

3.7 Actualización de Documentos con `updateOne()`, `update()`, y `replaceOne()`

En esta sección, exploraremos cómo actualizar documentos en una colección de MongoDB utilizando tres métodos diferentes: `updateOne()`, `update()`, y `replaceOne()`. Estos métodos te permiten modificar registros existentes en tu base de datos de MongoDB.

1. Actualización con `updateOne()`

1.1. Sintaxis de `updateOne()`

`updateOne()` te permite actualizar un solo documento que cumple con ciertos criterios de búsqueda. Su sintaxis básica es la siguiente:

```
db.nombreDeLaColección.updateOne(filtro, actualización)
```

- `filtro`: Es un objeto que define los criterios para seleccionar el documento que deseas actualizar.
- `actualización`: Es un objeto que especifica las modificaciones que deseas realizar en el documento seleccionado.

Ejemplo de uso de `updateOne()`:

Supongamos que deseamos actualizar la edad de un usuario específico:

```
// Actualizar la edad de un usuario
db.usuarios.updateOne(
  { nombre: "Juan Pérez" },
  { $set: { edad: 36 } }
)
```

2. Actualización con `update()`

2.1. Sintaxis de `update()`

`update()` te permite actualizar múltiples documentos que cumplen con ciertos criterios de búsqueda. Su sintaxis básica es la siguiente:

```
db.nombreDeLaColección.update(filtro, actualización, opciones)
```

- `filtro`: Es un objeto que define los criterios para seleccionar los documentos que deseas actualizar.
- `actualización`: Es un objeto que especifica las modificaciones que deseas realizar en los documentos seleccionados.
- `opciones` (opcional): Permite especificar opciones adicionales para la operación de actualización.

Ejemplo de uso de `update()`:

Actualicemos la edad de todos los usuarios mayores de 30 años:

```
// Actualizar la edad de todos los usuarios mayores de 30 años
db.usuarios.update(
  { edad: { $gt: 30 } },
  { $set: { edad: 31 } },
  { multi: true }
)
```

3. Reemplazo de Documentos con `replaceOne()`

3.1. Sintaxis de `replaceOne()`

`replaceOne()` te permite reemplazar un solo documento que cumple con ciertos criterios de búsqueda por completo con un nuevo documento. Su sintaxis básica es la siguiente:

```
db.nombreDeLaColección.replaceOne(filtro, nuevoDocumento)
```

- `filtro` : Es un objeto que define los criterios para seleccionar el documento que deseas reemplazar.
- `nuevoDocumento` : Es el nuevo documento que reemplazará al documento seleccionado.

Ejemplo de uso de `replaceOne()` :

Reemplacemos completamente el documento de "Juan Pérez" con uno nuevo:

```
// Reemplazar el documento de "Juan Pérez" con uno nuevo
db.usuarios.replaceOne(
  { nombre: "Juan Pérez" },
  { nombre: "Juan Pérez", edad: 36, correo: "juan@example.com" }
)
```

4. Ejemplos de Aplicación

A continuación, te proporcionamos ejemplos de aplicación de estos métodos de actualización utilizando la base de datos de prueba de usuarios que hemos cargado previamente:

- **Actualizar la Edad de un Usuario:**

```
// Actualizar la edad de "Juan Pérez" con updateOne()
db.usuarios.updateOne(
  { nombre: "Juan Pérez" },
  { $set: { edad: 36 } }
)
```

- **Actualizar la Edad de Usuarios Mayores de 30:**

```
// Actualizar la edad de todos los usuarios mayores de 30 con update()
db.usuarios.update(
  { edad: { $gt: 30 } },
  { $set: { edad: 31 } },
  { multi: true }
)
```

- **Reemplazar un Documento de Usuario:**

```
// Reemplazar el documento de "Juan Pérez" con uno nuevo con replaceOne()
db.usuarios.replaceOne(
  { nombre: "Juan Pérez" },
  { nombre: "Juan Pérez", edad: 36, correo: "juan@example.com" }
)
```

Estos ejemplos te ayudarán a comprender cómo utilizar `updateOne()`, `update()`, y `replaceOne()` para actualizar documentos en MongoDB. Puedes utilizarlos durante tus lecciones para mostrar a tus estudiantes cómo realizar operaciones de actualización en una base de datos MongoDB.