

Manual de usuario

Sistema de Gestión de  
Reservas

<b>Manual de Usuario para la Aplicación de Gestión de Mobiliario.....</b>	<b>3</b>
1. Descripción General.....	3
2. Estructura de la Aplicación.....	3
2.1 JavaApplication1.....	4
2.1.1 Descripción.....	4
2.1.2 Componentes Clave.....	4
2.1.3 Uso.....	6
2.2 FXMLController.....	6
2.2.1 Descripción.....	6
2.2.2 Componentes Clave.....	6
2.2.3 Uso.....	7
2.3 tablaController.....	8
2.3.1 Descripción.....	8
2.3.2 Componentes Clave.....	9
2.3.3 Uso.....	9
3. Conexión a la Base de Datos.....	10
4. Contacto y Soporte.....	11

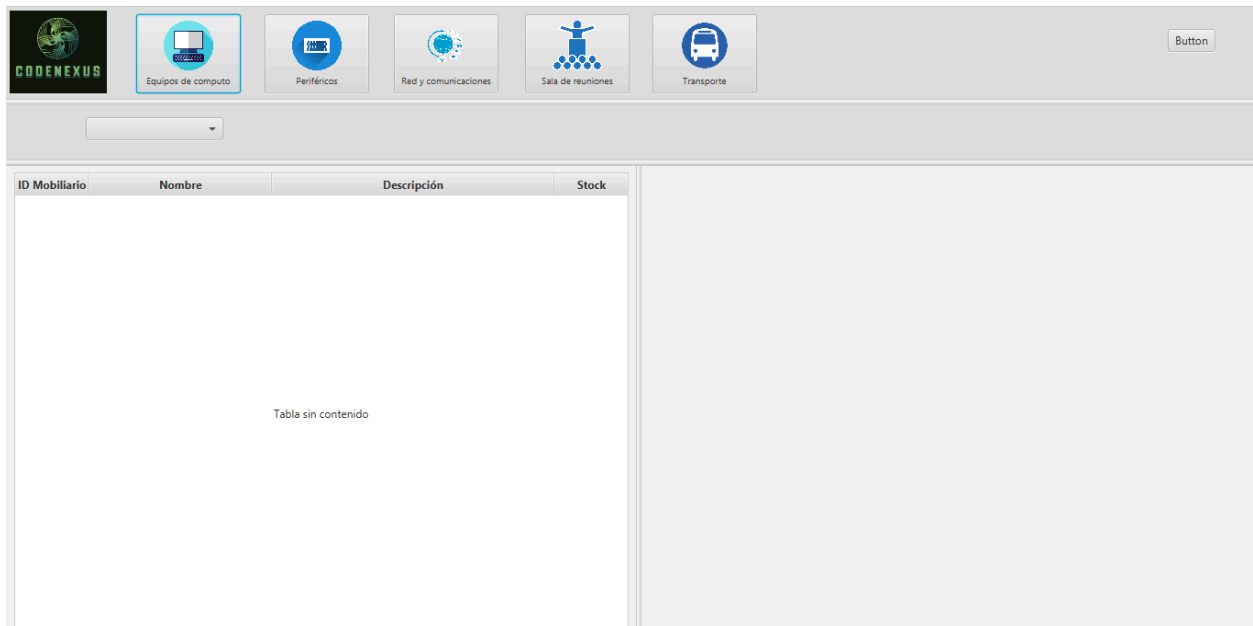
# Manual de Usuario para la Aplicación de Gestión de Mobiliario

## 1. Descripción General

Este manual está diseñado para ayudar a los usuarios a entender y utilizar la aplicación de gestión de mobiliario desarrollada en JavaFX. La aplicación permite conectarse a una base de datos MySQL, gestionar diferentes paneles de opciones y visualizar información de mobiliario a través de una tabla interactiva.

## 2. Estructura de la Aplicación

La aplicación se compone de tres clases principales:



JavaApplication1: Clase principal que inicializa la aplicación y gestiona la conexión a la base de datos.

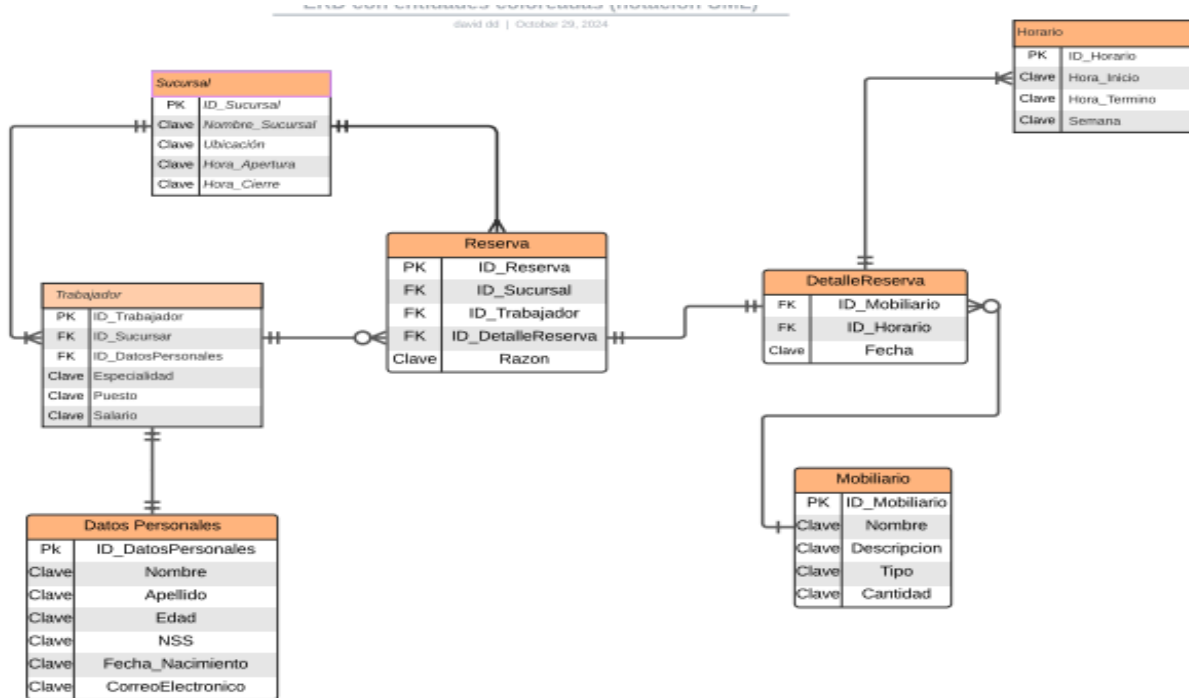
FXMLController: Controlador que gestiona la navegación entre diferentes paneles de opciones.

tablaController: Controlador que maneja la visualización y gestión de datos en una tabla de mobiliario.

## 2.1 JavaApplication1

### 2.1.1 Descripción

La clase JavaApplication1 es el punto de entrada de la aplicación. Se encarga de iniciar la interfaz gráfica, establecer la conexión con la base de datos y gestionar los ciclos de vida de la aplicación.



### 2.1.2 Componentes Clave

Método start(Stage stage): Descripción: Se ejecuta al iniciar la aplicación. Este método establece la conexión a la base de datos y carga el archivo FXML principal.

La imagen muestra una interfaz de desarrollo con la siguiente información:

**Navigation:** bd\_ventas, mydatabase, Tables (datospersonales, detalle\_reserva, horario, mobiliario, reserva, sucursal, trabajador), Views, Stored Procedures, Administration, Schemas, Information.

**Table: datospersonales**

Columns:

- ID\_DatosPersonales (int PK)
- Nombre (varchar(45))
- Apellido (varchar(45))
- ApellidoM (varchar(45))
- Fecha\_Nacimiento (varchar(45))
- Cellar1 (varchar(45))
- Cellar2 (varchar(45))
- Correo (varchar(45))

**SQL Query:**

```
1 SELECT * FROM mydatabase.mobiliario;
```

**Result Grid:**

id_Mobiliario	Nombre	Descripción	Tipo	Subtipo	Cantidad
3000	Computadora HP	Procesador i5, 8GB de RAM, 1TB HDD	Equipos de computo	Computadora de Escritorio	15
3001	Computadora Dell	Procesador i7, 16GB de RAM, 512GB SSD	Equipos de computo	Computadora de Escritorio	10
3002	iMac 24"	Pantalla Retina 4.5K, M1 chip, 16GB RAM	Equipos de computo	Computadora de Escritorio	5
3003	Lenovo ThinkCentre	Procesador i5, compacto, ideal para oficinas	Equipos de computo	Computadora de Escritorio	8
3004	ASUS VivoPC	8GB RAM, 1TB HDD, Procesador i3	Equipos de computo	Computadora de Escritorio	12
3005	Acer Aspire TC	16GB RAM, 512GB SSD, i5, Gráficos dedicados	Equipos de computo	Computadora de Escritorio	7
3006	HP ProDesk 600 G6	i7, 16GB RAM, 1TB HDD, Seguridad avanzada	Equipos de computo	Computadora de Escritorio	9
3007	MSI Trident 3	GTX 1660, i5, 16GB RAM, Compacta para juegos	Equipos de computo	Computadora de Escritorio	4
3008	Samsung All-in-One	i5, 8GB RAM, Pantalla táctil, Diseño elegante	Equipos de computo	Computadora de Escritorio	6
3009	Gigabyte Brix Gaming	i7, GTX 1650, Compacto para juegos, 16GB RAM	Equipos de computo	Computadora de Escritorio	3
3010	Lenovo ThinkCentre	i5, 8GB RAM, 1TB HDD, Compacta y eficiente	Equipos de computo	Computadora de Escritorio	8
3011	iMac 24"	M1 chip, 16GB RAM, Pantalla Retina 4.5K	Equipos de computo	Computadora de Escritorio	5
3012	Dell OptiPlex 7080	i7, 16GB RAM, SSD 512GB, Seguridad avanzada.	Equipos de computo	Computadora de Escritorio	10
3013	HP Pavilion 15	i5, 8GB RAM, 512GB SSD, Ligera y rápida	Equipos de computo	Computadora portátil	20
3014	Dell XPS 13	i7, 16GB RAM, Pantalla InfinityEdge	Equipos de computo	Computadora portátil	15
3015	Lenovo ThinkPad X1	i7, 16GB RAM, 1TB SSD, Teclado retroiluminado	Equipos de computo	Computadora portátil	10
3016	Acer Aspire 5	i5, 8GB RAM, 256GB SSD, Batería duradera	Equipos de computo	Computadora portátil	25
3017	ASUS ZenBook 14	i7, 16GB RAM, Pantalla OLED, Diseño delgado	Equipos de computo	Computadora portátil	12
3018	Microsoft Surface Lapto...	i5, 8GB RAM, Pantalla táctil	Equipos de computo	Computadora portátil	8

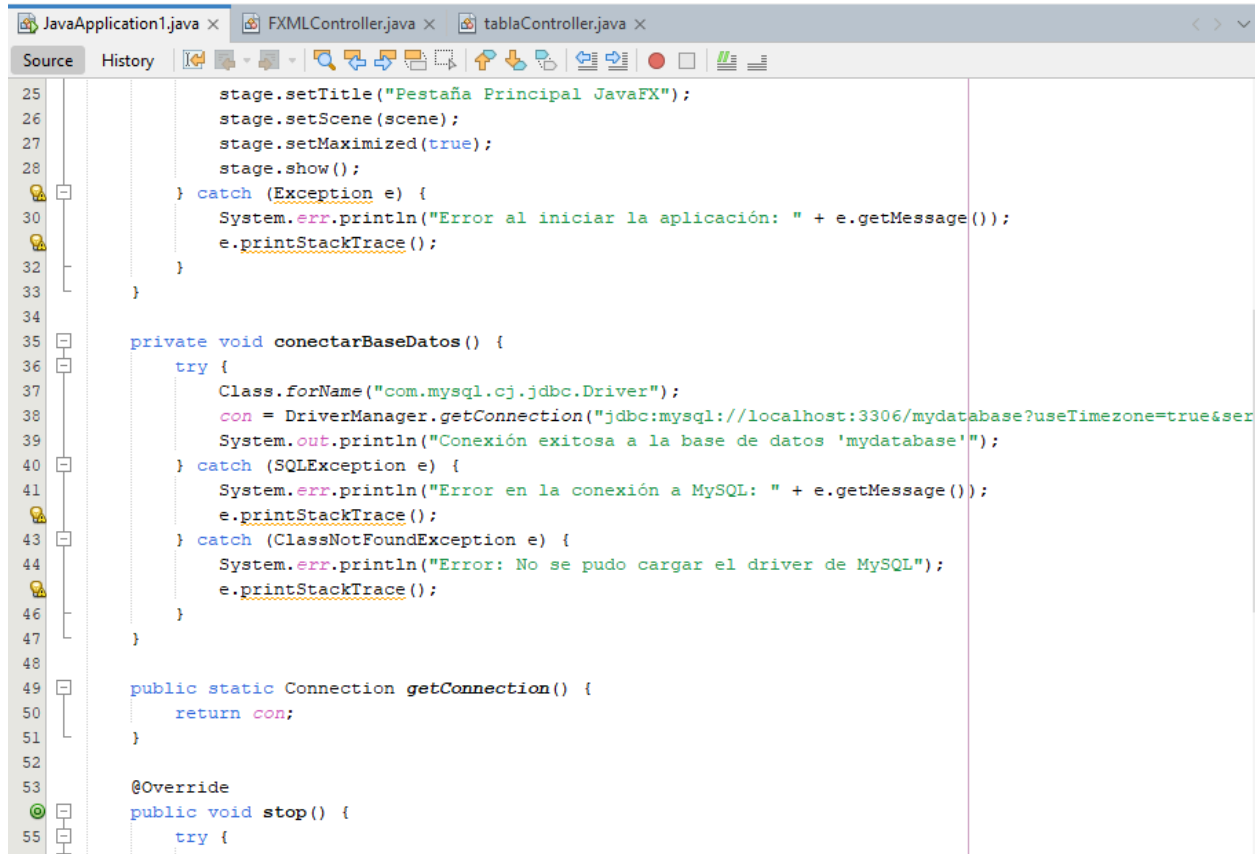
Flujo de trabajo:

Llama al método conectarBaseDatos() para establecer la conexión con la base de datos.

Carga el archivo FXML.fxml utilizando FXMLLoader.

Configura la escena y la muestra en el escenario.

Método conectarBaseDatos():

The image shows a screenshot of an IDE with three tabs: JavaApplication1.java, FXMLController.java, and tablaController.java. The FXMLController.java tab is active, showing the following code:

```
25 stage.setTitle("Pestaña Principal JavaFX");
26 stage.setScene(scene);
27 stage.setMaximized(true);
28 stage.show();
29 } catch (Exception e) {
30     System.err.println("Error al iniciar la aplicación: " + e.getMessage());
31     e.printStackTrace();
32 }
33 }
34
35 private void conectarBaseDatos() {
36     try {
37         Class.forName("com.mysql.cj.jdbc.Driver");
38         con = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydatabase?useTimezone=true&ser
39         System.out.println("Conexión exitosa a la base de datos 'mydatabase'");
40     } catch (SQLException e) {
41         System.err.println("Error en la conexión a MySQL: " + e.getMessage());
42         e.printStackTrace();
43     } catch (ClassNotFoundException e) {
44         System.err.println("Error: No se pudo cargar el driver de MySQL");
45         e.printStackTrace();
46     }
47 }
48
49 public static Connection getConnection() {
50     return con;
51 }
52
53 @Override
54 public void stop() {
55     try {
```

Descripción: Establece una conexión con la base de datos MySQL. Utiliza el controlador JDBC para realizar la conexión.

Flujo de trabajo:

Intenta cargar el controlador JDBC de MySQL.

Establece la conexión utilizando la URL de la base de datos, el nombre de usuario y la contraseña.

Imprime un mensaje de éxito o error en la consola.

Método stop():

Descripción: Se ejecuta al cerrar la aplicación. Cierra la conexión a la base de datos para liberar recursos.

Flujo de trabajo:

Verifica si la conexión no es nula y está abierta.

Cierra la conexión e imprime un mensaje de cierre en la consola.

### 2.1.3 Uso

Inicio de la aplicación:

Cuando inicias la aplicación, se ejecuta el método start().

La aplicación intenta conectarse a la base de datos mydatabase.

Si la conexión es exitosa, se mostrará la interfaz principal.

Mensajes de consola:

---

```
ant -f C:\\Users\\parca\\OneDrive\\Documentos\\git\\JavaApplication1 -Dnb.internal.action.name=run run
init:
Deleting: C:\\Users\\parca\\OneDrive\\Documentos\\git\\JavaApplication1\\build\\built-jar.properties
deps-jar:
Updating property file: C:\\Users\\parca\\OneDrive\\Documentos\\git\\JavaApplication1\\build\\built-jar.properties
compile:
run:
Conexión exitosa a la base de datos 'mydatabase'
|
```

Los mensajes de éxito o error se mostrarán en la consola para ayudarte a diagnosticar problemas.

## 2.2 FXMLController

### 2.2.1 Descripción

FXMLController es responsable de gestionar la interacción del usuario y la navegación entre diferentes paneles de la aplicación. Carga diferentes vistas FXML según las opciones seleccionadas por el usuario.

### 2.2.2 Componentes Clave

Método initialize(URL url, ResourceBundle rb):

Descripción: Este método se llama automáticamente al inicializar el controlador. Aquí se configura el panel inicial.

Flujo de trabajo:

Llama al método loadTablaPane() para cargar inicialmente la tabla de mobiliario.

Métodos para cargar diferentes paneles:

Métodos: btn\_Equipos(), btn\_Perifericos(), btn\_Comunicaciones(), btn\_Sala\_Reunion(), btn\_Transporte().

Descripción: Cada uno de estos métodos se ejecuta al hacer clic en los botones correspondientes en la interfaz.

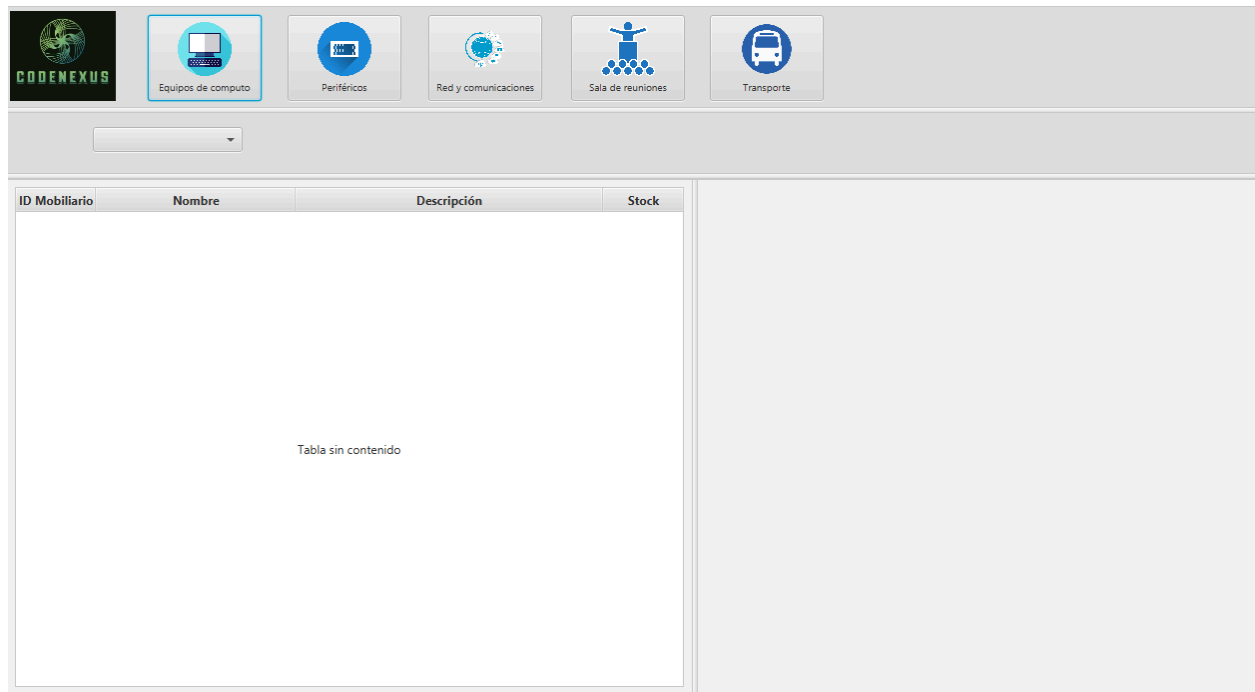
Flujo de trabajo:

Carga el archivo FXML correspondiente al panel seleccionado utilizando FXMLLoader.

Obtiene el controlador del archivo FXML cargado.

Si el controlador tiene un método setTablaPane(), se le pasa el panel de la tabla.

Limpia el contenido actual del panel 'opciones' y añade el nuevo panel.  
Método loadCalendarioPane():



Descripción: Carga el panel de calendario en la interfaz.

Flujo de trabajo:

Carga el archivo FXML MainView.fxml y lo añade al panel calendario.

Método loadTablaPane():

Descripción: Carga la tabla inicial de mobiliario en la interfaz.

Flujo de trabajo:

Carga el archivo FXML table.fxml y lo añade al panel tabla.


### 2.2.3 Uso






Navegación entre paneles:

Al hacer clic en los botones correspondientes en la interfaz, se cargará el panel adecuado en el área de opciones.

La tabla se cargará automáticamente al iniciar la aplicación.

Visualización de datos:



 Equipos de computo
  Periféricos
  Red y comunicaciones
  Sala de reuniones
  Transporte

Computadora de Escritorio

Computadora portátil

Equipos Telefónicos

Servidores

Dispositivos de almacenamiento

ID Mobiliario	Nombre	Descripción	Stock
Tabla sin contenido			

id:   
FECHA   
Hora de inicio:   
Hora de termino:   

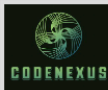
Confirmar






Los datos mostrados en los diferentes paneles pueden variar según la selección realizada.

## 2.3 tablaController

### 2.3.1 Descripción

tablaController gestiona la visualización de datos en una tabla, específicamente información sobre mobiliario. Se conecta a la base de datos para recuperar y mostrar datos en la TableView.



 Equipos de computo
  Periféricos
  Red y comunicaciones
  Sala de reuniones
  Transporte

Computadora de Escritorio

Computadora portátil

Equipos Telefónicos

Servidores

Dispositivos de almacenamiento

ID Mobiliario	Nombre	Descripción	Stock
3000	Computadora HP	Procesador i5, 8GB de RAM, 1TB HDD	15
3001	Computadora Dell	Procesador i7, 16GB de RAM, 512GB SSD	10
3002	iMac 24"	Pantalla Retina 4.5K, M1 chip, 16GB RAM	5
3003	Lenovo ThinkCentre	Procesador i5, compacto, ideal para oficinas	8
3004	ASUS VivoPC	8GB RAM, 1TB HDD, Procesador i3	12
3005	Acer Aspire TC	16GB RAM, 512GB SSD, i5, Gráficos dedicados	7
3006	HP ProDesk 600 G6	i7, 16GB RAM, 1TB HDD, Seguridad avanzada	9
3007	MSI Trident 3	GTX 1660, i5, 16GB RAM, Compacta para juegos	4
3008	Samsung All-in-One	i5, 8GB RAM, Pantalla táctil, Diseño elegante	6
3009	Gigabyte Brix Gaming	i7, GTX 1650, Compacto para juegos, 16GB RAM	3
3010	Lenovo ThinkCentre	i5, 8GB RAM, 1TB HDD, Compacta y eficiente	8
3011	iMac 24"	M1 chip, 16GB RAM, Pantalla Retina 4.5K	5
3012	Dell OptiPlex 7080	i7, 16GB RAM, SSD 512GB, Seguridad avanzada.	10

id:   
FECHA   
Hora de inicio:   
Hora de termino:   

Confirmar



### 2.3.2 Componentes Clave

Método initialize():

Descripción: Configura las columnas de la tabla para que muestren los atributos del mobiliario.

Flujo de trabajo:

Asigna cada columna de la tabla a la propiedad correspondiente del objeto tablaContenido.

Método realizarConsulta():

Descripción: Ejecuta una consulta SQL para recuperar datos del mobiliario filtrados por un subtipo específico.

Flujo de trabajo:

Obtiene la conexión desde JavaApplication1.

Limpia la lista de mobiliario para evitar duplicados.

Ejecuta la consulta SQL y agrega los resultados a listaMobiliario.

Asigna la lista a la TableView.

Método setFileName(String fileName):

Descripción: Establece el subtipo de mobiliario que se desea consultar y llama a realizarConsulta() para cargar los datos.

Flujo de trabajo:

Establece el nombre del archivo y ejecuta la consulta.

### 2.3.3 Uso

Visualización de la tabla:

Al cargar la aplicación, la tabla mostrará datos sobre el mobiliario almacenado en la base de datos.

Filtrado de datos:

```

JavaApplication1.java x FXMLController.java x tablaController.java x
Source History
42 private void realizarConsulta() {
43     Connection con = JavaApplication1.getConnection(); // Obtener conexión desde JavaApplication1
44     listaMobiliario.clear(); // Limpiar la lista para evitar datos duplicados
45     try {
46         if (con != null) {
47             Statement st = con.createStatement();
48             System.out.println("Conectado a la base de datos 'mydatabase'");
49
50             ResultSet rs = st.executeQuery("SELECT * FROM mobiliario WHERE Subtipo = '" + fileName + "'");
51
52             while (rs.next()) {
53                 int idMobiliario = rs.getInt("id_Mobiliario");
54                 String nombre = rs.getString("Nombre");
55                 String descripcion = rs.getString("Descripcion");
56                 String cantidad = rs.getString("Cantidad");
57
58                 listaMobiliario.add(new tablaContenido(idMobiliario, nombre, descripcion, cantidad));
59             }
60         }
61     } catch (SQLException e) {
62         e.printStackTrace();
63     }
64 }

```

Cuando se selecciona un subtipo de mobiliario en un panel, se llama a setFileName() para cargar solo los datos relevantes.

### 3. Conexión a la Base de Datos

La aplicación utiliza una conexión a una base de datos MySQL. Asegúrate de que:

Base de datos: La base de datos mydatabase esté creada y funcionando.

Tabla de Mobiliario: La tabla mobiliario contenga los campos necesarios:

id\_Mobiliario: Identificador único del mobiliario (tipo INTEGER).

Nombre: Nombre del mobiliario (tipo VARCHAR).

Descripcion: Descripción del mobiliario (tipo VARCHAR).

Cantidad: Cantidad disponible del mobiliario (tipo VARCHAR).

Subtipo: Categoría del mobiliario (tipo VARCHAR).

The screenshot shows the CODENEXUS application interface. At the top, there is a navigation bar with icons for various categories: Equipos de computo, Periféricos, Red y comunicaciones, Sala de reuniones, and Transporte. Below this, there is a sub-navigation bar with buttons for Router, Módems, Auriculares, Camara web (which is currently selected), and Projectores. The main content area features a table with the following data:

ID Mobiliario	Nombre	Descripción	Stock
3071	Logitech C920	Cámara HD 1080p, micrófono integrado	30
3072	Razer Kiyo	Cámara 720p, luz ajustable	15
3073	Microsoft LifeCam HD-3000	Cámara web 720p, micrófono integrado	20
3074	AverMedia Live Streamer CAM 313	Cámara 1080p, ideal para streaming	12
3075	Logitech StreamCam	Cámara Full HD, USB-C, ideal para creadores	18

To the right of the table, there is a search and filter section with the following fields:

- id:
- FECHA:
- Hora de inicio:
- Hora de termino:
- Confirmar button

## 4. Contacto y Soporte

Si tienes preguntas adicionales o necesitas soporte, por favor contacta al desarrollador o al equipo de soporte técnico.

[david.guzman.martinez@alumnos.uacm.edu.mx](mailto:david.guzman.martinez@alumnos.uacm.edu.mx)

[brayan.campos@alumnos.uacm.edu.mx](mailto:brayan.campos@alumnos.uacm.edu.mx)