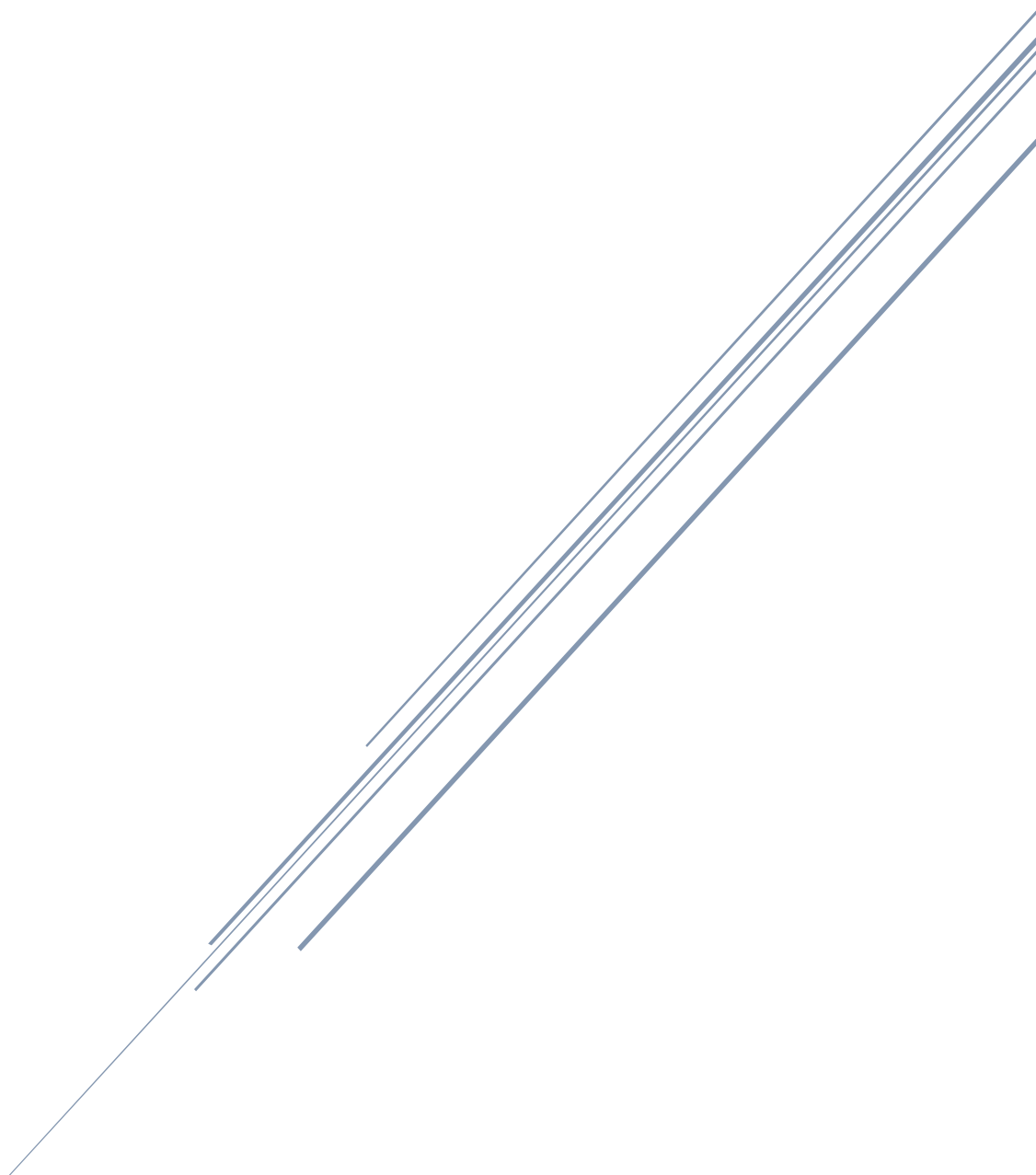


ACTIVIDAD 0.16

Héctor Redondo Jurado




CampusFP

Para empezar, al crear el proyecto he agregado la dependencia del Maven Repository de MongoDB.

```
<dependency>
  <groupId>org.mongodb</groupId>
  <artifactId>mongo-java-driver</artifactId>
  <version>3.12.12</version>
</dependency>
```

Luego he tenido que descargarme la librería de Mongo que es la siguiente:

 mongodb.mongo.java.driver

Compile ▾

Una vez tenemos esto, he tenido que cambiar de nombre a HelloAplication a Main.java para que no me diera un error que me estaba dando con una clase que así se llamaba.

```
public class Main extends Application { new *
```

Que es esta de aquí.

Luego solo he tenido que pedirle a chatgpt que me ayudase con el código para ya estando conectados que me saliera la tabla con los productos, ids y precios.

```
TableView<Document> tableView = new TableView<>();
ObservableList<Document> data = FXCollections.observableArrayList();
```

Para empezar, tenemos la creación de la tabla y una lista observable. En este caso lo datos son de tipo document, que es una clase proporcionada por el controlador de MongoDB.

```
try (MongoClient mongoClient = MongoClient.create(MONGO_URI)) {
    MongoDBDatabase database = mongoClient.getDatabase(DATABASE_NAME);
    MongoCollection<Document> collection = database.getCollection(COLLECTION_NAME);

    FindIterable<Document> documents = collection.find();
    for (Document document : documents) {
        data.add(document);
    }
} catch (Exception e) {
    e.printStackTrace();
}
```

Aquí establecemos la conexión a la base de datos de MongoDB Atlas utilizando la cadena de conexión que nos ha proporcionado. Luego accedemos a la base de datos y a la colección específica.

```
private static final String MONGO_URI = "mongodbsrv://lector:lector@cluster0.lxjdu9.mongodb.net/";
private static final String DATABASE_NAME = "ejemplo"; 1 usage
private static final String COLLECTION_NAME = "productos"; 1 usage
```

Que están guardados al principio del código, debajo de los import.

```
for (String key : data.get(0).keySet()) {  
    TableColumn<Document, String> column = new TableColumn<>(key);  
    column.setCellValueFactory(param -> {  
        Object value = param.getValue().get(key);  
        return value == null ? new SimpleStringProperty("") : new SimpleStringProperty(value.toString());  
    });  
    tableView.getColumns().add(column);  
}  
  
tableView.setItems(data);
```

En esta parte creamos las columnas para la tabla basadas en las claves de los documentos. En cada una creamos una columna con el nombre de la clave como encabezado.

```
tableView.setItems(data);
```

Y para acabar, configuramos la lista observable como el conjunto de datos de la tabla, de modo que los documentos que hemos recuperado se muestran en la tabla.