



## Bases de datos - No Relacionales

# » Conceptos NoSQL

// Definiciones, características, SGBD NoSQL y usos de Bases de Datos

No Relacionales

## ¿Qué es una Base de Datos NoSQL?

El término **NoSQL** se refiere a la denominación en inglés *Not Only SQL*. Plantea modelos de datos específicos de esquemas flexibles que se adaptan a los requisitos de las aplicaciones más modernas.



## ¿Por qué usar bases de datos NoSQL?

Las bases de datos NoSQL se adaptan muy bien a las aplicaciones modernas, como dispositivos móviles, web y juegos, que requieren bases de datos **flexibles**, **escalables**, de **alto rendimiento** y **altamente funcionales** para proporcionar excelentes experiencias de usuario. Estas características, de forma más específica, hacen referencia a:

**Flexibilidad:** Generalmente presentan esquemas flexibles que permiten un desarrollo más rápido e iterativo. El modelo de datos flexible hace que las bases de datos NoSQL sean ideales para datos semiestructurados y no estructurados.

**Escalabilidad:** las bases de datos NoSQL generalmente están diseñadas para escalar usando clústeres distribuidos de hardware, en vez de escalar añadiendo servidores caros y sólidos. Algunos proveedores de la nube manejan estas operaciones en segundo plano, como un servicio completamente administrado.



**Alto rendimiento:** Están optimizadas para modelos de datos específicos y patrones de acceso que permiten un mayor rendimiento, en vez de intentar lograr una funcionalidad similar a bases de datos relacionales.

**Altamente funcional:** Proporcionan API's altamente funcionales y tipos de datos que están diseñados específicamente para cada uno de sus respectivos modelos de datos.

## Desventajas de las bases de datos no relacionales

Como toda tecnología, existen algunas desventajas a tener en cuenta a la hora de utilizar bases de datos NoSQL:

**Funciones de Fiabilidad:** Se puede decir que es una de las debilidades más grandes de los modelos NoSQL. Estas bases de datos al no admitir funciones de fiabilidad lleva a los desarrolladores a implementar su propio código, lo que agrega mayor complejidad a los sistemas.

**Aplicabilidad:** Esta desventaja está íntimamente relacionada con la anterior. La falta de inclusión de funciones de fiabilidad limita la aplicabilidad de estas base de datos a funciones delicadas como las del sector bancario, limitando mucho su potencial.

**Incompatibilidad en consultas SQL:** En la mayoría de los casos las bases de datos NoSQL son incompatibles a consultas SQL, lo que agrega una dificultad mayor, ya que en caso de ser necesario se debe incluir una consulta manual, que puede hacer los procesos más lentos y complejos.

## Algunas características y usos

Existen algunas características de un proyecto que se pueden tener en cuenta a la hora de elegir por qué tipo de base de datos optar:

- La velocidad de lectura y escritura es más importante que mantener la consistencia.
  - No se sabe cómo se verá la estructura de los datos al final y probablemente cambie mucho.
  - Resulta más fácil adquirir varias máquinas modestas a través del tiempo que invertir en una sola muy potente.
  - Se prefiere facilidad de uso como programador, que no exige demasiados compromisos y el almacenamiento se puede realizar en cualquier lugar.
  - En general, no se exige definir el tipo de datos que se va a almacenar. Un día un campo puede ser un número y al otro un String o Array, incluso un JSON.
- Más que saber qué es la data, NoSQL pone mayor prioridad en cómo acceder dicha data.

- Cuando no se tiene la consistencia de datos como prioridad, distribuir y replicar la base de datos en múltiples máquinas es trivial, y por eso se considera que NoSQL es excelente para bases de datos que necesitan escalar horizontalmente (Por ejemplo: en Big Data, donde una sola máquina queda limitada con facilidad).

Todo dependerá de las necesidades del proyecto. Si se necesita escalabilidad, hay escasez de recursos físicos y, sobre todo, no se necesita mantener la integridad de los datos, se podría elegir una base de datos como esta. También se debe tener en cuenta el gran volumen de los datos, cómo leerlos de manera rápida y sencilla. Ya que algunas NoSQL pueden devolver esa cantidad de registros en un archivo JSON y después son manipulados desde ahí.

Los principales proyectos donde se sugiere utilizar NoSQL son:

- Redes sociales
- Desarrollo Web
- Desarrollo Móvil
- Big Data
- Cloud (XaaS)

## Ejemplo de utilización de una Base de Datos NoSQL

Las bases de datos NoSQL poseen múltiples funciones, marcas, tipos y ventajas.

A modo de ejemplo, se podría utilizar un **JSON**, como es posible emplear en diferentes *Bases de datos documentales*.

Suponiendo que se van a registrar diferentes personas en una colección perteneciente a una base de datos NoSQL con algunos campos especiales. Estos no necesariamente tienen que seguir un *patrón* específico:

```
[
  {
    "nombre": "Mariela",
    "apellido": "Barrios",
    "edad": 48,
    "aficiones": ["vinos", "libros", "running"],
    "amigos": [
      {
        "nombre": "Ricardo",
        "edad": 45
      },
      {
        "nombre": "Federico",
        "edad": 35
      }
    ]
  }
]
```



Ahora, si se quiere añadir otros datos de otra persona, con algunas características diferentes, la podemos hacer sin mayores problemas de la siguiente forma:

```
{ "nombre": "Eduardo",  
  "estudios": "Periodismo",  
  "amigos": 30  
}
```

Claramente, en un modelo relacional esto sería imposible de hacer, estando entre una de las tantas ventajas.