

Venta de entradas. Parte 1.

Instalación de BD

Comenzamos creando la base de datos y el usuario para acceder a esta, lo haremos en Oracle con la shell de MongoDB:

```
USE entradas_db;
```

```
db.createUser({
  user: "entradas",
  pwd: "1234",
  roles:
  [
    {role: "readWrite", db: "entradas_db" },
    {role: "dbAdmin", db: "entradas_db" }
  ]
});
```

Una vez creada, ya podremos crear nuestro programa en java. Para ello, creo un proyecto Maven en Eclipse:

```
package entradas;

import java.util.Scanner;

import com.mongodb.client.MongoClient;
import com.mongodb.client.MongoClients;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;
import org.bson.Document;

import java.util.logging.Level;
import java.util.logging.Logger;

public class TalonarioEntradas {

    private static final String MONGO_URI =
"mongodb://entradas:1234@79.72.63.217:27017/entradas_db?authSource=entradas_db";
    private static final String DATABASE_NAME = "entradas_db";

    public static void main(String[] args) {
        // Esto es para deshabilitar los logs de MongoDB
        Logger mongoLogger = Logger.getLogger("org.mongodb.driver");
        mongoLogger.setLevel(Level.WARNING);

        Scanner scanner = new Scanner(System.in);

        // Pedimos los datos del evento
        System.out.print("Ingrese el nombre del evento: ");
        String nombreEvento = scanner.nextLine();

        System.out.print("Ingrese el número de localidades: ");
        int numLocalidades = scanner.nextInt();
```

```

// Esto protege una futura colección de usuarios
if (nombreEvento.equals("usuarios")) {
    System.out.println("Error. No se permite un evento llamado 'usuarios'.");
    return;
}

// Nos conectamos a MongoDB
try (MongoClient mongoClient = MongoClient.create(MONGO_URI)) {
    MongoDB database = mongoClient.getDatabase(DATABASE_NAME);

    // Obtenemos la colección para el evento
    MongoCollection<Document> collection = database.getCollection(nombreEvento);

    // Verificamos si la colección existe y tiene documentos
    if (collection != null && collection.countDocuments() > 0) {
        collection.deleteMany(new Document());
        System.out.println("Colección existente eliminada.");
    }

    // Creamos entradas y las agregamos a la colección
    for (int i = 1; i ≤ numLocalidades; i++) {
        Document entrada = new Document("numeroEntrada",
i).append("observaciones", "").append("nombreCliente",
        null);
        collection.insertOne(entrada);
    }

    System.out.println("Talónario creado exitosamente para el evento: " +
nombreEvento);
} catch (Exception e) {
    System.err.println("Error al conectar con la base de datos o crear el
talónario: " + e.getMessage());
    e.printStackTrace();
}
}
}

```

Y en pom.xml me aseguro de configurar las dependencias:

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>1</groupId>
    <artifactId>AD404_VentaDeEntradas11</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <dependencies>
        <dependency>
            <groupId>org.mongodb</groupId>
            <artifactId>mongo-java-driver</artifactId>
            <version>3.12.14</version>
        </dependency>
    </dependencies>
</project>

```

Y al ejecutarlo, veremos que se conecta y genera la colección correctamente en nuestra base de datos:

```
entradas_db> db.getCollectionNames()
[]
entradas_db> db.getCollectionNames()
[ 'monologo' ]
entradas_db>
```

Problems Javadoc Declaration Console X

<terminated> TalonarioEntradas (1) [Java Application] C:\Users\dam2\p2

Ingrese el nombre del evento: monologo
 Ingrese el número de localidades: 50
 Talonario creado exitosamente para el evento: monologo

```
entradas_db> db.monologo.find()
[
  {
    _id: ObjectId('67990515588ef33638c0341b'),
    numeroEntrada: 1,
    observaciones: '',
    nombreCliente: null
  },
  {
    _id: ObjectId('67990515588ef33638c0341c'),
    numeroEntrada: 2,
    observaciones: '',
    nombreCliente: null
  },
  {
    _id: ObjectId('67990515588ef33638c0341d'),
    numeroEntrada: 3,
    observaciones: '',
    nombreCliente: null
  },
  {
    _id: ObjectId('67990515588ef33638c0341e'),
    numeroEntrada: 4,
```