontalmente (mayor disponibilidad)
  - Desnormalización (redundancia de datos)
  - En sus premisas no está garantizar restricciones de integridad
  - Permite almacenar y gestionar Big Data
- Conocidas también como bases de datos **Big Data**

# Panorámica de las bases de datos

## NoSQL

- A nivel lógico, la principal diferencia radica en cómo guardan los datos (p.ej., el almacenamiento de una factura):
  - En RDBMS separaríamos la información en diferentes tablas (cliente, factura, líneas\_factura,...) y luego el aplicativo, ejecutaría el *JOIN* y mapearía esta consulta SQL para mostrárselo al usuario.
  - En NoSQL, simplemente se guarda la factura como una unidad, sin separar los datos.

# Panorámica de las bases de datos NoSQL

Cuando usar ...



- El volumen de los datos no es extremadamente grande
- Es suficiente con un único servidor, o unos pocos en esquema de clustering.
- Necesitamos modelo de datos rígido, normalizado, y control ACID (transacciones)
- Se requieren implementar restricciones de integridad a nivel del SGBD.

# Panorámica de las bases de datos NoSQL

Cuando usar ...

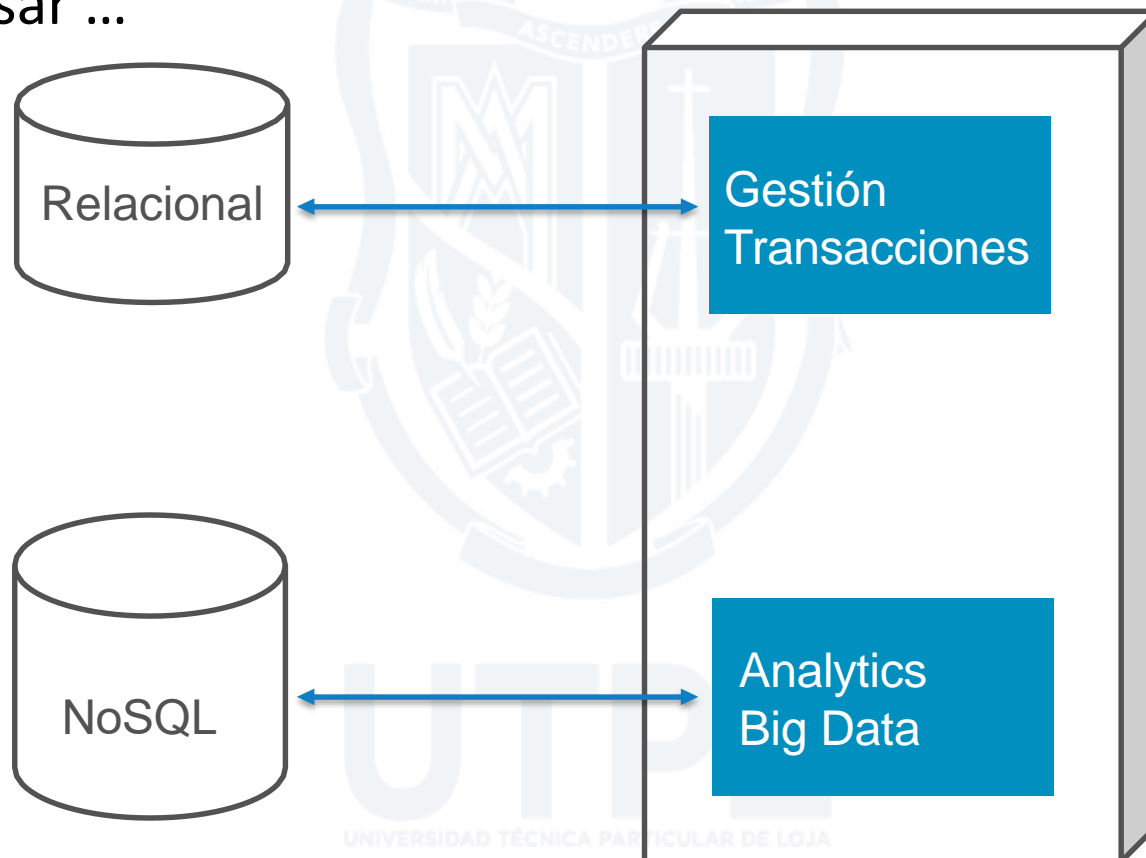


- Volumen de datos previsiblemente muy alto.
- Necesitamos un modelo **dinámico**.
- Necesitamos **escalabilidad** y **rendimiento**.
- No se requiere consistencia ACID
- Frecuencia muy alta de accesos de lectura y escritura.

# Panorámica de las bases de datos

## NoSQL

Cuando usar ...





# Panorámica de las bases de datos

## NoSQL

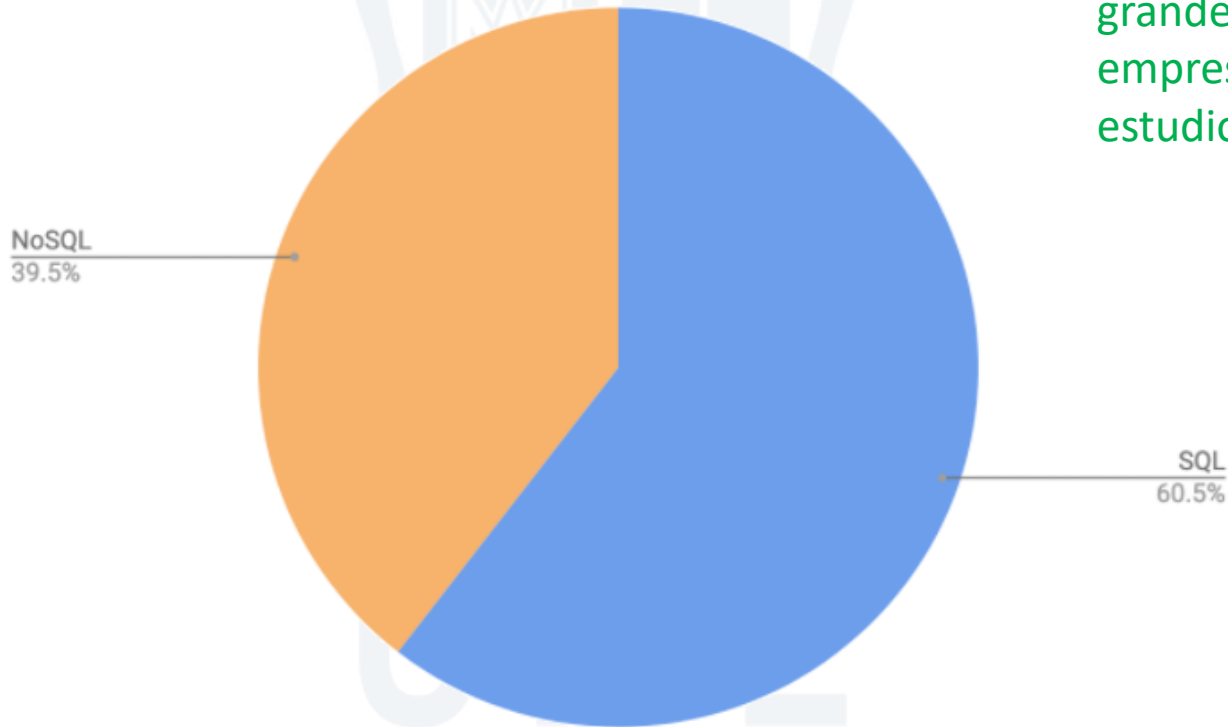
- Casos de aplicación NoSQL:
  - Servicios Web2.0 (redes sociales, blogs, etc.)
  - Aplicaciones IoT
  - Comercio electrónico (sistemas recomendadores)
  - Análisis de grandes volúmenes de datos
  - Sistemas de mensajería y chat
  - Smart Cities - Smart Land
  - Gestión de contenidos y medios
  - Juegos en línea
  - Finanzas y Trading
  - Salud y medicina (registros médicos, imágenes, datos genómicos)

# Bases de datos NoSQL

## Estado actual

SQL Database Use: 60.48%  
NoSQL Database Use: 39.52%

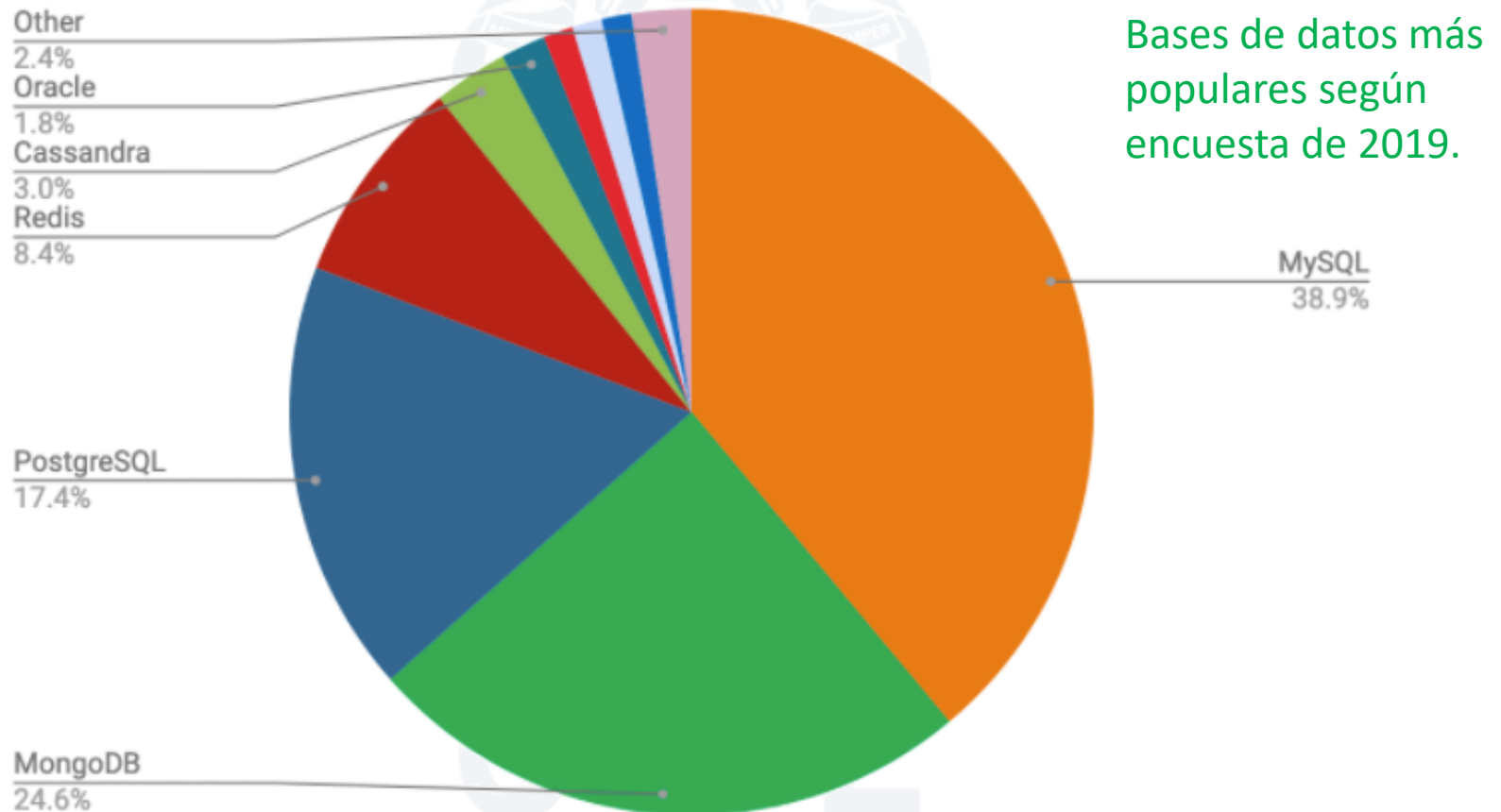
Bases de datos  
más usadas en  
grandes  
empresas según  
estudio de 2019.



<https://scalegrid.io/blog/2019-database-trends-sql-vs-nosql-top-databases-single-vs-multiple-database-use/>

# Bases de datos NoSQL

Estado actual



<https://scalegrid.io/blog/2019-database-trends-sql-vs-nosql-top-databases-single-vs-multiple-database-use/>

# Bases de datos NoSQL

## Estado actual

Popularidad según conocimiento/experiencia por parte de DBAs, según encuesta de 2019.

<https://www.explore-group.com/blog/the-most-popular-databases-2019/bp46/>

Most Popular Database Platforms				♥	Love	💔	Dread	★	Want
	2019	2018	%Change	2019	2018	2019	2018	2019	2018
MySQL	52%	59%	-7%	54%	49%	46%	51%	8%	8%
PostgreSQL	36%	33%	3%	70%	62%	30%	38%	14%	11%
MS SQL Server	34%	42%	-8%	58%	52%	43%	48%	3%	4%
SQLite	30%	20%	10%	56%	48%	45%	52%	7%	53%
MongoDB	26%	26%	0%	60%	55%	41%	45%	18%	45%
Redis	20%	19%	1%	71%	65%	29%	36%	11%	35%
MariaDB	17%	14%	3%	59%	53%	41%	47%	4%	47%
Oracle	16%	11%	5%	38%	37%	62%	63%	3%	63%
Elasticsearch	16%	14%	2%	63%	60%	67%	40%	11%	40%
Firebase	12%			61%		39%		8%	
DynamoDB	7%			55%		45%		4%	
Cassandra	4%			47%		53%		6%	
Couchbase	2%			37%		63%		2%	

# Bases de datos NoSQL

## Estado actual

421 systems in ranking, July 2024

Bases de datos más relevantes según información y soporte disponibles, a Julio 2024.

Rank			DBMS	Database Model	Score		
Jul 2024	Jun 2024	Jul 2023			Jul 2024	Jun 2024	Jul 2023
1.	1.	1.	Oracle +	Relational, Multi-model ⓘ	1240.37	-3.72	-15.64
2.	2.	2.	MySQL +	Relational, Multi-model ⓘ	1039.46	-21.89	-110.89
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model ⓘ	807.65	-13.91	-113.95
4.	4.	4.	PostgreSQL +	Relational, Multi-model ⓘ	638.91	+2.66	+21.08
5.	5.	5.	MongoDB +	Document, Multi-model ⓘ	429.83	+8.75	-5.67
6.	6.	6.	Redis +	Key-value, Multi-model ⓘ	156.77	+0.82	-7.00
7.	↑ 8.	↑ 11.	Snowflake +	Relational	136.53	+6.17	+18.84
8.	↓ 7.	8.	Elasticsearch	Search engine, Multi-model ⓘ	130.82	-2.01	-8.77
9.	9.	↓ 7.	IBM Db2	Relational, Multi-model ⓘ	124.40	-1.50	-15.41
10.	10.	10.	SQLite +	Relational	109.95	-1.46	-20.25
11.	11.	↓ 9.	Microsoft Access	Relational	100.63	-0.53	-30.09
12.	12.	12.	Cassandra +	Wide column, Multi-model ⓘ	99.13	+0.30	-7.40
13.	↑ 14.	↑ 14.	Splunk	Search engine	92.92	+3.82	+5.80
14.	↓ 13.	↓ 13.	MariaDB +	Relational, Multi-model ⓘ	90.58	-0.45	-5.52
15.	15.	↑ 18.	Databricks +	Multi-model ⓘ	83.29	+2.21	+14.83
16.	16.	↓ 15.	Microsoft Azure SQL Database	Relational, Multi-model ⓘ	76.75	-0.03	-2.21
17.	17.	↓ 16.	Amazon DynamoDB +	Multi-model ⓘ	70.95	-3.50	-7.86
18.	↑ 19.	↑ 20.	Google BigQuery +	Relational	57.82	-0.28	+2.40
19.	↓ 18.	↓ 17.	Hive	Relational	57.29	-2.46	-15.58
20.	20.	↑ 21.	FileMaker	Relational	48.59	+0.68	-4.73

<https://db-engines.com/en/ranking>

# Formatos de ficheros

- CSV
- XML
- YAML
- JSON



# Formatos de ficheros

## CSV

automoviles.csv

```
Año,Marca,Modelo,Descripción,Precio
1997,Ford,E350,"ac, abs, moon",3000.00
1999,Chevy,Venture,Extended Edition,4900.00
1999,Chevy,Venture,"Extended Edition, Very Large",5000.00
1996,Jeep,Grand Cherokee,"MUST SELL! air, moon roof",4799.00
```

### CSV (Comma-Separated Values)

- Permite transferir y compartir datos tabulares entre distintas aplicaciones y usuarios.
- Usado para almacenar datos de una forma sencilla y compatible.
- Usado para carga de datos en herramientas de ciencia de datos.

# Formatos de ficheros

## XML

```
<autos>
  <auto>
    <Anio>1997</Anio>
    <Marca>Ford</Marca>
    <Modelo>E350</Modelo>
    <Descripción>ac, abs, moon</Descripción>
    <Precio>3000.00</Precio>
  </auto>
  <auto>
    <Anio>1999</Anio>
    <Marca>Chevy</Marca>
    <Modelo>Venture</Modelo>
    <Descripción>Extended Edition</Descripción>
    <Precio>4900.00</Precio>
  </auto>
  <auto>
    <Anio>1999</Anio>
    <Marca>Chevy</Marca>
    <Modelo>Venture</Modelo>
    <Descripción>Extended Edition, Very Large</Descripción>
    <Precio>5000.00</Precio>
  </auto>
</autos>
```

### XML (eXtensible Markup Language)

- Altamente estructurado y permite definir etiquetas personalizadas.
- Usado para intercambiar datos entre sistemas heterogéneos (aplicaciones web, servicios web).
- También se usa para almacenar configuraciones y parámetros de aplicaciones.



# Formatos de ficheros

## YAML

**Autos:**

**Auto:**

**Anio:** 1997

**Marca:** Ford

**Modelo:** E350

**Descripcion:** "ac, abs, moon"

**Precio:** 3000.00

**Auto:**

**Anio:** 1999

**Marca:** Chevy

**Modelo:** Venture

**Descripcion:** Extended Edition

**Precio:** 4900.00

**Auto:**

**Anio:** 1999

**Marca:** Chevy

**Modelo:** Venture

**Descripcion:** "Extended Edition, Very Large"

**Precio:** 5000.00

### YAML (Yet Another Markup Language)

- Formato de serialización de datos que es legible para los humanos y fácil de escribir.
- Usado para archivos de configuración de múltiples plataformas.
- También es usado para intercambiar datos donde la legibilidad del archivo es importante.

# Formatos de ficheros

## JSON

```
{
  "autos": [
    {
      "Año": 1997,
      "Marca": "Ford",
      "Modelo": "E350",
      "Descripción": "ac, abs, moon",
      "Precio": 3000.00
    },
    {
      "Año": 1999,
      "Marca": "Chevy",
      "Modelo": "Venture",
      "Descripción": "Extended Edition",
      "Precio": 4900.00
    },
    {
      "Año": 1999,
      "Marca": "Chevy",
      "Modelo": "Venture",
      "Descripción": "Extended Edition, Very Large",
      "Precio": 5000.00
    }
  ]
}
```

### JSON (JavaScript Object Notation)

- Formato de intercambio de datos ligero y fácil de leer/escribir tanto para humanos como para máquinas.
- Utilizado en el desarrollo web y aplicaciones debido a su compatibilidad con JavaScript.
- En aplicaciones Web permite intercambiar datos entre el cliente y el servidor a través de APIs RESTful.
- También permite almacenar configuraciones y datos de aplicaciones debido a su simplicidad y compatibilidad
- Y además, es utilizado como formato de almacenamiento en bases de datos NoSQL, como MongoDB y CouchDB.

# Formatos de ficheros

## JSON

- El formato JSON está muy ampliamente extendido para intercambio de información a través de web:

```
{  
  "name": "Alderaan",  
  "rotation_period": 24,  
  "diameter": 12500,  
  "climate": "temperate",  
  "terrain": "grasslands, mountains",  
  "population": 2000000000,  
  "residents": [  
    "https://swapi.co/api/people/5/",  
    "https://swapi.co/api/people/68/"  
  ],  
  "films": [  
    "https://swapi.co/api/films/6/",  
    "https://swapi.co/api/films/1/"  
  ],  
  "created": "2014-12-10T11:35:48.479000Z",  
  "edited": "2014-12-20T20:58:18.420000Z",  
  "url": "https://swapi.co/api/planets/2/"  
}
```