

BASE DE DATOS

```
package Configuracion;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

public class Conexion {

    Connection con;

    String url="jdbc:mysql://localhost:3306/final";

    String user="root";

    String pass="";

    PreparedStatement ps=null;

    public Connection getConnection(){

        try {

            Class.forName("com.mysql.cj.jdbc.Driver");

            con=DriverManager.getConnection(url,user,pass);

            System.out.println("Conectado");

        }

        catch (ClassNotFoundException ex) {

            Logger.getLogger(Conexion.class.getName()).log(Level.SEVERE, null, ex);

        } catch (SQLException ex) {

            Logger.getLogger(Conexion.class.getName()).log(Level.SEVERE, null, ex);

        }

        return con;

    }

}
```

CLIENTE

```
package DAO;

import Configuracion.Conexion;

import Modelo.Cliente;
import Modelo.Usuario;

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 con;

    Conexion cn = new Conexion();

    PreparedStatement ps;

    ResultSet rs;

    public boolean registrar(Cliente us) {

        String sql = "INSERT INTO cliente (Ruc,
RazonSocial,Direccion,Telefono,RepresentanteLegal,IdUsuarioRegistro,estado) VALUES (?, ?, ?, ?, ?, ?, ?)";

        con = cn.getConnection();

        try {

            ps = con.prepareStatement(sql);

            ps.setString(1, us.getRuc());

            ps.setString(2, us.getRazonSocial());

            ps.setString(3, us.getDireccion());

            ps.setString(4, us.getTelefono());

            ps.setString(5, us.getRepresentanteLegal());

            ps.setInt(6, us.getIdUsuario());

            ps.setInt(7, us.getEstado());

            ps.execute();

            return true;

        }

    }

}
```

```

    } catch (SQLException ex) {

        System.out.println(ex.toString());

        return false;

    }

}

```

```

public boolean actualizar(Cliente reg) {

    boolean res;

    String sql = "UPDATE cliente SET Ruc=?,RazonSocial=?,
Direccion=?,Telefono=?,RepresentanteLegal=?,estado=? WHERE idcliente = ?";

    con = cn.getConnection();

    try {

        ps = con.prepareStatement(sql);

        ps.setString(1, reg.getRuc());

        ps.setString(2, reg.getRazonSocial());

        ps.setString(3, reg.getDireccion());

        ps.setString(4, reg.getTelefono());

        ps.setString(5, reg.getRepresentanteLegal());

        ps.setInt(6, reg.getEstado());

        ps.setInt(7, reg.getIdCliente());

        ps.execute();

        res = true;

    } catch (SQLException ex) {

        System.out.println(ex.toString());

        res = false;

    }

    return res;

}

```

```

public List Listar(String valor) {

    List<Cliente> lista = new ArrayList();

    try {

        con = cn.getConnection();

        if ("".equalsIgnoreCase(valor)) {

```

```

        String sql = "SELECT * FROM cliente ORDER BY IdCliente DESC";

        ps = con.prepareStatement(sql);

    } else {

        String sql = "SELECT * FROM cliente WHERE ruc LIKE '%" + valor + "%' OR razonsocial LIKE '%" +
valor + "%'";

        ps = con.prepareStatement(sql);

    }

    rs = ps.executeQuery();

    while (rs.next()) {

        Cliente user = new Cliente();

        user.setIdCliente(rs.getInt("idcliente"));

        user.setRuc(rs.getString("ruc"));

        user.setRazonSocial(rs.getString("RazonSocial"));

        user.setDireccion(rs.getString("direccion"));

        user.setTelefono(rs.getString("Telefono"));

        user.setRepresentanteLegal(rs.getString("RepresentanteLegal"));

        user.setEstado(rs.getInt("estado"));

        lista.add(user);

    }

} catch (SQLException e) {

    System.out.println(e.toString());

}

return lista;

}

```

```

public List ListarAyuda(String valor) {

    List<Cliente> lista = new ArrayList();

    try {

        con = cn.getConnection();

        if ("".equalsIgnoreCase(valor)) {

            String sql = "SELECT * FROM cliente where estado<>0 ORDER BY IdCliente DESC";

            ps = con.prepareStatement(sql);

        } else {

```

```
String sql = "SELECT * FROM cliente WHERE ruc LIKE '%" + valor + "%' OR razonsocial LIKE '%" + valor + "%' and estado<>0";
```

```
    ps = con.prepareStatement(sql);
}
rs = ps.executeQuery();
while (rs.next()) {
    Cliente user = new Cliente();
    user.setIdCliente(rs.getInt("idcliente"));
    user.setRuc(rs.getString("ruc"));
    user.setRazonSocial(rs.getString("RazonSocial"));
    user.setDireccion(rs.getString("direccion"));
    user.setTelefono(rs.getString("Telefono"));
    user.setRepresentanteLegal(rs.getString("RepresentanteLegal"));
    user.setEstado(rs.getInt("estado"));
    lista.add(user);
}
} catch (SQLException e) {
    System.out.println(e.toString());
}
return lista;
}
```

```
public int ExisteCliente( String Id) {
int resultado = 0;
Connection conexion = null;
```

```
String sql = "select count(*) as valor from cliente where ruc='" + Id + "'";
con = cn.getConnection();
try {
```

```
    ps = con.prepareStatement(sql);
```

```
    rs = ps.executeQuery();
```

```
    if (rs.next()) {
```

```
        if (rs.getRow() > 0) {  
            resultado =rs.getInt("valor");  
  
        }  
    }  
    conexion.close();  
} catch (Exception e) {  
    System.out.println(e);  
}  
return resultado;  
}  
}
```

DESTAJERO

```
package DAO;
```

```
import Configuracion.Conexion;
```

```
import Modelo.Cliente;
```

```
import Modelo.Destajero;
```

```
import java.sql.CallableStatement;
```

```
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 DestajeroDAO {
```

```
    Connection con;
```

```
    Conexion cn = new Conexion();
```

```
    PreparedStatement ps;
```

```
    ResultSet rs;
```

```
    public boolean registrar(Destajero us) {
```

```
        String sql = "INSERT INTO empresaDestajero (Ruc,  
RazonSocial,Direccion,Telefono,RepresentanteLegal,IdUsuarioRegistro,estado) VALUES (?, ?, ?, ?, ?, ?, ?)";
```

```
        con = cn.getConnection();
```

```
        try {
```

```
            ps = con.prepareStatement(sql);
```

```
            ps.setString(1, us.getRuc());
```

```
            ps.setString(2, us.getRazonSocial());
```

```
            ps.setString(3, us.getDireccion());
```

```
            ps.setString(4, us.getTelefono());
```

```
            ps.setString(5, us.getRepresentanteLegal());
```

```
            ps.setInt(6, us.getIdUsuario());
```

```
            ps.setInt(7, us.getEstado());
```

```

        ps.execute();

        return true;
    } catch (SQLException ex) {

        System.out.println(ex.toString());

        return false;
    }
}

```

```

public boolean actualizar(Destajero reg) {

    boolean res;

    String sql = "UPDATE EmpresaDestajero SET Ruc=?,RazonSocial=?,
Direccion=?,Telefono=?,RepresentanteLegal=? WHERE idEmpresaDestajero = ?";

    con = cn.getConnection();

    try {

        ps = con.prepareStatement(sql);

        ps.setString(1, reg.getRuc());

        ps.setString(2, reg.getRazonSocial());

        ps.setString(3, reg.getDireccion());

        ps.setString(4, reg.getTelefono());

        ps.setString(5, reg.getRepresentanteLegal());

        ps.setInt(6, reg.getIdDestajero());

        ps.execute();

        res = true;
    } catch (SQLException ex) {

        System.out.println(ex.toString());

        res = false;
    }

    return res;
}

```

```

public List Listar(String valor) {

    List<Destajero> lista = new ArrayList();

    try {

```



```

con = cn.getConnection();

if ("".equalsIgnoreCase(valor)) {

    String sql = "SELECT * FROM EmpresaDestajero ORDER BY IdEmpresaDestajero DESC";

    ps = con.prepareStatement(sql);

} else {

    String sql = "SELECT * FROM EmpresaDestajero WHERE ruc LIKE '%" + valor + "%' OR razonsocial LIKE '%" + valor + "%'";

    ps = con.prepareStatement(sql);

}

rs = ps.executeQuery();

while (rs.next()) {

    Destajero user = new Destajero();

    user.setIdDestajero(rs.getInt("IdEmpresaDestajero"));

    user.setRuc(rs.getString("ruc"));

    user.setRazonSocial(rs.getString("RazonSocial"));

    user.setDireccion(rs.getString("direccion"));

    user.setTelefono(rs.getString("Telefono"));

    user.setRepresentanteLegal(rs.getString("RepresentanteLegal"));

    user.setEstado(rs.getInt("estado"));

    lista.add(user);

}

} catch (SQLException e) {

    System.out.println(e.toString());

}

return lista;

}

public List ListarAyudaDestajero(String valor) {

    List<Destajero> lista = new ArrayList();

    try {

        con = cn.getConnection();

        if ("".equalsIgnoreCase(valor)) {

            String sql = "SELECT IdEmpresaDestajero,RUC,RazonSocial,estado, CASE WHEN Estado=1 THEN 'Libre' ELSE 'Ocupado' END AS EstadoNombre FROM `empresadestajero` where estado<>0 ORDER BY IdEmpresaDestajero DESC";

```

```

        ps = con.prepareStatement(sql);

    } else {

        String sql = "SELECT IdEmpresaDestajero,RUC,RazonSocial,estado, CASE WHEN Estado=1 THEN
'Libre' ELSE 'Ocupado' END AS EstadoNombre FROM `empresadestajero` where estado<>0 and ruc LIKE
'%" + valor + "%' OR razonsocial LIKE '%" + valor + "%'";

        ps = con.prepareStatement(sql);

    }

    rs = ps.executeQuery();
    while (rs.next()) {

        Destajero user = new Destajero();

        user.setIdDestajero(rs.getInt("IdEmpresaDestajero"));

        user.setRuc(rs.getString("ruc"));

        user.setRazonSocial(rs.getString("RazonSocial"));

        user.setEstadoNombre(rs.getString("EstadoNombre"));

        lista.add(user);

    }

} catch (SQLException e) {

    System.out.println(e.toString());

}

return lista;

}

```

```

public void actualizarEstadoDestajero(String Id,String estado){

    try{

        con = cn.getConnection();

        CallableStatement statement=con.prepareCall("{call
SP_ActualizarEstadoEmpresaDestajero(?,?)");

        statement.setString("piddestajero",Id);

        statement.setString("pldEstado",estado);

        statement.executeUpdate();

    }catch(SQLException ex){

        ex.printStackTrace();

    }

}

```

```

    }

    public int ExisteDestajero( String Id) {

        int resultado = 0;

        Connection conexion = null;

        String sql = "select count(*) as valor from empresadestajero where ruc='" + Id + "'";

        con = cn.getConnection();

        try {

            ps = con.prepareStatement(sql);

            rs = ps.executeQuery();

            if (rs.next()) {

                if (rs.getRow() > 0) {

                    resultado =rs.getInt("valor");

                }

            }

            conexion.close();

        } catch (Exception e) {

            System.out.println(e);

        }

        return resultado;

    }

}

```

MODELO PRODUCTO

```
package DAO;
```

```
import Configuracion.Conexion;
```

```
import Modelo.Producto;
```

```
import Modelo.ModeloProducto;
```

```
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 ModeloProductoDAO {
```

```
    Connection con;
```

```
    Conexion cn = new Conexion();
```

```
    PreparedStatement ps;
```

```
    ResultSet rs;
```

```
    public List ListarAyuda(String valor) {
```

```
        List<ModeloProducto> lista = new ArrayList();
```

```
        try {
```

```
            con = cn.getConnection();
```

```
            if ("".equalsIgnoreCase(valor)) {
```

```
                String sql = "SELECT * FROM Modelo where estado=1 ORDER BY IdModelo DESC";
```

```
                ps = con.prepareStatement(sql);
```

```
            } else {
```

```
                String sql = "SELECT * FROM Modelo WHERE nombre LIKE '%" + valor + "%' and estado=1";
```

```
                ps = con.prepareStatement(sql);
```

```
            }
```

```
            rs = ps.executeQuery();
```

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

```
                ModeloProducto user = new ModeloProducto();
```

```
        user.setIdModelo(rs.getInt("IdModelo"));
        user.setNombre(rs.getString("nombre"));
        user.setEstado(rs.getInt("estado"));
        lista.add(user);
    }
} catch (SQLException e) {
    System.out.println(e.toString());
}
return lista;
}
}
```

PEDIDO

```
package DAO;

import Configuracion.Conexion;
import Modelo.Cliente;
import Modelo.Pedido;
import Modelo.PedidoDetalle;
import java.sql.CallableStatement;
import java.sql.Connection;
import java.util.Date;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.List;
import javax.accessibility.AccessibleRole;
import javax.swing.JOptionPane;

public class PedidoDAO {
    Connection con;
    Conexion cn = new Conexion();
    PreparedStatement ps;
    ResultSet rs;
    public List Listar(String valor) {
        List<Pedido> lista = new ArrayList();
        try {
            con = cn.getConnection();
            if ("".equalsIgnoreCase(valor)) {
```

```
String sql = "SELECT p.IdPedido,p.Comentario,c.IdCliente,c.RazonSocial as  
Cliente,p.fecha,p.fechaDevolucion,p.TotalGeneral ,p.Estado FROM Pedido p inner join cliente c  
on p.IdCliente=c.IdCliente ORDER BY p.IdPedido DESC";
```

```
ps = con.prepareStatement(sql);
```

```
} else {
```

```
String sql = "SELECT p.IdPedido,p.Comentario,c.IdCliente,c.RazonSocial as  
Cliente,p.fecha,p.fechaDevolucion,p.TotalGeneral ,p.Estado FROM Pedido p inner join cliente c  
on p.IdCliente=c.IdCliente WHERE c.razonsocial LIKE '%" + valor + "%'";
```

```
ps = con.prepareStatement(sql);
```

```
}
```

```
rs = ps.executeQuery();
```

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

```
Pedido user = new Pedido();
```

```
user.setIdPedido(rs.getInt("IdPedido"));
```

```
user.setDescripcion(rs.getString("Comentario"));
```

```
user.setIdCliente(rs.getInt("IdCliente"));
```

```
user.setCliente(rs.getString("Cliente"));
```

```
user.setFecha(rs.getString("fecha"));
```

```
user.setFechaDevolucion(rs.getString("fechaDevolucion"));
```

```
user.setTotalGeneral(rs.getString("TotalGeneral"));
```

```
user.setEstado(rs.getInt("estado"));
```

```
lista.add(user);
```

```
}
```

```
} catch (SQLException e) {
```

```
System.out.println(e.toString());
```

```
}
```

```
return lista;
```

```
}
```

```
public List ListarAyudaPedido(String valor) {
```

```
List<Pedido> lista = new ArrayList();
```

```
try {
```

```
con = cn.getConnection();
```

```

        if ("".equalsIgnoreCase(valor)) {

            String sql = "select p.IdPedido,p.Comentario,c.IdCliente,c.RazonSocial as
Cliente,p.Estado, case when p.Estado=1 then 'PENDIENTE' when p.estado=2 THEN 'EN
PROCESO' WHEN P.Estado=3 THEN 'CULMINADO' WHEN P.Estado=4 THEN 'ENTREGADO' END
AS EstadoNombre FROM Pedido p inner join cliente c on p.IdCliente=c.IdCliente where
p.estado<>0 ORDER BY p.IdPedido DESC";

            ps = con.prepareStatement(sql);

        } else {

            String sql = "select p.IdPedido,p.Comentario,c.IdCliente,c.RazonSocial as
Cliente,p.Estado, case when p.Estado=1 then 'PENDIENTE' when p.estado=2 THEN 'EN
PROCESO' WHEN P.Estado=3 THEN 'CULMINADO' WHEN P.Estado=4 THEN 'ENTREGADO' END
AS EstadoNombre FROM Pedido p inner join cliente c on p.IdCliente=c.IdCliente WHERE
p.estado<>0 and c.razonsocial LIKE '%" + valor + "%'";

            ps = con.prepareStatement(sql);

        }

        rs = ps.executeQuery();

        while (rs.next()) {

            Pedido user = new Pedido();

            user.setIdPedido(rs.getInt("IdPedido"));

            user.setDescripcion(rs.getString("Comentario"));

            user.setCliente(rs.getString("Cliente"));

            user.setEstado(rs.getInt("estado"));

            user.setEstadoNombre(rs.getString("EstadoNombre"));

            lista.add(user);

        }

    } catch (SQLException e) {

        System.out.println(e.toString());

    }

    return lista;

}

public List ListarPedidoProducto(String valor) {

    List<PedidoDetalle> lista = new ArrayList();

    try {

        con = cn.getConnection();

```



```
String sql = "SELECT p.IdPedido,m.Nombre as
ProductoNombre,p.Cantidad,p.precio,p.Total FROM pedidodetalle p inner JOIN producto pr on
p.IdProducto=pr.IdProducto inner join modelo m on pr.idmodelo=m.idmodelo WHERE
p.idpedido = '" + valor + "'";
```

```
ps = con.prepareStatement(sql);
```

```
rs = ps.executeQuery();
```

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

```
PedidoDetalle user = new PedidoDetalle();
```

```
user.setIdPedido(rs.getInt("IdPedido"));
```

```
user.setProducto(rs.getString("ProductoNombre"));
```

```
user.setCantidad(rs.getString("Cantidad"));
```

```
user.setPrecio(rs.getString("Precio"));
```

```
user.setTotal(rs.getString("Total"));
```

```
lista.add(user);
```

```
}
```

```
} catch (SQLException e) {
```

```
System.out.println(e.toString());
```

```
}
```

```
return lista;
```

```
}
```

```
public List ObtenerPedidoProducto(String valor) {
```

```
List<PedidoDetalle> lista = new ArrayList();
```

```
try {
```

```
con = cn.getConnection();
```

```
String sql = "SELECT p.IdPedido,pr.IdProducto ,m.Nombre as
ProductoNombre,p.Cantidad,pr.PrecioElaboracionDestajero,p.cantidad *
pr.PrecioElaboracionDestajero as Total FROM pedidodetalle p inner JOIN producto pr on
p.IdProducto=pr.IdProducto inner join modelo m on pr.idmodelo=m.idmodelo WHERE
p.idpedido = '" + valor + "'";
```

```

        ps = con.prepareStatement(sql);

        rs = ps.executeQuery();
        while (rs.next()) {
            PedidoDetalle user = new PedidoDetalle();
            user.setIdPedido(rs.getInt("IdPedido"));
            user.setIdProducto(rs.getInt("IdProducto"));
            user.setProducto(rs.getString("ProductoNombre"));
            user.setCantidad(rs.getString("Cantidad"));
            user.setPrecio(rs.getString("PrecioElaboracionDestajero"));
            user.setTotal(rs.getString("Total"));

            lista.add(user);
        }
    } catch (SQLException e) {
        System.out.println(e.toString());
    }
    return lista;
}

public boolean registrarPedido(Pedido us) {
    String sql = "INSERT INTO pedido
(Comentario,IdCliente,Fecha,FechaDevolucion,TotalGeneral,IdUsuarioRegistro,estado) VALUES
(?,?,?,?,?,?,?)";

    con = cn.getConnection();
    try {

        ps = con.prepareStatement(sql);
        ps.setString(1,us.getDescripcion());
        ps.setInt(2, us.getIdCliente());
        ps.setString(3,us.getFecha() );
        ps.setString(4, us.getFechaDevolucion());
        ps.setString(5, us.getTotalGeneral());
    }
}

```

```

        ps.setInt(6, us.getIdUsuario());

        ps.setInt(7, us.getEstado());

        ps.execute();

        return true;
    } catch (SQLException ex) {

        System.out.println(ex.toString());

        return false;
    }
}

public ResultSet obtenerUltimoIdPedido() throws Exception{
    ResultSet rs = null;

    try{

        rs = ps.executeQuery("SELECT max(IdPedido) as IdPedido FROM pedido");

        return rs;

    }catch(SQLException SQLex){

        throw SQLex;

    }

}

```

```

    public int GuardarDetallePedido(String Idpedido, String IdProducto, String Cantidad, String
Precio, String Total) {

```

```

        int resultado = 0;

```

```

        Connection conexion = null;

```

```

        String sql = "INSERT INTO pedidodetalle (IdPedido, IdProducto, Cantidad,Precio, total)
VALUES (?, ?, ?, ?, ?)";

```

```

        con = cn.getConnection();

```

```

        try {

```

```

            ps = con.prepareStatement(sql);

```

```

            ps.setString(1, Idpedido);

```

```

            ps.setString(2, IdProducto);

```

```

        ps.setString(3, Cantidad);
        ps.setString(4, Precio);
        ps.setString(5, Total);
        resultado = ps.executeUpdate();
        if (resultado > 0) {

        }

        conexion.close();
    } catch (Exception e) {
        System.out.println(e);
    }
    return resultado;
}

```

```

public void actualizarEstadoPedido(String Id,String estado){
try{
    con = cn.getConnection();
    CallableStatement statement=con.prepareCall("{call SP_ActualizarEstadoPedido(?,?)}");
    statement.setString("pIdPedido",Id);
    statement.setString("pIdEstado",estado);
    statement.executeUpdate();

}catch(SQLException ex){
    ex.printStackTrace();
}

}

}

```

PRODUCTO

```
package DAO;
```

```
import Configuracion.Conexion;
```

```
import Modelo.Producto;
```

```
import java.sql.CallableStatement;
```

```
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 ProductoDAO {
```

```
    Connection con;
```

```
    Conexion cn = new Conexion();
```

```
    PreparedStatement ps;
```

```
    ResultSet rs;
```

```
    public boolean registrar(Producto us) {
```

```
        String sql = "INSERT INTO  
producto(IdModelo,Descripcion,CostoPrenda,PrecioFabricacion,PrecioElaboracionDestajero,St  
ock,IdUsuarioRegistro,estado) VALUES (?, ?, ?, ?, ?, ?, ?, ?)";
```

```
        con = cn.getConnection();
```

```
        try {
```

```
            ps = con.prepareStatement(sql);
```

```
            ps.setInt(1, us.getIdModelo());
```

```
            ps.setString(2, us.getDescripcion());
```

```
            ps.setString(3, us.getPrecio());
```

```
            ps.setDouble(4, us.getPrecioFabricacion());
```

```
            ps.setDouble(5, us.getPrecioElaboracionDestajero());
```

```

        ps.setString(6, us.getStock());

        ps.setInt(7, us.getIdUsuario());

        ps.setInt(8, us.getEstado());

        ps.execute();

        return true;
    } catch (SQLException ex) {

        System.out.println(ex.toString());

        return false;
    }
}

```

```

public boolean actualizar(Producto reg) {

    boolean res;

    String sql = "UPDATE producto SET
    IdModelo=?,Descripcion=?,CostoPrenda=?,PrecioFabricacion=?,PrecioElaboracionDestajero=?,
    Stock=?,estado=? WHERE idproducto = ?";

    con = cn.getConnection();

    try {

        ps = con.prepareStatement(sql);

        ps.setInt(1, reg.getIdModelo());

        ps.setString(2, reg.getDescripcion());

        ps.setString(3, reg.getPrecio());

        ps.setDouble(4, reg.getPrecioFabricacion());

        ps.setDouble(5, reg.getPrecioElaboracionDestajero());

        ps.setString(6, reg.getStock());

        ps.setInt(7, reg.getEstado());

        ps.setInt(8, reg.getIdProducto());

        ps.execute();

        res = true;
    } catch (SQLException ex) {

        System.out.println(ex.toString());

        res = false;
    }
}

```

```

    }

    return res;
}

public List Listar(String valor) {

    List<Producto> lista = new ArrayList();

    try {

        con = cn.getConnection();

        if ("".equalsIgnoreCase(valor)) {

            String sql = "SELECT
p.IdProducto,m.idmodelo,m.nombre,p.Descripcion,p.costoprenda,p.PrecioFabricacion,p.Precio
ElaboracionDestajero,p.stock,p.estado FROM Producto p inner join modelo m on
p.idmodelo=m.idmodelo ORDER BY p.IdProducto DESC";

            ps = con.prepareStatement(sql);

        } else {

            String sql = "SELECT
p.IdProducto,m.idmodelo,m.nombre,p.Descripcion,p.costoprenda,p.PrecioFabricacion,p.Precio
ElaboracionDestajero,p.stock,p.estado FROM Producto p inner join modelo m on
p.idmodelo=m.idmodelo WHERE m.nombre LIKE '%" + valor + "%'";

            ps = con.prepareStatement(sql);

        }

        rs = ps.executeQuery();

        while (rs.next()) {

            Producto user = new Producto();

            user.setIdProducto(rs.getInt("IdProducto"));

            user.setIdModelo(rs.getInt("IdModelo"));

            user.setModelo(rs.getString("nombre"));

            user.setDescripcion(rs.getString("Descripcion"));

            user.setPrecio(rs.getString("CostoPrenda"));

            user.setPrecioFabricacion(rs.getDouble("PrecioFabricacion"));

            user.setPrecioElaboracionDestajero(rs.getDouble("PrecioElaboracionDestajero"));

            user.setStock(rs.getString("stock"));

            user.setEstado(rs.getInt("estado"));

            lista.add(user);

```

```

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

public List ListarAyuda(String valor) {
    List<Producto> lista = new ArrayList();
    try {
        con = cn.getConnection();
        if ("".equalsIgnoreCase(valor)) {
            String sql = "SELECT
p.IdProducto,m.nombre,p.Descripcion,p.costoprenda,p.stock,p.estado FROM Producto p inner
join modelo m on p.idmodelo=m.idmodelo where p.estado<>0 ORDER BY p.IdProducto
DESC";
            ps = con.prepareStatement(sql);
        } else {
            String sql = "SELECT
p.IdProducto,m.nombre,p.Descripcion,p.costoprenda,p.stock,p.estado FROM Producto p inner
join modelo m on p.idmodelo=m.idmodelo WHERE m.nombre LIKE '%" + valor + "%' and
p.estado<>0 ";
            ps = con.prepareStatement(sql);
        }
        rs = ps.executeQuery();
        while (rs.next()) {
            Producto user = new Producto();
            user.setIdProducto(rs.getInt("IdProducto"));
            user.setModelo(rs.getString("nombre"));
            user.setDescripcion(rs.getString("Descripcion"));
            user.setPrecio(rs.getString("CostoPrenda"));
            user.setStock(rs.getString("stock"));
            user.setEstado(rs.getInt("estado"));
            lista.add(user);
        }
    }
}

```



```

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

public void actualizarProductoStock(String Id,String cantidad){
    try{
        con = cn.getConnection();

        CallableStatement statement=con.prepareCall("{call sp_aumentar_stock (?,?)}");
        statement.setString("pidproducto",Id);
        statement.setString("pcantidad",cantidad);
        statement.executeUpdate();

    }catch(SQLException ex){
        ex.printStackTrace();
    }
}
}

```

```

public int ExisteProductoModelo( String Id) {
    int resultado = 0;
    Connection conexion = null;

    String sql = "select * from producto where idmodelo=" + Id +"";
    con = cn.getConnection();
    try {

        ps = con.prepareStatement(sql);
        resultado = ps.executeUpdate();
        if (resultado > 0) {

```

```

    }

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

return resultado;
}

public void actualizarEstadoModeloProducto(String Id){
    try{
        con = cn.getConnection();

        CallableStatement statement=con.prepareCall("{call
sp_actualizar_estado_modelo_producto (?)}");

        statement.setString("pidmodelo",Id);

        statement.executeUpdate();

    }catch(SQLException ex){
        ex.printStackTrace();
    }
}
}

```

SERVICIO DE DESTAJERO

```
package DAO;
```

```
import Configuracion.Conexion;
```

```
import Modelo.Pedido;
```

```
import Modelo.ServicioDestajero;
```

```
import Modelo.ServicioDestajeroDetalle;
```

```
import java.sql.CallableStatement;
```

```
import java.sql.Connection;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
```

```
import java.sql.SQLException;
```

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

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

```
import javax.swing.JOptionPane;
```

```
public class ServicioDestajeroDAO {
```

```
    Connection con;
```

```
    Conexion cn = new Conexion();
```

```
    PreparedStatement ps;
```

```
    ResultSet rs;
```

```
    public List Listar(String valor) {
```

```
        List<ServicioDestajero> lista = new ArrayList();
```

```
        try {
```

```
            con = cn.getConnection();
```

```
            if ("".equalsIgnoreCase(valor)) {
```

```
                String sql = "SELECT
```

```
s.IdServicioDestajero,p.IdPedido,cl.razonsocial,e.IdEmpresaDestajero,e.RazonSocial as  
EmpresaDestajero,s.Fecha,s.FechaDevolucion,s.CantidadTotalEntregado,s.CantidadTotalCulmin  
ado,s.TotalPagar,s.Estado FROM serviciodestajero s inner join pedido p on
```

```

s.IdPedido=p.IdPedido inner join empresadestajero e on
s.IdEmpresDestajero=e.IdEmpresaDestajero inner join cliente cl on p.idcliente=cl.idcliente
ORDER BY s.IdServicioDestajero DESC";

        ps = con.prepareStatement(sql);

        } else {

            String sql = "SELECT
s.IdServicioDestajero,p.IdPedido,cl.razonsocial,e.IdEmpresaDestajero,e.RazonSocial as
EmpresaDestajero,s.Fecha,s.FechaDevolucion,s.CantidadTotalEntregado,s.CantidadTotalCulmin
ado,s.TotalPagar,s.Estado FROM serviciodestajero s inner join pedido p on
s.IdPedido=p.IdPedido inner join empresadestajero e on
s.IdEmpresDestajero=e.IdEmpresaDestajero inner join cliente cl on p.idcliente=cl.idcliente
WHERE p.Descripcion like '%" + valor + "%' or e.razonsocial LIKE '%" + valor + "%'";

            ps = con.prepareStatement(sql);

        }

        rs = ps.executeQuery();
        while (rs.next()) {

            ServicioDestajero user = new ServicioDestajero();

            user.setIdServicioDestajero(rs.getInt("IdServicioDestajero"));

            user.setIdPedido(rs.getInt("IdPedido"));

            user.setIPedido(rs.getString("razonsocial"));

            user.setIdDestajero(rs.getInt("IdEmpresaDestajero"));

            user.setDestajero(rs.getString("EmpresaDestajero"));

            user.setFecha(rs.getString("fecha"));

            user.setFechaDevolucion(rs.getString("fechaDevolucion"));

            user.setCantidadTotalEntregado(rs.getString("CantidadTotalEntregado"));

            user.setCantidadTotalCulminado(rs.getString("CantidadTotalCulminado"));

            user.setTotalPagar(rs.getString("TotalPagar"));

            user.setEstado(rs.getInt("estado"));

            lista.add(user);

        }

    } catch (SQLException e) {

        System.out.println(e.toString());

    }

    return lista;

```

```

}

public List ListarPedidoProducto(String valor) {

    List<ServicioDestajeroDetalle > lista = new ArrayList();

    try {

        con = cn.getConnection();

        String sql = "SELECT p.IdServicioDestajero,m.Nombre as
ProductoNombre,p.Cantidad,p.precio,p.Total FROM serviciodestajerodetalle p inner JOIN
producto pr on p.IdProducto=pr.IdProducto inner join modelo m on pr.idmodelo=m.idmodelo
WHERE p.IdServicioDestajero = '" + valor + "'";

        ps = con.prepareStatement(sql);

        rs = ps.executeQuery();
        while (rs.next()) {

            ServicioDestajeroDetalle user = new ServicioDestajeroDetalle ();

            user.setIdServicio(rs.getInt("IdServicioDestajero"));

            user.setProducto(rs.getString("ProductoNombre"));

            user.setCantidad(rs.getString("Cantidad"));

            user.setPrecio(rs.getString("Precio"));

            user.setTotal(rs.getString("Total"));

            lista.add(user);

        }
    } catch (SQLException e) {

        System.out.println(e.toString());

    }

    return lista;

}

public List ListarServicioProducto(String valor) {

    List<ServicioDestajeroDetalle > lista = new ArrayList();

    try {

        con = cn.getConnection();

```

```
String sql = "SELECT s.IdServicioDestajero,p.IdProducto,m.Nombre as
ProductoNombre,p.CostoPrenda,s.Cantidad,s.CantidadEntregada,s.precio,s.SubTotal,s.Descuen
to,s.Total,s.Totalpagar FROM serviciodestajeroDetalle s INNER JOIN producto p on
s.IdProducto=p.IdProducto inner join modelo m on p.idmodelo=m.idmodelo WHERE
s.IdServicioDestajero = '" + valor + "'";
```

```
ps = con.prepareStatement(sql);
```

```
rs = ps.executeQuery();
```

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

```
ServicioDestajeroDetalle user = new ServicioDestajeroDetalle ();
```

```
user.setIdServicio(rs.getInt("IdServicioDestajero"));
```

```
user.setIdProducto(rs.getInt("IdProducto"));
```

```
user.setProducto(rs.getString("ProductoNombre"));
```

```
user.setCostoPrenda(rs.getString("CostoPrenda"));
```

```
user.setCantidad(rs.getString("Cantidad"));
```

```
user.setCantidadEntregada(rs.getString("CantidadEntregada"));
```

```
user.setPrecio(rs.getString("Precio"));
```

```
user.setSubtotal(rs.getString("SubTotal"));
```

```
user.setDescuento(rs.getString("Descuento"));
```

```
user.setTotal(rs.getString("Total"));
```

```
user.setTotalPagar(rs.getString("TotalPagar"));
```

```
lista.add(user);
```

```
}
```

```
} catch (SQLException e) {
```

```
System.out.println(e.toString());
```

```
}
```

```
return lista;
```

```
}
```

```
public boolean registrarServicioDestajero(ServicioDestajero us) {
```

```
String sql = "INSERT INTO serviciodestajero (IdPedido,IdEmpresDestajero
,Fecha,FechaDevolucion,TotalGeneral,IdUsuarioRegistro,CantidadTotalEntregado,estad
o) VALUES (?, ?, ?, ?, ?, ?, ?, ?)";
```

```

con = cn.getConnection();

try {

    ps = con.prepareStatement(sql);
    ps.setInt(1,us.getIdPedido());
    ps.setInt(2, us.getIdDestajero());
    ps.setString(3,us.getFecha() );
    ps.setString(4, us.getFechaDevolucion());
    ps.setString(5, us.getTotalGeneral());
    ps.setInt(6, us.getIdUsuario());
    ps.setString(7, us.getCantidadTotalEntregado());
    ps.setInt(8, us.getEstado());
    ps.execute();
    return true;
} catch (SQLException ex) {
    System.out.println(ex.toString());
    return false;
}
}

public ResultSet obtenerUltimoIdServicioDestajero() throws Exception{
    ResultSet rs = null;
    try{
        rs = ps.executeQuery("SELECT max(IdServicioDestajero) as IdServicioDestajero FROM
serviciodestajero");
        return rs;
    }catch(SQLException SQLex){
        throw SQLex;
    }
}

public int GuardarDetalleServicio(String IdServicioDestajero, String IdProducto, String
Cantidad, String Precio, String Total) {

```

```
int resultado = 0;
```

```
Connection conexion = null;
```

```
String sql = "INSERT INTO serviciodestajerodetalle(IdServicioDestajero, IdProducto,  
Cantidad,Precio, total) VALUES (?, ?, ?, ?, ?)";
```

```
con = cn.getConnection();
```

```
try {
```

```
    ps = con.prepareStatement(sql);
```

```
    ps.setString(1, IdServicioDestajero);
```

```
    ps.setString(2, IdProducto);
```

```
    ps.setString(3, Cantidad);
```

```
    ps.setString(4, Precio);
```

```
    ps.setString(5, Total);
```

```
    resultado = ps.executeUpdate();
```

```
    if (resultado > 0) {
```

```
    }
```

```
    conexion.close();
```

```
    } catch (Exception e) {
```

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

```
    }
```

```
    return resultado;
```

```
}
```

```
public boolean actualizarServicioDetalle(ServicioDestajeroDetalle reg) {
```

```
    boolean res;
```

```
    String sql = "UPDATE serviciodestajerodetalle SET  
CantidadEntregada=?,SubTotal=?,Descuento=?,TotalPagar=? WHERE IdServicioDestajero=? and  
idproducto = ?";
```

```
    con = cn.getConnection();
```

```
    try {
```



```

        ps = con.prepareStatement(sql);
        ps.setString(1, reg.getCantidadEntregada());
        ps.setString(2, reg.getSubtotal());
        ps.setString(3, reg.getDescuento());
        ps.setString(4, reg.getTotalPagar());
        ps.setInt(5, reg.getIdServicio());
        ps.setInt(6, reg.getIdProducto());
        ps.execute();

        res = true;
    } catch (SQLException ex) {
        System.out.println(ex.toString());
        res = false;
    }
    return res;
}

```

```

    public boolean actualizarServicioCal(ServicioDestajero reg) {
        boolean res;

        String sql = "UPDATE serviciodestajero SET
CantidadTotalCulminado=?,Descuento=?,TotalPagar=? WHERE IdServicioDestajero=? ";

        con = cn.getConnection();
        try {
            ps = con.prepareStatement(sql);

            ps.setDouble(1, Double.parseDouble(reg.getCantidadTotalCulminado()));
            ps.setDouble(2, Double.parseDouble(reg.getDescuento()));
            ps.setDouble(3, Double.parseDouble( reg.getTotalPagar()));

            ps.setInt(4, reg.getIdServicioDestajero());

            ps.execute();

            res = true;

```

```
} catch (SQLException ex) {  
    System.out.println(ex.toString());  
    res = false;  
}  
return res;  
}
```

```
public void actualizarEstadoServicio(String Id,String estado){  
try{  
    con = cn.getConnection();  
    CallableStatement statement=con.prepareCall("{call  
SP_ActualizarEstadoServicioDestajero(?,?)}");  
    statement.setString("ipservicio",Id);  
    statement.setString("pIdEstado",estado);  
    statement.executeUpdate();  
  
}catch(SQLException ex){  
    ex.printStackTrace();  
}  
  
}
```

USUARIO

```
package DAO;
```

```
import Configuracion.Conexion;
```

```
import Modelo.Usuario;
```

```
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 UsuarioDao {
```

```
    Connection con;
```

```
    Conexion cn = new Conexion();
```

```
    PreparedStatement ps;
```

```
    ResultSet rs;
```

```
    public boolean registrar(Usuario us) {
```

```
        String sql = "INSERT INTO usuario (usuario, nombres,clave,estado) VALUES (?, ?, ?, ?)";
```

```
        con = cn.getConnection();
```

```
        try {
```

```
            ps = con.prepareStatement(sql);
```

```
            ps.setString(1, us.getUsuario());
```

```
            ps.setString(2, us.getNombres());
```

```
            ps.setString(3, us.getClave());
```

```
            ps.setInt(4, us.getEstado());
```

```
            ps.execute();
```

```
            return true;
```

```

    } catch (SQLException ex) {

        System.out.println(ex.toString());

        return false;

    }

}

```

```

public boolean actualizar(Usuario reg) {

    boolean res;

    String sql = "UPDATE usuario SET usuario=?, nombres=?,clave=?,estado=? WHERE
idusuario = ?";

    con = cn.getConnection();

    try {

        ps = con.prepareStatement(sql);
        ps.setString(1, reg.getUsuario());
        ps.setString(2, reg.getNombres());
        ps.setString(3, reg.getClave());
        ps.setInt(4, reg.getEstado());
        ps.setInt(5, reg.getIdUsuario());

        ps.execute();

        res = true;

    } catch (SQLException ex) {

        System.out.println(ex.toString());

        res = false;

    }

    return res;

}

```

```

public List Listar(String valor) {

    List<Usuario> lista = new ArrayList();

    try {

        con = cn.getConnection();

        if ("".equalsIgnoreCase(valor)) {

```

```

        String sql = "SELECT * FROM usuario ORDER BY IdUsuario DESC";

        ps = con.prepareStatement(sql);

    } else {

        String sql = "SELECT * FROM usuario WHERE nombres LIKE '%" + valor + "%' OR
usuario LIKE '%" + valor + "%'";

        ps = con.prepareStatement(sql);

    }

    rs = ps.executeQuery();

    while (rs.next()) {

        Usuario user = new Usuario();

        user.setIdUsuario(rs.getInt("idusuario"));

        user.setUsuario(rs.getString("usuario"));

        user.setNombres(rs.getString("nombres"));

        user.setClave(rs.getString("clave"));

        user.setEstado(rs.getInt("estado"));

        lista.add(user);

    }

} catch (SQLException e) {

    System.out.println(e.toString());

}

return lista;

}

public Usuario login(String user, String pass) {

    String sql = "SELECT * FROM usuario WHERE usuario = ? AND clave = ?";

    Usuario l = new Usuario();

    try {

        con = cn.getConnection();

        ps = con.prepareStatement(sql);

        ps.setString(1, user);

        ps.setString(2, pass);

        rs = ps.executeQuery();

```

```
        if (rs.next()) {  
            if (rs.getRow() > 0) {  
                l.setIdUsuario(rs.getInt("IdUsuario"));  
                l.setUsuario(rs.getString("usuario"));  
                l.setNombres(rs.getString("nombres"));  
            }  
        }  
    } catch (SQLException e) {  
        System.out.println(e.toString());  
    }  
    return l;  
}  
}
```