

1-Cliente ingresa al sistema e inicia sesion ingresando las credenciales correspondientes. Sistema valida credenciales ingresadas.

ControladorLogin.java:

```
package controlador;
import java.io.IOException;
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 javax.servlet.http.HttpSession;
import modelo.ClienteDAO;
import modelo.Cliente;
import static validaciones.Encriptador.decrypt;
@WebServlet("/ControladorLogin")
public class ControladorLogin extends HttpServlet {
    private static final long serialVersionUID = 1L;
    ClienteDAO cDAO = new ClienteDAO();
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException, Exception {
    }
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String menu = request.getParameter("menu");
        String accion = request.getParameter("accion");
        HttpSession session = request.getSession();
        if (menu.equals("login")) {
            switch (accion) {
                case "login": {
                    request.getRequestDispatcher("login.jsp").forward(request,
response);
                    break;
                }
                case "logout": {
                    session.setAttribute("auth", null);
                    request.getRequestDispatcher("index.jsp").forward(request,
response);
                    break;
                }
                case "registro": {
                    request.getRequestDispatcher("registro.jsp").forward(request,
response);
                }
                default: {
```

```

request.getRequestDispatcher("ControladorInicio?menu=inicio&accion=inicio").forward(request,
response);
    }
    }
}
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    String menu = request.getParameter("menu");
    String accion = request.getParameter("accion");
    if (menu.equals("login")) {
        switch (accion) {
            case "loginCliente": {
                HttpSession session = request.getSession();
                String password = request.getParameter("password");
                Cliente c = cDAO.login(request.getParameter("email"));
                if (c != null) {
                    try {
                        if (password.equals(decrypt(c.getPassword()))) {
                            session.setAttribute("auth", c);

response.sendRedirect("ControladorInicio?menu=inicio&accion=inicio");
                        } else {
                            // Password incorrecto
                            session.setAttribute("mensaje", "Usuario o
Contraseña incorrecto");

response.sendRedirect("ControladorLogin?menu=login&accion=login");
                        }
                    } catch (Exception ex) {
                        Logger.getLogger(ControladorLogin.class.getName()).log(Level.SEVERE, null, ex);
                    }
                } else {
                    // Cliente no encontrado
                    session.setAttribute("mensaje", "Usuario o Contraseña
incorrecto");

response.sendRedirect("ControladorLogin?menu=login&accion=login");
                }
                break;
            }
        }
    }
}
}
}
}
}

```

Login.jsp:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Login</title>
<%@include file="estructura/head.jsp"%>
</head>
<body>
    <%@include file="estructura/nav.jsp"%>
    <div class="container">
        <%
            if (request.getSession().getAttribute("mensaje") != null) {
        %>
        <div class="alert alert-danger alert-dismissible fade show"
            role="alert">
            <strong><%=request.getSession().getAttribute("mensaje")%></strong>
            <%
                request.getSession().setAttribute("mensaje", null);
            %>
            <button type="button" class="btn-close" data-bs-dismiss="alert"
                aria-label="Close"></button>
        </div>
        <%
            }
        %>
        <div class="card w-50 mx-auto my-5">
            <div class="card-header text-center">Iniciar Sesión</div>
            <div class="card-body">
                <form action="ControladorLogin?menu=login&accion=loginCliente"
                    method="post">
                    <div class="form-group">
                        <label>Email</label> <input type="email"
class="form-control"
                                name="email" id="email"
placeholder="Ingrese su Email" required>
                    </div>
                    <div class="form-group">
                        <label>Contraseña</label> <input type="password"
class="form-control" name="password"
id="password"
                                placeholder="Ingrese su contraseña"
required>
                    </div>
                    <br>
                    <div class="d-flex justify-content-between
align-items-center">
```

```

<button type="submit" class="btn
btn-primary">Iniciar
session</button>
<a
href="ControladorLogin?menu=login&accion=registro"
class="btn btn-primary">Registrarse</a>
</div>
</form>
</div>
</div>
<%@include file="estructura/footer.jsp"%>
</body>
</html>

```

Cliente.java

```

package modelo;
public class Cliente {
    private int id;
    private String nombre;
    private String apellido;
    private String telefono;
    private String email;
    private String direccion;
    private String cuit;
    private int idCategoria;
    private String password;

    public Cliente(){
    }
    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 getApellido() {
        return apellido;
    }
    public void setApellido(String apellido) {

```

```
this.apellido = apellido;
}

public String getTelefono() {
    return telefono;
}
public void setTelefono(String telefono) {
    this.telefono = telefono;
}
public String getEmail() {
    return email;
}
public void setEmail(String email) {
    this.email = email;
}
public String getDireccion() {
    return direccion;
}
public void setDireccion(String direccion) {
    this.direccion = direccion;
}
public String getCuit() {
    return cuit;
}
public void setCuit(String cuit) {
    this.cuit = cuit;
}
public int getIdCategoria() {
    return idCategoria;
}
public void setIdCategoria(int idCategoria) {
    this.idCategoria = idCategoria;
}
public String getPassword() {
    return password;
}
public void setPassword(String password) {
    this.password = password;
}
}
```

ClienteDAO.java

```
package modelo;

import config.Conexion;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

public class ClienteDAO {

    Connection conexion;
    PreparedStatement ps;
    ResultSet rs;

    public ClienteDAO() {

    }

    public List<Cliente> getAll() {
        String sql = "SELECT * FROM cliente";
        List<Cliente> clientes = new ArrayList<>();
        try {
            Conexion con = new Conexion();
            conexion = con.getConexion();
            ps = conexion.prepareStatement(sql);
            rs = ps.executeQuery();

            while (rs.next()) {
                Cliente cli = new Cliente();
                cli.setId(rs.getInt(1));
                cli.setNombre(rs.getString(2));
                cli.setApellido(rs.getString(3));
                cli.setTelefono(rs.getString(4));
                cli.setEmail(rs.getString(5));
                cli.setDireccion(rs.getString(6));
                cli.setCuit(rs.getString(7));
                cli.setIdCategoria(rs.getInt(8));
                cli.setPassword(rs.getString(9));

                clientes.add(cli);
            }
            ps.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

```
    } catch (SQLException e) {
        System.out.println(e.toString());
    }

    return clientes;

}

public boolean save(Cliente c) {
    String sql = "INSERT INTO cliente(nombre, apellido, telefono, email, direccion, cuit,
idCategoria, password) VALUES(?,?,?,?,?,?,?,?)";

    try {
        Conexion con = new Conexion();
        conexion = con.getConexion();
        ps = conexion.prepareStatement(sql);
        ps.setString(1, c.getNombre());
        ps.setString(2, c.getApellido());

        ps.setString(3, c.getTelefono());
        ps.setString(4, c.getEmail());
        ps.setString(5, c.getDireccion());
        ps.setString(6, c.getCuit());
        ps.setInt(7, c.getIdCategoria());
        ps.setString(8, c.getPassword());
        ps.executeUpdate();
        ps.close();
        return true;
    } catch (SQLException e) {
        System.out.println(e.toString());
    }
    return false;
}

public void update(Cliente c) {
    String sql = "UPDATE cliente SET nombre=?, apellido=?, telefono=?, direccion=?,
cuit=?, idCategoria=? WHERE idCliente=?";
    try {
        Conexion con = new Conexion();
        conexion = con.getConexion();
        ps = conexion.prepareStatement(sql);
        ps.setString(1, c.getNombre());
        ps.setString(2, c.getApellido());

        ps.setString(3, c.getTelefono());
```

```
        ps.setString(4, c.getDireccion());
        ps.setString(5, c.getCuit());
        ps.setInt(6, c.getIdCategoria());
        ps.setInt(7, c.getId());
        ps.executeUpdate();
        ps.close();

    } catch (SQLException e) {
        System.out.println(e.toString());
    }
}

public boolean delete(int id) {
    String sql = "DELETE FROM cliente WHERE idCliente=?";
    try {
        Conexion con = new Conexion();
        conexion = con.getConexion();
        ps = conexion.prepareStatement(sql);
        ps.setInt(1, id);
        ps.executeUpdate();
        ps.close();
        return true;
    } catch (SQLException e) {
        System.out.println(e.toString());
        return false;
    }
}

public Cliente getById(int id) {
    String sql = "SELECT * FROM cliente WHERE idCliente=?";
    Cliente cli = null;
    try {
        Conexion con = new Conexion();
        conexion = con.getConexion();
        ps = conexion.prepareStatement(sql);
        ps.setInt(1, id);
        rs = ps.executeQuery();

        while (rs.next()) {
            cli = new Cliente();
            cli.setId(rs.getInt("idCliente"));
            cli.setNombre(rs.getString("nombre"));
            cli.setApellido(rs.getString("apellido"));
            cli.setTelefono(rs.getString("telefono"));
        }
    }
}
```



```
        cli.setEmail(rs.getString("email"));
        cli.setDireccion(rs.getString("direccion"));
        cli.setCuit(rs.getString("cuit"));
        cli.setIdCategoria(rs.getInt("idCategoria"));
        cli.setPassword(rs.getString("password"));

    }
    ps.close();
} catch (SQLException e) {
    System.out.println("Error al traer Cliente: "+e.toString());
}
return cli;
}

public Cliente login(String email) {
    String sql = "SELECT * FROM cliente WHERE email=?";
    Cliente cli = null;
    try {
        Conexion con = new Conexion();
        conexion = con.getConexion();
        ps = conexion.prepareStatement(sql);
        ps.setString(1, email);

        rs = ps.executeQuery();

        while (rs.next()) {

            cli = new Cliente();
            cli.setId(rs.getInt("idCliente"));
            cli.setNombre(rs.getString("nombre"));
            cli.setApellido(rs.getString("apellido"));
            cli.setTelefono(rs.getString("telefono"));
            cli.setEmail(rs.getString("email"));
            cli.setDireccion(rs.getString("direccion"));
            cli.setCuit(rs.getString("cuit"));
            cli.setIdCategoria(rs.getInt("idCategoria"));
            cli.setPassword(rs.getString("password"));
        }

        ps.close();
    } catch (SQLException e) {
        System.out.println(e.toString());
    }
    return cli;
}
```

Encriptador.java:

```
package validaciones;

import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;
import java.security.Key;
import java.util.Base64;

public class Encriptador {

    private static final String AES_ALGORITHM = "AES";
    private static final String ENCRYPTION_KEY = "nax3QSuveekMhA46ak4Tsg==";

    public static String encrypt(String data) throws Exception {
        Key key = generateKey();
        Cipher cipher = Cipher.getInstance(AES_ALGORITHM);
        cipher.init(Cipher.ENCRYPT_MODE, key);
        byte[] encryptedValue = cipher.doFinal(data.getBytes());
        return Base64.getEncoder().encodeToString(encryptedValue);
    }

    public static String decrypt(String encryptedData) throws Exception {
        Key key = generateKey();
        Cipher cipher = Cipher.getInstance(AES_ALGORITHM);
        cipher.init(Cipher.DECRYPT_MODE, key);
        byte[] decryptedValue = cipher.doFinal(Base64.getDecoder().decode(encryptedData));
        return new String(decryptedValue);
    }

    private static Key generateKey() throws Exception {
        byte[] keyValue = ENCRYPTION_KEY.getBytes("UTF-8");
        return new SecretKeySpec(keyValue, AES_ALGORITHM);
    }
}
```

2-Cliente agrega producto al carrito de compras. Paso 2 se repite tantas veces como el cliente desea.

Index.jsp:

```
<%@page import="modelo.Carrito"%>
<%@page import="java.util.ArrayList"%>
<%@page import="modelo.Cliente"%>
<%@page import="modelo.TipoProductoDAO"%>
<%@page import="modelo.TipoProducto"%>
<%@page import="modelo.Producto"%>
<%@page import="java.util.List"%>
<%@page import="modelo.ProductoDAO"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%
ProductoDAO pDAO = new ProductoDAO();
String buscador = (String) request.getAttribute("buscador");
List<Producto> productos;
if (buscador == null || buscador.equals("")) {
    productos = pDAO.getAll();
} else {
    productos = pDAO.getByDescripcion(buscador);
}
if (productos.isEmpty()) {
    request.setAttribute("mensajeExiste", "No se encontraron productos");
}
TipoProducto tp = new TipoProducto();
TipoProductoDAO tpDAO = new TipoProductoDAO();
ArrayList<Carrito> carrito_lista = (ArrayList<Carrito>) session.getAttribute("carrito-lista");
if (carrito_lista != null) {
    request.setAttribute("carrito_lista", carrito_lista);
}
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Ecommerce Java</title>
<%@include file="estructura/head.jsp"%>
</head>
<body>
    <%
        if (request.getSession().getAttribute("authEmpleado") != null) {
            %><%@include file="estructura/navAdmin.jsp"%>
        } else {
            %>
            <%@include file="estructura/nav.jsp"%>
        }
    %>
```

```

}
%>
<h1>Tienda</h1>
<%
if (request.getAttribute("mensajeAgregado") != null) {
%>
<div class="alert alert-success alert-dismissible fade show"
    role="alert">
    <strong><%=request.getAttribute("mensajeAgregado")%></strong>
    <%
    request.setAttribute("mensajeAgregado", null);
    %>
    <button type="button" class="btn-close" data-bs-dismiss="alert"
        aria-label="Close"></button>
</div>
<%
}
%>
<%
if (request.getAttribute("mensajeExiste") != null) {
%>
<div class="alert alert-danger alert-dismissible fade show"
    role="alert">
    <strong><%=request.getAttribute("mensajeExiste")%></strong>
    <%
    request.setAttribute("mensajeExiste", null);
    %>
    <button type="button" class="btn-close" data-bs-dismiss="alert"
        aria-label="Close"></button>
</div>
<%
}
%>
<%
if (request.getSession().getAttribute("mensajeExito") != null) {
%>
<div class="alert alert-success alert-dismissible fade show"
    role="alert">
    <strong><%=request.getSession().getAttribute("mensajeExito")%></strong>
    <%
    request.getSession().setAttribute("mensajeExito", null);
    %>
    <button type="button" class="btn-close" data-bs-dismiss="alert"
        aria-label="Close"></button>
</div>
<%
}
%>
<%
if (request.getSession().getAttribute("mensajeError") != null) {

```

```

%>
<div class="alert alert-danger alert-dismissible fade show"
    role="alert">
    <strong><%=request.getSession().getAttribute("mensajeError")%></strong>
    <%
    request.getSession().setAttribute("mensajeError", null);
    %>
    <button type="button" class="btn-close" data-bs-dismiss="alert"
        aria-label="Close"></button>
</div>
<%
}
%>
<!-- -----BUSCADOR----- -->
<div class="container text-center">
    <div class="row mt-3">
        <div class="col-12">
            <form action="ControladorInicio?menu=inicio&accion=buscador"
                method="post">
                <div class="input-group" style="max-width: 400px; margin: 0
auto;">
                    <input type="text" name="buscador" id="buscador"
                        class="form-control"
placeholder="Buscar...">
                    <button class="btn btn-primary"
type="submit">Buscar</button>
                </div>
            </form>
        </div>
    </div>
</div>
<div class="container">
    <div class="card-header my-3">Productos</div>
    <div class="row">
        <%
        if (!productos.isEmpty()) {
            for (Producto p : productos) {
                tp = tpDAO.getByld(p.getIdTipo());
            %>
            <div class="col-md-3 my-3">
                <div class="card w-100" style="width: 18rem;">
                    
                    <div class="card-body">
                        <h5 class="card-title">
                            <%=p.getDescripcion()%></h5>
                        <h6 class="price">
                            Precio: $
                            <%=p.getPrecio()%>
                        </h6>

```

```

        <h6 class="categoria">
            Categoria:
            <%=tp.getDescripcion()%>
        </h6>
        <%
            if (p.getStock() > 0) {
                <%>
                <div class="mt-3 d-flex justify-content-between">
                    <a
href="ControladorCarrito?menu=carrito&accion=agregar&id=<%=p.getIdProducto()%>"
class="btn btn-primary">Agregar al
carrito</a>

                    </div>
                    <%
                } else {
                <%>
                <div class="mt-3 d-flex justify-content-between">
                    <a href="#" class="btn btn-danger"
tabindex="-1" role="button"
aria-disabled="true">Agotado</a>

                    </div>
                    <%
                }
                <%>
            }
        </div>
    </div>
</div>
<%@include file="estructura/footer.jsp"%>
</body>
</html>

```

Producto.java

```
package modelo;

public class Producto {
    private int idProducto;
    private String descripcion;
    private double precio;
    private int idTipo;
    private int stock;
    private String imagen;

    public Producto(){
    }

    public int getIdProducto() {
        return idProducto;
    }

    public void setIdProducto(int idProducto) {
        this.idProducto = idProducto;
    }

    public String getDescripcion() {
        return descripcion;
    }

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }

    public double getPrecio() {
        return precio;
    }

    public void setPrecio(double precio) {
        this.precio = precio;
    }

    public int getIdTipo() {
        return idTipo;
    }

    public void setIdTipo(int idTipo) {
        this.idTipo = idTipo;
    }

    public int getStock() {
        return stock;
    }

    public void setStock(int stock) {
        this.stock = stock;
    }

    public String getImagen() {
        return imagen;
    }

    public void setImagen(String imagen) {
        this.imagen = imagen;
    }
}
```

ProductoDAO.java

```
package modelo;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import config.Conexion;

import java.util.List;
import java.util.ArrayList;

public class ProductoDAO {

    Connection conexion;
    PreparedStatement ps;
    ResultSet rs;

    public ProductoDAO() {

    }

    public List<Producto> getAll() {
        String sql = "SELECT * FROM producto";
        List<Producto> productos = new ArrayList<>();
        try {
            Conexion con = new Conexion();
            conexion = con.getConexion();

            ps = conexion.prepareStatement(sql);
            rs = ps.executeQuery();

            while (rs.next()) {
                Producto p = new Producto();
                p.setIdProducto(rs.getInt("idProducto"));
                p.setDescripcion(rs.getString("descripcion"));
                p.setPrecio(rs.getDouble("precio"));
                p.setIdTipo(rs.getInt("idTipo"));
                p.setStock(rs.getInt("stock"));
                p.setImagen(rs.getString("imagen"));

                productos.add(p);
            }
            ps.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```



```
    } catch (SQLException e) {
        System.out.println(e.toString());
    }
    return productos;
}

public void save(Producto p) {
    String sql = "INSERT INTO producto(descripcion, precio, idTipo, stock, imagen)
VALUES(?,?,?,?,?)";
    try {
        Conexion con = new Conexion();
        conexion = con.getConexion();
        ps = conexion.prepareStatement(sql);
        ps.setString(1, p.getDescripcion());
        ps.setDouble(2, p.getPrecio());
        ps.setInt(3, p.getIdTipo());
        ps.setInt(4, p.getStock());
        ps.setString(5, p.getImagen());
        ps.executeUpdate();
        ps.close();
    } catch (SQLException e) {
        System.out.println(e.toString());
    }
}

//Devuelve lista tipo Carrito completando cada elemento con los datos del producto
public List<Carrito> getCarritoProductos(ArrayList<Carrito> listaCarrito) {
    List<Carrito> productos = new ArrayList<>();

    try {
        Conexion con = new Conexion();
        conexion = con.getConexion();
        if (!listaCarrito.isEmpty()) {
            for (Carrito item : listaCarrito) {
                String sql = "SELECT * FROM producto WHERE idProducto=?";
                ps = conexion.prepareStatement(sql);
                ps.setInt(1, item.getIdProducto());
                rs = ps.executeQuery();

                while (rs.next()) {
                    Carrito c = new Carrito();
                    c.setIdProducto(rs.getInt("idProducto"));
                    c.setDescripcion(rs.getString("descripcion"));
                    c.setIdTipo(rs.getInt("idTipo"));
                    c.setCantidad(1);
                }
            }
        }
    }
}
```

```
        c.setPrecio(rs.getDouble("precio"));
        c.setStock(rs.getInt("stock"));
        c.setCantidad(item.getCantidad());
        productos.add(c);
    }
    ps.close();
}
}
} catch (SQLException e) {
    System.out.println(e.toString());
}
return productos;
}

public double getTotal(ArrayList<Carrito> listaCarrito) {
    double total = 0;

    try {
        Conexion con = new Conexion();
        conexion = con.getConexion();
        if (!listaCarrito.isEmpty()) {
            for (Carrito c : listaCarrito) {
                String sql = "SELECT * FROM producto WHERE idProducto=?";
                ps = conexion.prepareStatement(sql);
                ps.setInt(1, c.getIdProducto());
                rs = ps.executeQuery();

                while (rs.next()) {
                    total = total + (c.getCantidad() * rs.getDouble("precio"));
                }
                ps.close();
            }
        }
    } catch (SQLException e) {
        System.out.println(e.toString());
    }
    return total;
}

public Producto getById(int id) {
```

```
String sql = "SELECT * FROM producto WHERE idProducto=?";
Producto producto = null;
try {
    Conexion con = new Conexion();
    conexion = con.getConexion();
    ps = conexion.prepareStatement(sql);
    ps.setInt(1, id);
    rs = ps.executeQuery();

    while (rs.next()) {
        producto = new Producto();
        producto.setIdProducto(rs.getInt("idProducto"));
        producto.setDescripcion(rs.getString("descripcion"));
        producto.setPrecio(rs.getDouble("precio"));
        producto.setIdTipo(rs.getInt("idTipo"));
        producto.setStock(rs.getInt("stock"));
        producto.setImagen(rs.getString("imagen"));
    }
    ps.close();

} catch (SQLException e) {
    System.out.println(e.toString());
}
return producto;
}

public void updateStock(int id, int stock) {
    String sql = "UPDATE producto SET stock=? WHERE idProducto=?";
    try {
        Conexion con = new Conexion();
        conexion = con.getConexion();
        ps = conexion.prepareStatement(sql);
        ps.setInt(1, stock);
        ps.setInt(2, id);
        ps.executeUpdate();
        ps.close();
    } catch (SQLException e) {
        System.out.println(e.toString());
    }
}

public List<Producto> getByDescripcion(String desc) {
    String sql = "SELECT * FROM producto WHERE descripcion LIKE '%" + desc + "%'";
```

```
List<Producto> productos = new ArrayList<>();
try {
    Conexion con = new Conexion();
    conexion = con.getConexion();
    ps = conexion.prepareStatement(sql);
    rs = ps.executeQuery();

    while (rs.next()) {
        Producto p = new Producto();
        p.setIdProducto(rs.getInt("idProducto"));
        p.setDescripcion(rs.getString("descripcion"));
        p.setPrecio(rs.getDouble("precio"));
        p.setIdTipo(rs.getInt("idTipo"));
        p.setStock(rs.getInt("stock"));
        p.setImagen(rs.getString("imagen"));
        productos.add(p);
    }
    ps.close();
} catch (SQLException e) {
    System.out.println(e.toString());
}

return productos;
}
```

```
public void update(Producto p){
    String sql="UPDATE producto SET descripcion=?, precio=?, idTipo=?, stock=? WHERE
idProducto=?";
    try{
        Conexion con=new Conexion();
        conexion=con.getConexion();
        ps=conexion.prepareStatement(sql);
        ps.setString(1, p.getDescripcion());
        ps.setDouble(2, p.getPrecio());
        ps.setInt(3, p.getIdTipo());
        ps.setInt(4, p.getStock());
        ps.setInt(5, p.getIdProducto());
        ps.executeUpdate();
        ps.close();
    } catch(SQLException e){
        System.out.println(e.toString());
    }
}
```

```
}

public boolean delete(int id) {
    String sql="DELETE FROM producto WHERE idProducto=?;";
    try {
        Conexion con=new Conexion();
        conexion=con.getConexion();
        ps=conexion.prepareStatement(sql);
        ps.setInt(1, id);
        ps.executeUpdate();
        ps.close();
        return true;

    } catch(SQLException e) {
        System.out.println(e.toString());
        return false;
    }

}

}
```

ControladorCarrito.java

```
package controlador;
import java.io.IOException;
import java.text.NumberFormat;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;
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 javax.servlet.http.HttpSession;
import modelo.Carrito;
import modelo.ProductoDAO;
import modelo.Producto;

@WebServlet("/ControladorCarrito")
public class ControladorCarrito extends HttpServlet {
    private static final long serialVersionUID = 1L;
    ProductoDAO pDAO = new ProductoDAO();
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    }
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String menu = request.getParameter("menu");
        String accion = request.getParameter("accion");
        HttpSession session = request.getSession();
        if (menu.equals("carrito")) {
            switch (accion) {
                case "agregar": {
                    ArrayList<Carrito> carritoLista = new ArrayList<>();
                    Carrito carrito = new Carrito();
                    int id = Integer.parseInt(request.getParameter("id"));
                    Producto producto = pDAO.getById(id);
                    carrito.setIdProducto(id);
                    carrito.setCantidad(1);
                    carrito.setStock(producto.getStock());
                    carrito.setPrecio(producto.getPrecio());
                    ArrayList<Carrito> carrito_lista = (ArrayList<Carrito>)
session.getAttribute("carrito-lista");
                    // Primer producto al carrito
                    if (carrito_lista == null) {
                        carritoLista.add(carrito);
                        session.setAttribute("carrito-lista", carritoLista);
                        request.setAttribute("mensajeAgregado", "Producto
agregado al carrito");
                        request.getRequestDispatcher("index.jsp").forward(request,
response);
                    }
                }
            }
        }
    }
}
```

```

    } // Ya existe la lista carrito
    else {
        carritoLista = carrito_lista;
        boolean existe = false;
        // Compruebo que el producto no este en el carrito
        for (Carrito c : carritoLista) {
            if (c.getIdProducto() == id) {
                existe = true;
            }
        }
        // Si el producto no existe en la lista, lo agrego
        if (!existe) {
            carritoLista.add(carrito);
            request.setAttribute("mensajeAgregado", "Producto
agregado al carrito");

request.getRequestDispatcher("index.jsp").forward(request, response);
        } // Producto ya se encuentra en el carrito
        else {
            request.setAttribute("mensajeExiste", "Producto ya
se encuentra en el carrito");

request.getRequestDispatcher("index.jsp").forward(request, response);
        }
    }
    break;
}
case "carrito": {
    ArrayList<Carrito> carrito_lista = (ArrayList<Carrito>)
session.getAttribute("carrito-lista");
    List<Carrito> carritoProducto = null;
    double total = 0;
    if (carrito_lista != null) {
        ProductoDAO pDAO = new ProductoDAO();
        carritoProducto = pDAO.getCarritoProductos(carrito_lista);
        total = pDAO.getTotal(carrito_lista);
        request.setAttribute("carrito_lista", carrito_lista);
        request.setAttribute("carritoProducto", carritoProducto);
        NumberFormat formatoDecimales =
NumberFormat.getCurrencyInstance(Locale.US);
        formatoDecimales.setMaximumFractionDigits(2);
        String totalFormateado = formatoDecimales.format(total);
        request.setAttribute("total", totalFormateado);
    } else {
        request.setAttribute("total", total);
    }

    request.getRequestDispatcher("carrito.jsp").forward(request,
response);
    break;
}

```

```

    }
    case "incrementar": {
        int id = Integer.parseInt(request.getParameter("id"));
        ArrayList<Carrito> carrito_lista = (ArrayList<Carrito>)
request.getSession().getAttribute("carrito-lista");
        if (carrito_lista != null) {
            for (Carrito c : carrito_lista) {
                if (id == c.getIdProducto()) {
                    Producto producto = pDAO.getById(id);
                    c.setStock(producto.getStock());
                    if (c.getStock() < c.getCantidad() + 1) {
                        request.setAttribute("mensaje",
                            "La cant pedida
excede el stock disponible. Stock: " + c.getStock());

request.getRequestDispatcher("ControladorCarrito?menu=carrito&accion=carrito")
                            .forward(request,
response);

                                break;
                            } else {
                                c.setCantidad(c.getCantidad() + 1);
                                break;
                            }
                        }
                    }
                }
            }
        }

request.getRequestDispatcher("ControladorCarrito?menu=carrito&accion=carrito").forward(request,
response);

        break;
    }
    case "decrementar": {
        int id = Integer.parseInt(request.getParameter("id"));
        ArrayList<Carrito> carrito_lista = (ArrayList<Carrito>)
request.getSession().getAttribute("carrito-lista");
        if (carrito_lista != null) {
            for (Carrito c : carrito_lista) {
                if (id == c.getIdProducto()) {
                    if (c.getCantidad() > 1) {
                        c.setCantidad(c.getCantidad() - 1);
                    }
                }
            }
        }

request.getRequestDispatcher("ControladorCarrito?menu=carrito&accion=carrito").forward(request,
response);

        break;
    }
    case "eliminar": {

```



```

        int id = Integer.parseInt(request.getParameter("id"));
        ArrayList<Carrito> carrito_lista = (ArrayList<Carrito>)
request.getSession().getAttribute("carrito-lista");
        if (carrito_lista != null) {
            for (Carrito c : carrito_lista) {
                if (id == c.getIdProducto()) {
                    carrito_lista.remove(carrito_lista.indexOf(c));
                    break;
                }
            }
        }

request.getRequestDispatcher("ControladorCarrito?menu=carrito&accion=carrito").forward(request,
response);

        break;
    }
}

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
}

@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
}

```

Carrito.java

```

package modelo;

public class Carrito extends Producto {
    private int cantidad;

    public Carrito(){
    }

    public int getCantidad() {
        return cantidad;
    }

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

```

3-Cliente accede al carrito de compras y define unidades de cada producto a comprar. Sistema valida cantidades seleccionadas.

Carrito.jsp:

```
<%@page import="java.text.NumberFormat"%>
<%@page import="java.util.Locale"%>
<%@page import="modelo.TipoProductoDAO"%>
<%@page import="modelo.TipoProducto"%>
<%@page import="modelo.ProductoDAO"%>
<%@page import="java.util.List"%>
<%@page import="modelo.Carrito"%>
<%@page import="java.util.ArrayList"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Carrito</title>
<%@include file="estructura/head.jsp"%>
<style type="text/css">
table tbody td {
    vertical-align: middle;
}
.btn-incre, .btn-decre {
    box-shadow: none;
    font-size: 25px;
}
</style>
</head>
<body>
    <%@include file="estructura/nav.jsp"%>
    <div class="container">
        <%
            if (request.getAttribute("mensaje") != null) {
                <div class="alert alert-danger alert-dismissible fade show"
                    role="alert">
                        <strong><%=request.getAttribute("mensaje")%></strong>
                        <%request.setAttribute("mensaje", null);%>
                        <button type="button" class="btn-close" data-bs-dismiss="alert"
                            aria-label="Close"></button>
                    </div>
                <%
            }
        %>

        <%
            if (request.getSession().getAttribute("mensaje") != null) {
```

```

%>
<div class="alert alert-danger alert-dismissible fade show"
    role="alert">
    <strong><%=request.getSession().getAttribute("mensaje")%></strong>
    <%request.getSession().setAttribute("mensaje", null);%>
    <button type="button" class="btn-close" data-bs-dismiss="alert"
        aria-label="Close"></button>
</div>
<%
}
%>

<div class="d-flex py-3">
    <h3>Total:$ ${total}</h3>
    <a class="mx-3 btn btn-primary"
href="ControladorVenta?menu=venta&accion=nuevaVenta">Confirmar
        pedido</a>
</div>
<table class="table table-loght">
    <thead>
        <tr>
            <th scope="col">DESCRIPCION</th>
            <th scope="col">CATEGORIA</th>
            <th scope="col">PRECIO</th>
            <th scope="col">CANTIDAD</th>
            <th scope="col">ACCION</th>
        </tr>
    </thead>
    <tbody>
        <%
            ArrayList<Carrito> carrito_lista = (ArrayList<Carrito>)
request.getAttribute("carrito_lista");
            ArrayList<Carrito> carritoProducto = (ArrayList<Carrito>)
request.getAttribute("carritoProducto");
            TipoProductoDAO tpDAO = new TipoProductoDAO();
            if (carrito_lista != null) {
                for (Carrito c : carritoProducto) {
                    TipoProducto tp = new TipoProducto();
                    tp = tpDAO.getByld(c.getldTipo());
                }
            }
        <%>
        <tr>
            <td><%=c.getDescripcion()%></td>
            <td><%=tp.getDescripcion()%></td>
            <td><%=c.getPrecio()*c.getCantidad()%></td>
            <td>
                <form action="" method="post"
class="form-inline">
                    <input type="hidden" name="id"
value="<%=c.getldProducto()%>"

```

```

class="form-input">
        <div
            class="form-group d-flex
align-items-center justify-content-between">
            <a class="btn btn-sm btn-decre"
href="ControladorCarrito?menu=carrito&accion=decrementar&id=<%=c.getIdProducto()%>"><i
i
            class="fas
fa-minus-square"></i></a> <input type="text"
            name="cantidad"
class="form-control text-center"
value="<%=c.getCantidad()%>" style="width: 100px" readonly>
            <a class="btn btn-sm btn-incre"
href="ControladorCarrito?menu=carrito&accion=incrementar&id=<%=c.getIdProducto()%>"><i
            class="fas
fa-plus-square"></i></a>
        </div>
    </form>
</td>
<td><a class="btn btn-sm btn-danger"
href="ControladorCarrito?menu=carrito&accion=eliminar&id=<%=c.getIdProducto()%>">Elimi
nar</a>
    </td>
</tr>
<%
}
}
%>
</tbody>
</table>
</div>
<%@include file="estructura/footer.jsp"%>
</body>
</html>

```

ControladorCarrito.java

4-Cliente confirma pedido.

Sistema valida que el cliente se encuentre logueado, valida que haya al menos un producto en el carrito, valida nuevamente stock de cada producto, registra venta e informa al Cliente.

ControladorVenta.java

```
package controlador;
import java.io.IOException;
import java.time.LocalDateTime;
import java.util.ArrayList;
import java.util.List;
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 modelo.Carrito;
import modelo.Cliente;
import modelo.ClienteDAO;
import modelo.LineaVenta;
import modelo.LineaVentaDAO;
import modelo.Venta;
import modelo.VentaDAO;
import modelo.ProductoDAO;
import modelo.Producto;
@WebServlet("/ControladorVenta")
public class ControladorVenta extends HttpServlet {
    private static final long serialVersionUID = 1L;
    VentaDAO vDAO = new VentaDAO();
    LineaVentaDAO lvDAO = new LineaVentaDAO();
    ProductoDAO pDAO = new ProductoDAO();
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    }
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String menu = request.getParameter("menu");
        String accion = request.getParameter("accion");
        if (menu.equals("venta")) {
            switch (accion) {

                case "venta":{
                    if(request.getSession().getAttribute("authEmpleado")!=null){
                        String buscador=request.getParameter("buscador");
                        request.setAttribute("buscador", buscador);
                        System.out.println("Contenido buscador: "+buscador);
```

```

        request.getRequestDispatcher("admin/ventas.jsp").forward(request, response);

        break;
    } else {
        request.getRequestDispatcher("Admin").forward(request, response);
        break;
    }
}

case "nuevaVenta": {
    if (request.getSession().getAttribute("carrito-lista") == null) {
        request.getSession().setAttribute("mensaje", "Debe ingresar al menos un producto en
el carrito");
        response.sendRedirect("ControladorCarrito?menu=carrito&accion=carrito");
        break;
    }
    if (request.getSession().getAttribute("auth") == null) {
        response.sendRedirect("ControladorLogin?menu=login&accion=login");
        break;
    }
}

//Controlo Stock nuevamente
ArrayList<Carrito> lista_carrito = (ArrayList<Carrito>)
request.getSession().getAttribute("carrito-lista");
for(Carrito item : lista_carrito) {
    Producto producto = pDAO.getByld(item.getIdProducto());
    item.setStock(producto.getStock());
    if(item.getCantidad()>item.getStock()) {
        System.out.println("LA CANTIDAD EXECEDEEEEEEEEEEE");
        request.getSession().setAttribute("mensaje", "La cantidad pedida excede el
stock disponible en el producto: "+item.getDescripcion());
        response.sendRedirect("ControladorCarrito?menu=carrito&accion=carrito");
        return;
    }
}

Venta venta = new Venta();
LocalDateTime fecha = LocalDateTime.now();
venta.setFecha(fecha);
Cliente cliente = (Cliente) request.getSession().getAttribute("auth");
venta.setCliente(cliente);
int nroVenta = vDAO.getMaxNroVenta();
venta.setNroVenta(nroVenta);
if (vDAO.save(venta)) {

    for (Carrito item : lista_carrito) {

        LineaVenta lv = new LineaVenta();
        lv.setCantProducto(item.getCantidad());
        lv.setNroVenta(nroVenta);
    }
}

```

```

        lv.setProducto(item.getProducto());
        lv.setPrecio(item.getPrecio());
        lvDAO.save(lv);
        pDAO.updateStock(item.getIdProducto(), item.getStock() - item.getCantidad());
    }
    request.getSession().setAttribute("carrito-lista", null);
    request.getSession().setAttribute("mensajeExito", "Gracias por su compra! Su pedido
fue cargado exitosamente");
    response.sendRedirect("index.jsp");

    break;
} else {
    request.getSession().setAttribute("mensajeError", "Error al generar el pedido.Intente
nuevamente");
    response.sendRedirect("index.jsp");
    break;
}
}

case "detalle":{
    int id=Integer.parseInt(request.getParameter("id"));
    Venta venta=vDAO.getByid(id);
    ClienteDAO cDAO=new ClienteDAO();
    venta.setCliente(cDAO.getByid(venta.getCliente().getId()));
    venta.setLineas(lvDAO.getAllByVenta(venta.getNroVenta()));
    request.setAttribute("venta", venta);
    if(request.getSession().getAttribute("auth")!=null) {
request.getRequestDispatcher("detalleVentaCliente.jsp").forward(request, response);
        break;
    }

    if(request.getSession().getAttribute("authEmpleado")!=null) {
request.getRequestDispatcher("admin/detalleVenta.jsp").forward(request, response);
        break;
    }

    break;
}

case "misPedidos":{
    if (request.getSession().getAttribute("auth") != null) {
        Cliente cliente=(Cliente)request.getSession().getAttribute("auth");
        List<Venta> ventas=vDAO.getByCliente(cliente);
        for(Venta venta: ventas) {
            venta.setLineas(lvDAO.getAllByVenta(venta.getNroVenta()));

```

```
        }
        request.setAttribute("ventas", ventas);
        request.getRequestDispatcher("misPedidos.jsp").forward(request, response);

        break;
    } else {
        request.getRequestDispatcher("login.jsp").forward(request, response);
        break;
    }
    }
}

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    String cliente=request.getParameter("cliente");

    response.sendRedirect("ControladorVenta?menu=venta&accion=venta&buscador="+cliente);
}

@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
}
```


Venta.java

```
package modelo;

import java.time.LocalDateTime;
import java.util.ArrayList;

public class Venta {
    private int id;
    private LocalDateTime fecha;
    private Cliente cliente;
    private int nroVenta;
    private ArrayList<LineaVenta> lineas;

    public Venta() {
    }

    public int getId() {
        return id;
    }

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

    public LocalDateTime getFecha() {
        return fecha;
    }

    public void setFecha(LocalDateTime fecha) {
        this.fecha = fecha;
    }

    public int getNroVenta() {
        return nroVenta;
    }

    public void setNroVenta(int nroVenta) {
        this.nroVenta = nroVenta;
    }

    public ArrayList<LineaVenta> getLineas() {
        return lineas;
    }
}
```

```
public void setLineas(ArrayList<LineaVenta> lineas) {
    this.lineas = lineas;
}

public Cliente getCliente() {
    return cliente;
}

public void setCliente(Cliente cliente) {
    this.cliente = cliente;
}

public double getTotal() {
    double total=0;
    for(LineaVenta lv: lineas) {
        total=total+(lv.getPrecio()*lv.getCantProducto());
    }

    return total;
}

}
```

VentaDAO.java

```
package modelo;
```

```
import java.time.format.DateTimeFormatter;
```

```
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;
```

```
import config.Conexion;  
import java.time.LocalDateTime;
```

```
import java.util.ArrayList;  
import java.util.List;
```

```
public class VentaDAO {
```

```
    Connection conexion;  
    PreparedStatement ps;  
    ResultSet rs;  
    DateTimeFormatter formato = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");  
    LocalDateTime dateTime;
```

```
    public VentaDAO() {
```

```
    }
```

```
    public boolean save(Venta v) {
```

```
        String sql = "INSERT INTO venta(fechaVenta, idCliente, nroVenta) VALUES(?,?,?)";
```

```
        try {
```

```
            Conexion con = new Conexion();  
            conexion = con.getConexion();  
            ps = conexion.prepareStatement(sql);
```

```
            ps.setString(1, v.getFecha().format(formato));  
            ps.setInt(2, v.getId());  
            ps.setInt(3, v.getNroVenta());  
            ps.executeUpdate();  
            ps.close();  
            return true;
```

```
        } catch (SQLException e) {  
            System.out.println(e.toString());  
            return false;
```

```
    }  
}  
  
public List<Venta> getAll() {  
    String sql = "SELECT * FROM venta";  
    List<Venta> ventas = new ArrayList<>();  
    try {  
        Conexion con = new Conexion();  
        conexion = con.getConexion();  
        ps = conexion.prepareStatement(sql);  
        rs = ps.executeQuery();  
  
        while (rs.next()) {  
            Venta v = new Venta();  
            v.setId(rs.getInt(1));  
            LocalDateTime fechaVenta = LocalDateTime.parse(rs.getString(2), formato);  
  
            v.setFecha(fechaVenta);  
            Cliente cliente=new Cliente();  
            cliente.setId(rs.getInt(3));  
            v.setCliente(cliente);  
            v.setNroVenta(rs.getInt("nroVenta"));  
            ventas.add(v);  
  
        }  
        ps.close();  
    } catch (SQLException e) {  
        System.out.println(e.toString());  
    }  
    return ventas;  
}  
  
public Venta getById(int id) {  
    String sql = "SELECT * FROM venta WHERE idVenta=?";  
    LocalDateTime fecha;  
    Venta v = new Venta();  
    try {  
        Conexion con = new Conexion();  
        conexion = con.getConexion();  
        ps = conexion.prepareStatement(sql);  
        ps.setInt(1, id);  
        rs = ps.executeQuery();  
  
        while (rs.next()) {  
            v.setId(rs.getInt("idVenta"));
```

```
        fecha = LocalDateTime.parse(rs.getString("fechaVenta"), formato);
        v.setFecha(fecha);
        Cliente cliente=new Cliente();
        cliente.setId(rs.getInt("idCliente"));
        v.setCliente(cliente);
        v.setNroVenta(rs.getInt("nroVenta"));

    }
    ps.close();
} catch (SQLException e) {
    System.out.println("Error al traer venta: "+e.toString());
}
return v;
}

public void delete(int id) {
    String sql = "DELETE FROM venta WHERE idVenta=?";
    try {
        Conexion con = new Conexion();
        conexion = con.getConexion();
        ps = conexion.prepareStatement(sql);
        ps.setInt(1, id);
        ps.executeUpdate();
        ps.close();
    } catch (SQLException e) {
        System.out.println(e.toString());
    }
}

public int getMaxNroVenta() {
    String sql = "SELECT MAX(nroVenta) AS nroVenta FROM venta";
    int nroVenta = 1;

    try {
        Conexion con = new Conexion();
        conexion = con.getConexion();
        ps = conexion.prepareStatement(sql);
        rs = ps.executeQuery();

        while (rs.next()) {

            nroVenta = rs.getInt("nroVenta");
        }
        ps.close();
    }
```

```
} catch (SQLException e) {
    System.out.println("Error getMaxNroVenta: "+e.toString());
}
return (nroVenta + 1);
}

public List<Venta> getByEmail(String email){
    String sql="SELECT * FROM venta INNER JOIN cliente ON
venta.idCliente=cliente.idCliente WHERE email LIKE '%" +email+ "%' ";
    List<Venta> ventas=new ArrayList<>();
    LocalDateTime fecha;
    try {
        Conexion con = new Conexion();
        conexion = con.getConexion();
        ps = conexion.prepareStatement(sql);
        rs=ps.executeQuery();
        while(rs.next()) {
            Venta venta=new Venta();
            venta.setId(rs.getInt("idVenta"));
            fecha=LocalDateTime.parse(rs.getString("fechaVenta"), formato);

            venta.setFecha(fecha);
            Cliente cliente=new Cliente();
            cliente.setId(rs.getInt(3));
            venta.setCliente(cliente);
            venta.setNroVenta(rs.getInt("nroVenta"));
            ventas.add(venta);

        }
        ps.close();

    } catch(SQLException e) {
        System.out.println(e.toString());
    }
    return ventas;
}

public List<Venta> getByCliente(Cliente cliente) {
```

```
String sql = "SELECT * FROM venta WHERE idCliente=?;";
List<Venta> ventas = new ArrayList<>();
try {
    Conexion con = new Conexion();
    conexion = con.getConexion();
    ps = conexion.prepareStatement(sql);
    ps.setInt(1, cliente.getId());
    rs = ps.executeQuery();

    while (rs.next()) {
        Venta v = new Venta();
        v.setId(rs.getInt(1));
        LocalDateTime fechaVenta = LocalDateTime.parse(rs.getString(2), formato);

        v.setFecha(fechaVenta);

        v.setCliente(cliente);
        v.setNroVenta(rs.getInt("nroVenta"));
        ventas.add(v);
    }
    ps.close();
} catch (SQLException e) {
    System.out.println(e.toString());
}
return ventas;
}
}
```

LineaVenta.java

```
package modelo;

public class LineaVenta {
    private int idLineaVenta;
    private int cantProducto;
    private int nroVenta;
    private int idProducto;
    private double precio;

    public LineaVenta(){
    };
    public int getIdLineaVenta() {
        return idLineaVenta;
    }
    public void setIdLineaVenta(int idLineaVenta) {
        this.idLineaVenta = idLineaVenta;
    }
    public int getCantProducto() {
        return cantProducto;
    }
    public void setCantProducto(int cantProducto) {
        this.cantProducto = cantProducto;
    }
    public int getNroVenta() {
        return nroVenta;
    }
    public void setNroVenta(int idVenta) {
        this.nroVenta = idVenta;
    }
    public int getIdProducto() {
        return idProducto;
    }
    public void setIdProducto(int idProducto) {
        this.idProducto = idProducto;
    }
    public double getPrecio() {
        return precio;
    }
    public void setPrecio(double precio) {
        this.precio = precio;
    }
}
```


LineaVentaDAO.java

```
package modelo;
```

```
import config.Conexion;  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.util.ArrayList;
```

```
public class LineaVentaDAO {
```

```
    Connection conexion;  
    PreparedStatement ps;  
    ResultSet rs;
```

```
    public LineaVentaDAO() {  
  
    }  
}
```

```
    public void save(LineaVenta lv) {  
        String sql = "INSERT INTO linea_de_venta(cantProducto,nroVenta,idProducto, precio)  
VALUES(?,?,?,?); ";  
        try {  
            Conexion con = new Conexion();  
            conexion = con.getConexion();  
            ps = conexion.prepareStatement(sql);  
            ps.setInt(1, lv.getCantProducto());  
            ps.setInt(2, lv.getNroVenta());  
            ps.setInt(3, lv.getIdProducto());  
            ps.setDouble(4, lv.getPrecio());  
            ps.executeUpdate();  
            ps.close();  
        } catch (SQLException e) {  
            System.out.println(e.toString());  
        }  
    }  
}
```

```
    public ArrayList<LineaVenta> getAllByVenta(int nroVenta){  
        String sql="SELECT * FROM linea_de_venta WHERE nroVenta=?;";
```

```
ArrayList<LineaVenta> lineas= new ArrayList<>();
try {
    Conexion con=new Conexion();
    conexion=con.getConexion();
    ps=conexion.prepareStatement(sql);
    ps.setInt(1, nroVenta);
    rs=ps.executeQuery();
    while(rs.next()) {
        LineaVenta lv=new LineaVenta();
        lv.setIdLineaVenta(rs.getInt("idLineaVenta"));
        lv.setCantProducto(rs.getInt("cantProducto"));
        lv.setNroVenta(rs.getInt("nroVenta"));
        lv.setIdProducto(rs.getInt("idProducto"));
        lv.setPrecio(rs.getDouble("precio"));
        lineas.add(lv);
    }

} catch(SQLException e) {
    System.out.println(e.toString());
}

return lineas;

}

}
```