**Métodos de MongoDB en Java**

En este documento se explican los métodos específicos de **MongoDB en Java** utilizando la biblioteca **MongoDB Java Driver**.

**1. Conectar a MongoDB**

**Método:**

MongoClient mongoClient = MongoClients.create("mongodb://usuario:password@host:27017");

MongoDatabase database = mongoClient.getDatabase("miBaseDeDatos");

**Explicación:**

* MongoClients.create(String uri): Crea una conexión con la base de datos.
* getDatabase(String name): Obtiene una base de datos específica.

**2. Obtener una colección**

**Método:**

MongoCollection<Document> collection = database.getCollection("miColeccion");

**Explicación:**

* getCollection(String name): Obtiene una colección dentro de la base de datos.

**3. Insertar documentos**

**Método:**

**Insertar un solo documento:**

Document doc = new Document("nombre", "Juan").append("edad", 30);

collection.insertOne(doc);

**Insertar múltiples documentos:**

List<Document> documentos = Arrays.asList(

new Document("nombre", "Maria").append("edad", 25),

new Document("nombre", "Pedro").append("edad", 40)

);

collection.insertMany(documentos);

**Explicación:**

* insertOne(Document doc): Inserta un solo documento en la colección.
* insertMany(List<Document> docs): Inserta múltiples documentos a la vez.

**4. Buscar documentos**

**Método:**

**Buscar todos los documentos:**

FindIterable<Document> documentos = collection.find();

for (Document doc : documentos) {

System.out.println(doc.toJson());

}

**Buscar con filtro:**

Document resultado = collection.find(Filters.eq("nombre", "Juan")).first();

**Explicación:**

* find(): Devuelve todos los documentos de la colección.
* find(Filters.eq("campo", valor)): Filtra documentos por un valor específico.
* .first(): Obtiene el primer documento que coincide con el filtro.

**5. Actualizar documentos**

**Método:**

**Actualizar un solo documento:**

collection.updateOne(Filters.eq("nombre", "Juan"), Updates.set("edad", 35));

**Actualizar múltiples documentos:**

collection.updateMany(Filters.gt("edad", 30), Updates.inc("edad", 1));

**Explicación:**

* updateOne(Filters.eq(campo, valor), Updates.set(campo, nuevoValor)): Modifica el primer documento encontrado.
* updateMany(Filters.gt(campo, valor), Updates.inc(campo, incremento)): Modifica todos los documentos que cumplen la condición.

**6. Eliminar documentos**

**Método:**

**Eliminar un solo documento:**

collection.deleteOne(Filters.eq("nombre", "Juan"));

**Eliminar múltiples documentos:**

collection.deleteMany(Filters.lt("edad", 25));

**Explicación:**

* deleteOne(Filters.eq(campo, valor)): Elimina el primer documento que cumple la condición.
* deleteMany(Filters.lt(campo, valor)): Elimina todos los documentos que cumplen la condición.

**7. Contar documentos**

**Método:**

long total = collection.countDocuments();

long filtrados = collection.countDocuments(Filters.gt("edad", 30));

**Explicación:**

* countDocuments(): Cuenta todos los documentos en la colección.
* countDocuments(Filters.gt(campo, valor)): Cuenta los documentos que cumplen la condición.

**8. Uso de Agregaciones**

**Método:**

AggregateIterable<Document> resultado = collection.aggregate(Arrays.asList(

Aggregates.group("$nombre", Accumulators.sum("totalEdad", "$edad"))

));

for (Document doc : resultado) {

System.out.println(doc.toJson());

}

**Explicación:**

* aggregate(List<AggregationStage>): Ejecuta una agregación.
* Aggregates.group("$campo", Accumulators.sum("nombreCampo", "$campo")): Agrupa por un campo y suma los valores de otro.

Estos son los métodos principales de **MongoDB en Java** y cómo utilizarlos en diferentes escenarios. ¡Espero que te ayude!