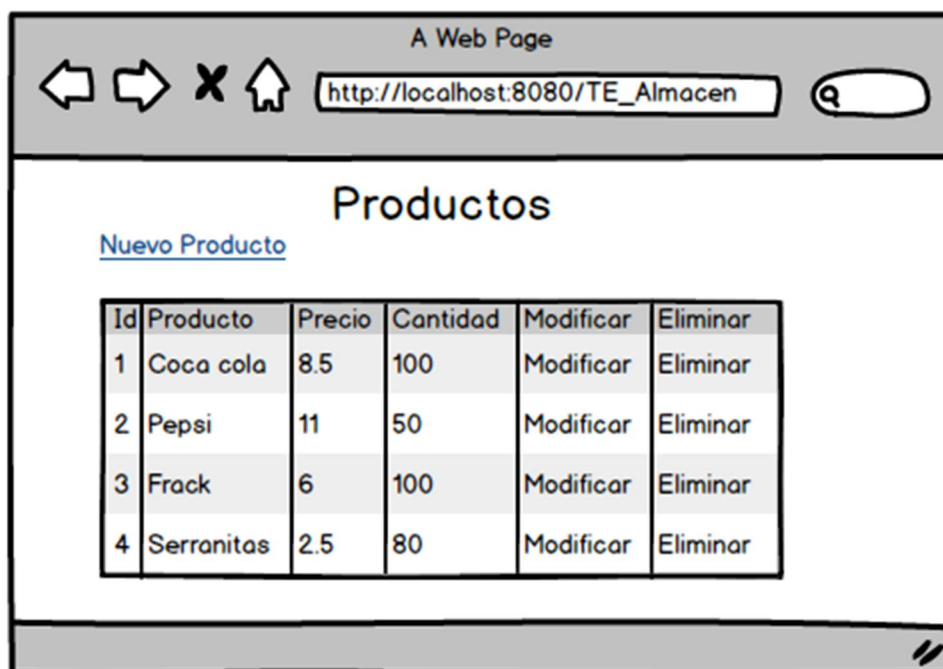


Práctica N° 4

Conexión a Base de Datos MySQL

Crear una aplicación Web en Java para gestionar los productos de un almacén.

1. Crear la base de datos **bd_almacen**
2. Crear la tabla **productos** con los campos (**id**:int(11) PK, **producto**:varchar(80), **precio**:float(11,2), **cantidad**:int(11))
3. Crear el CRUD para la tabla **productos**



Límite de entrega: 5 de octubre de 2021

Formato de entrega en plataforma virtual:

- Un archivo PDF con capturas de la base de datos creada, código fuente y el funcionamiento de la aplicación.

Estudiante: Mery Judith Mamani Mamani	Fecha: 04/11/2021	M
Paralelo: 7 c	Materia: Tec. Emergentes II	

PRÁCTICA 4

Base de datos- bd_almacen

Bases de datos

Crear base de datos

bd_almacen utf8_spanish2_ci

Base de datos	Cotejamiento	Acción
<input type="checkbox"/> bd_agenda	utf8_general_ci	Seleccionar privilegios

Estructura SQL Buscar Generar una consulta Exportar Importar

Ejecutar la(s) consulta(s) SQL en la base de datos bd_almacen:

```

1 CREATE TABLE productos (
2 id INT(11) AUTO_INCREMENT PRIMARY KEY,
3 producto varchar(80),
4 precio float(11,2),
5 cantidad INT(11)
6 );
7
8 INSERT INTO productos (producto,precio,cantidad) VALUES ('Coca Cola',8.5,100);
9 INSERT INTO productos (producto,precio,cantidad) VALUES ('Pepsi',11,50);
10 INSERT INTO productos (producto,precio,cantidad) VALUES ('Frack',6,100);
11 INSERT INTO productos (producto,precio,cantidad) VALUES ('Serranitas',2.5,80);

```

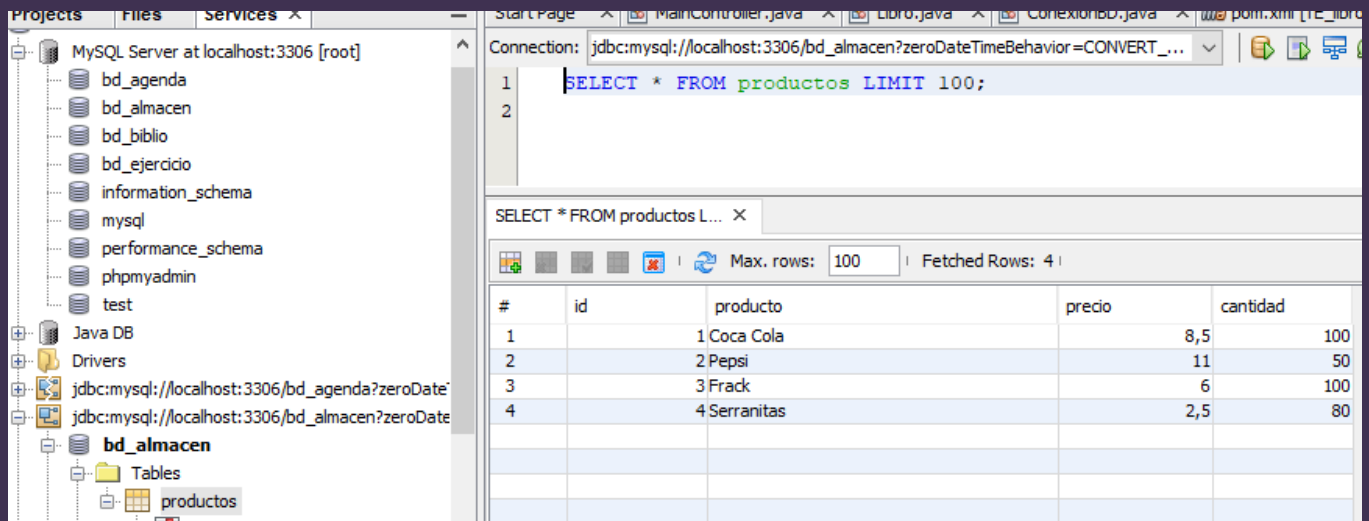
SELECT * FROM `productos`

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

+ Opciones

					id	producto	precio	cantidad		
<input type="checkbox"/>		Editar		Copiar		Borrar	1	Coca Cola	8.50	100
<input type="checkbox"/>		Editar		Copiar		Borrar	2	Pepsi	11.00	50
<input type="checkbox"/>		Editar		Copiar		Borrar	3	Frack	6.00	100
<input type="checkbox"/>		Editar		Copiar		Borrar	4	Serranitas	2.50	80



index.jsp (menú principal)

Código:

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@page import="com.emergentes.modelo.Producto"%>
<%@page import="java.util.List"%>
<%
    List<Producto> lista = (List<Producto>) request.getAttribute("lista");
%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="es">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Index</title>
    </head>
    <body bgcolor="cadetblue">
        <font face="Candara" size="4" color=LightYellow>
            <h1>PRODUCTOS</h1>
        <p><a href="MainController?op=nuevo">Nuevo Producto</a></p>
        <table border="1">
            <tr>
                <th>ID</th>
                <th>PRODUCTO</th>
                <th>PRECIO</th>
                <th>CANTIDAD</th>
                <th colspan="2">ACCIONES</th>
            </tr>
            <c:forEach var="item" items="${lista}">
                <tr>
                    <td>${item.id}</td>
                    <td>${item.producto}</td>
                    <td>${item.precio}</td>
                    <td>${item.cantidad}</td>
                    <td><a href="MainController?op=editar&id=${item.id}">Modificar</a></td>
                    <td><a href="MainController?op=eliminar&id=${item.id}"
                        onclick="return(confirm('¿Está seguro de eliminar?'))">Eliminar</a></td>
                </tr>
            </c:forEach>
        </table>
    </body>
</html>
```

editar.jsp

Código:

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@page import="com.emergentes.modelo.Producto"%>
<%
    Producto prod = (Producto)request.getAttribute("prod");
%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Editor</title>
    </head>
    <body bgcolor="cadetblue">
        <font face="Candara" size="3.5" color=LightYellow>
            <h1><c:if test="${prod.id==0}">
                Nuevo Producto
            </c:if>
            <c:if test="${prod.id!=0}">
                Edita Producto
            </c:if>
        </h1>
        <form action="MainController" method="post">
            <input type="hidden" name="id" value=${prod.id}>
            <table>
                <tr>
                    <td>PRODUCTO</td>
                    <td><input type="text" name="producto" value="${prod.producto}"></td>
                </tr>
                <tr>
                    <td>PRECIO</td>
                    <td><input type="text" name="precio" value="${prod.precio}"></td>
                </tr>
                <td>CANTIDAD</td>
                <td><input type="number" name="cantidad" min="0"
value="${prod.cantidad}"></td>
                <tr>
                    <td colspan="2"><input type="submit" value="Enviar" ></td>
                </tr>
            </table>
        </form>
    </body>
</html>
```

MainController.java

Código:

```
package com.emergentes.controlador;
import com.emergentes.modelo.Producto;
import com.emergentes.utiles.ConexionBD;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
```

```
import java.util.ArrayList;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import jdk.internal.net.http.hpack.HPACK;
```

```
@WebServlet(name = "MainController", urlPatterns = {"/MainController"})
public class MainController extends HttpServlet {
```

```
    @Override
```

```
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        try {
            String op;
            op = (request.getParameter("op") != null) ? request.getParameter("op") : "list";
            ArrayList<Producto> lista = new ArrayList<Producto>();
            int id;
            ConexionBD canal = new ConexionBD();
            Connection conn = canal.conectar();
            PreparedStatement ps;
            ResultSet rs;
            if (op.equals("list")) {
                // Operaciones para listar los datos
                String sql = "select * from productos";
                // Consulta de selección y almacenarlo en una colección
                ps = conn.prepareStatement(sql);
                rs = ps.executeQuery();
                while (rs.next()){
                    Producto prod = new Producto();
                    prod.setId(rs.getInt("id"));
                    prod.setProducto(rs.getString("producto"));
                    prod.setPrecio(rs.getFloat("precio"));
                    prod.setCantidad(rs.getInt("cantidad"));
                    lista.add(prod);
                }
                request.setAttribute("lista", lista);
                request.getRequestDispatcher("index.jsp").forward(request, response);
            }
            if (op.equals("nuevo")) {
                Producto li = new Producto();
                request.setAttribute("prod", li);
                request.getRequestDispatcher("editar.jsp").forward(request, response);
            }
            if (op.equals("editar")) {

                id = Integer.parseInt(request.getParameter("id"));
                try {
                    Producto prod1 = new Producto();
                    ps = conn.prepareStatement("select * from productos where id=?");
                    ps.setInt(1, id);
                    rs=ps.executeQuery();
                    if(rs.next()){
```

```

        prod1.setId(rs.getInt("id"));
        prod1.setProducto(rs.getString("producto"));
        prod1.setPrecio(rs.getFloat("precio"));
        prod1.setCantidad(rs.getInt("cantidad"));
    }
    request.setAttribute("prod", prod1);
    request.getRequestDispatcher("editar.jsp").forward(request, response);
} catch (SQLException ex) {
    Logger.getLogger(MainController.class.getName()).log(Level.SEVERE, null,
ex);
}

}
if (op.equals("eliminar")) {
    id = Integer.parseInt(request.getParameter("id"));
    String sql = "delete from productos where id = ?";
    ps = conn.prepareStatement(sql);
    ps.setInt(1, id);
    ps.executeUpdate();
    response.sendRedirect("MainController");
}
} catch (SQLException ex) {
    System.out.println("¡¡ ERROR AL CONECTAR !!" + ex.getMessage());
}
}

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    try {
        int id = Integer.parseInt(request.getParameter("id"));
        String producto = request.getParameter("producto");
        float precio = Float.parseFloat(request.getParameter("precio"));
        int cantidad = Integer.parseInt(request.getParameter("cantidad"));

        Producto prod = new Producto();
        prod.setId(id);
        prod.setProducto(producto);
        prod.setPrecio(precio);
        prod.setCantidad(cantidad);

        ConexionBD canal = new ConexionBD();
        Connection conn = canal.conectar();
        PreparedStatement ps;
        ResultSet rs;

        if (id == 0) {
            // Nuevo registro
            String sql = "insert into productos (producto, precio, cantidad) values
(?,?,?)";

            ps = conn.prepareStatement(sql);
            ps.setString(1, prod.getProducto());
            ps.setFloat(2, prod.getPrecio());
            ps.setInt(3, prod.getCantidad());
            ps.executeUpdate();

        } else {
            // Edición de registro
            String sql1 = "update productos set producto=?, precio=?, cantidad=? where
id=?";

```

```

        try {
            ps = conn.prepareStatement(sql1);
            ps.setString(1, prod.getProducto());
            ps.setFloat(2, prod.getPrecio());
            ps.setInt(3, prod.getCantidad());
            ps.setInt(4, prod.getId());
            ps.executeUpdate();
        } catch (SQLException ex) {
            Logger.getLogger(MainController.class.getName()).log(Level.SEVERE, null,
ex);
        }
    }
    response.sendRedirect("MainController");
} catch (SQLException ex) {
    System.out.println("Error en SQL " + ex.getMessage());
}
}
}

```

Producto.java

Código:

```

package com.emergentes.modelo;

public class Producto {
    private int id;
    private String producto;
    private float precio;
    private int cantidad;

    public Producto() {
        this.id = 0;
        this.producto = "";
        this.precio = 0;
        this.cantidad = 0;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getProducto() {
        return producto;
    }

    public void setProducto(String producto) {
        this.producto = producto;
    }

    public float getPrecio() {
        return precio;
    }

    public void setPrecio(float precio) {

```

```

        this.precio = precio;
    }

    public int getCantidad() {
        return cantidad;
    }

    public void setCantidad(int cantidad) {
        this.cantidad = cantidad;
    }

    @Override
    public String toString() {
        return "Producto{" + "id=" + id + ", producto=" + producto + ", precio=" + precio + ", cantidad=" + cantidad + '}';
    }
}

```

ConexionBD.java

Código:

```

package com.emergentes.utiles;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

public class ConexionBD {

    static String driver = "com.mysql.jdbc.Driver";
    static String url = "jdbc:mysql://localhost:3306/bd_almacen";
    static String usuario = "root";
    static String password = "";

    Connection conn = null;

    public ConexionBD() {
        try {
            Class.forName(driver);
            conn = DriverManager.getConnection(url, usuario, password);
            if (conn != null) {
                System.out.println("Conexión OK:" + conn);
            }
        } catch (SQLException ex) {
            System.out.println("Error de SQL" + ex.getMessage());
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(ConexionBD.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    public Connection conectar() {
        return conn;
    }

    public void desconectar() {
        try {
            conn.close();
        }
    }
}

```



```

    } catch (SQLException ex) {
        Logger.getLogger(ConexionBD.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}

```

pom.xml

```

<properties>
    <endorsed.dir>${project.build.directory}/endorsed</endorsed.dir>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
</properties>

<dependencies>
    <dependency>
        <groupId>javax</groupId>
        <artifactId>javaee-web-api</artifactId>
        <version>7.0</version>
        <scope>provided</scope>
    </dependency>
    <dependency>
        <groupId>javax.servlet</groupId>
        <artifactId>jstl</artifactId>
        <version>1.2</version>
    </dependency>
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.26</version>
    </dependency>
</dependencies>

```

web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
    <welcome-file-list>
        <welcome-file>MainController</welcome-file>
    </welcome-file-list>
</web-app>

```

FUNCIONAMIENTO

Visualización de registros:



ID	PRODUCTO	PRECIO	CANTIDAD	ACCIONES	
1	Coca Cola	8.5	100	Modificar	Eliminar
2	Pepsi	11.0	50	Modificar	Eliminar
3	Frack	6.0	100	Modificar	Eliminar
4	Serranitas	2.5	80	Modificar	Eliminar

SELECT * FROM `productos`

☐

Perfilando

[\[Editar en línea \]](#)

[\[Editar \]](#)

[\[Explicar SQL \]](#)

[\[Crear código PHP \]](#)

☐

Mostrar todo

|

Número de filas:

25

▼

Filtrar filas:

Buscar

+ Opciones

←

T

→

▼

					id	producto	precio	cantidad		
<input type="checkbox"/>		Editar		Copiar		Borrar	1	Coca Cola	8.50	100
<input type="checkbox"/>		Editar		Copiar		Borrar	2	Pepsi	11.00	50
<input type="checkbox"/>		Editar		Copiar		Borrar	3	Frack	6.00	100
<input type="checkbox"/>		Editar		Copiar		Borrar	4	Serranitas	2.50	80

Nuevo registro:

← → ↻ ⓘ localhost:8080/p4/MainController?op=nuevo

Nuevo Producto

PRODUCTO

PRECIO

CANTIDAD

← → ↻ ⓘ localhost:8080/p4/MainController

PRODUCTOS

[Nuevo Producto](#)

ID	PRODUCTO	PRECIO	CANTIDAD	ACCIONES	
1	Coca Cola	8.5	100	Modificar	Eliminar
2	Pepsi	11.0	50	Modificar	Eliminar
3	Frack	6.0	100	Modificar	Eliminar
4	Serranitas	2.5	80	Modificar	Eliminar
6	Leche	6.5	15	Modificar	Eliminar

← → ↻ ⓘ localhost:8080/p4/MainController?op=editar&id=4

Edita Producto

PRODUCTO

PRECIO

CANTIDAD

← → ↻ ⓘ localhost:8080/p4/MainController

PRODUCTOS

[Nuevo Producto](#)

ID	PRODUCTO	PRECIO	CANTIDAD	ACCIONES	
1	Coca Cola	8.5	100	Modificar	Eliminar
2	Pepsi	11.0	50	Modificar	Eliminar
3	Frack	6.0	100	Modificar	Eliminar
4	Serranitasssssssssss	2.5	80	Modificar	Eliminar
6	Leche	6.5	15	Modificar	Eliminar

Eliminación de registro:

← → ↻ ⓘ localhost:8080/p4/MainController

PRODUCTOS

[Nuevo Producto](#)

localhost:8080 dice

¿Está seguro de eliminar?

Aceptar Cancelar

ID	PRODUCTO	PRECIO	CANTIDAD	ACCIONES	
1	Coca Cola	8.5	100	Modificar	Eliminar
2	Pepsi	11.0	50	Modificar	Eliminar
3	Frack	6.0	100	Modificar	Eliminar
4	Serranitasssssssssss	2.5	80	Modificar	Eliminar
6	Leche	6.5	15	Modificar	Eliminar

← → ↻ ⓘ localhost:8080/p4/MainController

PRODUCTOS

[Nuevo Producto](#)

ID	PRODUCTO	PRECIO	CANTIDAD	ACCIONES	
1	Coca Cola	8.5	100	Modificar	Eliminar
2	Pepsi	11.0	50	Modificar	Eliminar
3	Frack	6.0	100	Modificar	Eliminar
4	Serranitasssssssssss	2.5	80	Modificar	Eliminar

LINK - GITHUB

<https://github.com/ByMeyJudy/practica-4.git>