



TECNOLOGO ANALISIS Y DESARROLLO DE SOFTWARE

FICHA 2758368



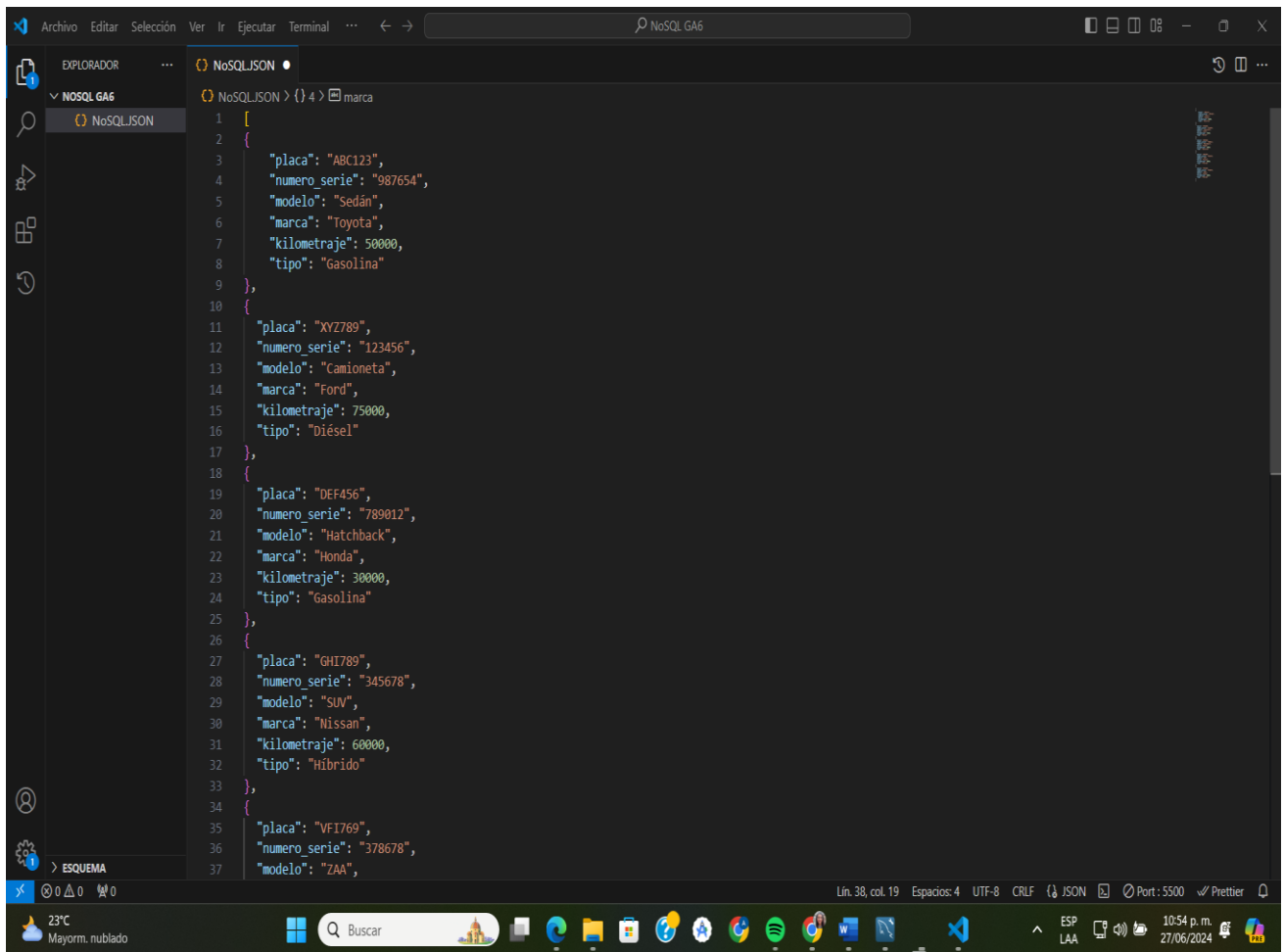
GA6-220501096-AA1-EV03- Creación de los objetos de la Base de Datos.

Instructor Ricardo Alfonso González Vargas.

Aprendiz Andrea Villarreal Meglan

Introducción

NoSQL (también conocido como “no sólo SQL” o “no SQL”) es un enfoque utilizado en el diseño de bases de datos que permite el almacenamiento y consulta de datos fuera de las estructuras tradicionales que se encuentran en las bases de datos relacionales. A diferencia de las bases de datos relacionales, que utilizan una estructura tabular, las bases de datos NoSQL albergan datos dentro de una estructura de datos, como un documento JSON. Esto significa que NoSQL no requiere un esquema fijo, lo que permite una escalabilidad rápida para administrar grandes conjuntos de datos no estructurados. Además, las bases de datos NoSQL son distribuidas, lo que garantiza la disponibilidad y fiabilidad de los datos al almacenar copias en varios servidores. En resumen, NoSQL es una opción popular para aplicaciones web modernas debido a su rendimiento y facilidad de uso.



```
1  [
2  {
3    "placa": "ABC123",
4    "numero_serie": "987654",
5    "modelo": "Sedán",
6    "marca": "Toyota",
7    "kilometraje": 50000,
8    "tipo": "Gasolina"
9  },
10 {
11   "placa": "XYZ789",
12   "numero_serie": "123456",
13   "modelo": "Camioneta",
14   "marca": "Ford",
15   "kilometraje": 75000,
16   "tipo": "Diésel"
17 },
18 {
19   "placa": "DEF456",
20   "numero_serie": "789012",
21   "modelo": "Hatchback",
22   "marca": "Honda",
23   "kilometraje": 30000,
24   "tipo": "Gasolina"
25 },
26 {
27   "placa": "GHI789",
28   "numero_serie": "345678",
29   "modelo": "SUV",
30   "marca": "Nissan",
31   "kilometraje": 60000,
32   "tipo": "Híbrido"
33 },
34 {
35   "placa": "VFI769",
36   "numero_serie": "378678",
37   "modelo": "ZAA",
```

sentencias de colecciones y consultas:

- **Creación de colecciones:** En lugar de tablas, las bases de datos NoSQL almacenan datos en colecciones. Puedes crear una colección para agrupar documentos relacionados. Por ejemplo, si estás modelando datos de carros, podrías tener una colección llamada “Carros” donde cada documento representa un carro con sus atributos (placa, número de serie, modelo, marca, etc.).
- **Consultas:** Puedes buscar datos dentro de una colección utilizando consultas. Por ejemplo, podrías consultar todos los carros de una marca específica o encontrar los carros con un kilometraje superior a cierto valor.
- **Inserciones:** Para agregar datos a una colección, puedes insertar nuevos documentos. Por ejemplo, si un usuario registra un nuevo carro, insertarías un documento con la información relevante.

- **Actualizaciones:** Si necesitas modificar datos existentes, puedes actualizar documentos en la colección. Por ejemplo, si un carro cambia de dueño, actualizarías el campo correspondiente en el documento.

Bases de datos NoSQL son una categoría de bases de datos que difieren de las **bases de datos relacionales** en varios aspectos. Permíteme explicarte las diferencias clave:

Modelo de datos:

- **Relacionales (SQL):** Utilizan tablas para almacenar datos, con filas y columnas. Cada tabla tiene un esquema fijo.
- **No relacionales (NoSQL):** No siguen un esquema rígido. Pueden usar modelos como documentos, grafos o columnas.

Lenguaje de consulta:

- **SQL:** Las bases de datos relacionales utilizan SQL para consultas.
- **NoSQL:** NoSQL no se basa en SQL, aunque algunas ofrecen lenguajes de consulta similares

Escalabilidad:

- **SQL:** Escalabilidad vertical (agregar más recursos a un servidor existente).
- **NoSQL:** Escalabilidad horizontal (distribución en múltiples servidores).

Consistencia y disponibilidad:

- **SQL:** Enfoca la consistencia (datos siempre coherentes).
- **NoSQL:** Prioriza la disponibilidad (datos accesibles incluso si algunos servidores fallan).

En resumen, NoSQL es ideal para aplicaciones modernas que requieren manejo eficiente de grandes volúmenes de datos. SQL sigue siendo popular, pero NoSQL ofrece flexibilidad y escalabilidad

Objetivos

- **Escalabilidad:** Diseñar una base de datos que pueda manejar un alto volumen de datos y crecer sin problemas a medida que aumenta la carga de trabajo.
- **Rendimiento:** Optimizar las consultas y operaciones para obtener respuestas rápidas, especialmente en aplicaciones web o móviles.
- **Flexibilidad:** Modelar datos de manera más libre y sin un esquema fijo, lo que permite cambios y adaptaciones más sencillas.
- **Distribución:** Garantizar que los datos estén disponibles incluso si algunos servidores fallan, mediante la distribución en múltiples nodos.
- **Eficiencia en costos:** Reducir los gastos asociados con infraestructura y administración de bases de datos

Conclusiones

trabajar con datos en bases de datos no relacionales mediante sentencias de creación, inserción y consulta. Es fundamental aplicar estos conocimientos en proyectos reales para optimizar la gestión de información