**Apuntes para MongoDB y Java**

**1. Crear Usuarios en MongoDB**

**En Java:**

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

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

MongoCollection<Document> coleccionUsuarios = database.getCollection("usuarios");

Document usuario = new Document("nombre", "usuario1")

.append("contrasena", "hashedPassword");

coleccionUsuarios.insertOne(usuario);

**En MongoDB Shell:**

use espectaculos

db.usuarios.insertOne({nombre: "usuario1", contrasena: "hashedPassword"})

**2. Gestión de Entradas en Eventos**

**En Java (Ver entradas disponibles):**

MongoCollection<Document> collection = database.getCollection("evento1");

long entradasDisponibles = collection.countDocuments(Filters.eq("nombre\_cliente", null));

System.out.println("Entradas disponibles: " + entradasDisponibles);

**En MongoDB Shell:**

use espectaculos

db.evento1.countDocuments({nombre\_cliente: null})

**Comprar entrada en Java:**

collection.updateOne(Filters.eq("nombre\_cliente", null),

Updates.set("nombre\_cliente", "Cliente"));

**Comprar entrada en MongoDB Shell:**

db.evento1.updateOne({nombre\_cliente: null}, {$set: {nombre\_cliente: "Cliente"}})

**3. Facturas y Ventas en MongoDB**

**Contar Facturas por Cliente en Java:**

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

Aggregates.match(Filters.eq("cliente", "Juan")),

Aggregates.group("$cliente", Accumulators.sum("totalFacturas", 1))

));

**En MongoDB Shell:**

db.facturas.aggregate([

{ $match: { cliente: "Juan" } },

{ $group: { \_id: "$cliente", totalFacturas: { $sum: 1 } } }

])

**Obtener total de ventas por cliente en Java:**

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

Aggregates.match(Filters.eq("cliente", "Juan")),

Aggregates.unwind("$lineas\_factura"),

Aggregates.group("$cliente", Accumulators.sum("totalVentas",

new Document("$multiply", Arrays.asList("$lineas\_factura.precio\_unitario", "$lineas\_factura.cantidad"))))

));

**En MongoDB Shell:**

db.facturas.aggregate([

{ $match: { cliente: "Juan" } },

{ $unwind: "$lineas\_factura" },

{ $group: { \_id: "$cliente", totalVentas: { $sum: { $multiply: ["$lineas\_factura.precio\_unitario", "$lineas\_factura.cantidad"] } } } }

])

**4. Biblioteca: Libros y Autores**

**Agregar Autor en Java:**

Document autor = new Document("nombre", "Gabriel García Márquez").append("nacionalidad", "Colombiano");

autoresCollection.insertOne(autor);

**En MongoDB Shell:**

db.autores.insertOne({nombre: "Gabriel García Márquez", nacionalidad: "Colombiano"})

**Agregar Libro en Java:**

Document libro = new Document("titulo", "Cien años de soledad")

.append("isbn", "123456789")

.append("autor\_id", autor.getObjectId("\_id"))

.append("disponible", true);

librosCollection.insertOne(libro);

**En MongoDB Shell:**

db.libros.insertOne({titulo: "Cien años de soledad", isbn: "123456789", disponible: true})

**Prestar Libro en Java:**

librosCollection.updateOne(Filters.eq("titulo", "Cien años de soledad"),

Updates.set("disponible", false));

**En MongoDB Shell:**

db.libros.updateOne({titulo: "Cien años de soledad"}, {$set: {disponible: false}})

Estos comandos resumen las operaciones más importantes que has implementado en tus proyectos. ¡Espero que te ayuden en tu examen!

**Apuntes para MongoDB y Java**

**1. Crear Usuarios en MongoDB**

**En Java:**

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

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

MongoCollection<Document> coleccionUsuarios = database.getCollection("usuarios");

Document usuario = new Document("nombre", "usuario1")

.append("contrasena", "hashedPassword");

coleccionUsuarios.insertOne(usuario);

**En MongoDB Shell:**

use espectaculos

db.usuarios.insertOne({nombre: "usuario1", contrasena: "hashedPassword"})

**2. Gestión de Entradas en Eventos**

**En Java (Ver entradas disponibles):**

MongoCollection<Document> collection = database.getCollection("evento1");

long entradasDisponibles = collection.countDocuments(Filters.eq("nombre\_cliente", null));

System.out.println("Entradas disponibles: " + entradasDisponibles);

**En MongoDB Shell:**

use espectaculos

db.evento1.countDocuments({nombre\_cliente: null})

**Comprar entrada en Java:**

collection.updateOne(Filters.eq("nombre\_cliente", null),

Updates.set("nombre\_cliente", "Cliente"));

**Comprar entrada en MongoDB Shell:**

db.evento1.updateOne({nombre\_cliente: null}, {$set: {nombre\_cliente: "Cliente"}})

**3. Facturas y Ventas en MongoDB**

**Contar Facturas por Cliente en Java:**

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

Aggregates.match(Filters.eq("cliente", "Juan")),

Aggregates.group("$cliente", Accumulators.sum("totalFacturas", 1))

));

**En MongoDB Shell:**

db.facturas.aggregate([

{ $match: { cliente: "Juan" } },

{ $group: { \_id: "$cliente", totalFacturas: { $sum: 1 } } }

])

**Obtener total de ventas por cliente en Java:**

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

Aggregates.match(Filters.eq("cliente", "Juan")),

Aggregates.unwind("$lineas\_factura"),

Aggregates.group("$cliente", Accumulators.sum("totalVentas",

new Document("$multiply", Arrays.asList("$lineas\_factura.precio\_unitario", "$lineas\_factura.cantidad"))))

));

**En MongoDB Shell:**

db.facturas.aggregate([

{ $match: { cliente: "Juan" } },

{ $unwind: "$lineas\_factura" },

{ $group: { \_id: "$cliente", totalVentas: { $sum: { $multiply: ["$lineas\_factura.precio\_unitario", "$lineas\_factura.cantidad"] } } } }

])

**4. Biblioteca: Libros y Autores**

**Agregar Autor en Java:**

Document autor = new Document("nombre", "Gabriel García Márquez").append("nacionalidad", "Colombiano");

autoresCollection.insertOne(autor);

**En MongoDB Shell:**

db.autores.insertOne({nombre: "Gabriel García Márquez", nacionalidad: "Colombiano"})

**Agregar Libro en Java:**

Document libro = new Document("titulo", "Cien años de soledad")

.append("isbn", "123456789")

.append("autor\_id", autor.getObjectId("\_id"))

.append("disponible", true);

librosCollection.insertOne(libro);

**En MongoDB Shell:**

db.libros.insertOne({titulo: "Cien años de soledad", isbn: "123456789", disponible: true})

**Prestar Libro en Java:**

librosCollection.updateOne(Filters.eq("titulo", "Cien años de soledad"),

Updates.set("disponible", false));

**En MongoDB Shell:**

db.libros.updateOne({titulo: "Cien años de soledad"}, {$set: {disponible: false}})

Estos comandos resumen las operaciones más importantes que has implementado en tus proyectos. ¡Espero que te ayuden en tu examen!