

# UNIVERSIDAD TECNOLÓGICA DE TULA-TEPEJI



## Documentacion de sistema CRUD

**Materia: Programacion orientada a objetos**

**Docente: Paola García Chávez**

**Equipo:**

**Alan Cruz Reyes**

**Angel Eduardo Jimenez Garcia**

**Ethan Rene Almazan Rivera**

**Yaretzi Hernandez Alonso**

**3TIDSM-G4**

**Mayo - Agosto 2025**

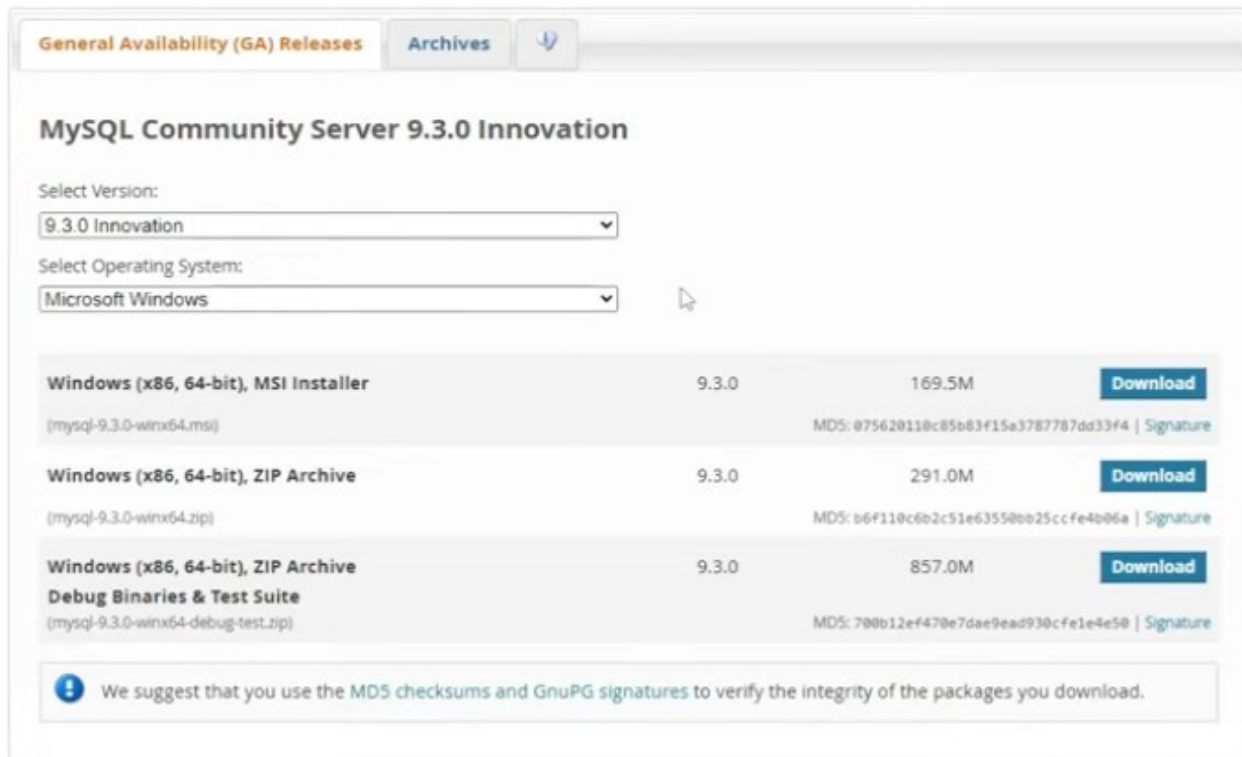
# Documentacion de sistema CRUD

Trabajo final del cuatrimestre Mayo – Agosto 2025

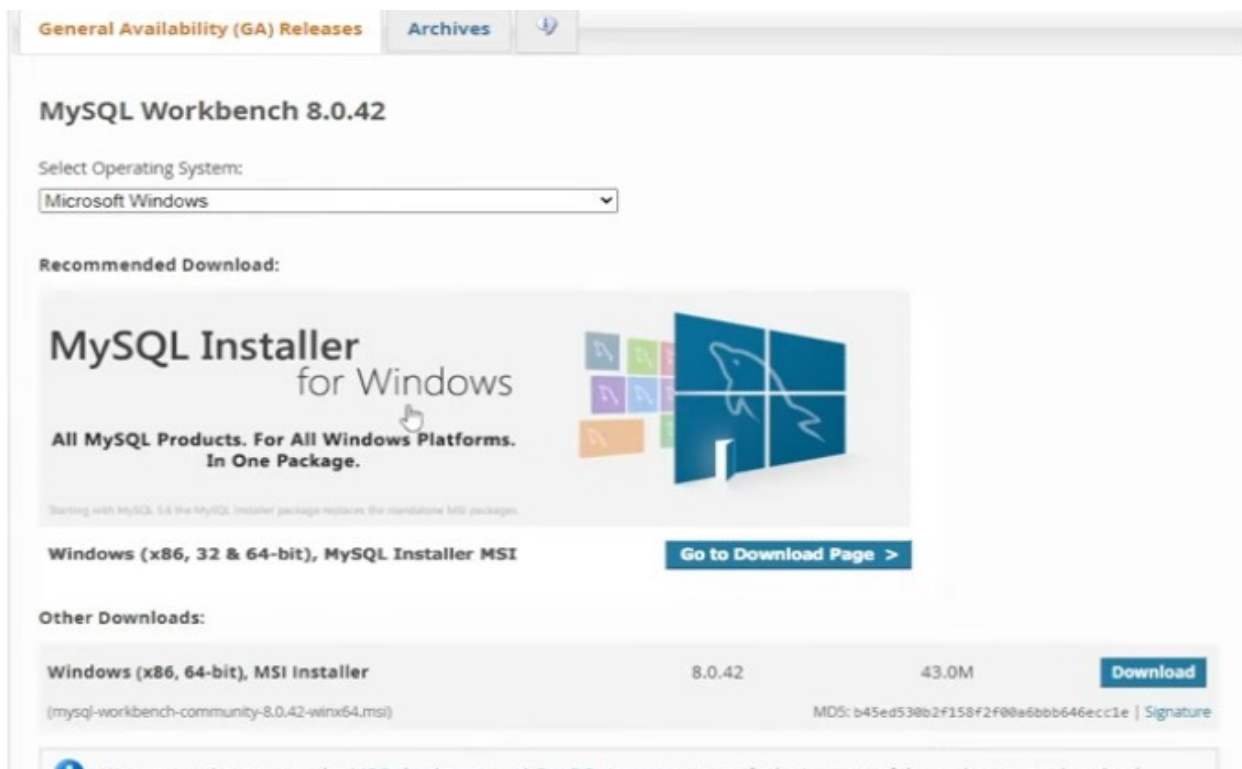
## Preparacion del ambiente de desarrollo

### 1.- Instalacion de MySQL

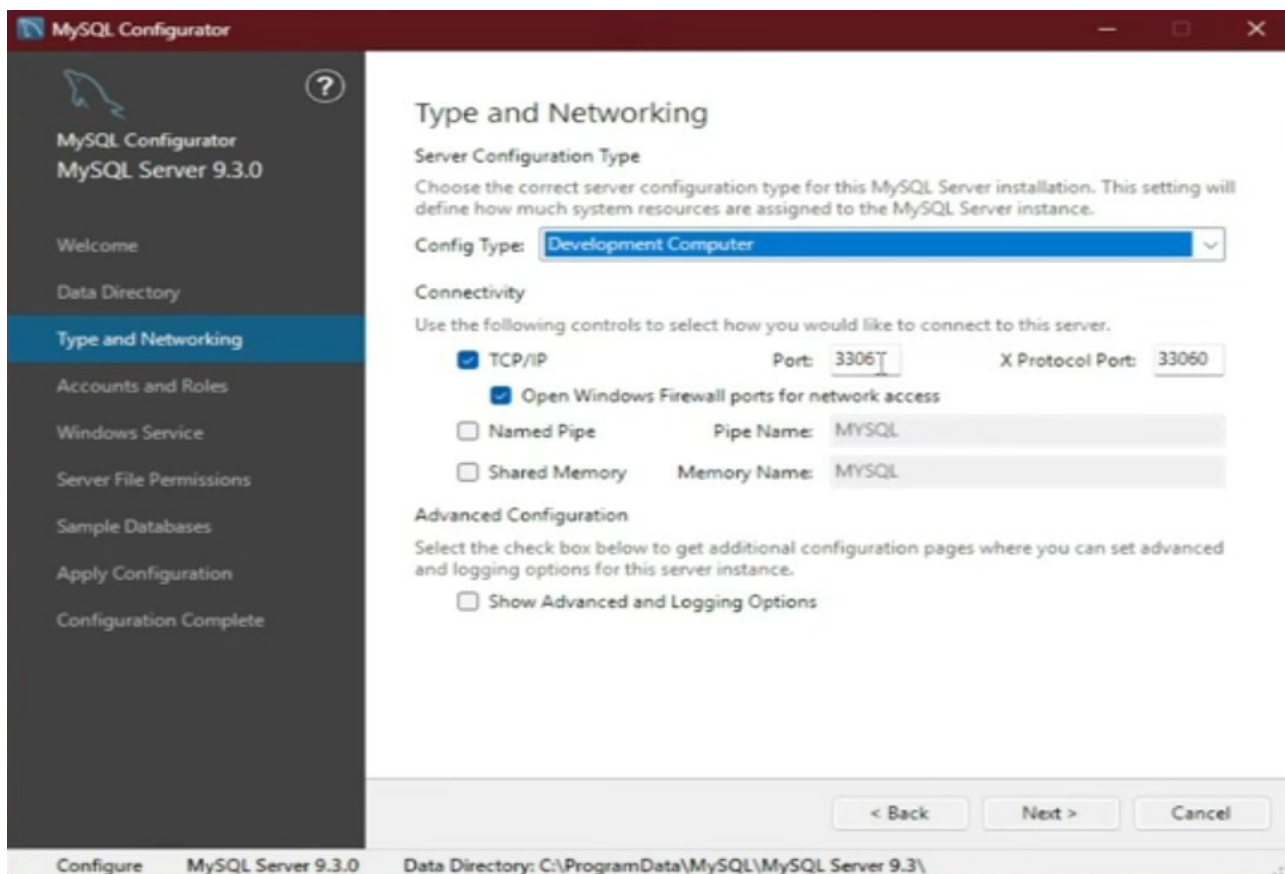
- Entramos a la pagina de MySQL oficial <https://www.mysql.com/> y daremos clic en MySQL Community (GPL) Downloads
- Seleccionaremos MySQL Community Server



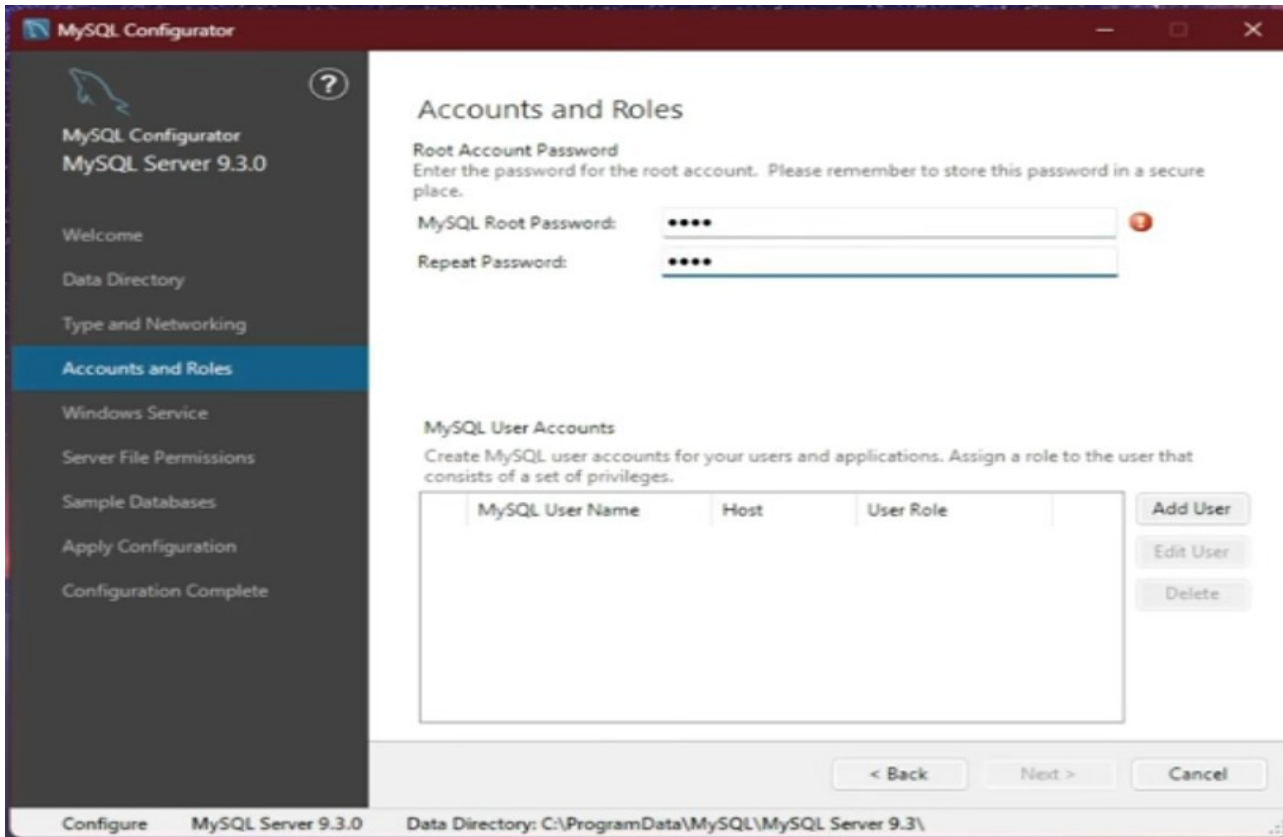
- Descargaremos la version de MSI installer
- Regresaremos a MySQL Community (GPL) Downloads y daremos clic en la opcion MySQL Workbench para tener un mejor entorno de desarrollo de bases de datos en lugar de usar la linea de comandos



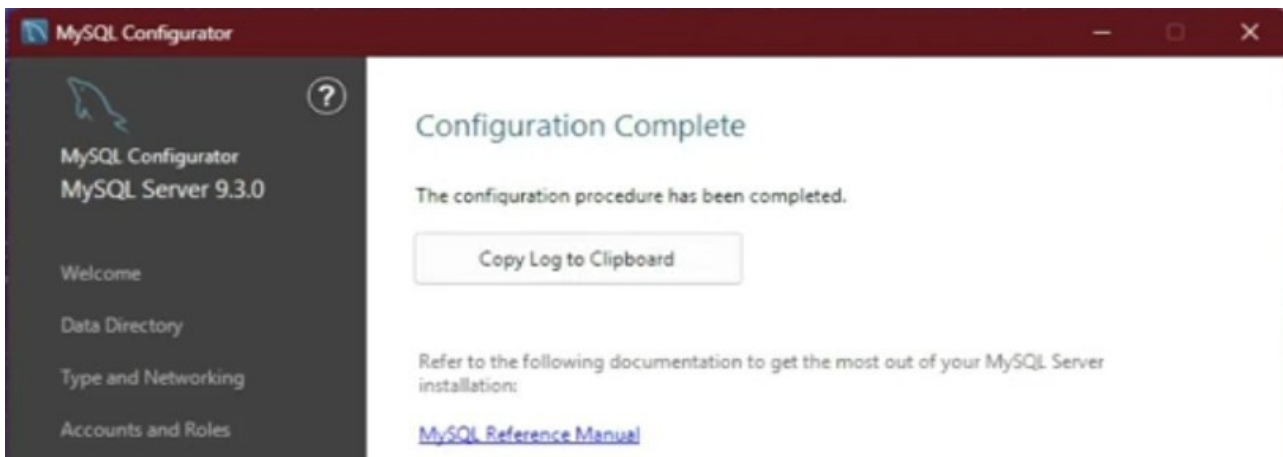
- Descargamos e instalamos
- Durante la inslacion tomaremos nota del puerto que vamos a usar para nuestra base de datos



- Seleccionamos la configuracion de Development Computer



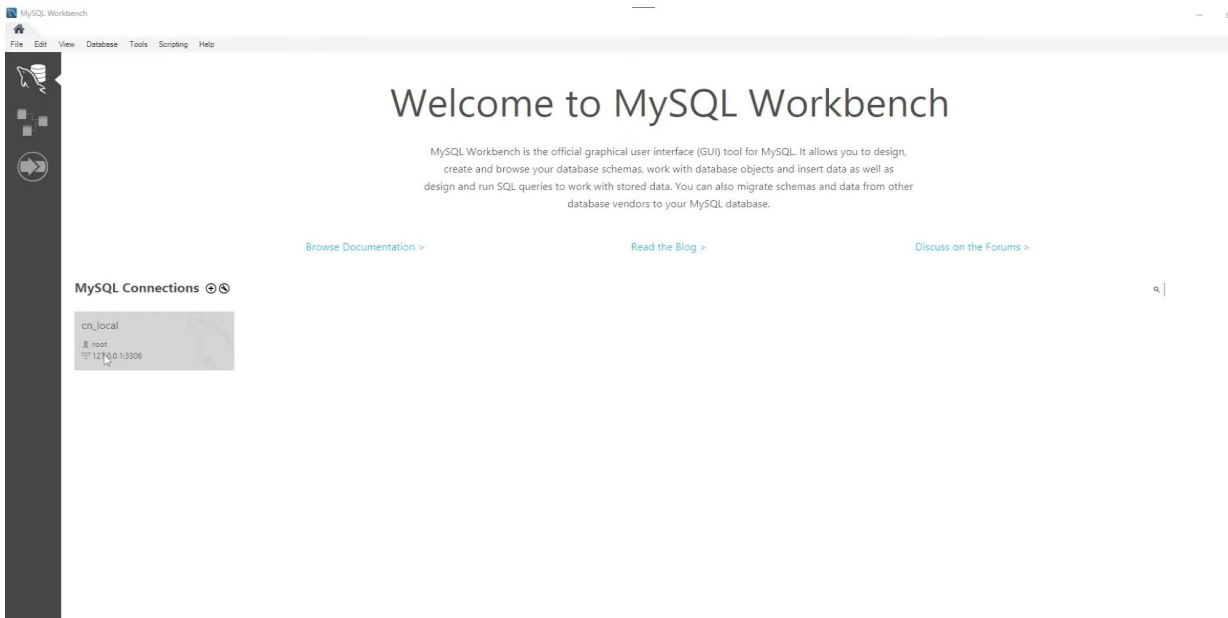
- Creamos una contraseña para poder usar nuestra base de datos. Esta sera la contraseña de nuestro perfil root
- Seleccionamos todas las opciones y damos en continuar



- Terminamos la instalacion de MySQL
- Hacemos clic en el ainstalador de MySQL workbench
- Simplemente hacemos clic en Siguiente a todo y terminamos la instalacion

## 1.1.- Base de datos

- Entramos a MySQL workbench



- Entramos a la conexión que configuramos e ingresamos nuestro usuario (root) y nuestra clave



- Entramos y crearemos la siguiente base de datos con estos atributos:

////////////////////////////////////

```
create database dbUsuarios;
```

```
use dbUsuarios;
```

```
create table sexo(
```

id int auto\_increment not null primary key,

sexo varchar(50)

```
insert into sexo (sexo) values ("Masculino");
insert into sexo (sexo) values ("Femenino");
select * from sexo;
```

```
create table Usuarios(  
id int auto_increment not null primary key,  
nombre varchar(50),  
apellidos varchar(50),  
fkSexo int,  
fechaNac date,  
correo varchar(50),  
foreign key (fkSexo) references sexo(id) on DELETE CASCADE ON UPDATE CASCADE  
);
```

## 2.- Instalacion de JavaFX

- 

Create beautiful user interfaces and turn your design into an interactive prototype. Scene Builder closes the gap between designers and developers by creating user interfaces which can be directly used in a JavaFX application.

- Instalamos dando a siguiente hasta terminar la instalacion
- Instalamos el Java Developer Kit (JDK) a travez del la pagina oficial de Oracle

ORACLE

ProductsIndustriesResourcesCustomersPartnersDevelopersCompany

🔍🇺🇸👤View Accounts📄Contact Sales

Tools and resourcesJava downloadsJava archive

JDK 24 is the latest release of the Java SE Platform.  
JDK 21 is the latest *Long-Term Support (LTS)* release of the Java SE Platform.  
Earlier JDK versions are available below.

Learn about Java SE Subscription

JDK 24JDK 21GraalVM for JDK 24GraalVM for JDK 21

Java SE Development Kit 24.0.2 downloads

JDK 24 binaries are free to use in production and free to redistribute, at no cost, under the [Oracle No-Fee Terms and Conditions](#) (NFTC).  
JDK 24 will receive updates under these terms, until September 2025, when it will be superseded by JDK 25.

LinuxmacOSWindows

Product/file description	File size	Download
ARM64 Compressed Archive	229.37 MB	<a href="https://download.oracle.com/java/24/latest/jdk-24_linux-aarch64_bin.tar.gz">https://download.oracle.com/java/24/latest/jdk-24_linux-aarch64_bin.tar.gz</a> (sha256)
ARM64 RPM Package	228.98 MB	<a href="https://download.oracle.com/java/24/latest/jdk-24_linux-aarch64_bin.rpm">https://download.oracle.com/java/24/latest/jdk-24_linux-aarch64_bin.rpm</a> (sha256) (OL 9 GPG Key)
x64 Compressed Archive	232.15 MB	<a href="https://download.oracle.com/java/24/latest/jdk-24_linux-x64_bin.tar.gz">https://download.oracle.com/java/24/latest/jdk-24_linux-x64_bin.tar.gz</a> (sha256)
x64 Debian Package	200.21 MB	<a href="https://download.oracle.com/java/24/latest/jdk-24_linux-x64_bin.deb">https://download.oracle.com/java/24/latest/jdk-24_linux-x64_bin.deb</a> (sha256)
x64 RPM Package	231.71 MB	<a href="https://download.oracle.com/java/24/latest/jdk-24_linux-x64_bin.rpm">https://download.oracle.com/java/24/latest/jdk-24_linux-x64_bin.rpm</a> (sha256) (OL 9 GPG Key)

- Luego descargamos el SDK seleccionando la opcion correcta en la lista. Posteriormente instalaremos de manera manual esos archivos

Downloads

JavaFX versionOperating SystemArchitectureType

21.0.1 [LTS][any][any][any]

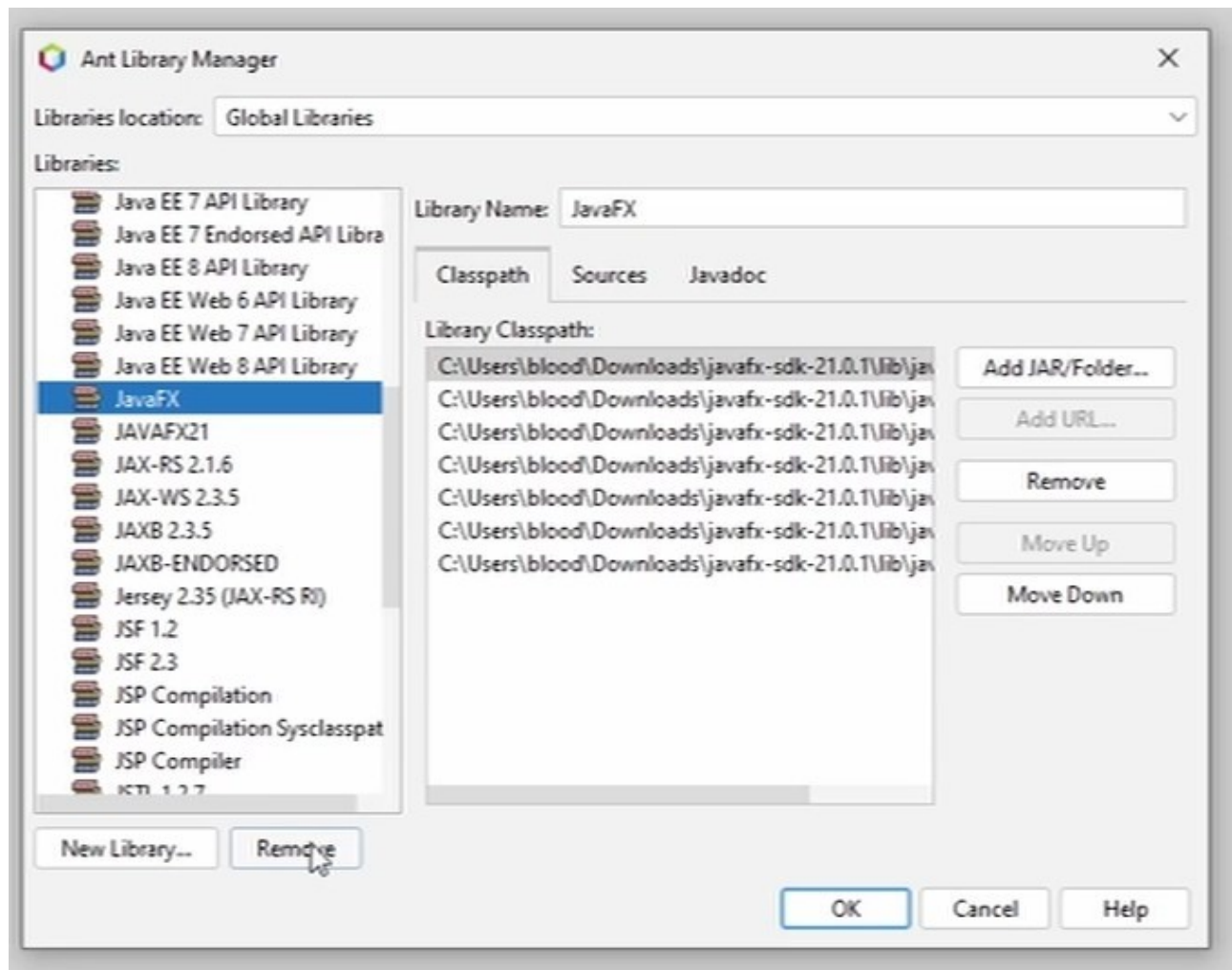
☐ Include older versions

Supported Platforms

OS	Version	Architecture	Type	Download
Linux	21.0.1	x64	SDK	<a href="#">Download</a> (SHA256)
Linux	21.0.1	x64	jmods	<a href="#">Download</a> (SHA256)
macOS	21.0.1	aarch64	SDK	<a href="#">Download</a> (SHA256)
macOS	21.0.1	aarch64	jmods	<a href="#">Download</a> (SHA256)
macOS	21.0.1	x64	SDK	<a href="#">Download</a> (SHA256)
macOS	21.0.1	x64	jmods	<a href="#">Download</a> (SHA256)
Windows	21.0.1	x64	SDK	<a href="#">Download</a> (SHA256)
Windows	21.0.1	x64	jmods	<a href="#">Download</a> (SHA256)
Javadoc	21.0.1		Javadoc	<a href="#">Download</a> (SHA256)

- Entramos y seleccionamos la carpeta javaFX y la pasamos a nuestro proyecto de NetBeans. Los archivos de se deberan de ver asi:

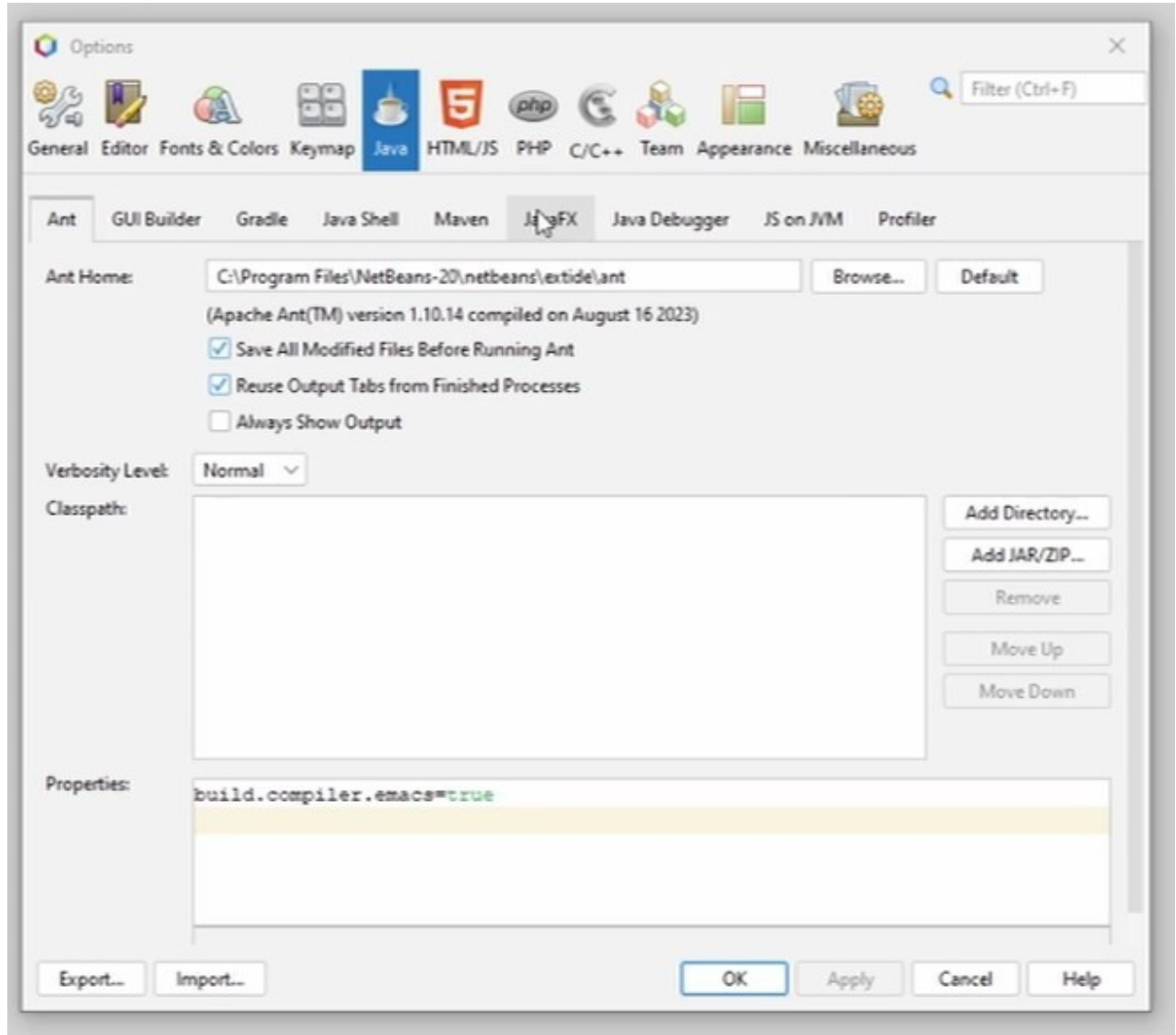
- 



- Debemos hacer la conexión entre Java y SM Builder
- Entramos a Tools, Opciones y seleccionamos JavaFX



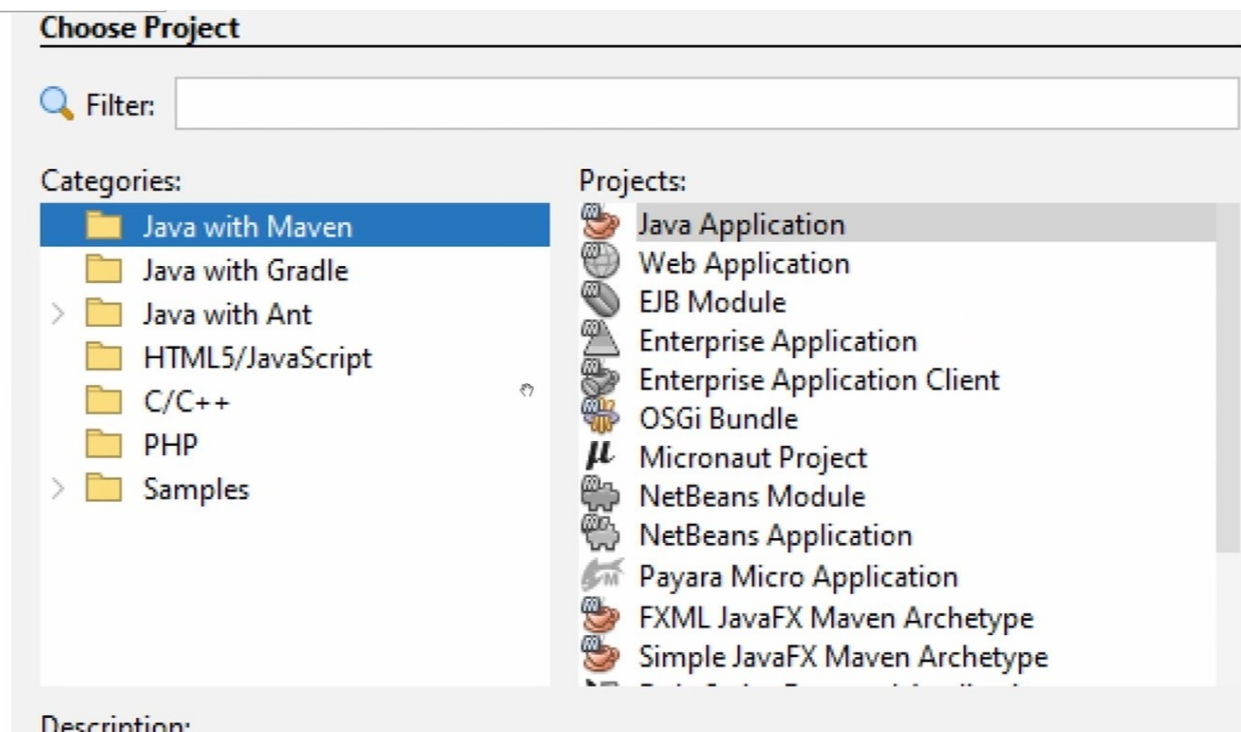
- 



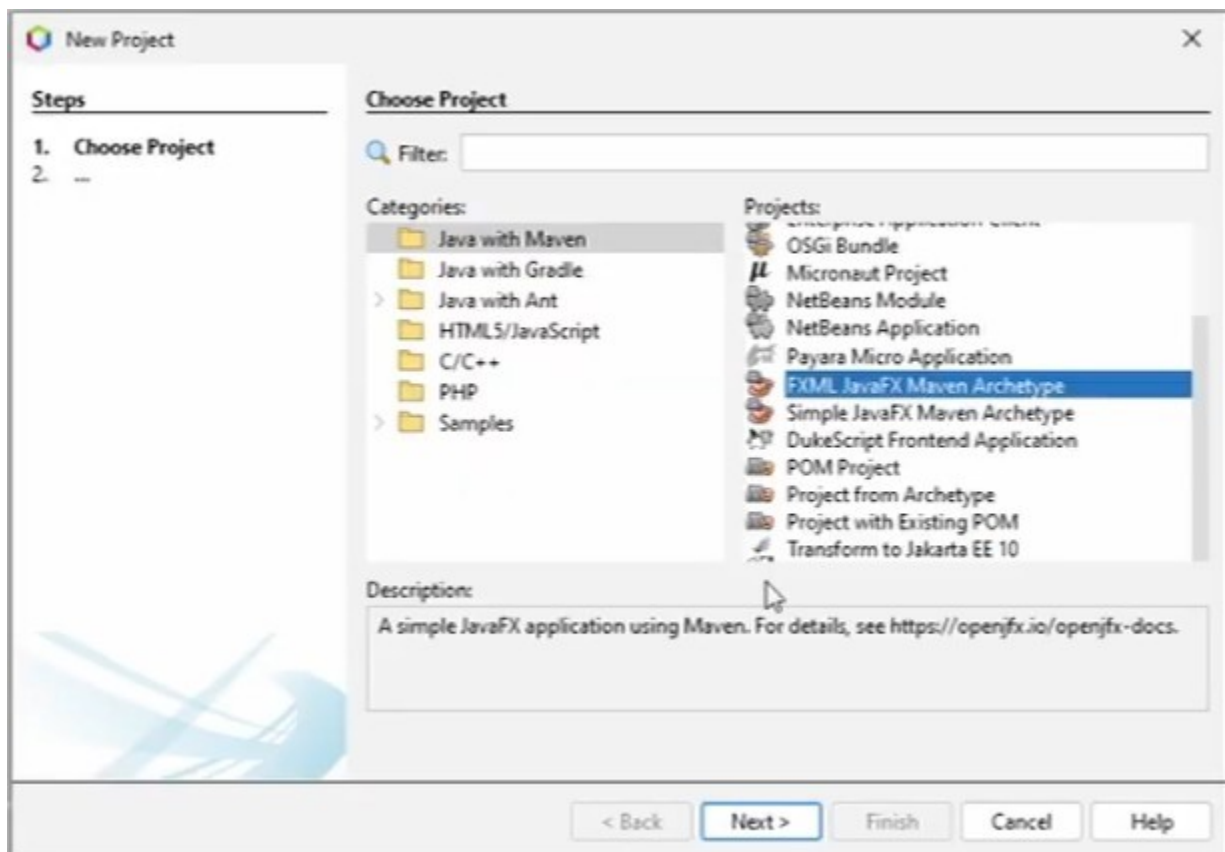
- Le damos a Ok, Siguiente e Instalar

### 3.- Creacion del proyecto

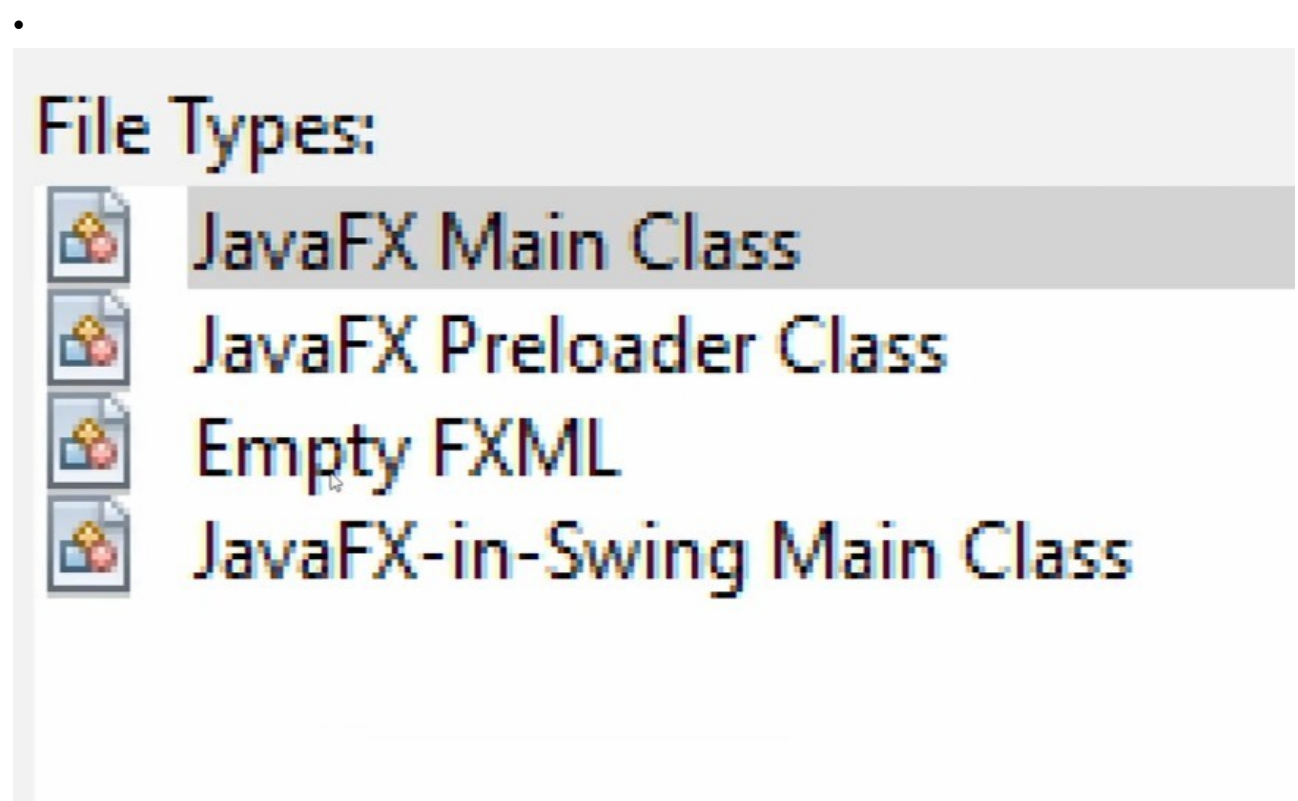
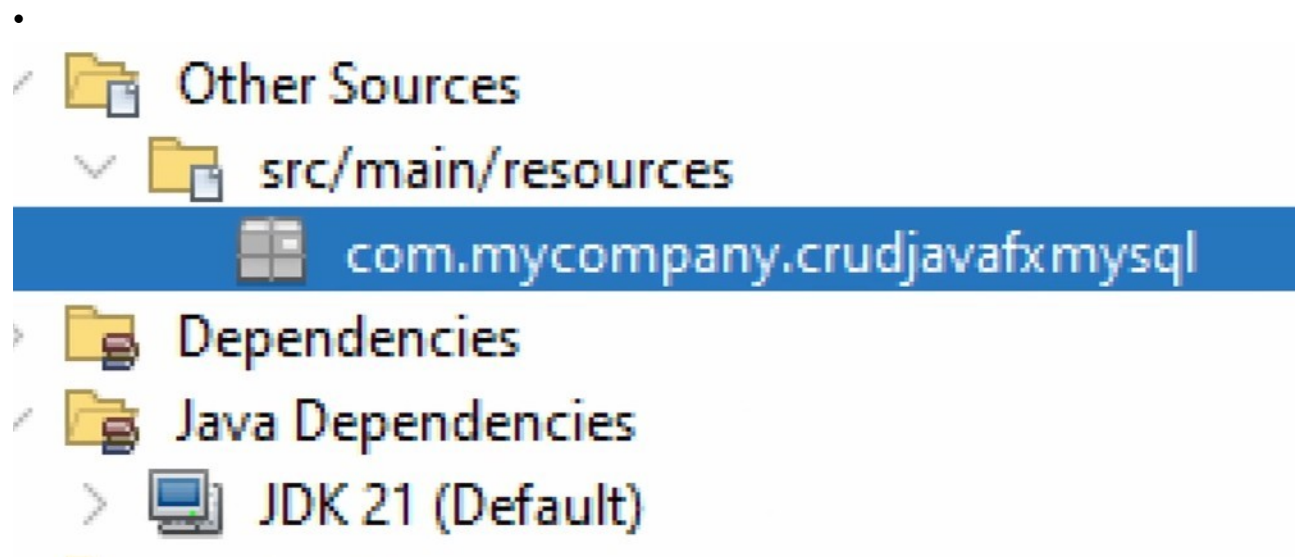
- Creamos un proyecto nuevo de la categoria Maven



- Seleccionamos un proyecto FXML JavaFX Maven Archetype



- Después de crearlo, seleccionamos la carpeta dentro del proyecto llamada src/main/resources y creamos un nuevo archivo del tipo Empty FMXL



- Seleccionamos la opción para que se creen de manera automática los controladores en el mismo proyecto

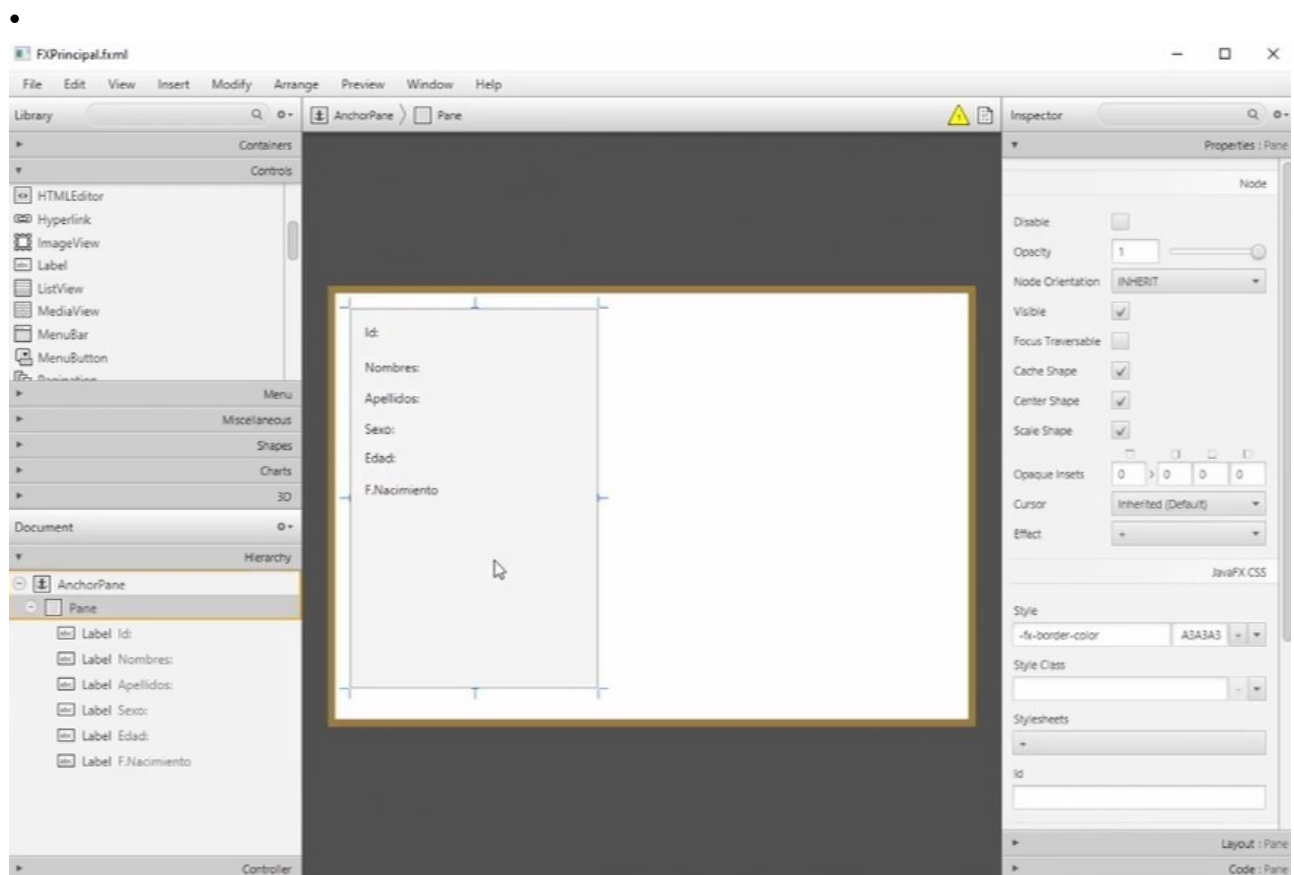
# FXML Controller Class Name and Location

☐ Use Java Controller

☒ Create New

Controller Name: FXPrincipal

- Con el archivo recién creado pondremos la interfaz utilizando las herramientas de Java FX en la forma en la que sea necesaria



- Vemos que para hacer conexión entre la base de datos y el SNK descargamos el siguiente archivo y lo guardamos dentro de las dependencias de nuestro proyecto

- 

Categories	JDBC Drivers
Tags	database sql jdbc driver connector rdbms mysql connection
Organization	Oracle Corporation
HomePage	<a href="http://dev.mysql.com/doc/connector-j/en/">http://dev.mysql.com/doc/connector-j/en/</a>
Date	Oct 25, 2023
Files	<a href="#">pom (3 KB)</a> <a href="#">jar (2.4 MB)</a> <a href="#">View All</a>
Repositories	Central
Ranking	#1102 in MvnRepository (See Top Artifacts) #9 in JDBC Drivers
Used By	438 artifacts

Maven

Gradle

Gradle (Short)

Gradle (Kotlin)

SBT

Ivy

Grape

Leiningen

Buildr

```
<!-- https://mvnrepository.com/artifact/com.mysql/mysql-connector-j -->
<dependency>
  <groupId>com.mysql</groupId>
  <artifactId>mysql-connector-j</artifactId>
  <version>8.2.0</version>
</dependency>
```

☒ Include comment with link to declaration

- Luego creamos una conexion con la base de datos usando el siguiente comando en una nueva clase:

```
public class FXPrincipalController implements Initializable {

    /**
     * Initializes the controller class.
     */
    @Override
    public void initialize(URL url, ResourceBundle rb) {
        Clases.CConexion objetoConexion = new Clases.CConexion();
        objetoConexion.estableceConexion();
    }
}
```

- Creamos los siguientes elementos con sus controladores:

```
<ComboBox id="cbSexo" fx:id="cbSexo" layoutX="124.0" layoutY="243.0" prefHeight="25.0" prefWidth="119.0" style="-fx-background-color: FAD59A; -fx-border-color: black;" />
<DatePicker fx:id="dateNacimiento" layoutX="123.0" layoutY="210.0" prefHeight="25.0" prefWidth="119.0" style="-fx-border-color: black;" />
<TextField fx:id="txtCorreo" layoutX="124.0" layoutY="176.0" prefHeight="25.0" prefWidth="119.0" style="-fx-border-color: black;" />
<TextField fx:id="txtApellidos" layoutX="123.0" layoutY="141.0" prefHeight="25.0" prefWidth="119.0" style="-fx-border-color: black;" />
<TextField fx:id="txtNombre" layoutX="123.0" layoutY="110.0" prefHeight="25.0" prefWidth="119.0" style="-fx-border-color: black;" />
<Button layoutX="98.0" layoutY="252.0" mnemonicParsing="false" onAction="#guardarUsuario" prefHeight="25.0" prefWidth="79.0" style="-fx-background-color: F08B51; text="GUARDAR" textFill="WHITE" />
<Button layoutX="145.0" layoutY="331.0" mnemonicParsing="false" onAction="#eliminarUsuario" prefHeight="25.0" prefWidth="79.0" style="-fx-background-color: F08B51; text="ELIMINAR" textFill="WHITE" />
<Button layoutX="41.0" layoutY="331.0" mnemonicParsing="false" onAction="#modificarUsuario" style="-fx-background-color: F08B51; text="MODIFICAR" textFill="WHITE" />
<Label layoutX="24.0" layoutY="79.0" text="ID:" />
```

```

        <font>
            <font name="Franklin Gothic Medium" size="12.0" />
        </font></Label>
        <TextField fx:id="Id" layoutX="123.0" layoutY="75.0" prefHeight="25.0" prefWidth="119.0" style="-fx-border-color: black;" />
    </children>
</Pane>
<Pane layoutX="295.0" layoutY="13.0" prefHeight="378.0" prefWidth="506.0">
    <children>
        <TableView fx:id="tbUsuarios" layoutX="14.0" layoutY="14.0" onMouseClicked="#seleccionarUsuario" prefHeight="355.0" prefWidth="486.0" style="-fx-background-color: DDFED2; -fx-border-color: B3A398;" />
    </children></Pane>
</children>
</children>

```

- Crearemos una clase Usuarios para crear todos los metodos que usaremos para las funciones CRUD teniendo el siguiente codigo:



```

7  import java.sql.CallableStatement;
8  import java.sql.Date;
9  import java.sql.ResultSet;
10 import java.sql.Statement;
11 import java.text.SimpleDateFormat;
12 import java.time.LocalDate;
13 import java.time.format.DateTimeFormatter;
14 import javafx.beans.property.SimpleStringProperty;
15 import javafx.scene.control.Alert;
16 import javafx.scene.control.ComboBox;
17 import javafx.scene.control.DatePicker;
18 import javafx.scene.control.TableColumn;
19 import javafx.scene.control.TableView;
20 import javafx.scene.control.TextField;
21 import javax.xml.transform.Result;
22 /**
23  *
24  * @author angel
25  */
26 public class Usuarios {
27
28     public void MostrarSexo(ComboBox<String> comboSexo){
29         Clases.Conexion objetoConexion = new Clases.Conexion();
30         comboSexo.getItems().clear();
31         comboSexo.setValue("Seleccione Sexo: ");
32
33         String sql = "select * from sexo;";
34
35         try {
36             Statement st = objetoConexion.establecerConexion().createStatement();
37
38             ResultSet rs = st.executeQuery(sql);
39
40             while (rs.next()) {
41                 int idSexo = rs.getInt("id");
42                 String nombreSexo = rs.getString("sexo");
43
44                 comboSexo.getItems().add(nombreSexo);
45                 comboSexo.getProperties().put(nombreSexo, idSexo);
46
47             }
48
49             } catch (Exception e) {
50                 mostrarAlerta("ERROR ", "Error al mostrar los sexos: "+e.toString());
51             }
52             finally{
53                 objetoConexion.cerrarConexion();
54             }
55
56         }
57

```

```

58 //Agregar los usuarios desde JavaFX y se agreguen al SQL
59 public void AgregarUsuarios(TextField nombres, TextField apellidos, TextField correo, DatePicker nacimiento, ComboBox<String> comboSexo){
60     Conexion objetoConexion = new Conexion();
61
62     String consulta = "INSERT INTO usuarios (nombre, apellidos, fkSexo, fechaNac, correo) VALUES (?, ?, ?, ?, ?)";
63     try (CallableStatement cs = objetoConexion.establecerConexion().prepareCall(consulta)){
64
65         cs.setString(1, nombres.getText());
66         cs.setString(2, apellidos.getText());
67
68         String nombreSexoSeleccionado = comboSexo.getSelectionModel().getSelectedItem();
69
70         int idSexo = (int) comboSexo.getProperties().get(nombreSexoSeleccionado);
71
72         cs.setString(5, correo.getText());
73
74         LocalDate fechaSeleccionada = nacimiento.getValue();
75         Date fechaSQL = Date.valueOf(fechaSeleccionada);
76         cs.setDate(4, fechaSQL);
77
78         cs.setInt(3, idSexo);
79
80         cs.execute();
81
82         mostrarAlerta("Informacion", "se guardo correctamente ");
83
84     } catch (Exception e) {
85         mostrarAlerta("Informacion", "Error al mostrar la informacion "+e.toString());
86     } finally {
87         objetoConexion.cerrarConexion();
88     }
89 }
90
91 //Codigo que hace que la base de datos se vea en el TableView
92 public void mostrarUsuarios(TableView<Object[]> tablaTotalUsuarios) {
93
94     Clases.Conexion objetoConexion = new Clases.Conexion();
95
96     TableColumn<Object[], String> idColumn = new TableColumn<>("Id");
97     TableColumn<Object[], String> nombresColumn = new TableColumn<>("Nombres");
98     TableColumn<Object[], String> apellidosColumn = new TableColumn<>("Apellidos");
99     TableColumn<Object[], String> correoColumn = new TableColumn<>("Correo");
100     TableColumn<Object[], String> fechaNacColumn = new TableColumn<>("FNacimiento");
101     TableColumn<Object[], String> sexoColumn = new TableColumn<>("Sexo");
102
103     idColumn.setCellValueFactory(cellData -> new SimpleStringProperty(cellData.getValue()[0].toString()));
104     nombresColumn.setCellValueFactory(cellData -> new SimpleStringProperty(cellData.getValue()[1].toString()));
105     apellidosColumn.setCellValueFactory(cellData -> new SimpleStringProperty(cellData.getValue()[2].toString()));
106     correoColumn.setCellValueFactory(cellData -> new SimpleStringProperty(cellData.getValue()[3].toString()));
107     fechaNacColumn.setCellValueFactory(cellData -> new SimpleStringProperty(cellData.getValue()[4].toString()));
108     sexoColumn.setCellValueFactory(cellData -> new SimpleStringProperty(cellData.getValue()[5].toString()));

```

```

109     correoColumn.setCellValueFactory(cellData -> new SimpleStringProperty(cellData.getValue()[3].toString()));
110     fechaNacColumn.setCellValueFactory(cellData -> new SimpleStringProperty(cellData.getValue()[4].toString()));
111     sexoColumn.setCellValueFactory(cellData -> new SimpleStringProperty(cellData.getValue()[5].toString()));
112
113     tablaTotalUsuarios.getColumns().addAll(idColumn, nombresColumn, apellidosColumn, correoColumn, fechaNacColumn, sexoColumn);
114
115     String sql = "SELECT usuarios.id, usuarios.nombre, usuarios.apellidos, sexo.sexo, usuarios.correo, usuarios.fechaNac from usuarios inner join sexo on usuarios.fkSexo = sexo.id";
116
117     try {
118         Statement st = objetoConexion.establecerConexion().createStatement();
119         ResultSet rs = st.executeQuery(sql);
120
121         //Se guarden los datos en la columna de fecha de nacimiento con un formato 00/00/0000
122         SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
123
124         while (rs.next()) {
125             java.sql.Date fechaSql = rs.getDate("fechaNac");
126             String nuevaFecha = (fechaSql != null) ? sdf.format(fechaSql) : null;
127
128             Object[] rowData = {
129                 rs.getString("id"),
130                 rs.getString("Nombre"),
131                 rs.getString("Apellidos"),
132                 rs.getString("Correo"),
133                 nuevaFecha,
134                 rs.getString("Sexo")
135             };
136
137             tablaTotalUsuarios.getItems().add(rowData);
138         }
139     } catch (Exception e) {
140         mostrarAlerta("Error", "Error al guardar"+e.toString());
141     } finally {
142         objetoConexion.cerrarConexion();
143     }
144 }
145
146 //Seleccionar los usuarios desde la tabla
147 public void seleccionarUsuario(TableView<Object[]> tablaTotalUsuarios, TextField id, TextField nombres, TextField apellidos, TextField correo, DatePicker nacimiento, ComboBox<String> comboSexo){
148
149     int fila = tablaTotalUsuarios.getSelectionModel().getSelectedIndex();
150     if (fila >= 0) {
151         Object[] filaSeleccionada = tablaTotalUsuarios.getItems().get(fila);
152
153         id.setText(filaSeleccionada[0].toString());
154         nombres.setText(filaSeleccionada[1].toString());
155         apellidos.setText(filaSeleccionada[2].toString());
156         correo.setText(filaSeleccionada[3].toString());
157         nacimiento.setValue(LocalDate.parse(filaSeleccionada[4].toString(), DateTimeFormatter.ofPattern("dd/MM/yyyy")));
158         comboSexo.setValue(filaSeleccionada[5].toString());

```



```

154         id.setText(filaSeleccionada[0].toString());
155         nombres.setText(filaSeleccionada[1].toString());
156         apellidos.setText(filaSeleccionada[2].toString());
157         correo.setText(filaSeleccionada[3].toString());
158
159         String fechaString = filaSeleccionada[4].toString();
160         DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd/MM/yyyy");
161         LocalDate fechaLocalDate = LocalDate.parse(fechaString, formatter);
162         nacimiento.setValue(fechaLocalDate);
163         comboSexo.getSelectionModel().select(filaSeleccionada[5].toString());
164     }
165 }
166
167 //Modificar los usuarios desde javaFX
168 public void modificarUsuarios(TextField id, TextField nombres, TextField apellidos, TextField correo, DatePicker nacimiento, ComboBox<String> comboSexo){
169     Conexion objetoConexion = new Conexion();
170     String consulta = "update usuarios set usuarios.nombre=?, usuarios.apellidos=?, usuarios.correo=?, usuarios.fkSexo=?, usuarios.fechaNac=? where usuarios.id=?";
171
172     try {
173
174         CallableStatement cs = objetoConexion.establishConexion().prepareCall(consulta);
175         cs.setString(1, nombres.getText());
176         cs.setString(2, apellidos.getText());
177         cs.setString(3, correo.getText());
178         String nombreSexoSeleccionado = comboSexo.getSelectionModel().getSelectedItem();
179
180         int idSexo = (int) comboSexo.getProperties().get(nombreSexoSeleccionado);
181         cs.setInt(4, idSexo);
182
183         LocalDate fechaSeleccionada = nacimiento.getValue();
184         Date fechaSQL = Date.valueOf(fechaSeleccionada);
185         cs.setDate(5, fechaSQL);
186
187         cs.setInt(6, Integer.parseInt(id.getText()));
188         cs.execute();
189
190         mostrarAlerta("Informacion ", "Se modifiko el registro");
191     } catch (Exception e) {
192         mostrarAlerta("Informacion ", "No se modifiko el registro"+e.toString());
193     } finally {
194         objetoConexion.cerrarConexion();
195     }
196 }
197
198 public void eliminarUsuario(TextField id){
199     Conexion objetoConexion = new Conexion();
200     String consulta = "DELETE FROM usuarios WHERE usuarios.id=?";

```

```

201 public void eliminarUsuario(TextField id){
202     Conexion objetoConexion = new Conexion();
203
204     String consulta = "DELETE FROM usuarios WHERE usuarios.id=?";
205
206     try {
207         CallableStatement cs = objetoConexion.establishConexion().prepareCall(consulta);
208         cs.setInt(1, Integer.parseInt(id.getText()));
209         cs.execute();
210
211         mostrarAlerta("Informacion ", "Se elimino el registro");
212     } catch (Exception e) {
213         mostrarAlerta("Informacion ", "No se elimino el registro"+e.toString());
214     } finally {
215         objetoConexion.cerrarConexion();
216     }
217 }
218
219 private void mostrarAlerta(String tittle, String content){
220     Alert alerta = new Alert(Alert.AlertType.INFORMATION);
221     alerta.setTitle(tittle);
222     alerta.setHeaderText(null);
223     alerta.setContentText(content);
224     alerta.showAndWait();
225 }
226
227 public void limpiarCampos (TextField id, TextField nombres, TextField apellidos, TextField correo, DatePicker nacimiento, ComboBox<String> comboSexo){
228     id.setText("");
229     nombres.setText("");
230     apellidos.setText("");
231     correo.setText("");
232     nacimiento.setValue(LocalDate.now());
233 }
234
235 }
236
237 }
238
239 }

```

## 4.- Final

- Finalmente tenemos como resultado el siguiente programa:



