



Universidad Del Centro Educativo Latinoamericano

Carrera: Ingeniería en Sistemas de Información

Materia: Proyecto Final

Profesor: Luciano Ripani

VISTA DE ARQUITECTURA

ÍNDICE

Historia de Versiones del Documento

- 1. Introducción 4
 - 1.1. Propósito del Documento 4
 - 1.2. Alcance del Documento 4
 - 1.3. Definiciones, Abreviaturas y Acrónimos 4
 - 1.4. Documentos Relacionados 5
 - 1.5. Visión General del Documento 5
- 2. Vistas de Arquitectura 6
 - 2.1 Arquitectura de Red
 - 2.2 Arquitectura del Software
 - 2.3 Modelo de Datos
 - 2.3.1 Estructura de la Base de Datos de Sistema
 - 2.3.2 Modelo de Datos y Tipos de Datos

Historia de Versiones del Documento

Fecha	Versión	Descripción	Autor
07/12/2019	1.0	Versión inicial del documento	Paula Garcia Morales
05/02/2019	1.1	Creación de la introducción a la metodología aplicada.Modificaciones varias en documento	Paula Garcia Morales
21/02/2020	1.2	Resaltado de componentes importantes + corrección del listado.	Paula Garcia Morales

1.Introducción

1.1. Propósito del Documento

El propósito de este documento es presentar diferentes vistas de arquitectura del sistema BidNex.

1.2. Alcance del Documento

Este documento se refiere al sistema BidNex , el cual se construye como parte del proyecto final de los alumnos Paula Garcia Morales y Maximiliano Mata Córdoba, como parte de la carrera de Ingeniería en Sistemas de Información de la Universidad del Centro Educativo Latinoamericano, en la ciudad de Rosario, Argentina.

1.3. Definiciones, Abreviaturas y Acrónimos

Ver documento Glosario.doc, cada palabra que se aclare su significado tiene una referencia la cual nos indica en qué lugar se encuentra en el glosario.

1.4. Documentos Relacionados

Documento	Nombre/Ubicación del Archivo	Fuente
Especificación de requerimientos	Nombre: Especificación de requerimientos.docx Ubicación: \\google drive\Proyecto Final\Especificación de requerimientos	Garcia Morales / Mata Córdoba
Plan de proyecto	Nombre: Plan de proyecto.docx Ubicación: \\google drive\Proyecto Final\Plan de proyecto	Garcia Morales / Mata Córdoba

1.5. Visión General del Documento

En las secciones subsiguientes se detallan todos los aspectos relacionados con las vistas de arquitectura utilizados en el proyecto de BidNex.

2. Vistas de Arquitectura

2.1 - Metodología Aplicada

El patrón Modelo-Vista-Controlador (MVC) surge con el objetivo de reducir el esfuerzo de programación, necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, a partir de estandarizar el diseño de las aplicaciones. El patrón MVC es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo. A partir del uso de frameworks basados en el patrón MVC se puede lograr una mejor organización del trabajo y mayor especialización de los desarrolladores y diseñadores.

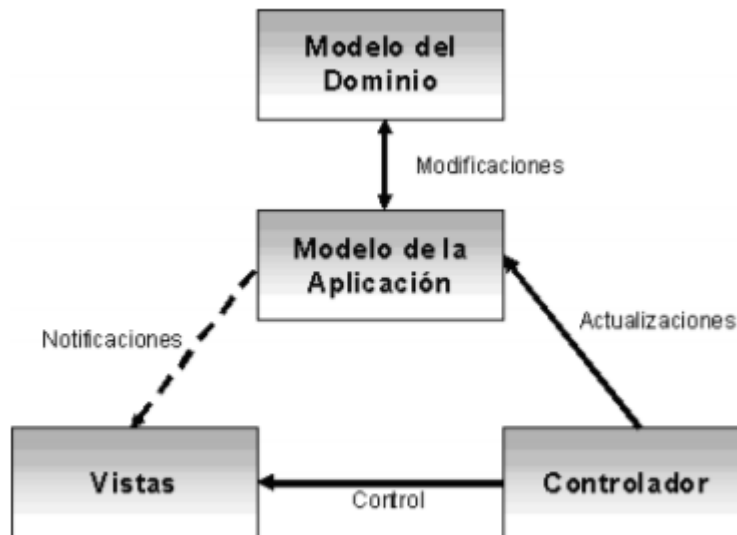


Figura 1: Relación entre los módulos del patrón MVC.

Modelo de la aplicación

El modelo de la aplicación es un conjunto de clases que se relacionan con el modelo del dominio, que tienen conocimiento de las vistas y que implementan los mecanismos necesarios para notificar a estas últimas sobre los cambios que se pudieren dar en el modelo del dominio. El modelo de la aplicación es llamado también coordinador de la aplicación.

Las vistas

Las vistas son el conjunto de clases que se encargan de mostrar al usuario la información contenida en el modelo. Una vista está asociada a un modelo, pudiendo existir varias vistas asociadas al mismo modelo; así por ejemplo, se puede tener una vista mostrando la hora del sistema como un reloj analógico y otra vista mostrando la misma información como un reloj digital. Una vista obtiene del modelo solamente la información que necesita para desplegar y se actualiza cada vez que el modelo del dominio cambia por medio de notificaciones generadas por el modelo de la aplicación. [

El controlador

El controlador es un objeto que se encarga de dirigir el flujo del control de la aplicación debido a mensajes externos, como datos introducidos por el usuario u opciones del menú seleccionadas por él. A partir de estos mensajes, el controlador se encarga de modificar el modelo o de abrir y cerrar vistas. El controlador tiene acceso al modelo y a las vistas, pero las vistas y el modelo no conocen de la existencia del controlador.

Ventajas

Desarrollar una aplicación siguiendo este patrón de diseño tiene muchas ventajas:

- La aplicación está implementada modularmente. Sus vistas muestran información actualizada siempre.
- El programador no debe preocuparse de solicitar que las vistas se actualicen, ya que este proceso es realizado automáticamente por el modelo de la aplicación.
- Si se desea hacer una modificación al modelo del dominio, como aumentar métodos o datos contenidos, sólo debe modificarse el modelo y las interfaces del mismo con las vistas, no todo el mecanismo de comunicación y de actualización entre modelos.
- Las modificaciones a las vistas no afectan en absoluto a los otros módulos de la aplicación.
- MVC es bastante utilizado en la actualidad en marcos de aplicación orientados a objeto desarrollados para construir aplicaciones de gran tamaño
- MVC está demostrando ser un patrón de diseño bien elaborado pues las aplicaciones que lo implementan presentan una extensibilidad y una mantenibilidad únicas comparadas con otras aplicaciones basadas en otros patrones.

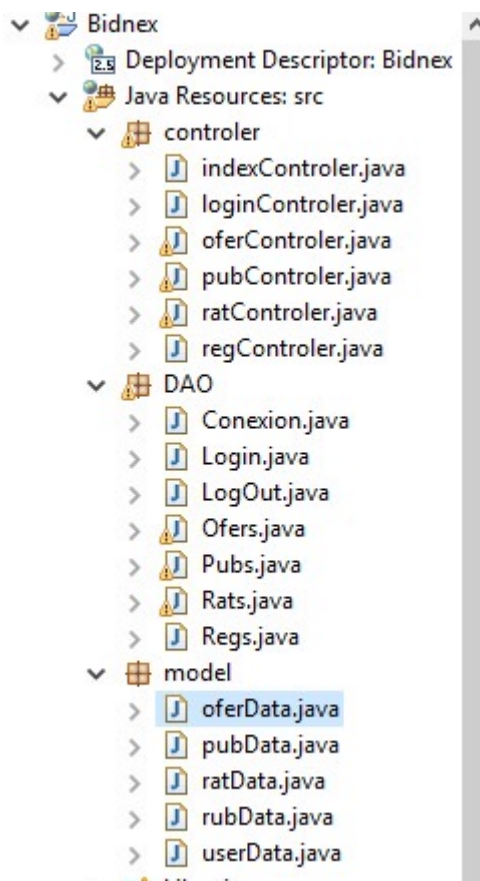
Desventajas




























- El tiempo de desarrollo de una aplicación que implementa el patrón de diseño MVC es mayor, al menos en la primera etapa, que el tiempo de desarrollo de una aplicación que no lo implementa, ya que MVC requiere que el programador implementa una mayor cantidad de clases que en un entorno de desarrollo común no son necesarias. Sin embargo, esta desventaja es muy relativa ya que posteriormente, en la etapa de mantenimiento de la aplicación, una aplicación MVC es muchísimo más mantenible, extensible y modificable que una aplicación que no lo implementa.
- MVC requiere la existencia de una arquitectura inicial sobre la que se deben construir clases e interfaces para modificar y comunicar los módulos de una aplicación. Esta arquitectura inicial debe incluir, por lo menos: un mecanismo de eventos para poder proporcionar las notificaciones que genera el modelo de aplicación; una clase Modelo, otra clase Vista y una clase Controlador genéricas que realicen todas las tareas de comunicación, notificación y actualización que serán luego transparentes para el desarrollo de la aplicación.
- MVC es un patrón de diseño orientado a objetos por lo que su implementación es sumamente costosa y difícil en lenguajes que no siguen este paradigma.

2.2 Arquitectura de Red

La plataforma requiere conexión a internet para funcionar.

2.3 Arquitectura del Software



- ▼  WebContent
 - >  contactform
 - >  css
 - >  fonts
 - >  img
 - >  js
 - >  lib
 - >  META-INF
 - >  vendor
 - >  WEB-INF
 -  about.jsp
 -  adminMenu.jsp
 -  cliAlta.jsp
 -  cliList.jsp
 -  cliMenu.jsp
 -  cliOffer.jsp
 -  cliUpdate.jsp
 -  index.jsp
 -  provAlta.jsp
 -  provList.jsp
 -  provMenu.jsp
 -  provUpdate.jsp
 -  rating.jsp
 -  register.jsp
 -  userUpdate.jsp
 -  userUpdateAdmin.jsp
 -  userUpdateCli.jsp

LOGIN



Index.jsp

...
<form class="login100-form validate-form" id="login-form" action="loginControler" method="POST" role="form" style="display: block;">*//esta linea indica que hace el boton iniciar sesion*

Inicia Con Tu Usuario y Contraseña

<!-- mensaje ok-->

```
<%if (session.getAttribute("msg")=="1"){%>
    <script>
        swal("Listo!", "Ahora podes loguearte!", "success", "ok");
    </script>
    <%session.setAttribute("msg", "null");;%>
<%}%>
```

```
<%if (session.getAttribute("msg")=="2"){%>
    <script>
        swal("Perfecto!", "Cerraste sesion correctamente!", "success", "ok");
    </script>
    <%session.setAttribute("msg", "null");;%>
```

<%}%>

<!-- mensaje error-->

<%if (session.getAttribute("msg")=="0"){%>

<script>

swal("Ups!", "Revisa tu usuario y contraseña. Tienes que ser un usuario activo", "error", "ok");

</script>

<%session.setAttribute("msg", "null");%>

<%}%>

//campo para ingresar el usuario

<div class="wrap-input100" data-validate = "Por favor

Ingrese un Usuario">

<input class="input100" type="text" name="user"

id="user">

Usuario

</div>

// campo para ingresar la clave

<div class="wrap-input100 validate-input" data-validate="Por favor Ingrese una contraseña">

<input class="input100" type="password"

name="password" id="password">

Contraseña

</div>

<div class="container-login100-form-btn">

<button class="login100-form-btn" type="submit">

Iniciar sesion *// boton iniciar sesion que te*

lleva al logincontroler (controlador) con la información ingresada

</button>

</div>

<div class="container-login100-form-btn">

Quiero Registrarme

</div>

</form>

...

loginControler.java

protected void doPost(HttpServletRequest request, *// ingresa por el doPost //utilizamos este método como se indicó en el jsp*

```
        HttpServletResponse response) throws ServletException, IOException {  
        String userName = request.getParameter("user"); // inicializamos las variables  
        String password = request.getParameter("password");
```

userData userdata = new userData(userName, password); *//inicializamos el modelo de datos luego utilizado*

```
        Login login = new Login();
```

```
        try {
```

```
            String userValidate = login.LoginUser(userdata); //verificamos la sesión  
            if (userValidate.equals("null")) {  
                HttpSession session = request.getSession();  
                session.setMaxInactiveInterval(10 * 60); //la finalizamos luego de
```

60 segundos sin usar

```
                session.setAttribute("msg", "0");  
                request.getRequestDispatcher("index.jsp").forward(  
                    request, response);
```

```
            } else {  
                String[] parts = userValidate.split("-");  
                String partRole = parts[0]; // identificamos el rol del usuario para
```

dirigirlo a donde corresponde

```
                String partIduser = parts[1];  
                if (partRole.equals("Admin_Role")) { // si es tipo admin
```

```
                    HttpSession session = request.getSession();  
                    session.setMaxInactiveInterval(10 * 60);  
                    session.setAttribute("user", userName);  
                    session.setAttribute("iduser", partIduser);  
                    session.setAttribute("role", partRole);
```

```
                    request.getRequestDispatcher("adminMenu.jsp").forward(  
                        request, response); // lo dirigimos al admin
```

menu

```
                } else if (partRole.equals("Prov_Role")) { // si es tipo proveedor
```

```
                    HttpSession session = request.getSession();  
                    session.setMaxInactiveInterval(10 * 60);  
                    session.setAttribute("user", userName);  
                    session.setAttribute("iduser", partIduser);  
                    session.setAttribute("role", partRole);
```

```

        request.getRequestDispatcher("/provMenu.jsp").forward(
            request, response); // lo dirigimos al
provMenu
    } else if (partRole.equals("Cli_Role")) { // si es tipo cliente
        HttpSession session = request.getSession();
        session.setMaxInactiveInterval(10 * 60);
        session.setAttribute("user", userName);
        session.setAttribute("idUser", partIduser);
        session.setAttribute("role", partRole);

        request.getRequestDispatcher("/cliMenu.jsp").forward(
            request, response); // lo dirigimos al cliMenu
    }
}
} catch (IOException e) { //esta es una excepción por si hay algun fallo no
contemplado
    System.err.print(e);
}
}
}
}

```

Login.java /*DAO*/

```

public class Login {

    private static String FindAll = "Select * from usuarios"; // esta query nos identifica los
    usuarios

    public String LoginUser(userData userdata) {

        String userName = userdata.getUser(); //inicializa variables
        String password = userdata.getPass();

        Connection con = null; // coneccion con la base de datos
        Statement statement = null;
        ResultSet resultSet = null;

        String userNameDB = ""; //limpia e inicializa variables locales
        String passwordDB = "";
        String roleDB = "";

        try {

```

```

con = Conexion.conexion(); //conecta con la base
statement = con.createStatement();
resultSet = statement.executeQuery(FindAll);

while (resultSet.next()) { //compara variables
    userNameDB = resultSet.getString("usuario");
    passwordDB = resultSet.getString("contraseña");
    roleDB = resultSet.getString("rol");
    String status = resultSet.getString("estado_user");
    int iduser = resultSet.getInt("idUsuario");
    if (userName.equals(userNameDB) &&
password.equals(passwordDB) //compara pass
&& roleDB.equals("admin")&&
status.equals("activo")) //verifica que este activo
        return "Admin_Role-"+iduser; // devuelve el rol + el id del
usuario

        else if (userName.equals(userNameDB)
&& password.equals(passwordDB)
&& roleDB.equals("proveedor") &&
status.equals("activo"))
            return "Prov_Role-"+iduser; // devuelve el rol + el id del
usuario

        else if (userName.equals(userNameDB)
&& password.equals(passwordDB) &&
roleDB.equals("cliente") && status.equals("activo")) //verifica que este acitvo
            return "Cli_Role-"+iduser; // devuelve el rol + el id del
usuario
    }
} catch (SQLException e) { //excepción por si sucede algun error inesperado
    System.err.print(e);
}
return "null"; // si no encontro nada sale con el null para rechazar el login
}

private static Login instance = null;

public Login() {
}

public static Login getInstace() {
    if (instance == null) {
        instance = new Login();
    }
    return instance;
}

```

```

    }

    public static void main(String[] args) {

    }

}

```

UserData.javab /*model*/

```

package model;

public class userData {

    int iduser;
    String user;
    String pass;
    String role;
    String email;
    String statusUser;

    public userData(int iduser, String user, String role) {
        super();
        this.iduser = iduser;
        this.user = user;
        this.role = role;
    }

    public userData(int iduser, String user, String pass, String role) {
        super();
        this.iduser = iduser;
        this.user = user;
        this.pass = pass;
        this.role = role;
    }

    public userData(String user, String pass) {
        super();
        this.user = user;
        this.pass = pass;
    }
}

```

```

public userData(String user, String pass, String role, String statusUser) {
    super();
    this.user = user;
    this.pass = pass;
    this.role = role;
    this.statusUser = statusUser;
}

```

```

public userData(int iduser, String user, String pass, String role,
    String email, String statusUser) {
    super();
    this.iduser = iduser;
    this.user = user;
    this.pass = pass;
    this.role = role;
    this.email = email;
    this.statusUser = statusUser;
}

```

```

public userData(String user, String pass, String role, String email,
    String statusUser) {
    super();
    this.user = user;
    this.pass = pass;
    this.role = role;
    this.email = email;
    this.statusUser = statusUser;
}

```

```

}

```

ALTA PUBLICACION

Aca podes ingresar todo sobre tu nueva publicacion

Título de la publicacion

Lugar donde se hace la entrega

Fecha Fin de la Publicacion

dd/mm/aaaa

Descripcion de la publicacion

Seleccionar un Rubro

Ferreteria

Crear Publicacion

cliAlta.jsp

...

```
<form class="login100-form validate-form" id="altaPub-form" action="pubControler"
method="POST" role="form" style="display: block;">
```

```
<span class="login100-form-title p-b-
```

```
43">
```

Aca podes ingresar todo sobre tu nueva publicacion

```
</span>
```



```

        <input class="input100" type="text" id="idUser" value="<%=idUser %>">
name="idUser" id="idUser" value="<%=idUser %>">

        <div class="wrap-input100 validate-input" data-validate =
"Por favor Ingrese el titulo de la publicacion">
            <input class="input100" type="text"
name="titlePub" id="titlePub" >
            <span class="focus-input100"></span>
            <span class="label-input100">Titulo de la
publicacion</span>
        </div>

        <div class="wrap-input100 validate-input" data-validate =
"Por favor Ingrese el lugar de entrega">
            <input class="input100" type="text"
name="lugarPub" id="lugarPub" >
            <span class="focus-input100"></span>
            <span class="label-input100">Lugar donde se hace la
entrega</span>
        </div>

        <center>
        <a class="input100">Fecha Fin de la Publicacion</a>
        </center>
        <br>
        <div class="wrap-input100 validate-input" data-validate =
"Por favor Ingrese Fecha de Fin">
            <input class="input100" type="date"
name="fechaFinPub" id="fechaFinPub" >
            <span class="focus-input100"></span>
            <span class="label-input100"></span>
        </div>

        <div class="wrap-input100 validate-input" data-
validate="Por favor Ingrese una breve descripcion">
            <input class="input100" type="text"
name="descripcionPub" id="descripcionPub" >
            <span class="focus-input100"></span>
            <span class="label-input100">Descripcion de la
publicacion</span>
        </div>

        <br></br>
        <center>
        <span class="input100">Seleccionar un Rubro</span>
        <select name="itemPub" class="input100">

```

```

value="ferreteria">Ferreteria</option>
value="limpieza">Limpieza</option>
value="informatica">Informatica</option>
value="almacen">Almacen</option>
value="libreria">Libreria</option>
</select>
</center>
<br></br>
<br></br>
<div class="container-login100-form-btn">
    <button type="submit" name="action"
value="create" id="create-submit" tabindex="4" class="btn btn-outline-primary">Crear
Publicacion</button>
</div>

</form>

```

...

pubControler.java

....

```

protected void doPost(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    HttpSession sesion = request.getSession(true);
    Pubs pub = Pubs.getInstance();
    String fechaFinBusqueda = request.getParameter("fechaFin");
    String fechaIniBusqueda = request.getParameter("fechaIni");
    String tituloPub = request.getParameter("titlePub");
    String lugarEntregaPub = request.getParameter("lugarPub");
    String rubroPub = request.getParameter("itemPub");
    String fechaFinPub = request.getParameter("fechaFinPub");
    String descripcionPub = request.getParameter("descripcionPub");
    String id = request.getParameter("idUser");
    String statusStr = (String) request.getParameter("status");
    String itemStr = (String) request.getParameter("item");
    String idPubStr = (String) request.getParameter("id");
    String action = request.getParameter("action");

```

```

String listDetails = request.getParameter("withDetails");
String details = "false";
pubData pubdata = null;

....

else if ("create".equals(action)) {
    if (id != null) {
        int iduser = Integer.parseInt(id);
        String status = "publicada";
        pubdata = new pubData(iduser, tituloPub, lugarEntregaPub,
                               rubroPub, fechaFinPub, descripcionPub, status);
        Object result = pub.CreatePub(pubdata);
        if (result.equals("ok")) {
            sesion.setAttribute("msg", "1");
        } else {
            sesion.setAttribute("msg", "0");
        }
        String nextJSP = "/cliAlta.jsp";
        RequestDispatcher dispatcher = getServletContext()
            .getRequestDispatcher(nextJSP);
        dispatcher.forward(request, response);
    }
}

....

```

Pubs.java /*DAO*/

```

...
public class Pubs {
    ...
    private static String Insert = "insert into publicaciones
(idUsuario,titulo_pub,lugarEntrega_pub,rubro_pub,fechaFin_pub,descripcion_pub, estado_pub)"
        + " values (?, ?, ?, ?, ?, ?, ?)";

    ...

    public String CreatePub(pubData pubdata) {

        String result = "ok";
        int valres = 0;
        try {
            // conexion
            Connection conn = null;
            conn = Conexion.conexion();
            // query

```

```

        // alta

        statement = conn.prepareStatement(Insert);
        statement.setInt(1, pubdata.getIdUsuario());
        statement.setString(2, pubdata.getTitulo_pub());
        statement.setString(3, pubdata.getLugarEntrega_pub());
        statement.setString(4, pubdata.getRubro_pub());
        statement.setString(5, pubdata.getFechaFin_pub());
        statement.setString(6, pubdata.getDescripcion_pub());
        statement.setString(7, "publicada");
        valres = statement.executeUpdate();
        statement.close();
        conn.close();

        if (valres == 0) {
            result = "error";
        }
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        System.err.print("Error Al conectar Con El Servidor");
    }

    return result;
}
...

```

pubData.java /*model*/

```

public class pubData {
    int idPublicacion;
    int idUsuario;
    String titulo_pub;
    String lugarEntrega_pub;
    String rubro_pub;
    String fechaFin_pub;
    String descripcion_pub;
    String estado_pub;

    String fechaIniBusqueda;
    String fechaFinBusqueda;

```

```

public pubData(int idPublicacion , String estadoPub) {
    super();
    this.idPublicacion = idPublicacion;
    estado_pub = estadoPub;
}

public pubData(int idUsuario, String estadoPub, String fechaIniBusqueda,
    String fechaFinBusqueda) {
    super();
    this.idUsuario = idUsuario;
    estado_pub = estadoPub;
    this.fechaIniBusqueda = fechaIniBusqueda;
    this.fechaFinBusqueda = fechaFinBusqueda;
}

public pubData(int idUsuario, String tituloPub, String lugarEntregaPub,
    String rubroPub, String fechaFinPub, String descripcionPub,
    String estadoPub) {
    super();
    this.idUsuario = idUsuario;
    titulo_pub = tituloPub;
    lugarEntrega_pub = lugarEntregaPub;
    rubro_pub = rubroPub;
    fechaFin_pub = fechaFinPub;
    descripcion_pub = descripcionPub;
    estado_pub = estadoPub;
}

public pubData(int idPublicacion, int idUsuario, String tituloPub,
    String lugarEntregaPub, String rubroPub, String fechaFinPub,
    String descripcionPub, String estadoPub) {
    super();
    this.idPublicacion = idPublicacion;
    this.idUsuario = idUsuario;
    titulo_pub = tituloPub;
    lugarEntrega_pub = lugarEntregaPub;
    rubro_pub = rubroPub;
    fechaFin_pub = fechaFinPub;
    descripcion_pub = descripcionPub;
    estado_pub = estadoPub;
}
}

```

cliAlta.jsp - Msj

```
...
<body>
    <%if (session.getAttribute("user")== null){
        response.sendRedirect("index.jsp");} %><!-- verifica la existencia de inicio de
session-->

    <!-- mensaje ok-->

    <%if (session.getAttribute("msg")== "1"){%>
        <script>
            swal("Listo!", "Ya creaste tu Publicacion!", "success", "ok");
        </script>
        <%session.setAttribute("msg", "null");;%>
    <%}%>

    <!-- mensaje error-->
    <%if (session.getAttribute("msg")== "0"){%>
        <script>
            swal("Ups!", "Hubo un error, vuelve a intentarlo", "error", "ok");
        </script>
        <%session.setAttribute("msg", "null");;%>
    <%}%>

    ...
```

BAJA PUBLICACIÓN

Publicacion	Finaliza	Rubro			
tuercas	18/11/2017	ferreteria	Cancelar 	Ofertas 	Editar 

cliUpdate.jsp

```
...
<tbody>

        <%

                ArrayList<pubData> pubdatas =
(ArrayList<pubData>)request.getAttribute("pubdatas");
                if (pubdatas!=null) {
                for (int i=0; i<pubdatas.size(); i++){
                        pubData pubdatalist = pubdatas.get(i);
                        %>
                <tr>
                        <th><%= pubdatalist.getTitulo_pub() %></th>
                        <th><%= pubdatalist.getFechaFin_pub() %></th>
                        <th><%= pubdatalist.getRubro_pub() %></th>
                        <th class ="btn btn-outline-
danger" onclick="location.href='pubControler?action=delete&idPub=<
%=pubdatalist.getIdPublicacion()%>'>Cancelar <i class="fas fa-window-close"></i></a></th>
                        <th class ="btn btn-outline-
dark" onclick="location.href='oferControler?action=selectForCli&id=<
%=pubdatalist.getIdPublicacion()%>'>Ofertas <i class="fas fa-shipping-fast"></i></a></th>
                        <th class ="btn btn-outline-dark"
onclick="location.href='pubControler?action=select&idPub=<%=pubdatalist.getIdPublicacion()
%>'> Editar <i class="fas fa-edit"></i></th>

                </tr>

        <%}

                }%>
</tbody>

...

```

pubControler.java

...

```
public class pubControler extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
  
    public pubControler() {  
        super();  
    }  
}
```

...

```
protected void doGet(HttpServletRequest request,  
    HttpServletResponse response) throws ServletException, IOException {  
  
    HttpSession sesion = request.getSession(true);  
    Pubs pub = Pubs.getInstace();  
    String action = request.getParameter("action");  
    String idPubstr = request.getParameter("idPub");  
    String idUserstr = request.getParameter("idUser");  
    String itemStr = (String) request.getParameter("item");  
    String details = "true";  
    pubData pubdata = null;
```

...

```
if ("delete".equals(action)) {  
    if (idPubstr != null) {  
        int idPub = Integer.parseInt(idPubstr);  
        Object result = pub.DeletePub(idPub);  
  
        if (result.equals("ok")) {  
            sesion.setAttribute("msg", "1");  
        } else {  
            sesion.setAttribute("msg", "0");  
        }  
    }  
}
```

...

Pubs.java /*DAO*/

...

```
public class Pubs {
```



```

...
private static String Delete = "Update publicaciones set estado_pub=? where idPublicacion=?";
...
public String DeletePub(int idPub) {
    String result = "ok";
    try {
        // conexion
        Connection conn = null;
        conn = Conexion.conexion();

        String statusPub = "cancelada";
        statement = conn.prepareStatement(Delete);
        statement.setString(1, statusPub);
        statement.setInt(2, idPub);
        int valres = statement.executeUpdate();
        statement.close();
        conn.close();

        if (valres == 0) {
            result = "error";
        }
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        System.err.print(e);
    }
    return result;
}

```

MODIFICAR PUBLICACIÓN

Publicacion	Finaliza	Rubro	
tuercas	18/11/2017	ferreteria	Cancelar  Ofertas  Editar 

Aca Podes editar tu publicacion

tuercas

rosario

Fecha Fin de la Publicacion

dd/mm/aaaa

tuercas x 100

Seleccionar un Rubro

Ferreteria ▼

Modificar

cliUpdate.jsp

...

<tbody>

<%

```

        ArrayList<pubData> pubdatas =
(ArrayList<pubData>)request.getAttribute("pubdatas");
        if (pubdatas!=null) {
        for (int i=0; i<pubdatas.size(); i++){
            pubData pubdatalist = pubdatas.get(i);
            %>
        <tr>
            <th><%= pubdatalist.getTitulo_pub() %></th>
    
```

```

<th><%= pubdatalist.getFechaFin_pub() %></th>
<th><%= pubdatalist.getRubro_pub() %></th>
<th class ="btn btn-outline-
danger" onclick="location.href='pubControler?action=delete&idPub=<
%=pubdatalist.getIdPublicacion()%>'>Cancelar <i class="fas fa-window-close"></i></a></th>
<th class ="btn btn-outline-
dark" onclick="location.href='oferControler?action=selectForCli&id=<
%=pubdatalist.getIdPublicacion()%>'>Ofertas <i class="fas fa-shipping-fast"></i></a></th>
<th class ="btn btn-outline-dark"
onclick="location.href='pubControler?action=select&idPub=<%=pubdatalist.getIdPublicacion()
%>'> Editar <i class="fas fa-edit"></i></th>

</tr>

<%}
    }%>
</tbody>

```

...

<!--Editor-->

```

<form class="login100-form validate-form" id="Editor-form"
action="pubControler" method="post" role="form" style="display: block;">

```

```

<%
    pubData pubdata=

```

```

(pubData)request.getAttribute("pubdata");

```

```

String
    bannerDescription= "";

    String name= "";
    String place= "";
    String dateF= "";
    String desc= "";
    int idPub = -1;

String
    String Tname= "Titulo
de la publicacion";

    String Tplace= "Lugar
donde se hace la entrega";

```

```

fin de la publicacion";

"Descripcion de la publicacion";

null) {

pubdata.getIdPublicacion();

pubdata.getTitulo_pub();

pubdata.getLugarEntrega_pub();

pubdata.getFechaFin_pub();

pubdata.getDescripcion_pub();

pubdata.getIdUsuario();

String TdateF= "Fecha

String Tdesc=

if (pubdata !=

idPub =

name =

place =

dateF =

desc =

idUser =

bannerDescription= "Aca Podes editar tu publicacion";

Tname= "";
Tplace= "";
TdateF= "";
Tdesc= "";
disabled="";

}

%>

<span class="login100-form-title p-b-43">
    <%=bannerDescription %></span>
<center>
    <a class="input100" <%=hideStatus %>>id usuario</a>
</center>
    <div class="wrap-input100 validate-input" data-validate =
"Por favor Ingrese el titulo de la publicacion" <%=hideStatus %>>
        <input class="input100" type=<%=hideIdUser %>
name="idUser" id="idUser" value="<%=idUser %>">
        <span class="focus-input100"></span>
        <span class="label-input100"></span>
    </div>
    <input class="input100" type=hidden name="idPub"
id="idPub" value="<%=idPub %>">

```

```

<div class="wrap-input100 validate-input" data-validate =
"Por favor Ingrese el titulo de la publicacion">
    <input class="input100" type="text"
name="titlePub" id="titlePub" value="<%=name %>" <%=disabled%>>
    <span class="focus-input100"></span>
    <span class="label-input100"><%=Tname %></span>
</div>

```

```

<div class="wrap-input100 validate-input" data-validate =
"Por favor Ingrese el lugar de entrega">
    <input class="input100" type="text"
name="lugarPub" id="lugarPub" value="<%=place %>" <%=disabled%>>
    <span class="focus-input100"></span>
    <span class="label-input100"><%=Tplace%></span>
</div>

```

```

<center>
<a class="input100">Fecha Fin de la Publicacion</a>
</center>
<br>

```

```

<div class="wrap-input100 validate-input" data-validate =
"Por favor Ingrese Fecha de Fin">
    <input class="input100" type="date"
name="fechaFinPub" id="fechaFinPub" <%=disabled%> value="<%=dateF%>">
    <span class="focus-input100"></span>
    <span class="label-input100"></span>
</div>

```

```

<div class="wrap-input100 validate-input" data-
validate="Por favor Ingrese una breve descripcion">
    <input class="input100" type="text"
name="descripcionPub" id="descripcionPub" value="<%=desc%>" <%=disabled%>>
    <span class="focus-input100"></span>
    <span class="label-input100"><%=Tdesc%></span>
</div>

```

```

<br></br>
<center>
<span class="input100">Seleccionar un Rubro</span>
<select name="itemPub" class="input100" <%=disabled%>>
    <option class="input100"
value="ferreteria">Ferreteria</option>
    <option class="input100"
value="limpieza">Limpieza</option>
    <option class="input100"
value="informatica">Informatica</option>

```

```

        <option class="input100"
value="almacen">Almacen</option>
        <option class="input100"
value="libreria">Libreria</option>
    </select>
</center>
<br></br>

    <center>
    <span class="input100" <%=hideStatus %> >Seleccionar un
Estado</span>

        <select name="status" class="input100" <%=hideStatus %>>
            <option class="input100"
value="publicada">Publicada</option>
            <option class="input100"
value="ofertada">Ofertada</option>
            <option class="input100"
value="cancelada">Cancelada</option>
            <option class="input100"
value="vencida">Vencida</option>
            <option class="input100"
value="finalizada">Finalizada</option>
        </select>
    </center>

    <br></br>
    <div class="container-login100-form-btn">
        <button type="submit" name="action"
value="update" id="create-submit" tabindex="4" class="btn btn-outline-primary" <%=disabled
%>>Modificar</button>

    </div>

    <br></br>

    <div class="container-login100-form-btn" <%=hideStatus %>>
        <a href="" onclick="this.href='pubControler?
action=serchForAdmin&idUser='+document.getElementById('idUser').value;" class="btn btn-
outline-primary">
            Buscar
        </a>
    </div>

</form>

<!--Editor fin-->

```

...

pubControler.java

...

```
public class pubControler extends HttpServlet {  
    private static final long serialVersionUID = 1L;
```

```
    public pubControler() {  
        super();
```

```
    }
```

...

```
protected void doGet(HttpServletRequest request,  
    HttpServletResponse response) throws ServletException, IOException {
```

```
    HttpSession sesion = request.getSession(true);  
    Pubs pub = Pubs.getInstace();  
    String action = request.getParameter("action");  
    String idPubstr = request.getParameter("idPub");  
    String idUserstr = request.getParameter("idUser");  
    String itemStr = (String) request.getParameter("item");  
    String details = "true";  
    pubData pubdata = null;
```

...

```
} else if ("select".equals(action)) {  
    if (idPubstr != null) {  
        int idPub = Integer.parseInt(idPubstr);  
        pubdata = pub.findById(idPub);  
        request.setAttribute("pubdata", pubdata);  
  
        String IdUserStr = (String) sesion.getAttribute("iduser");  
        int idUser = Integer.parseInt(IdUserStr);  
        ArrayList<pubData> pubdatas = pub.findByUser(idUser);  
        request.setAttribute("pubdatas", pubdatas);  
  
        String nextJSP = "/cliUpdate.jsp";  
        RequestDispatcher dispatcher = getServletContext()  
            .getRequestDispatcher(nextJSP);
```

```

        dispatcher.forward(request, response);
    }

```

...

Pubs.java /*DAO*/

...

```

public class Pubs {
    private static String FindById = "Select *, DATE_FORMAT(fechaFin_pub,'%d/%m/%Y') as
    fechaFin_pub_Order from publicaciones where idPublicacion=?";

```

...

```

    private static String Update = "Update publicaciones set
    idUsuario=?,titulo_pub=?,lugarEntrega_pub=?,rubro_pub=?,fechaFin_pub=?,descripcion_pub=?,
    estado_pub=? where idPublicacion=?";

```

...

```

    public String UpdatePub(int idPub, pubData pubdata) {
        String result = "ok";
        try {
            // conexion
            Connection conn = null;
            conn = Conexion.conexion();
            statement = conn.prepareStatement(Update);
            statement.setInt(1, pubdata.getIdUsuario());
            statement.setString(2, pubdata.getTitulo_pub());
            statement.setString(3, pubdata.getLugarEntrega_pub());
            statement.setString(4, pubdata.getRubro_pub());
            statement.setString(5, pubdata.getFechaFin_pub());
            statement.setString(6, pubdata.getDescripcion_pub());
            statement.setString(7, pubdata.getEstado_pub());
            statement.setInt(8, pubdata.getIdPublicacion());
            statement.executeUpdate();
            statement.close();
            conn.close();
            int valres = 1;
            if (valres == 0) {
                result = "error";
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            System.err.print(e);
        }

        return result;
    }

```

...

pubData.java /*model*/

```
public class pubData {
    int idPublicacion;
    int idUsuario;
    String titulo_pub;
    String lugarEntrega_pub;
    String rubro_pub;
    String fechaFin_pub;
    String descripcion_pub;
    String estado_pub;

    String fechaIniBusqueda;
    String fechaFinBusqueda;

    public pubData(int idPublicacion , String estadoPub) {
        super();
        this.idPublicacion = idPublicacion;
        estado_pub = estadoPub;
    }

    public pubData(int idUsuario, String estadoPub, String fechaIniBusqueda,
        String fechaFinBusqueda) {
        super();
        this.idUsuario = idUsuario;
        estado_pub = estadoPub;
        this.fechaIniBusqueda = fechaIniBusqueda;
        this.fechaFinBusqueda = fechaFinBusqueda;
    }

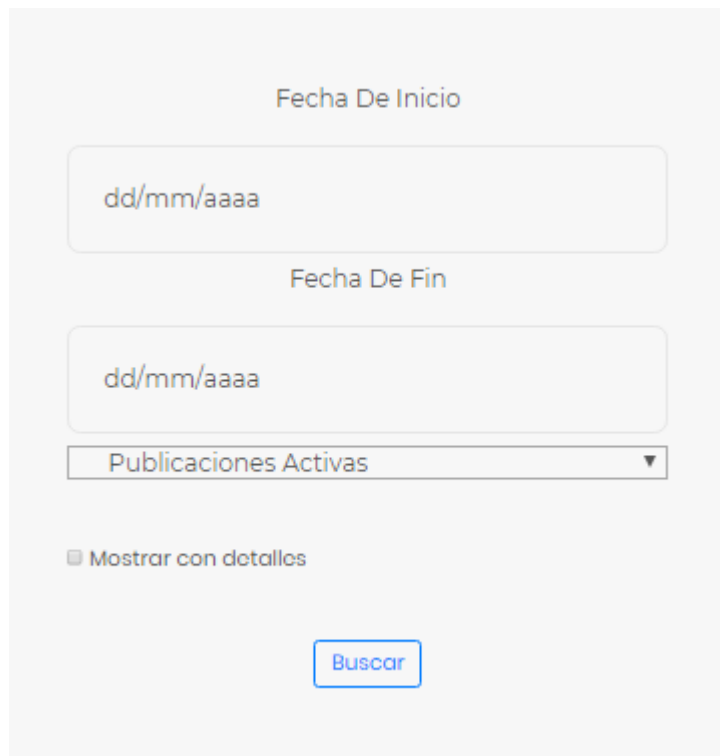
    public pubData(int idUsuario, String tituloPub, String lugarEntregaPub,
        String rubroPub, String fechaFinPub, String descripcionPub,
        String estadoPub) {
        super();
        this.idUsuario = idUsuario;
        titulo_pub = tituloPub;
        lugarEntrega_pub = lugarEntregaPub;
        rubro_pub = rubroPub;
        fechaFin_pub = fechaFinPub;
        descripcion_pub = descripcionPub;
        estado_pub = estadoPub;
    }
}
```

```
}
```

```
public pubData(int idPublicacion, int idUsuario, String tituloPub,  
               String lugarEntregaPub, String rubroPub, String fechaFinPub,  
               String descripcionPub, String estadoPub) {  
    super();  
    this.idPublicacion = idPublicacion;  
    this.idUsuario = idUsuario;  
    titulo_pub = tituloPub;  
    lugarEntrega_pub = lugarEntregaPub;  
    rubro_pub = rubroPub;  
    fechaFin_pub = fechaFinPub;  
    descripcion_pub = descripcionPub;  
    estado_pub = estadoPub;  
}
```

```
}
```

LISTADO DE PUBLICACIONES



Fecha De Inicio

dd/mm/aaaa

Fecha De Fin

dd/mm/aaaa

Publicaciones Activas ▼

☐ Mostrar con detalles

Buscar

cliList.jsp

```
<!--Filtros-->
<div class="limiter">
    <div class="container-login100">
        <div class="wrap-login100">
            <div class="login100-form validate-form">

                <form id="Cli-form" action="pubControler" method="post"
role="form" style="display: block;">

                    <div class="wrap-input100 validate-input" data-validate =
"Por favor Ingrese el titulo de la publicacion" <%=hideStatus %>>
                        <input class="input100" type="text" value="<%=hideIdUser %>
name="idUser" id="idUser" value="<%=idUser %>">
                        <span class="focus-input100"></span>
                        <span class="label-input100"></span>
                    </div>

                    <center>
                        <a class="input100">Fecha De Inicio</a>
                    </center>
```

```

<br>
<div class="wrap-input100 validate-input" data-validate =
"Por favor Ingrese Fecha de Inicio">
    <input class="input100" type="date"
name="fechalni" id="fechalni">
    <span class="focus-input100"></span>
    <span class="label-input100"></span>
</div>

<center>
<a class="input100">Fecha De Fin</a>
</center>
<br>
<div class="wrap-input100 validate-input" data-validate =
"Por favor Ingrese Fecha de Fin">
    <input class="input100" type="date"
name="fechaFin" id="fechaFin">
    <span class="focus-input100"></span>
    <span class="label-input100"></span>
</div>

<center>
<select name="status" class="input100">
    <option class="input100"
value="publicada">Publicaciones Activas</option>
    <option class="input100"
value="cancelada">Publicaciones Canceladas</option>
    <option class="input100"
value="vencida">Publicaciones Vencidas</option>
    <option class="input100"
value="finalizada">Publicaciones Finalizadas</option>
    <option class="input100"
value="ofertada">Publicaciones Ofertadas</option>
    <option class="input100"
value="all">Todas</option>
</select>
</center>
<br></br>

<center>
    <div class="btn-group-toggle" data-toggle="buttons">
        <label class="btn btn-outline-primary">
            <input type="checkbox" name="withDetails"
value="withDetails" > Mostrar con detalles<br>
        </label>
    </div>
</center>

```

```

        <br></br>
        <div class="container-login100-form-btn">
            <button type="submit" name="action"
value="filter" id="create-submit" tabindex="4" class="btn btn-outline-primary">Buscar</button>
        </div>

    </form>
</div>
<div class="login100-more">
    <!--Filtros fin-->

```

pubControler.java

```

protected void doPost(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    HttpSession sesion = request.getSession(true);
    Pubs pub = Pubs.getInstace();
    String fechaFinBusqueda = request.getParameter("fechaFin");
    String fechalniBusqueda = request.getParameter("fechalni");
    String tituloPub = request.getParameter("titlePub");
    String lugarEntregaPub = request.getParameter("lugarPub");
    String rubroPub = request.getParameter("itemPub");
    String fechaFinPub = request.getParameter("fechaFinPub");
    String descripcionPub = request.getParameter("descripcionPub");
    String id = request.getParameter("idUser");
    String statusStr = (String) request.getParameter("status");
    String itemStr = (String) request.getParameter("item");
    String idPubStr = (String) request.getParameter("id");
    String action = request.getParameter("action");
    String listDetails = request.getParameter("withDetails");
    String details = "false";
    pubData pubdata = null;

    ...

    ....
    else if ("filter".equals(action)) {
        if (id != null) {
            int iduser = Integer.parseInt(id);
            if (statusStr.equals("all")) {
                ArrayList<pubData> pubdatas = pub.FilterAllPub(iduser,
                    fechalniBusqueda, fechaFinBusqueda);
            }
        }
    }
}

```

```

        if (listDetails != null) {
            details = "list";
        }

        request.setAttribute("pubdatas", pubdatas);
        request.setAttribute("details", details);
        String nextJSP = "/cliList.jsp";
        RequestDispatcher dispatcher = getServletContext()
            .getRequestDispatcher(nextJSP);
        dispatcher.forward(request, response);
    } else {
        ArrayList<pubData> pubdatas = pub.FilterPub(iduser,
            fechaIniBusqueda, fechaFinBusqueda,
statusStr);

        if (listDetails != null) {
            details = "list";
        }

        request.setAttribute("pubdatas", pubdatas);
        request.setAttribute("details", details);
        request.setAttribute("statusStr", statusStr);
        String nextJSP = "/cliList.jsp";
        RequestDispatcher dispatcher = getServletContext()
            .getRequestDispatcher(nextJSP);
        dispatcher.forward(request, response);
    }
}
} else if ("filterForProv".equals(action)) {
    if (itemStr.equals("all")) {
        ArrayList<pubData> pubdatas = pub.FilterAllProv();

        if (listDetails != null) {
            details = "list";
        }

        request.setAttribute("details", details);
        request.setAttribute("pubdatas", pubdatas);
        String nextJSP = "/provAlta.jsp";
        RequestDispatcher dispatcher = getServletContext()
            .getRequestDispatcher(nextJSP);
        dispatcher.forward(request, response);
    } else {
        ArrayList<pubData> pubdatas = pub.FilterProv(itemStr);

```

```

        if (listDetails != null) {
            details = "list";
        }
        request.setAttribute("details", details);
        request.setAttribute("pubdatas", pubdatas);
        String nextJSP = "/provAlta.jsp";
        RequestDispatcher dispatcher = getServletContext()
            .getRequestDispatcher(nextJSP);
        dispatcher.forward(request, response);
    }
}
doGet(request, response);
}
}

```

...

Pubs.java /*DAO*/

```

...
public class Pubs {
...
    private static String ByDate = "Select *, DATE_FORMAT(fechaFin_pub,'%d/%m/%Y') as
    fechaFin_pub_Order from publicaciones where idUsuario=? and fechaFin_pub BETWEEN ? AND ?
    and estado_pub=?";
    private static String ByDateAll = "Select *, DATE_FORMAT(fechaFin_pub,'%d/%m/%Y') as
    fechaFin_pub_Order from publicaciones where idUsuario=? and fechaFin_pub BETWEEN ?
    AND ?";
    private static String ForProv = "Select *, DATE_FORMAT(fechaFin_pub,'%d/%m/%Y') as
    fechaFin_pub_Order from publicaciones where rubro_pub=? and (estado_pub=? or
    estado_pub=?)";
    private static String ForProvAll = "Select *, DATE_FORMAT(fechaFin_pub,'%d/%m/%Y') as
    fechaFin_pub_Order from publicaciones a inner join calificaciones b on a.idUsuario=b.idUsuario
    where estado_pub=? or estado_pub=?";

```

...

```

Connection conn = null;
private PreparedStatement statement = null;
private ResultSet resultSet = null;

```

...

```

public ArrayList<pubData> FilterPub(int idUser, String iniBusqueda,
    String finBusqueda, String status) {
    ArrayList<pubData> pubdatas = new ArrayList<pubData>();
    try {
        conn = Conexion.conexion();
        statement = conn.prepareStatement(ByDate);
        statement.setInt(1, idUser);
        statement.setString(2, iniBusqueda);
        statement.setString(3, finBusqueda);
        statement.setString(4, status);
        resultSet = statement.executeQuery();
        while (resultSet.next()) {
            int idPub = resultSet.getInt("idPublicacion");
            int idUsuariobd = resultSet.getInt("idUsuario");
            String tituloPub = resultSet.getString("titulo_pub");
            String lugarEntregaPub = resultSet.getString("lugarEntrega_pub");
            String rubroPub = resultSet.getString("rubro_pub");
            String fechaFinPub = resultSet.getString("fechaFin_pub_Order");
            String descripcionPub = resultSet.getString("descripcion_pub");
            String statusPub = resultSet.getString("estado_pub");
            pubData pubdata = new pubData(idPub, idUsuariobd, tituloPub,
                lugarEntregaPub, rubroPub, fechaFinPub,
descripcionPub,
                statusPub);
            pubdatas.add(pubdata);
        }
        resultSet.close();
        statement.close();
        conn.close();
    } catch (Exception e) {
        // TODO Auto-generated catch block
        System.err.print(e);
    }
    return pubdatas;
}

```

```

public ArrayList<pubData> FilterProv(String item) {
    ArrayList<pubData> pubdatas = new ArrayList<pubData>();
    try {
        conn = Conexion.conexion();
        statement = conn.prepareStatement(ForProv);
        statement.setString(1, item);
        statement.setString(2, "publicada");
    }
}

```



```

statement.setString(3, "ofertada");
resultSet = statement.executeQuery();
while (resultSet.next()) {
    int idPub = resultSet.getInt("idPublicacion");
    int idUsuariobd = resultSet.getInt("idUsuario");
    String tituloPub = resultSet.getString("titulo_pub");
    String lugarEntregaPub = resultSet.getString("lugarEntrega_pub");
    String rubroPub = resultSet.getString("rubro_pub");
    String fechaFinPub = resultSet.getString("fechaFin_pub_Order");
    String descripcionPub = resultSet.getString("descripcion_pub");
    String statusPub = resultSet.getString("estado_pub");
    pubData pubdata = new pubData(idPub, idUsuariobd, tituloPub,
        lugarEntregaPub, rubroPub, fechaFinPub,
descripcionPub,
        statusPub);
    pubdatas.add(pubdata);
}
resultSet.close();
statement.close();
conn.close();
} catch (Exception e) {
    // TODO Auto-generated catch block
    System.err.print(e);
}
return pubdatas;
}

public ArrayList<pubData> FilterAllProv() {
    ArrayList<pubData> pubdatas = new ArrayList<pubData>();
    try {
        conn = Conexion.conexion();
        statement = conn.prepareStatement(ForProvAll);
        statement.setString(1, "publicada");
        statement.setString(2, "ofertada");
        resultSet = statement.executeQuery();
        while (resultSet.next()) {
            int idPub = resultSet.getInt("idPublicacion");
            int idUsuariobd = resultSet.getInt("idUsuario");
            String tituloPub = resultSet.getString("titulo_pub");
            String lugarEntregaPub =
resultSet.getString("lugarEntrega_pub");
            String rubroPub = resultSet.getString("rubro_pub");
            String fechaFinPub =
resultSet.getString("fechaFin_pub_Order");

```

```

        String descripcionPub =
resultSet.getString("descripcion_pub");
        String statusPub = resultSet.getString("estado_pub");
        int calificacion = resultSet.getInt("calificacion");
        int cantTransacciones =
resultSet.getInt("cantTransacciones");
        pubData pubdata = new pubData(idPub, idUsuariobd,
tituloPub,
                                lugarEntregaPub, rubroPub, fechaFinPub,
descripcionPub,
                                statusPub,calificacion,cantTransacciones);
        pubdatas.add(pubdata);
    }
    resultSet.close();
    statement.close();
    conn.close();
} catch (Exception e) {
    // TODO Auto-generated catch block
    System.err.print(e);
}
return pubdatas;
}

```

```

public ArrayList<pubData> FilterAllPub(int idUser, String iniBusqueda,
String finBusqueda) {
    ArrayList<pubData> pubdatas = new ArrayList<pubData>();
    try {
        conn = Conexion.conexion();
        statement = conn.prepareStatement(ByDateAll);
        statement.setInt(1, idUser);
        statement.setString(2, iniBusqueda);
        statement.setString(3, finBusqueda);
        resultSet = statement.executeQuery();
        while (resultSet.next()) {
            int idPub = resultSet.getInt("idPublicacion");
            int idUsuariobd = resultSet.getInt("idUsuario");
            String tituloPub = resultSet.getString("titulo_pub");
            String lugarEntregaPub =
resultSet.getString("lugarEntrega_pub");
            String rubroPub = resultSet.getString("rubro_pub");
            String fechaFinPub =
resultSet.getString("fechaFin_pub_Order");
            String descripcionPub =
resultSet.getString("descripcion_pub");

```

```

        String statusPub = resultSet.getString("estado_pub");
        pubData pubdata = new pubData(idPub, idUsuariobd,
tituloPub,
                                lugarEntregaPub, rubroPub, fechaFinPub,
descripcionPub,
                                statusPub);
        pubdatas.add(pubdata);
    }
    resultSet.close();
    statement.close();
    conn.close();
} catch (Exception e) {
    // TODO Auto-generated catch block
    System.err.print(e);
}
return pubdatas;
}

```

pubData.java /*model*/

```

public class pubData {
    int idPublicacion;
    int idUsuario;
    String titulo_pub;
    String lugarEntrega_pub;
    String rubro_pub;
    String fechaFin_pub;
    String descripcion_pub;
    String estado_pub;

    String fechaIniBusqueda;
    String fechaFinBusqueda;

    public pubData(int idPublicacion , String estadoPub) {
        super();
        this.idPublicacion = idPublicacion;
        estado_pub = estadoPub;
    }

    public pubData(int idUsuario, String estadoPub, String fechaIniBusqueda,

```

```

        String fechaFinBusqueda) {
    super();
    this.idUsuario = idUsuario;
    estado_pub = estadoPub;
    this.fechaIniBusqueda = fechaIniBusqueda;
    this.fechaFinBusqueda = fechaFinBusqueda;
}

public pubData(int idUsuario, String tituloPub, String lugarEntregaPub,
               String rubroPub, String fechaFinPub, String descripcionPub,
               String estadoPub) {
    super();
    this.idUsuario = idUsuario;
    titulo_pub = tituloPub;
    lugarEntrega_pub = lugarEntregaPub;
    rubro_pub = rubroPub;
    fechaFin_pub = fechaFinPub;
    descripcion_pub = descripcionPub;
    estado_pub = estadoPub;
}

public pubData(int idPublicacion, int idUsuario, String tituloPub,
               String lugarEntregaPub, String rubroPub, String fechaFinPub,
               String descripcionPub, String estadoPub) {
    super();
    this.idPublicacion = idPublicacion;
    this.idUsuario = idUsuario;
    titulo_pub = tituloPub;
    lugarEntrega_pub = lugarEntregaPub;
    rubro_pub = rubroPub;
    fechaFin_pub = fechaFinPub;
    descripcion_pub = descripcionPub;
    estado_pub = estadoPub;
}

}

```

2.3 Modelo de Datos

2.3.1 Estructura de la Base de Datos de Sistema

La persistencia física de los datos se realiza sobre una base de datos, la misma consta de diversas tablas, donde se aloja la información requerida para el funcionamiento correcto del sistema BidNex.

- Nombre de la base de datos: BidNex.
- Tipo de base de datos: MySQL.

2.3.2 Modelo de Datos y Tipos de Datos

