



Reto

¡Bienvenidos al reto de programación en NestJS!

En este reto tendrás la oportunidad de poner en práctica tus habilidades en programación orientada a objetos y en el framework NestJS, al crear una aplicación que permita realizar operaciones CRUD en entidades relacionadas entre sí.

Para este reto, te hemos proporcionado un modelo de entidad de Automóvil y otros dos modelos de entidad de Cliente y Vendedor que se relacionan con el modelo de Automóvil.

En este ejercicio, tendrás que implementar la persistencia de los datos utilizando arreglos en lugar de una base de datos. Además, tendrás que asegurarte de utilizar los conceptos fundamentales de programación orientada a objetos, como la encapsulación, la herencia, el polimorfismo y la abstracción para diseñar e implementar tus clases de manera efectiva.

Estamos emocionados de ver lo que puedes hacer con este reto. ¡Buena suerte!

Crea una entidad **automovil** que tenga los siguientes atributos:

- **id**: número de identificación del automóvil.
- **marca**: marca del automóvil.
- **modelo**: modelo del automóvil.
- **año**: año del automóvil.
- **vendedor**: referencia al vendedor del automóvil.
- **cliente**: referencia al cliente que ha comprado el automóvil (opcional).

Crea una entidad **Cliente** que tenga los siguientes atributos:

- **id**: número de identificación del cliente.
- **nombre**: nombre del cliente.
- **automovil**: referencia al automóvil que ha comprado el cliente (opcional).

Crea una entidad **Vendedor** que tenga los siguientes atributos:

- **id**: número de identificación del vendedor.
- **nombre**: nombre completo del vendedor.
- **autos**: arreglo de referencias a los automóviles que el vendedor ha vendido.

Información adicional

1. Los id, deben ser generados con `uuidv4()`
2. Revisare bien que es opcional

3. Se debe usar de forma obligatoria, los conceptos de POO (Herencia, polimorfismo, encapsulación y abstracción)
4. Evitar código repetido (obligatorio)
5. `CLIENTE AUTOMOVIL Y VENDEDOR` deben tener sus modulos.
6. La información se debe guardar en arreglos.
7. Todos los métodos deben interactuar con los arreglos ya sea búsqueda, actualizar eliminar o crear.
8. Se deben implementar DTO
9. Se deben implementar interfaces

Metodos

Para la entidad `Automovil`:

- Crear un nuevo automóvil (`createCar`).
- Actualizar la información de un automóvil existente (`updateCar`).
- Eliminar un automóvil (`deleteCar`).
- Obtener todos los automóviles (`getAllCars`).
- Obtener un automóvil por su identificador (`getCarById`).
- Asignar un coche a un cliente (`assignCarToClient`).
- Eliminar la asignación de un coche a un cliente (`unassignCarFromClient`).

Para la entidad `Cliente`:

- Crear un nuevo cliente (`createClient`).
- Actualizar la información de un cliente existente (`updateClient`).
- Eliminar un cliente (`deleteClient`).
- Obtener todos los clientes (`getAllClients`).
- Obtener un cliente por su identificador (`getClientById`).

- Obtener el automóvil comprado por un cliente (`getClientCar`).
- Asignar un automóvil a un cliente (`assignCarToClient`).
- Eliminar la asignación de un automóvil a un cliente (`unassignCarFromClient`).

Para la entidad `vendedor` :

- Crear un nuevo vendedor (`createSeller`).
- Actualizar la información de un vendedor existente (`updateSeller`).
- Eliminar un vendedor (`deleteSeller`).
- Obtener todos los vendedores (`getAllSellers`).
- Obtener un vendedor por su identificador (`getSellerById`).
- Obtener todos los automóviles vendidos por un vendedor (`getSoldCarsBySellerId`).
- Agregar un nuevo automóvil vendido por un vendedor (`addSoldCar`).

Ejemplos

Automóvil

```
{
  "id": "a1d1c8b3-2a3e-42ed-bfe6-78d064b7c707",
  "marca": "Toyota",
  "modelo": "Corolla",
  "año": 2022,
  "vendedor": {
    "id": "e6b23c8f-24f1-42c9-b712-cf55d8c82c0b",
    "nombre": "Juan Pérez"
  },
  "cliente": {
    "id": "6b81f8d9-8827-4c12-bd99-5fa90c16df04",
    "nombre": "Ana Gómez"
  }
}
```

Cliente

```
{  
  "id": "6b81f8d9-8827-4c12-bd99-5fa90c16df04",  
  "nombre": "Ana Gómez",  
  "cocheComprado": {  
    "id": "a1d1c8b3-2a3e-42ed-bfe6-78d064b7c707",  
    "marca": "Toyota",  
    "modelo": "Corolla",  
    "año": 2022,  
    "vendedor": {  
      "id": "e6b23c8f-24f1-42c9-b712-cf55d8c82c0b",  
      "nombre": "Juan Pérez"  
    }  
  }  
}
```

Vendedor

```
{  
  "id": "e6b23c8f-24f1-42c9-b712-cf55d8c82c0b",  
  "nombre": "Juan Pérez",  
  "cochesVendidos": [  
    {  
      "id": "a1d1c8b3-2a3e-42ed-bfe6-78d064b7c707",  
      "marca": "Toyota",  
      "modelo": "Corolla",  
      "año": 2022,  
      "cliente": {  
        "id": "6b81f8d9-8827-4c12-bd99-5fa90c16df04",  
        "nombre": "Ana Gómez"  
      }  
    },  
    {  
      "id": "c62a977a-6d03-43b3-b6b3-b6b80f1ac9f6",  
      "marca": "Honda",  
      "modelo": "Civic",  
      "año": 2021,  
      "cliente": null  
    }  
  ]  
}
```