1. **¿Qué son las bases de datos NoSQL y en qué se diferencian de las bases de datos relacionales?**

Las bases de datos NoSQL (Not Only SQL) son sistemas de almacenamiento de datos que no utilizan el modelo relacional tradicional basado en tablas. Se diferencian de las bases de datos relacionales en varios aspectos clave:

* + No requieren esquemas rígidos, lo que permite mayor flexibilidad en la estructura de los datos.
  + Son altamente escalables y están diseñadas para manejar grandes volúmenes de datos de manera eficiente.
  + No utilizan SQL como lenguaje principal de consulta, sino APIs y otros métodos específicos.
  + Priorizan la disponibilidad y la partición de datos sobre la consistencia inmediata, siguiendo el teorema CAP.

1. **Mencione y describa brevemente los principales tipos de bases de datos NoSQL.**

Existen cuatro tipos principales de bases de datos NoSQL:

* + **Basadas en documentos**: Almacenan datos en documentos JSON o BSON, como MongoDB y CouchDB.
  + **Basadas en clave-valor**: Usan un modelo simple de pares clave-valor, como Redis y DynamoDB.
  + **Basadas en columnas**: Diseñadas para manejar grandes volúmenes de datos estructurados en columnas en lugar de filas, como Cassandra y HBase.
  + **Basadas en grafos**: Optimizadas para gestionar relaciones complejas entre datos, como Neo4j y ArangoDB.

1. **Explique en qué casos es recomendable utilizar una base de datos NoSQL en lugar de una relacional.**

Es recomendable usar NoSQL en los siguientes casos:

* + Cuando se manejan grandes volúmenes de datos que requieren escalabilidad horizontal.
  + En aplicaciones web y móviles con crecimiento rápido y esquemas de datos cambiantes.
  + Para sistemas que necesitan alta disponibilidad y tolerancia a fallos.
  + En proyectos con datos semiestructurados o sin estructura fija.
  + Para aplicaciones en tiempo real que requieren respuestas rápidas, como redes sociales o sistemas de análisis de datos.

1. **¿Qué es la consistencia eventual en las bases de datos NoSQL y cómo difiere de la consistencia fuerte en las bases de datos SQL?**

La **consistencia eventual** es un modelo en el que los datos se replican en diferentes nodos de un sistema distribuido, y aunque pueden no estar actualizados en todos los nodos de inmediato, con el tiempo convergen a un estado consistente.

En cambio, la **consistencia fuerte**, característica de bases de datos SQL, garantiza que todas las transacciones se reflejen de inmediato en todas las copias de los datos, asegurando integridad pero sacrificando disponibilidad y rendimiento en sistemas distribuidos.

1. **Explique el modelo CAP y cómo afecta a las bases de datos NoSQL.**

El **teorema CAP** establece que en un sistema distribuido solo se pueden garantizar dos de las tres propiedades siguientes simultáneamente:

* + **Consistencia** (Consistency): Todos los nodos ven la misma información en todo momento.
  + **Disponibilidad** (Availability): Siempre se puede acceder a los datos, incluso si hay fallos en algunos nodos.
  + **Tolerancia a particiones** (Partition Tolerance): El sistema sigue funcionando a pesar de fallos en la red.

Las bases de datos NoSQL suelen priorizar disponibilidad y tolerancia a particiones sobre consistencia inmediata, lo que les permite ser más escalables y resilientes en entornos distribuidos.

1. **Describa el funcionamiento y los casos de uso de una base de datos basada en documentos.**

Una base de datos basada en documentos almacena información en documentos JSON, BSON o XML, organizados en colecciones en lugar de tablas. Cada documento puede contener datos con estructuras flexibles y anidadas.

**Casos de uso:**

* + Aplicaciones web y móviles con datos dinámicos y sin esquemas fijos.
  + Sistemas de gestión de contenido y comercio electrónico.
  + Almacenamiento de logs y analítica de datos en tiempo real.

1. **¿Cómo se almacenan y consultan los datos en una base de datos de tipo clave-valor?**

En una base de datos de clave-valor, los datos se almacenan como pares únicos, donde cada clave apunta a un valor asociado. Las consultas se realizan buscando la clave correspondiente, lo que permite respuestas rápidas.

**Ejemplo:** En Redis, un par clave-valor puede ser "usuario123": "Juan Pérez", y acceder al dato es tan simple como buscar usuario123.

1. **Mencione algunas bases de datos NoSQL populares y el tipo de datos que utilizan.**
   * **MongoDB** (documentos JSON/BSON)
   * **Redis** (clave-valor, estructuras en memoria)
   * **Cassandra** (almacenamiento basado en columnas)
   * **Neo4j** (base de datos de grafos)
   * **CouchDB** (documentos JSON con sincronización distribuida)
2. **Ventajas y desventajas de usar bases de datos NoSQL.**

**Ventajas:**

* + Escalabilidad horizontal eficiente.
  + Flexibilidad en el esquema de datos.
  + Mayor velocidad en consultas para grandes volúmenes de datos.
  + Buen rendimiento en sistemas distribuidos y tolerancia a fallos.

**Desventajas:**

* + Falta de estandarización y compatibilidad con SQL tradicional.
  + Requiere más esfuerzo en la gestión de la consistencia de datos.
  + No es ideal para transacciones complejas que necesitan ACID.

1. **Explique qué es el escalado horizontal y cómo se implementa en las bases de datos NoSQL.**

El **escalado horizontal** consiste en aumentar la capacidad del sistema añadiendo más servidores en lugar de mejorar el hardware de un solo servidor (escalado vertical). En bases de datos NoSQL, se implementa mediante **sharding**, donde los datos se distribuyen en múltiples nodos, y **replicación**, que copia los datos en varios servidores para mejorar disponibilidad y tolerancia a fallos.