

**Actividad:**

**Creación de los Objetos de la Base de Datos  
GA6-220501096-AA1-EV03**

**Aprendiz:**

Wilmer Jair Espinosa Silva

CC: 1.095.910.391

Instructor:

**ISRAEL ARBONA GUERRERO**

Servicio Nacional de aprendizaje-SENA

**Curso: TECNOLOGÍA EN ANÁLISIS Y DESARROLLO DE SOFTWARE**

Ficha: 2455285

1. Realice la sentencia para crear un objeto JSON que modele los datos de un carro (placa, número serie, modelo, marca, kilometraje, tipo).

RTA: Para crear un objeto JSON que modele los datos de un carro en MongoDB, puedes utilizar la función **db.collection.insertOne()** de la siguiente manera:

```
db.cars.insertOne({
  placa: "ABC-123",
  numero_serie: "123456",
  modelo: "Civic",
  marca: "Honda",
  kilometraje: 150000,
  tipo: "sedan"
})
```

Esto insertará un nuevo documento en la colección "cars" con los campos especificados. Puedes cambiar los valores de los campos según tus necesidades.

2. Realice la sentencia para insertar 5 datos en la colección creada.

RTA: Podemos utilizar la función **db.collection.insertMany()**, de la siguiente manera:

```
db.cars.insertMany([
  {
    placa: "DEF-456",
    numero_serie: "789123",
    modelo: "Accord",
    marca: "Honda",
    kilometraje: 200000,
    tipo: "sedan"
  },
  {
    placa: "GHI-789",
    numero_serie: "654123",
    modelo: "Camry",
    marca: "Toyota",
    kilometraje: 120000,
    tipo: "sedan"
  },
  {
    placa: "JKL-012",
    numero_serie: "321098",
    modelo: "Prius",
    marca: "Toyota",
    kilometraje: 90000,
    tipo: "hatchback"
  },
  {
    placa: "MNO-345",
    numero_serie: "987654",
    modelo: "Corolla",
    marca: "Toyota",
    kilometraje: 110000,
    tipo: "sedan"
  },
  {
    placa: "PQR-678",
    numero_serie: "456789",
    modelo: "Civic",
    marca: "Honda",
    kilometraje: 130000,
    tipo: "sedan"
  }
])
```

```
{
  placa: "JKL-012",
  numero_serie: "369258",
  modelo: "Mustang",
  marca: "Ford",
  kilometraje: 80000,
  tipo: "coupe"
},
{
  placa: "MNO-345",
  numero_serie: "147369",
  modelo: "Corolla",
  marca: "Toyota",
  kilometraje: 100000,
  tipo: "sedan"
},
```

```
{
  placa: "PQR-678",
  numero_serie: "258147",
  modelo: "Focus",
  marca: "Ford",
  kilometraje: 60000,
  tipo: "hatchback"
}
])
```

Esto insertará cinco documentos en la colección "cars" con los valores especificados. Puedes cambiar los valores de los campos según tus necesidades y agregar o quitar documentos según sea necesario.

3. Realice la sentencia para consultar los datos ingresados en la colección.

**RTA:** Puedo utilizar la función **db.collection.find()**, de la siguiente manera:

```
db.cars.find()
```

Esto te devolverá todos los documentos de la colección "cars". Si quieres filtrar la búsqueda por algún campo específico, puedes añadir un objeto JSON como parámetro, por ejemplo:

```
db.cars.find({marca: "Toyota"})
```

Esto te devolverá solo los documentos de la colección "cars" con el valor "Toyota" en el campo "marca".

También puedes proyectar solo los campos específicos de los documentos que necesites,

```
db.cars.find({}, {placa: 1, modelo: 1, _id: 0})
```

Esto te devolverá solo los campos "placa" y "modelo" de cada documento de la colección "cars" y ocultando el campo "\_id"

#### 4. Actualice los datos de la colección.

**RTA:** Puedo utilizar la función **db.collection.updateOne()** o **db.collection.updateMany()**. La función **updateOne()**

La función **updateOne()** se utiliza para actualizar un solo documento de la colección, mientras que la función **updateMany()** se utiliza para actualizar varios documentos.

Ambas funciones tienen dos parámetros: el primer parámetro es un objeto JSON que especifica el filtro para seleccionar los documentos que se van a actualizar, y el segundo parámetro es un objeto JSON que especifica las actualizaciones que se van a realizar.

Por ejemplo, si quieres actualizar el campo "kilometraje" de un carro con la placa "ABC-123" a 200000, puedes utilizar la función **updateOne()** de la siguiente manera:

```
db.cars.updateOne({placa: "ABC-123"}, {$set: {kilometraje: 200000}})
```

Si quieres actualizar el campo "kilometraje" de varios carros con la marca "Honda" a 200000, puedes utilizar la función **updateMany()** de la siguiente manera:

```
db.cars.updateMany({marca: "Honda"}, {$set: {kilometraje: 200000}})
```

Es importante mencionar que si no especificas el operador \$set en el segundo parámetro, MongoDB reemplazará el documento completo, en lugar de actualizar solo el campo especificado.

Hay varios operadores de actualización disponibles en MongoDB, como \$inc, \$mul, \$rename, \$unset, etc. que puedes utilizar para realizar diferentes tipos de actualizaciones en tus datos.