



Modelo No Relacional (NoSQL)

*Características, elementos y
ejemplos*

UPEMOR

Objetivo de Aprendizaje

- Al finalizar esta clase, el estudiante podrá:
 - Describir las características del modelo No Relacional
 - Identificar sus elementos y tipos
 - Distinguirlo del modelo relacional
 - Analizar ejemplos prácticos de uso

¿Qué es el Modelo No Relacional?

- Es un modelo de bases de datos que **no utiliza tablas** ni esquemas rígidos como SQL.
- Se basa en estructuras como documentos, grafos o pares clave-valor.
- *Ejemplo:* Un documento JSON que representa el perfil de un usuario con nombre, edad y gustos.

¿Por qué surgió NoSQL?

Las bases relacionales no podían escalar ni adaptarse bien a:

- Millones de usuarios simultáneos
- Datos no estructurados (imágenes, redes sociales)
- Crecimiento masivo de datos (Big Data)

Ejemplo: Facebook y Amazon crearon sus propios sistemas NoSQL.

Diferencias con el Modelo Relacional

Relacional (SQL)	No Relacional (NoSQL)
Tablas	Documentos, grafos, etc.
Esquema fijo	Esquema flexible
Lenguaje SQL	JSON/API/propio
ACID (consistencia fuerte)	BASE (eventual consistency)

Modelo ACID vs. BASE

ACID	BASE
Atomicidad	Basically Available
Consistencia	Soft-state
Aislamiento	Eventual Consistency
Durabilidad	

Características principales

- ✓ Escalabilidad horizontal (más servidores, no más potencia)
- ✓ Alta disponibilidad
- ✓ Soporte a datos no estructurados
- ✓ Esquema dinámico (sin necesidad de definir estructuras rígidas)
- ✓ Modelo BASE (consistencia eventual)

Esquema flexible

- NoSQL permite tener documentos con campos diferentes entre sí.

- *Ejemplo:*

- { "nombre": "Ana", "edad": 22 }
 - { "nombre": "Luis", "carrera": "Computación" }



Escalabilidad horizontal

- A diferencia del modelo relacional (que escala verticalmente), NoSQL escala **horizontalmente** agregando más nodos.
- Ideal para sistemas distribuidos, microservicios y nube.

Tipos de Bases de Datos NoSQL

- **Documentales** (ej: MongoDB)
- **Clave-valor** (ej: Redis)
- **Columnar** (ej: Cassandra)
- **Grafos** (ej: Neo4j)

Cada tipo resuelve problemas específicos.

Bases Documentales

- Almacenan documentos en formato JSON, BSON o XML.
- Estructura anidada y flexible.
- *Ejemplo:* MongoDB, CouchDB.
- *Uso:* perfiles de usuarios, catálogos de productos.

Bases Clave-Valor

- Cada entrada tiene una clave única y un valor asociado.
- Rápido acceso por clave.
- *Ejemplo:* Redis, DynamoDB.
- *Uso:* sesiones de usuario, caché.

Bases Columnar

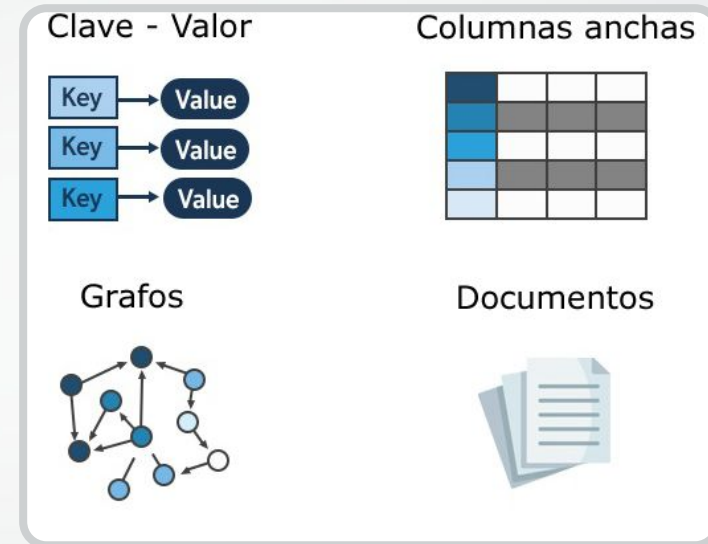
- Organizan los datos por columnas en lugar de filas.
- Buen rendimiento en consultas analíticas.
- *Ejemplo:* Cassandra, HBase.
- *Uso:* análisis de logs, sensores, datos financieros.

Bases de Grafos

- Representan nodos y sus relaciones.
- Ideal para analizar redes complejas.
- *Ejemplo:* Neo4j, Amazon Neptune.
- Uso: redes sociales, recomendaciones, mapas de rutas.

Ejemplos Visuales

- Documento JSON en MongoDB
- Par clave-valor en Redis
- Red de nodos en Neo4j



```
{
  name : "Beau Carnes",
  title : "Developer & Teacher",
  address : {
    address_1 : "123 Main Street",
    city : "Grand Rapids",
    state : "Michigan",
    postal_code : "49503"
  },
  topics : [ "MongoDB", "Python", "JavaScript", "Robots" ],
  employee_number : 1234,
  location : [ 44.9901, 123.0262 ]
}
```

String

Nested Document

Array

Integer

Geo-Spatial Coordinates

Ventajas del Modelo No Relacional

- ✓ Altamente escalable y distribuido
- ✓ Flexibilidad de esquema
- ✓ Rápido rendimiento en operaciones masivas
- ✓ Ideal para Big Data y nube
- ✓ Soporte a datos heterogéneos

Desventajas del Modelo No Relacional

- ⚠ Falta de estandarización
- ⚠ No adecuado para relaciones complejas entre datos
- ⚠ Menor soporte para transacciones críticas
- ⚠ Curva de aprendizaje si vienes de SQL

¿Cuándo usar y cuándo no usar NoSQL?

● Usar NoSQL cuando:

- Necesitas escalabilidad
- Manejas datos no estructurados
- Prioridad es rendimiento

● Evitar NoSQL cuando:

- Requieres transacciones complejas
- Necesitas relaciones estrictas (ej: facturación)

Criterio	Relacional (SQL)	No Relacional (NoSQL)
Estructura	Tablas	Documentos, grafos
Lenguaje	SQL	JSON/API
Escalabilidad	Vertical	Horizontal
Esquema	Fijo	Flexible
Consistencia	Alta (ACID)	Media (BASE)

Comparativa general final

Conclusión y preguntas

- **Conclusión:**

El modelo No Relacional es ideal para sistemas distribuidos, grandes volúmenes de datos y estructuras flexibles. No reemplaza al modelo relacional, sino que lo complementa.

- **Preguntas para investigar:**

- ¿Qué tipo de NoSQL usarías en una app tipo Uber?

- ¿Cuáles ventajas te parecen más importantes?

- ¿Combinarías SQL y NoSQL en un mismo sistema?