



REPORTE DE LA

LIBRERÍA DE CRUDs

Programación Orientada
a Objetos

Integrantes:

Alejandro Arriaga Perales

Luis Tadeo Betancourt Rodríguez

Brian Martin Jaramillo Hernández

Edgar Osvaldo Maldonado Campos

Juana Guadalupe Villasana Reyna

Librerías de los CRUDs en el Programa

Librería de Menú

```
Main main = new Main();
public void VentanaUsuarios(ActionEvent actionEvent) {
    main.profesor();
    Stage stage = (Stage) this.Evaluar.getScene().getWindow();
    stage.close();
}

public void VentanaEvaluar(ActionEvent actionEvent) {
    main.evaluacion();
    Stage stage = (Stage) this.Evaluar.getScene().getWindow();
    stage.close();
}

public void ventanaCarreras(ActionEvent actionEvent) {
    main.carrera();
    Stage stage = (Stage) this.Evaluar.getScene().getWindow();
    stage.close();
}

public void ventanaAulas(ActionEvent actionEvent) {
    main.nuevaAula();
    Stage stage = (Stage) this.Evaluar.getScene().getWindow();
    stage.close();
}

public void ventanaMateria(ActionEvent actionEvent) {
    main.nuevaMateria();
    Stage stage = (Stage) this.Evaluar.getScene().getWindow();
    stage.close();
}

public void ventanaPrestamo(ActionEvent actionEvent) {
    main.nuevoPrestamo();
    Stage stage = (Stage) this.Evaluar.getScene().getWindow();
    stage.close();
}

public void ventanaEquipo(ActionEvent actionEvent) {
    main.nuevoEquipo();
    Stage stage = (Stage) this.Evaluar.getScene().getWindow();
    stage.close();
}

public void ventanaPlan(ActionEvent actionEvent) {
    main.nuevoPlan();
    Stage stage = (Stage) this.Evaluar.getScene().getWindow();
    stage.close();
}
}
```

Librería de Carrera

```
5  import javafx.fxml.Initializable;
6  import javafx.scene.control.Button;
7  import javafx.scene.control.TextField;
8  import javafx.stage.Stage;
9
10 import java.net.URL;
11 import java.util.ResourceBundle;
12
13 import static com.boros.main.Controller.ekisde;
14
15 public class Controller_Carrera implements Initializable {
16
17     public Button agregarCarrera;
18     public TextField nombre;
19     public Button guardar;
20     public TextField id;
21     Main main = new Main();
22
23     @Override
24     public void initialize(URL location, ResourceBundle resources) {
25
26     }
27
28     public void ventAgregarCarrera(ActionEvent actionEvent) {
29         main.nuevaCarrera();
30         Stage stage = (Stage) this.agregarCarrera.getScene().getWindow();
31         stage.close();
32     }
33
34
35     public void guardarCarrera(ActionEvent actionEvent) {
36         SQLite.insercion_carrera(id.getText(), nombre.getText(), ekisde);
37     }
38 }
39
40 public void carrera(){
41     try {
42         Stage op = FXMLLoader.load(getClass().getResource("/Carreras.fxml"));
43         op.show();
44     } catch (IOException e) {
45     }
46 }
47
48 public void nuevaCarrera(){
49     try {
50         Stage op = FXMLLoader.load(getClass().getResource("/Carreras.fxml"));
51         op.show();
52     } catch (IOException e) {
53     }
54 }
55 }
```

Librería de Profesor

```
public class Controller_NuevoProfesor implements Initializable {

    public TableView<Usuarios> tablaProfesores = new TableView<>();
    public TableColumn claveP;
    public TableColumn nombreP;
    public TableColumn tipoP;
    public TableColumn carreraP;
    public TableColumn contratoP;
    public Button ventAgregarN;

    @FXML
    Button guardar;
    @FXML
    ComboBox id_carrera;
    @FXML
    TextField clv, nombre, contrato, ads;

    Main main = new Main();

    @Override
    public void initialize(URL location, ResourceBundle resources) {
        try{
            FilteredList<String> filteredData = new FilteredList<>(FXCollections.observableList(SQLite.consultaCarrera()));
            id_carrera.setItems(filteredData);
        }catch (Exception e){

        }
        try{
            FilteredList<Usuarios> filteredData = new FilteredList<>(FXCollections.observableList(SQLite.consultaProfesorTodo()));
        }catch (Exception e){

        }
    }

    public void guardarD(){
        SQLite.insercion_usuarios(clv.getText(),Integer.parseInt(String.valueOf(id_carrera.getValue())),nombre.getText(), ads.getText(),contr
    }

    public void ventAgregarN(ActionEvent actionEvent) {
        main.nuevoProfesor();
        Stage stage = (Stage) this.guardar.getScene().getWindow();
        stage.close();
    }
}
```

Librería de Auto-Evaluación

```
public class Controller_AutoEvaluacion implements Initializable {

    @FXML
    ComboBox plan, materia, profesor, PConfianza, PDirector;
    @FXML
    Button agregar;
    CRUD crud = new CRUD();

    @Override
    public void initialize(URL location, ResourceBundle resources) {
        FilteredList<String> filteredData = new FilteredList<>(FXCollections.observableList(SQLite.consultaPlan()));
        plan.setItems(filteredData);
        FilteredList<String> filteredData2 = new FilteredList<>(FXCollections.observableList(SQLite.consultaProfesor()));
        profesor.setItems(filteredData2);
        ObservableList list = FXCollections.observableArrayList();
        list.add("1");
        list.add("2");
        list.add("3");
        list.add("4");
        list.add("5");
        PConfianza.setItems(list);
        PDirector.setItems(list);
    }

    public void seleccion(javaafx.event.ActionEvent actionEvent) {
        FilteredList<String> filteredData = new FilteredList<>(FXCollections.observableList(SQLite.consultaMateria((String) plan.getValue())));
        materia.setItems(filteredData);
    }

    public void agregarDatos(){
        SQLite.insercion_materia_usuarios(String.valueOf(materia.getValue()),String.valueOf(plan.getValue()),String.valueOf(profesor.getValue
    }
}
```

Librería de Main

```
public class Main extends Application {

    public static void main(String[] args) throws SQLException {
        Controller.initialize();
        launch(args);
    }

    @Override
    public void start(Stage stage) throws Exception {
        stage = FXMLLoader.load(getClass().getResource("/menu.fxml"));
        stage.show();
    }

    public void evaluacion(){
        try {
            Stage op = FXMLLoader.load(getClass().getResource("/Evaluar.fxml"));
            op.show();
        } catch (IOException e) {
        }
    }

    public void profesor(){
        try {
            Stage op = FXMLLoader.load(getClass().getResource("/Nuevo_Profesor.fxml"));
            op.show();
        } catch (IOException e) {
        }
    }

    public void nuevoProfesor(){
        try {
            Stage op = FXMLLoader.load(getClass().getResource("/Formulario_Nuevo_Profesor.fxml"));
            op.show();
        } catch (IOException e) {
        }
    }
}
```

Librería de Controlador

```
import com.boros.rutas.RunRutas;
import com.boros.sqlite.SQLite;
import com.boros.txt.RunTXT;
import com.boros.xlsx.RunXLSX;
import javafx.fxml.Initializable;

import java.net.URL;
import java.sql.SQLException;
import java.util.ResourceBundle;

public class Controller{

    public static java.sql.Connection ekisde = SQLite.creacion_volatil();
    public static String rutaTXT;
    public static String rutaXLSX;

    public static void initialize() {
        RunRutas.cambioRuta();
        System.out.println(rutaTXT);
        System.out.println(rutaXLSX);
        try {
            RunTXT.lecturaTXT(rutaTXT);
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        }
        try {
            RunXLSX.lecturaXLSX(rutaXLSX);
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        }
    }
}
```


Librería de Controlador Aula

```
public class Controller_Aula{
    public TextField id;
    public TextField nombre;
    public TextField capacidad;
    public TextField descripcion;
    public TextField ubicacion;
    public TextField tipo;
    public Button guardar;
    public Button Evaluar;
    Main main = new Main();

    public void guardarAula(ActionEvent actionEvent) {

        SQLite.insercion_aulas(id.getText(),nombre.getText(),tipo.getText(),Integer.parseInt(capacidad.getText()),descripcion.getText(),ubicacion.getText(), ekisde);
    }

    public void ventanaProfesor(ActionEvent actionEvent) {
        main.nuevoProfesor();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }
    public void ventanaInicio(ActionEvent actionEvent) {
        main.nuevoInicio();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }
    public void ventanaMateria(ActionEvent actionEvent) {
        main.nuevaMateria();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }
    public void ventanaAula(ActionEvent actionEvent) {
        main.nuevaAula();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }
    public void ventanaPlan(ActionEvent actionEvent) {
        main.nuevoPlan();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }
    public void ventanaEvaluar(ActionEvent actionEvent) {
        main.evaluacion();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }
}
```

Librería de Categoría

```
public class Controller_Categoria implements Initializable {

    public Button Evaluar;
    public TextField id;
    public TextField nombre;
    public TextField descripcion;

    public TextArea descripcionZ;
    public ComboBox categoriaZ;
    public TextField nombreZ;
    public TextField idZ;

    Main main = new Main();
    public void guardarCategoria(ActionEvent actionEvent) {
        SQLite.insercion_categorias_equipo(Integer.parseInt(id.getText()), nombre.getText(), descripcion.getText(), ekisde);
    }

    public void guardarEquipo(ActionEvent actionEvent){
        SQLite.insercion_equipo(Integer.parseInt(idZ.getText()), Integer.parseInt(String.valueOf(categoriaZ.getValue())), nombreZ.getText(), descripcionZ.getText(), ekisde);
    }

    public void ventanaCategoria(ActionEvent actionEvent) {
        main.nuevaCategoria();
        Stage stage = (Stage) this.idZ.getScene().getWindow();
        stage.close();
    }

    @Override
    public void initialize(URL location, ResourceBundle resources) {
        try {
            FilteredList<String> filteredData = new FilteredList<>(FXCollections.observableList(SQLite.consultaCategoria()));
            categoriaZ.setItems(filteredData);
        } catch (Exception e) {
        }
    }

    public void regresarEquipo(ActionEvent actionEvent) {
        main.nuevoEquipo();
        Stage stage = (Stage) this.descripcion.getScene().getWindow();
        stage.close();
    }

    public void ventanaProfesor(ActionEvent actionEvent) {
        main.nuevoProfesor();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }

    public void ventanaInicio(ActionEvent actionEvent) {
        main.nuevoInicio();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }

    public void ventanaMateria(ActionEvent actionEvent) {
```

Librería de Materia

```
public class Controller_Materia implements Initializable {
    public Button guardar;
    public ComboBox plan;
    public TextField credito;
    public TextField nombre;
    public TextField clv;
    public ComboBox cuatrimestre;
    public TextField posicion;
    public TextField tipo;
    public TextField horas;
    public Button Evaluar;
    Main main = new Main();
    public void guardarMateria(ActionEvent actionEvent) {
        SQLite.Insercion_materia(nombre.getText(), clv.getText(), Integer.parseInt(credito.getText()), Integer.valueOf(String.valueOf(cuatrimestre.getValue())), Integer.valueOf(posicion.getText()), String.valueOf(tipo.getText()), Integer.parseInt(horas.getText()));
    }

    @Override
    public void initialize(URL location, ResourceBundle resources) {
        FilteredList<String> filteredData = new FilteredList<>(FXCollections.observableList(SQLite.consultaPlan()));
        plan.setItems(filteredData);
        ObservableList list = FXCollections.observableArrayList();
        list.add("1");
        list.add("2");
        list.add("3");
        list.add("4");
        list.add("5");
        list.add("6");
        list.add("7");
        list.add("8");
        list.add("9");
        list.add("10");
        cuatrimestre.setItems(list);
    }

    public void ventanaProfesor(ActionEvent actionEvent) {
        main.nuevoProfesor();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }

    public void ventanaInicio(ActionEvent actionEvent) {
        main.nuevoInicio();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }

    public void ventanaMateria(ActionEvent actionEvent) {
        main.nuevaMateria();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }

    public void ventanaAula(ActionEvent actionEvent) {
        main.nuevaAula();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }
}
```


Librería de Nuevo Profesor

```
public class Controller_NuevoProfesor implements Initializable {
    public Button Evaluar;
    @FXML
    Button guardar, AgregarN;
    @FXML
    ComboBox id_carrera;
    @FXML
    TextField clv, nombre, contrato, ads;
    public TableView<Usuarios> tabla = new TableView<>();

    Main main = new Main();
    @Override
    public void initialize(URL location, ResourceBundle resources) {
        try {
            FilteredList<String> filteredData = new FilteredList<>(FXCollections.observableList(SQLite.consultaCarrera()));
            id_carrera.setItems(filteredData);
        } catch (Exception e) {}

        try {
            FilteredList<Usuarios> filteredData = new FilteredList<>(FXCollections.observableList(SQLite.consultaProfesorTodo()));
            tabla.getColumns().add(column("Clave Usuario", Usuarios::getClv_usuario));
            tabla.getColumns().add(column("Id Carrera", Usuarios::getId_carrera));
            tabla.getColumns().add(column("Nombre Usuario", Usuarios::getNombre_usuario));
            tabla.getColumns().add(column("Nivel ADS", Usuarios::getNivel_ads));
            tabla.getColumns().add(column("Contrato", Usuarios::getContrato));
            tabla.setItems(filteredData);
        } catch (Exception e) {}
    }

    private static <S,T> TableColumn<S,T> column(String title, Function<S,T> property) {
        TableColumn<S,T> col = new TableColumn<>();
        col.setCellValueFactory(new Callback<CellData<T>, ObservableList<T>>() {
            @Override
            public ObservableList<T> call(CellData<T> cellData) {
                return property.apply(cellData.getValue());
            }
        });
        return col;
    }

    public void guardarD(){
        SQLite.insercion_usuarios(clv.getText(), Integer.parseInt(String.valueOf(id_carrera.getValue())), nombre.getText(), ads.getText(), contrato.getText(), ekisde);
    }

    public void ventAgregarN() {
        main.nuevoProfesor();
        Stage stage = (Stage) this.AgregarN.getScene().getWindow();
        stage.close();
    }

    public void ventanaProfesor(ActionEvent actionEvent) {
        main.nuevoProfesor();
    }
}
```

References:
[Controller_NuevoProfesor.java:41](#)
[Controller_NuevoProfesor.java:42](#)
[Controller_NuevoProfesor.java:43](#)
[Controller_NuevoProfesor.java:44](#)
[Controller_NuevoProfesor.java:45](#)

Librería de Plan de Estudios

```
public class Controller_Plan implements Initializable {
    public Button Evaluar;
    public TextField clv;
    public ComboBox carrera;
    public ComboBox nivel;
    public TextField nombre;
    Main main = new Main();

    public void guardarPlan(ActionEvent actionEvent) {
        SQLite.insercion_plan_estudios(clv.getText(), nombre.getText(), String.valueOf(nivel.getValue()), Integer.parseInt(String.valueOf(carrera.getValue())), ekisde);
    }

    @Override
    public void initialize(URL location, ResourceBundle resources) {
        try{
            FilteredList<String> filteredData = new FilteredList<>(FXCollections.observableList(SQLite.consultaCarrera()));
            carrera.setItems(filteredData);
            ObservableList list = FXCollections.observableArrayList();
            list.add("LIC");
            list.add("ING");
            list.add("MST");
            nivel.setItems(list);
        } catch (Exception e){
        }
    }

    public void ventanaProfesor(ActionEvent actionEvent) {
        main.nuevoProfesor();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }

    public void ventanaInicio(ActionEvent actionEvent) {
        main.nuevoInicio();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }

    public void ventanaMaterias(ActionEvent actionEvent) {
        main.nuevaMateria();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }

    public void ventanaAula(ActionEvent actionEvent) {
        main.nuevaAula();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }

    public void ventanaPlan(ActionEvent actionEvent) {
        main.nuevoPlan();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }
}
```

Librería de Prestamos

```
public class Controller_Prestamos implements Initializable {

    public Button Evaluar;
    public ComboBox usuario;
    public ComboBox carrera;
    Main main = new Main();
    @Override
    public void initialize(URL location, ResourceBundle resources) {
        FilteredList<String> filteredData = new FilteredList<>(FXCollections.observableList(SQLite.consultaProfesor()));
        usuario.setItems(filteredData);
        FilteredList<String> filteredData2 = new FilteredList<>(FXCollections.observableList(SQLite.consultaCarrera()));
        carrera.setItems(filteredData2);
    }

    public void guardarPrestamo(ActionEvent actionEvent) {
        SQLite.insercion_prestamos(String.valueOf(usuario.getValue()), Integer.parseInt(String.valueOf(carrera.getValue())), ekisde);
    }

    public void ventanaProfesor(ActionEvent actionEvent) {
        main.nuevoProfesor();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }

    public void ventanaInicio(ActionEvent actionEvent) {
        main.nuevoInicio();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }

    public void ventanaMateria(ActionEvent actionEvent) {
        main.nuevaMateria();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }

    public void ventanaAula(ActionEvent actionEvent) {
        main.nuevaAula();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }


    public void ventanaPlan(ActionEvent actionEvent) {
        main.nuevoPlan();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }

    public void ventanaEvaluar(ActionEvent actionEvent) {
        main.evaluacion();
        Stage stage = (Stage) this.Evaluar.getScene().getWindow();
        stage.close();
    }

}
```

Pruebas de Escritorio

Prueba 1: Agregar Profesor



UNIVERSIDAD POLITÉCNICA

[Inicio](#)
[Profesores](#)
[Materias](#)
[Aulas](#)
[Disponibilidad](#)
[Planes](#)
[Evaluar Profesor](#)

[Regresar](#)Cerrar s...

Nuevo Profesor

ID:

Ads:

Nom...

Contrato:

Guardar



UNIVERSIDAD POLITÉCNICA

[Inicio](#)
[Profesores](#)
[Materias](#)
[Aulas](#)
[Disponibilidad](#)
[Planes](#)
[Evaluar Profesor](#)

[Regresar](#)Cerrar s...


[Agregar Nuevo Prof...](#)

Profeso...

[Modificar](#)

Clave Usuario	Id Carrera	Nombre Usuario	Nivel ADS	Contrato
1830124	6	Edgar	ING	PTC
1830300	6	Edgar	ING	PTC
14	1	DhariusBodarius	hola	no

Prueba 2: Agregar Carrera



UNIVERSIDAD POLITÉCNICA

[Inicio](#)
[Profesores](#)
[Materias](#)
[Aulas](#)
[Disponibilidad](#)
[Planes](#)
[Evaluar Profesor](#)

[Regresar](#)Cerrar s...


Panel de administración

Nueva Carrera

ID: 5

Nom... Pymes

Guardar



UNIVERSIDAD POLITÉCNICA

[Inicio](#)
[Profesores](#)
[Materias](#)
[Aulas](#)
[Disponibilidad](#)
[Planes](#)
[Evaluar Profesor](#)

[Regresar](#)Cerrar s...

Panel de administración


Agregar Nueva Carr...

Carreras

Modificar

Clave Usuario	Id Carrera
1	ITI
5	Pymes

Prueba 3: Agregar Profesor



UNIVERSIDAD POLITÉCNICA

[Inicio](#)
[Profesores](#)
[Materias](#)
[Aulas](#)
[Disponibilidad](#)
[Planes](#)
[Evaluar Profesor](#)

[Regresar](#)Cerrar S...

Panel de administración


Nuevo Profesor

ID:

Ads:

Nom... Contrato:

Guardar



UNIVERSIDAD POLITÉCNICA

[Inicio](#)
[Profesores](#)
[Materias](#)
[Aulas](#)
[Disponibilidad](#)
[Planes](#)
[Evaluar Profesor](#)

[Cerrar S...](#)

Panel de administración

[Agregar Nuevo Prof...](#)

Profeso...

[Modificar](#)

Clave Usuario	Id Carrera	Nombre Usuario	Nivel ADS	Contrato
1830124	6	Edgar	ING	PTC
1830300	6	Edgar	ING	PTC
14	1	DhariusBodarius	hola	no
3	9	Millonario	Milli	no