

## App.java:

```
package edu.dam.app;

import java.nio.channels.Pipe.SourceChannel;
import java.util.Scanner;

import edu.dam.dao.DispositivosDAO;
import edu.dam.model.Dispositivos;
import edu.dam.util.DatabaseConnection;

public class App
{
    public static Scanner sc = new Scanner(System.in);
    public static DispositivosDAO dso = new DispositivosDAO();
    public static void main( String[] args ) {
        DatabaseConnection.iniciar();
        int num = 1;

        while (num != 6 || num > 6 || num < 1) {
            System.out.println("Introduce una opcion: "+
                "\n 1. Añadir Dispositivo"+
                "\n 2. Listar Dispositivo"+
                "\n 3. Buscar Por Categoria"+
                "\n 4. Actualizar precio/stock"+
                "\n 5. Eliminar dispositivo"+
                "\n 6. salir");
            switch (sc.nextInt()) {
                case 1 -> insertar();
                case 2 -> listar();
                case 3 -> buscar();
                case 4 -> actualizar();
                case 5 -> eliminar();
                case 6 -> salir();
                default -> salir();
            }
        }
    }
}
```

```
}

public static void insertar(){
    System.out.println("Digame el nombre del dispositivo: ");
    String n = sc.next();
    System.out.println("Digame la categoria del dispositivo");
    String c = sc.next();
    System.out.println("Digame el precio del dispositivo");
    double p = sc.nextDouble();
    System.out.println("Digame el stock del dispositivo");
    int s = sc.nextInt();
    Dispositivos d = new Dispositivos(n,c,p,s);
    try {
        dso.insertarDispositivo(d);
    } catch (Exception e) {
        System.out.println(e);
    }
}

public static void listar(){
    try {
        dso.listarDispositivos();
    } catch (Exception e) {
        System.out.println(e);
    }
}

public static void actualizar(){
    try {
        System.out.println("Digame el id del dispositivo: ");
        Dispositivos d = dso.buscarDispositivoID(sc.nextInt());
        System.out.println("Digame el nuevo stock: ");
        d.setStock(sc.nextInt());
        System.out.println("Digame el nuevo precio: ");
        d.setPrecio(sc.nextDouble());
        dso.actualizarDispositivos(d);
    } catch (Exception e) {
        System.out.println(e);
    }
}
```

```
public static void eliminar() {
    try {
        System.out.println("Digame el id del dispositivo que desea
eliminar: ");
        dso.eliminarDispositivo(sc.nextInt());
    } catch (Exception e) {
        System.out.println(e);
    }
}

public static void buscar() {
    try {
        System.out.println("Digame la categoria: ");
        String c = sc.next();
        System.out.println("Digame el precio minimo: ");
        double pmin = sc.nextDouble();
        System.out.println("Digame el precio maximo: ");
        double pmax = sc.nextDouble();
        dso.buscarPorCategorialyRangoPrecio(c, pmin, pmax);
    } catch (Exception e) {
        System.out.println(e);
    }
}

public static void salir(){
    System.out.println("Saliendo del programa ...");
    sc.close();
    System.exit(0);
}
```

## DispositivosDAO:

```
package edu.dam.dao;

import java.net.ConnectException;
import java.nio.channels.Pipe.SourceChannel;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

import edu.dam.model.Dispositivos;
import edu.dam.util.*;
import edu.dam.app.App;

public class DispositivosDAO {

    public void insertarDispositivo(Dispositivos d) throws SQLException{
        String sql = "INSERT INTO dispositivos(nombre, categoria, precio,
stock) VALUES (?, ?, ?, ?); ";
        try(Connection c = DatabaseConnection.getConection()) {
            PreparedStatement ps = c.prepareStatement(sql);

            ps.setString(1, d.getNombre());
            ps.setString(2, d.getCategoría());
            ps.setDouble(3, d.getPrecio());
            ps.setInt(4, d.getStock());

            ps.execute();

            System.out.println("Se inserto correctamente a la persona.");
            c.close();
        } catch(Exception e) {

        }
    }

    public void actualizarDispositivos(Dispositivos d) throws SQLException{
```

```
        String sql = "UPDATE dispositivos SET precio = ?, stock = ? WHERE id = ?;" ;
        try (Connection c = DatabaseConnection.getConection()) {
            PreparedStatement ps = c.prepareStatement(sql);
            ps.setDouble(1, d.getPrecio());
            ps.setInt(2, d.getStock());
            ps.setInt(3, d.getId());
            ps.executeUpdate();
            System.out.println("Se ejecuto correctamente el cambio");
            c.close();
        }

    } catch (Exception e) {
        System.out.println(e);
    }
}

public void eliminarDispositivo(int id) throws SQLException{
    String sql = "DELETE FROM dispositivos WHERE id = ?;" ;

    try (Connection c = DatabaseConnection.getConection()){
        PreparedStatement ps = c.prepareStatement(sql);
        ps.setInt(1, id);
        ps.execute();
        System.out.println("Se elimino correctamente el dispositivo ");
        c.close();
    } catch (Exception e) {
        System.out.println(e);
    }
}

public void listarDispositivos() throws SQLException{
    String sql = "SELECT * FROM dispositivos;" ;

    try( Connection c = DatabaseConnection.getConection() ) {
        PreparedStatement ps = c.prepareStatement(sql);
        ResultSet rs = ps.executeQuery();
        boolean comprob = false;

        while(rs.next()) {
```

```

        comprob = true;
        int id = rs.getInt("id");
        String nombre = rs.getString("nombre");
        String categoria = rs.getString("categoria");
        double precio = rs.getDouble("precio");
        int stock = rs.getInt("stock");
        Dispositivos ds = new Dispositivos(id, nombre, categoria,
precio, stock);
        System.out.println(ds);
    }

    if (comprob == false) {
        System.out.println("No hay resultado en la lista");
    }
    c.close();
} catch (Exception e) {
    System.out.println("Ha habido un error y el error es:
\n"+e.getMessage()+"\n");
}
}

public void buscarPorCategoríaYRangoPrecio(String categoria, double
precioMin, double precioMax) throws SQLException{
    String sql = "SELECT * FROM dispositivos;"

    try( Connection c = DatabaseConnection.getConection() ) {
        PreparedStatement ps = c.prepareStatement(sql);
        ResultSet rs = ps.executeQuery();
        boolean comprob = false;
        System.out.println("Aqui los tienes filtrados: ");

        while (rs.next()) {
            if (rs.getString("categoria").equalsIgnoreCase(categoria))
{
                double precio = rs.getDouble("precio");
                if (precio >= precioMin && precio <= precioMax) {
                    String n = rs.getString("nombre");
                    int s = rs.getInt("stock");
                    Dispositivos ds = new
Dispositivos(n,categoria,precio,s);

```

```
        System.out.println(ds);

        comprob = true;
    }
}

if (comprob == false) {
    System.out.println("No habia ningun dispositivo con esas
características");
}
c.close();
} catch (Exception e) {
    System.out.println(e);
}
}

// aparte que lo necesito
public Dispositivos buscarDispositivoID(int id) throws SQLException{
    try (Connection c = DatabaseConnection.getConection()) {
        Dispositivos d = new Dispositivos();

        PreparedStatement ps = c.prepareStatement("SELECT * FROM
dispositivos;");
        ResultSet rs = ps.executeQuery();

        boolean comprob = false;
        while (rs.next()) {
            if (rs.getInt("id") == id) {
                d.setNombre(rs.getString("nombre"));
                d.setCategoria(rs.getString("categoria"));
                d.setPrecio(rs.getDouble("precio"));
                d.setStock(rs.getInt("stock"));
                comprob = true;
            }
        }

        if (comprob = true) {
```

```
        System.out.println("Dispositivo encontrado");
    }else{
        System.out.println("El dispositivo no ha sido encontrado se
devolverá null;");
    }
    c.close();
    return d;
} catch (Exception e) {
    return null;
}
}
```

## Dispositivos:

```
package edu.dam.model;

public class Dispositivos {
    private int id;
    private String nombre;
    private String categoria;
    private double precio;
    private int stock;

    public Dispositivos(int id, String nombre, String categoria, double
precio, int stock) {
        this.id = id;
        this.nombre = nombre;
        this.categoria = categoria;
        this.precio = precio;
        this.stock = stock;
    }

    public Dispositivos(String nombre, String categoria, double precio, int
stock) {
        this.nombre = nombre;
        this.categoria = categoria;
        this.precio = precio;
        this.stock = stock;
    }

    public Dispositivos() {
    }
}
```

```
public int getId() {
    return id;
}
public void setId(int id) {
    this.id = id;
}
public String getNombre() {
    return nombre;
}
public void setNombre(String nombre) {
    this.nombre = nombre;
}
public String getCategoría() {
    return categoría;
}
public void setCategoría(String categoría) {
    this.categoría = categoría;
}
public double getPrecio() {
    return precio;
}
public void setPrecio(double precio) {
    this.precio = precio;
}
public int getStock() {
    return stock;
}
public void setStock(int stock) {
    this.stock = stock;
}

@Override
public String toString() {
    return "dispositivo [id=" + id + ", nombre=" + nombre +",
categoría=" + categoría + ", precio=" + precio
```

```
+ ", stock=" + stock + "]";  
}  
  
}  
  
}
```

# DatabaseConnection:

```
package edu.dam.util;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.*;

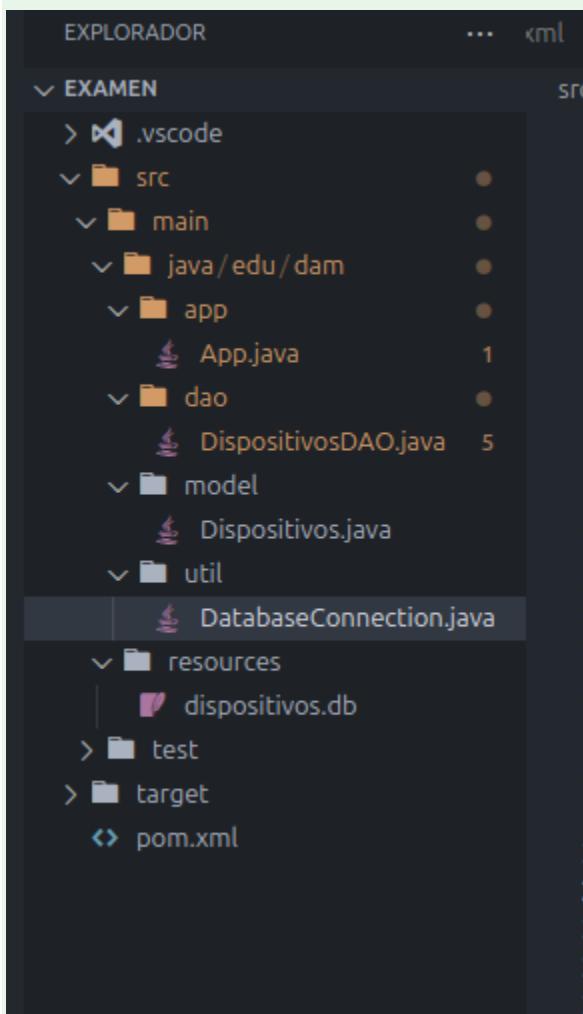
public class DatabaseConnection {
    private static final String url =
"jdbc:sqlite:src/main/resources/dispositivos.db";

    public static Connection getConexion() throws SQLException{
        return (Connection) (DriverManager.getConnection(url));
    }

    public static void iniciar(){
        String crear = "CREATE TABLE IF NOT EXISTS dispositivos (" +
        "id INTEGER PRIMARY KEY AUTOINCREMENT, " +
        "nombre TEXT NOT NULL, " +
        "categoria TEXT, " +
        "precio REAL, " +
        "stock INTEGER);";

        try (Connection c = DriverManager.getConnection(url); Statement st
= c.createStatement()){
            st.execute(crear);
            System.out.println("La base de datos se conecto correctamente");
            c.close();
        } catch (Exception e) {
            System.out.println("La base de datos falla por: " +
e.getMessage());
        }
    }
}
```

sistema de ficheros:



- ¿Qué has hecho y por qué lo has planteado de esa forma?

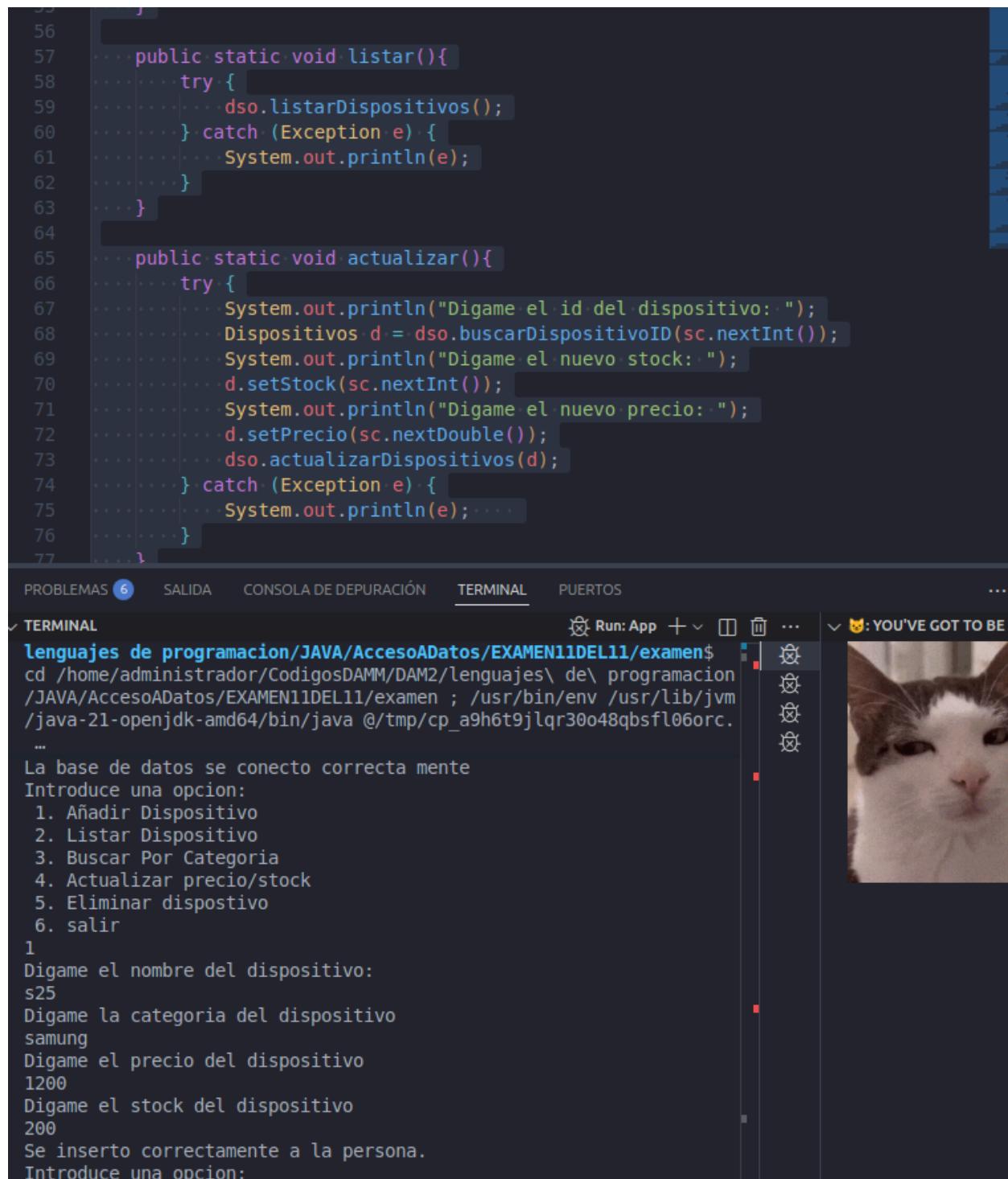
Lo he planteado con una base de datos en sqlite por que no es necesario tener nada en la nube y se puede tener en local. También es verdad que es la única manera que conozco de hacer un CRUD desde java.

- ¿Qué parte te ha resultado más complicada y cómo la has resuelto?

La parte de actualizar stock y precio por que no la terminaba de entender como pasarle los datos sin pedir ningún dato desde el DAO.

# Funcionamiento:

1:



The screenshot shows a Java IDE interface with the following components:

- Editor Area:** Displays Java code for methods `listar()` and `actualizar()`. The code interacts with a `dso` object to list or update devices based on user input from `System.out`.
- Terminal Tab:** Labeled "TERMINAL". It shows the execution of a Java application in a terminal window. The application connects to a database and presents a menu of options (1-6). It then prompts for device details (name, category, price, stock) and inserts the data.
- Terminal Output:**

```
lenguajes de programacion/JAVA/AccesoADatos/EXAMEN11DEL11/examen$ cd /home/administrador/CodigosDAMM/DAM2/lenguajes\ de\ programacion /JAVA/AccesoADatos/EXAMEN11DEL11/examen ; /usr/bin/env /usr/lib/jvm /java-21-openjdk-amd64/bin/java @/tmp/cp_a9h6t9jlqr30o48qbsfl06orc.
...
La base de datos se conecto correctamente
Introduce una opcion:
1. Añadir Dispositivo
2. Listar Dispositivo
3. Buscar Por Categoria
4. Actualizar precio/stock
5. Eliminar dispositivo
6. salir
1
Digame el nombre del dispositivo:
s25
Digame la categoria del dispositivo
samung
Digame el precio del dispositivo
1200
Digame el stock del dispositivo
200
Se inserto correctamente a la persona.
Introduce una opcion:
```
- Image:** A small image of a white and black cat is visible in the bottom right corner of the terminal window.

2:



```
55
56     public static void listar(){
57         try {
58             dso.listarDispositivos();
59         } catch (Exception e) {
60             System.out.println(e);
61         }
62     }
63 }
64
65     public static void actualizar(){
66     try {
67         System.out.println("Digame el id del dispositivo: ");
68         Dispositivos d = dso.buscarDispositivoID(sc.nextInt());
69         System.out.println("Digame el nuevo stock: ");
70         d.setStock(sc.nextInt());
71         System.out.println("Digame el nuevo precio: ");
72         d.setPrecio(sc.nextDouble());
73         dso.actualizarDispositivos(d);
74     } catch (Exception e) {
75         System.out.println(e);
76     }
77 }
```

PROBLEMAS 6 SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS ...

TERMINAL

```
lenguajes de programacion/JAVA/AccesoADatos/EXAMEN11DEL11/examen$ cd /home/administrador/CodigosDAMM/DAM2/lenguajes\ de\ programacion /JAVA/AccesoADatos/EXAMEN11DEL11/examen ; /usr/bin/env /usr/lib/jvm /java-21-openjdk-amd64/bin/java @/tmp/cp_a9h6t9jlqr30o48qbsfl06orc.
```

...  
Introduce una opcion:  
1. Añadir Dispositivo  
2. Listar Dispositivo  
3. Buscar Por Categoria  
4. Actualizar precio/stock  
5. Eliminar dispositivo  
6. salir  
2  
dispositivo [id=2, nombre=nacho2, categoria=carpintero, precio=213.  
0, stock=124]  
dispositivo [id=3, nombre=s25, categoria=samung, precio=1200.0, sto  
ck=200]  
Introduce una opcion:  
1. Añadir Dispositivo  
2. Listar Dispositivo  
3. Buscar Por Categoria  
4. Actualizar precio/stock  
5. Eliminar dispositivo  
6. salir

3:



```
55
56     public static void listar(){
57         try {
58             dso.listarDispositivos();
59         } catch (Exception e) {
60             System.out.println(e);
61         }
62     }
63
64     public static void actualizar(){
65         try {
66             System.out.println("Digame el id del dispositivo: ");
67             Dispositivos d = dso.buscarDispositivoID(sc.nextInt());
68             System.out.println("Digame el nuevo stock: ");
69             d.setStock(sc.nextInt());
70             System.out.println("Digame el nuevo precio: ");
71             d.setPrecio(sc.nextDouble());
72             dso.actualizarDispositivos(d);
73         } catch (Exception e) {
74             System.out.println(e);
75         }
76     }
77 }
```

PROBLEMAS 6 SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS ...

TERMINAL

```
lenguajes de programacion/JAVA/AccesoADatos/EXAMEN11DEL11/examen$ cd /home/administrador/CodigosDAMM/DAM2/lenguajes\ de\ programacion /JAVA/AccesoADatos/EXAMEN11DEL11/examen ; /usr/bin/env /usr/lib/jvm /java-21-openjdk-amd64/bin/java @/tmp/cp_a9h6t9jlqr30o48qbsfl06orc.
...
5. Eliminar dispositivo
6. salir
3
Digame la categoria:
carpintero
Digame el precio minimo:
213
Digame el precio maximo:
214
Aqui los tienes filtrados:
dispositivo [id=0, nombre=nacho2, categoria=carpintero, precio=213,
0, stock=124]
Introduce una opcion:
1. Añadir Dispositivo
2. Listar Dispositivo
3. Buscar Por Categoria
4. Actualizar precio/stock
5. Eliminar dispositivo
6. salir
```

4:

```
55
56     public static void listar(){
57         try {
58             dso.listarDispositivos();
59         } catch (Exception e) {
60             System.out.println(e);
61         }
62     }
63
64     public static void actualizar(){
65         try {
66             System.out.println("Digame el id del dispositivo: ");
67             Dispositivos d = dso.buscarDispositivoID(sc.nextInt());
68             System.out.println("Digame el nuevo stock: ");
69             d.setStock(sc.nextInt());
70             System.out.println("Digame el nuevo precio: ");
71             d.setPrecio(sc.nextDouble());
72             dso.actualizarDispositivos(d);
73         } catch (Exception e) {
74             System.out.println(e);
75         }
76     }
77 }
```

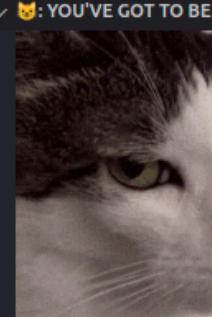
PROBLEMAS 6 SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS ...

TERMINAL

lenguajes de programacion/JAVA/AccesoADatos/EXAMEN11DEL11/examen\$ cd /home/administrador/CodigosDAMM/DAM2/lenguajes\ de\ programacion /JAVA/AccesoADatos/EXAMEN11DEL11/examen ; /usr/bin/env /usr/lib/jvm /java-21-openjdk-amd64/bin/java @/tmp/cp\_a9h6t9jlqr30o48qbsfl06orc.

...

4. Actualizar precio/stock  
5. Eliminar dispositivo  
6. salir  
4  
Digame el id del dispositivo:  
0  
Dispositivo encontrado  
Digame el nuevo stock:  
12  
Digame el nuevo precio:  
32  
Se ejecuto correctamente el cambio  
Introduce una opcion:  
1. Añadir Dispositivo  
2. Listar Dispositivo  
3. Buscar Por Categoria  
4. Actualizar precio/stock  
5. Eliminar dispositivo  
6. salir



5:



```
55
56     public static void listar(){
57         try {
58             dso.listarDispositivos();
59         } catch (Exception e) {
60             System.out.println(e);
61         }
62     }
63 }
64
65     public static void actualizar(){
66     try {
67         System.out.println("Digame el id del dispositivo: ");
68         Dispositivos d = dso.buscarDispositivoID(sc.nextInt());
69         System.out.println("Digame el nuevo stock: ");
70         d.setStock(sc.nextInt());
71         System.out.println("Digame el nuevo precio: ");
72         d.setPrecio(sc.nextDouble());
73         dso.actualizarDispositivos(d);
74     } catch (Exception e) {
75         System.out.println(e);
76     }
77 }
```

PROBLEMAS 6 SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS ...

TERMINAL

```
lenguajes de programacion/JAVA/AccesoADatos/EXAMEN11DEL11/examen$ cd /home/administrador/CodigosDAMM/DAM2/lenguajes\ de\ programacion /JAVA/AccesoADatos/EXAMEN11DEL11/examen ; /usr/bin/env /usr/lib/jvm /java-21-openjdk-amd64/bin/java @/tmp/cp_a9h6t9jlqr30o48qbsfl06orc.
...
Se ejecuto correctamente el cambio
Introduce una opcion:
1. Añadir Dispositivo
2. Listar Dispositivo
3. Buscar Por Categoria
4. Actualizar precio/stock
5. Eliminar dispositivo
6. salir
5
Digame el id del dispositivo que desea eliminar:
0
Se elimino correctamente el dispositivo
Introduce una opcion:
1. Añadir Dispositivo
2. Listar Dispositivo
3. Buscar Por Categoria
4. Actualizar precio/stock
5. Eliminar dispositivo
6. salir
```