

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

PROGRAMACIÓN 3
9na. práctica (tipo b)
(Segundo Semestre 2024)

Indicaciones Generales:

- Tiempo estimado: 1h 50 minutos
- Se les recuerda que, de acuerdo al reglamento disciplinario de nuestra institución, constituye una falta grave copiar del trabajo realizado por otro estudiante o cometer plagio para el desarrollo de esta práctica.
- Para el desarrollo de toda la práctica debe utilizar el sistema operativo Windows, Apache NETBEANS 21, JDK 21, .NET Framework 4.8.1, el Visual Studio 2022 y GlassFish Server 7.0.18.
- Está permitido el uso de apuntes de clase, diapositivas, ejercicios de clase y código fuente. (Debe descargarlos antes de iniciar con la solución del enunciado)
- Está permitido el uso de Internet (únicamente para consultar páginas oficiales de Microsoft y Oracle). No obstante, está prohibida toda forma de comunicación con otros estudiantes o terceros.

PARTE PRÁCTICA (20 puntos)

PUEDE UTILIZAR MATERIAL DE CONSULTA.

Antes de comenzar el laboratorio, descargue todos los proyectos, apuntes, diapositivas que utilizará.

Se considerará en la calificación el uso de buenas prácticas de programación (aquellas vistas en clase).

PREGUNTA 1 (Parte Guiada) (12.0 puntos):

1. Descargar el **Eclipse GlassFish 7.0.18** desde cualquiera de los siguientes enlaces.

https://www.eclipse.org/downloads/download.php?file=/ee4j/glassfish/glassfish-7.0.18.zip&mirror_id=576

O

https://drive.google.com/file/d/1n-V1Q5MPZzZEbRCT7sh0HwtQwFBqYYWe/view?usp=drive_link

2. Descomprima el archivo "glassfish-7.0.18.zip" en la computadora:

3. Descargue las siguientes librerías (**Jakarta XML WS API 4.0.2** y **MySQL Connector 9.1.0**) desde los enlaces:

https://drive.google.com/file/d/1Kz2euHul4_pcuceeVu1IVo-FY65DJRzs/view?usp=drive_link

https://drive.google.com/file/d/1LM1HSp-4xWLNhXlcMVNCqZpiCofqTb-/view?usp=drive_link

4. Abrimos el Apache Netbeans 22 y nos dirigimos a la pestaña de "**Services**". (En caso no aparezca pueden habilitarla con **Window - Services**). En esa pestaña damos clic derecho a "**Servers**" y luego "**Add Server**". Seleccionamos "**GlassFish Server**" y le damos clic a "**Next**".

5. En "**Installation Location**" colocamos como ruta la carpeta glassfish7 donde descomprimos el archivo. Si es que ha colocado la carpeta glassfish7 dentro de otra carpeta, debe ubicar la ruta hasta que encuentre la carpeta glassfish7.

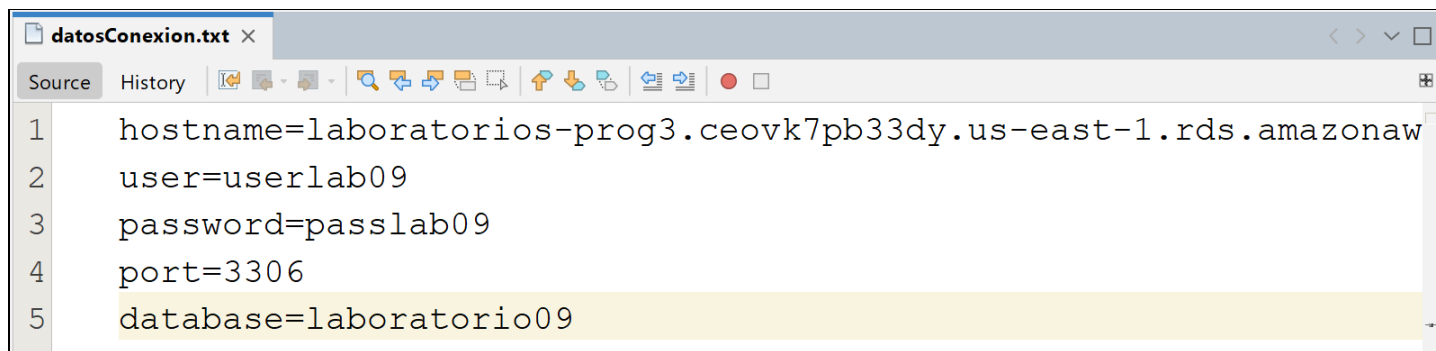
6. Aparecerá en la parte inferior un mensaje de que ha detectado una versión de glassfish y se habilitará el botón "**Next**".

7. En la pestaña de "**Domain Location**" dejamos la configuración por defecto y le damos clic a "**Finish**".

8. Con esta configuración realizada procedemos a descargar los proyectos en JAVA del Laboratorio 9 desde PAIDEIA y los abrimos con el Apache Netbeans 22. Son 4 proyectos: EventMasterSoftDA, EventMasterSoftDBManager, EventMasterSoftModel y EventMasterSoftWSBL.

9. Ejecute el SCRIPT SQL que se encuentra en PAIDEIA en su motor de base de datos de AWS Academy. El Script SQL genera un esquema llamado **laboratorio09**.

10. Ubique el archivo "**datosConexion.txt**" del proyecto EventMasterSoftWSBL y coloque los datos de conexión a su instancia de AWS Academy. En la Fig. 01 se muestra un ejemplo.



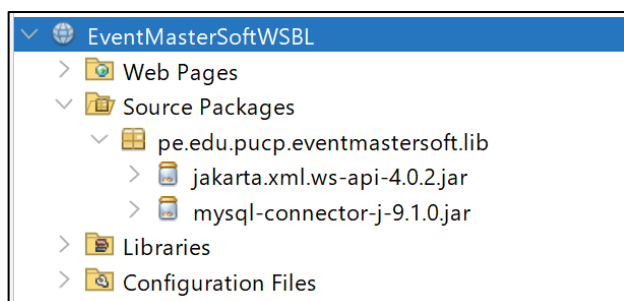
```
1 hostname=laboratorios-prog3.ceovk7pb33dy.us-east-1.rds.amazonaws
2 user=userlab09
3 password=passlab09
4 port=3306
5 database=laboratorio09
```

Fig. 01. Ejemplo de archivo "datosConexion.txt" con los datos asignados.

11. Ubique la clase ProductoraMySQL en el proyecto EventMasterSoftDA, e implemente el método "ArrayList<Productora> listarTodas()" utilizando los métodos definidos en el DBManager.

```
@Override
public ArrayList<Productora> listarTodas() {
    ArrayList<Productora> productoras = new ArrayList<>();
    rs = DBManager.getInstance().ejecutarProcedimientoLectura("LISTAR_PRODUCTORAS_TODAS", null);
    try{
        while(rs.next()){
            Productora productora = new Productora();
            productora.setIdProductora(rs.getInt("id_productora"));
            productora.setNombre(rs.getString("nombre"));
            productora.setActiva(true);
            productoras.add(productora);
        }
    }catch(SQLException ex){
        System.out.println("Error leyendo datos: " + ex.getMessage());
    }
    return productoras;
}
```

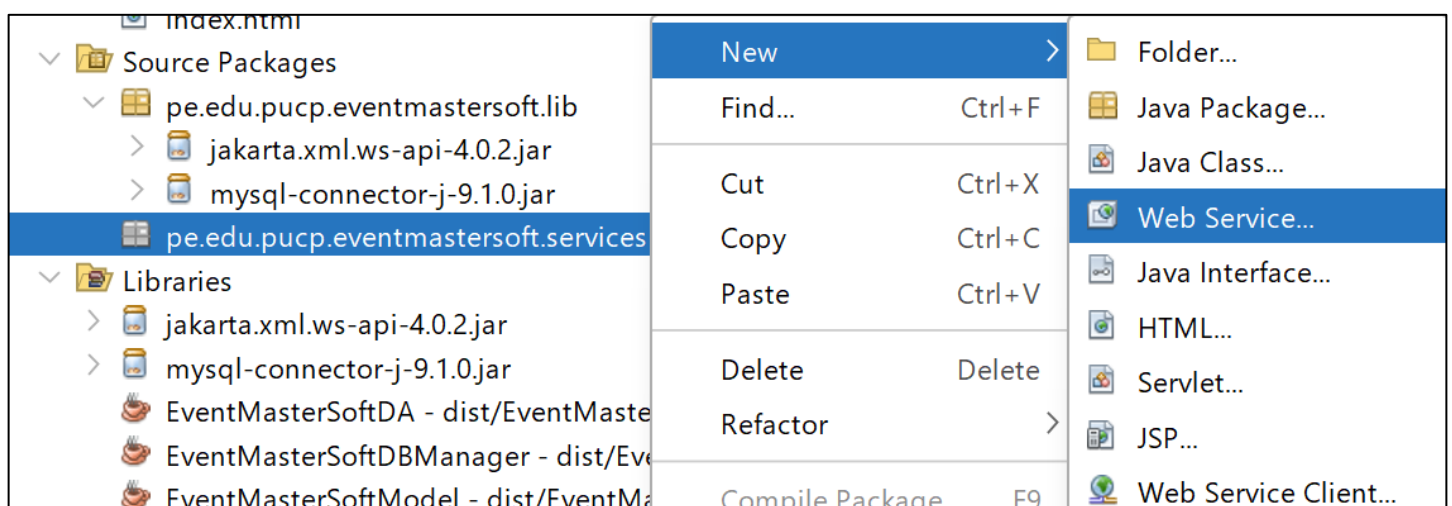
12. En el proyecto web, crearemos un paquete llamado "pe.edu.pucp.eventmastersoft.lib" y colocaremos ahí las dos librerías: "jakarta.xml.ws-api-4.0.2.jar" y "mysql-connector-j-9.1.0.jar".



13. Referenciamos esas librerías con ruta relativa en el proyecto EventMasterSoftWSBL.

14. Agregamos referencia a los proyectos "EventMasterSoftDBManager", "EventMasterSoftDA" y "EventMasterSoftModel" en el proyecto "EventMasterSoftWSBL".

15. En el proyecto web, ahora creamos un paquete llamado: "pe.edu.pucp.eventmastersoft.services" y creamos una clase de tipo "Web Service".



16. Colocamos de nombre "ProductoraWS" y le damos "Finish". Verificamos que se encuentre seleccionada la opción "Create Web Service from Scratch". Modificamos el contenido de la clase creada para que sea posible contar con un servicio que tenga una función que permita devolver todas las productoras desde la base de datos.

```
package pe.edu.pucp.eventmastersoft.services;

import jakarta.jws.WebService;
import jakarta.jws.WebMethod;
import jakarta.jws.WebParam;
import java.util.ArrayList;
import pe.edu.pucp.eventmastersoft.dao.ProductoraDAO;
import pe.edu.pucp.eventmastersoft.model.Productora;
import pe.edu.pucp.eventmastersoft.mysql.ProductoraMySQL;

@WebService(serviceName = "ProductoraWS", targetNamespace =
"http://services.eventmastersoft.pucp.edu.pe")
public class ProductoraWS {

    private ProductoraDAO daoProductora;

    @WebMethod(operationName = "listarTodasProductoras")
    public ArrayList<Productora> listarTodasProductoras() {
        ArrayList<Productora> productoras = null;
        try{
            daoProductora = new ProductoraMySQL();
            productoras = daoProductora.listarTodas();
        }catch(Exception ex){
            System.out.println(ex.getMessage());
        }
        return productoras;
    }
}
```

17. Ahora procedemos a implementar el método `listarPorNombre(String nombre)` de la clase `EventoMySQL`.

```
@Override
public ArrayList<Evento> listarPorNombre(String nombre) {
    ArrayList<Evento> eventos = new ArrayList<>();
    HashMap<String,Object> parametrosEntrada = new HashMap<>();
    parametrosEntrada.put("_nombre", nombre);
    rs = DBManager.getInstance().ejecutarProcedimientoLectura("LISTAR_EVENTOS_X_NOMBRE",
parametrosEntrada);
    try{
        while(rs.next()){
            Evento evento = new Evento();
            evento.setIdEvento(rs.getInt("id_evento"));
            evento.setNombre(rs.getString("nombre_evento"));
            evento.setFechaRealizacion(rs.getDate("fecha_realizacion"));
            Productora productora = new Productora();
            productora.setIdProductora(rs.getInt("id_productora"));
            productora.setNombre(rs.getString("nombre_productora"));
            evento.setProductora(productora);
            eventos.add(evento);
        }
    }catch(SQLException ex){
        System.out.println("Error leyendo datos: " + ex.getMessage());
    }
    return eventos;
}
```

20. Volvemos a crear una clase de tipo "Web Service" llamada "EventoWS" en el paquete "pe.edu.pucp.eventmastersoft.services" del proyecto EventMasterSoftWSBL, y modificamos el contenido para que exista un método que permita listar los eventos de la base de datos por nombre.

```
package pe.edu.pucp.eventmastersoft.services;

import jakarta.jws.WebService;
import jakarta.jws.WebMethod;
import jakarta.jws.WebParam;
import java.util.ArrayList;
import pe.edu.pucp.eventmastersoft.dao.EventoDAO;
import pe.edu.pucp.eventmastersoft.model.Evento;
import pe.edu.pucp.eventmastersoft.mysql.EventoMySQL;

@WebService(serviceName = "EventoWS", targetNamespace =
"http://services.eventmastersoft.pucp.edu.pe")
public class EventoWS {

    private EventoDAO daoEvento;

    @WebMethod(operationName = "listarEventosPorNombre")
    public ArrayList<Evento> listarEventosPorNombre(@WebParam(name = "nombre") String nombre) {
        ArrayList<Evento> eventos = null;
        try{
            daoEvento = new EventoMySQL();
            eventos = daoEvento.listarPorNombre(nombre);
        }catch(Exception ex){
            System.out.println(ex.getMessage());
        }
        return eventos;
    }
}
```

21. Ejecutamos el proyecto Web.

Probamos en la página de Tester que los métodos de listado estén devolviendo información de la base de datos:

<http://localhost:8080/EventMasterSoftWSBL/ProductoraWS?Tester>

<http://localhost:8080/EventMasterSoftWSBL/EventoWS?Tester>

listarTodasProductoras Method invocation

Method parameter(s)

Type	Value

Method returned

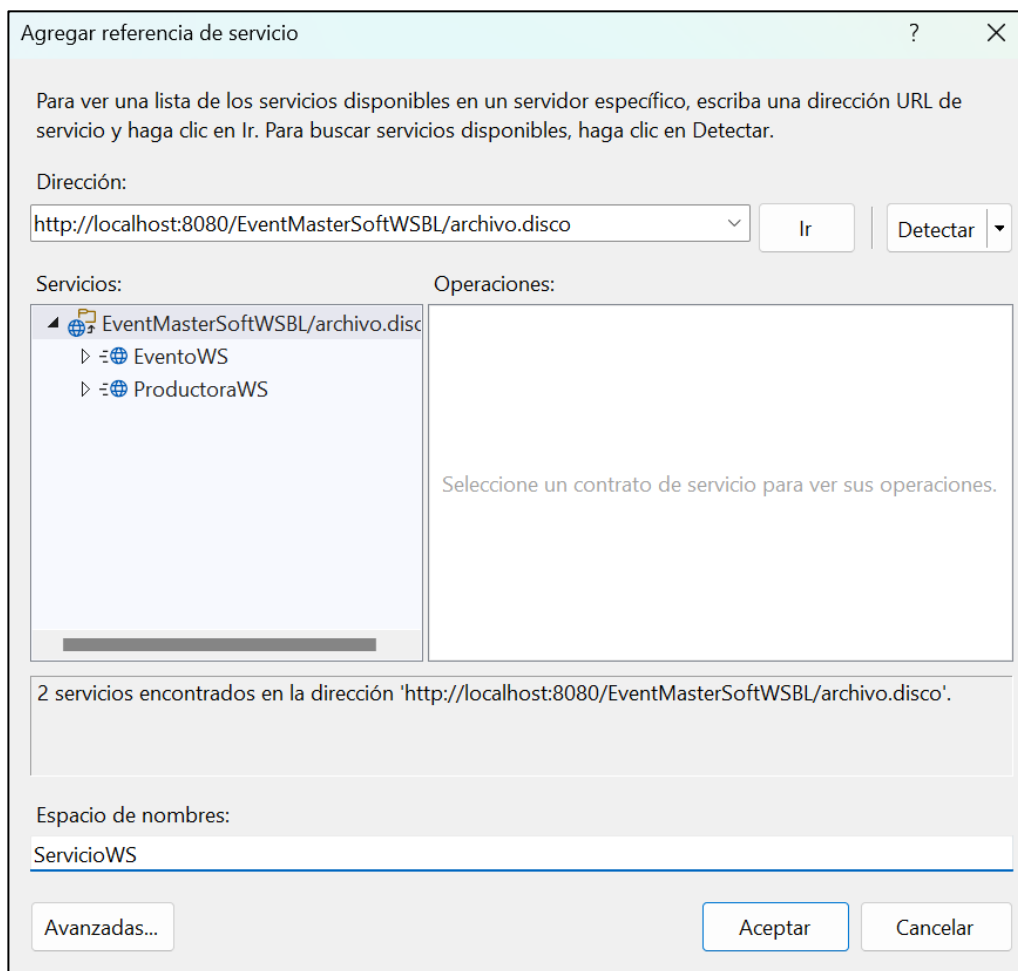
java.util.List : "[pe.edu.pucp.eventmastersoft.services.Productora@cbe64a1, pe.edu.pucp.eventmastersoft.services.Productora@1b3444c2, pe.edu.pucp.eventmastersoft.services.Productora@6702b02d, pe.edu.pucp.eventmastersoft.services.Productora@5245e7e, pe.edu.pucp.eventmastersoft.services.Productora@7a40a917, pe.edu.pucp.eventmastersoft.services.Productora@43ebf0c9, pe.edu.pucp.eventmastersoft.services.Productora@311aa341, pe.edu.pucp.eventmastersoft.services.Productora@16eadcb2, pe.edu.pucp.eventmastersoft.services.Productora@691dc769]"

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:listarTodasProductoras xmlns:ns2="http://services.eventmastersoft.pucp.edu.pe/" />
  </S:Body>
</S:Envelope>
```

SOAP Response

22. Abrimos el proyecto Front-end con Visual Studio 2022.
 23. Coloquen sus datos completos en las clases "ListarEventos.aspx.cs" y "RegistrarEvento.aspx.cs".
 24. Clic derecho a Referencias y "Agregar referencia de servicio".
 25. Colocar la siguiente dirección: <http://localhost:8080/EventMasterSoftWSBL/archivo.disco>
(Asegúrese de que el proyecto web de Netbeans se está ejecutando)
- Colocamos como espacio de nombres "ServicioWS" y Aceptamos.



26. Ubique el archivo "Web.config" y agregue las instrucciones a cada "binding" para dar soporte a la transferencia de archivos entre el cliente y el servidor.

```
<system.serviceModel>
  <bindings>
    <basicHttpBinding>
      <binding name="ProductoraWSPortBinding" receiveTimeout="00:1:00" sendTimeout="00:1:00"
maxBufferSize="2147483647" maxReceivedMessageSize="2147483647"/>
      <binding name="EventoWSPortBinding" receiveTimeout="00:1:00" sendTimeout="00:1:00"
maxBufferSize="2147483647" maxReceivedMessageSize="2147483647"/>
    </basicHttpBinding>
  </bindings>
  ...

```

27. Implemente lo siguiente en el código C# de la página ListarEventos.

```
public partial class ListarEventos : System.Web.UI.Page
{
    private EventoWSClient daoEvento = new EventoWSClient();
    protected void Page_Load(object sender, EventArgs e)
    {
        gvEventos.DataSource = daoEvento.listarEventosPorNombre(txtNombre.Text);
        gvEventos.DataBind();
    }
    ....
}
```

28. Implemente lo siguiente en el código C# de la página RegistrarEvento.

```
public partial class RegistrarEvento : System.Web.UI.Page
{
    private ProductoraWSCClient daoProductora = new ProductoraWSCClient();
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack) {
            ddlProductora.DataSource = daoProductora.listarTodasProductoras();
            ddlProductora.DataTextField = "nombre";
            ddlProductora.DataValueField = "idProductora";
            ddlProductora.DataBind();
        }
        ....
    }
    ....
}
```

29. Verifique que funciona tanto el listado de productoras como de eventos.

PREGUNTA 2 (Parte NO Guiada) (5.0 puntos):

30. Siguiendo los pasos realizados para el listado de productoras y eventos implemente todo lo necesario a nivel de front y back para que sea posible el registro de eventos.

Utilice lo siguiente en la clase **EventoMySQL**:

```
@Override
public int insertar(Evento evento) {
    HashMap<String,Object> parametrosEntrada = new HashMap<>();
    parametrosEntrada.put("_fid_productora", evento.getProductora().getIdProductora());
    parametrosEntrada.put("_fid_clasificacion", evento.getClasificacion());
    parametrosEntrada.put("_nombre", evento.getNombre());
    parametrosEntrada.put("_costo_realizacion", evento.getCostoRealizacion());
    parametrosEntrada.put("_tipo_evento", evento.getTipoEvento());
    parametrosEntrada.put("_fecha_realizacion", evento.getFechaRealizacion());
    parametrosEntrada.put("_descripcion", evento.getDescripcion());
    parametrosEntrada.put("_permite_reingreso", evento.isPermiteReingreso());
    parametrosEntrada.put("_permite_grabacion", evento.isPermiteGrabacion());
    parametrosEntrada.put("_banner_promocional", evento.getBannerPromocional());

    HashMap<String,Object> parametrosSalida = new HashMap<>();
    parametrosSalida.put("_id_evento", Types.INTEGER);

    DBManager.getInstance().ejecutarProcedimiento("INSERTAR_EVENTO", parametrosEntrada,
    parametrosSalida);
    return (int) parametrosSalida.get("_id_evento");
}
```

Utilice lo siguiente en la clase **EventoWS**:

```
@WebMethod(operationName = "insertarEvento")
public int insertarEvento(@WebParam(name = "evento") Evento evento) {
    int resultado = 0;
    try{
        daoEvento = new EventoMySQL();
        resultado = daoEvento.insertar(evento);
    }catch(Exception ex){
        System.out.println(ex.getMessage());
    }
    return resultado;
}
```

Utilice lo siguiente en la clase **RegistrarEvento.aspx.cs**.

```
protected void lbGuardar_Click(object sender, EventArgs e)
{
    evento evento = new evento();
    evento.nombre = txtNombreEvento.Text;
    evento.productora = new productora();
    evento.productora.idProductora = Int32.Parse(ddlProductora.SelectedValue);
    evento.descripcion = txtDescripcion.Value;
    evento.fechaRealizacion = DateTime.Parse(dtpFechaRealizacion.Value);
    evento.fechaRealizacionSpecified = true;
    evento.clasificacion =
        (rbAdultos.Checked ? 'A' : (rbJovenes.Checked ? 'J' : (rbNinhos.Checked ? 'N' : 'T')));
    evento.permiteGrabacion = cbGrabacion.Checked;
    evento.permiteReingreso = cbReingreso.Checked;
    evento.costoRealizacion = Double.Parse(txtCostoRealizacion.Text);
    evento.tipoEvento = rbConcierto.Checked ? tipoEvento.CONCIERTO : tipoEvento.OBRA_TEATRAL;
    evento.tipoEventoSpecified = true;
    evento.bannerPromocional = (byte[]) Session["foto"];
    int resultado = daoEvento.insertarEvento(evento);
    if (resultado != 0)
        Response.Redirect("ListarEventos.aspx");
}
```

PREGUNTA 2 (Parte NO Guiada) (3.0 puntos):

31. Siguiendo los pasos realizados para el listado de productoras y eventos implemente todo lo necesario a nivel de front y back para que sea posible la obtención y visualización de un evento.

Utilice el siguiente código en la clase **EventoMySQL**

```
@Override
public Evento obtenerPorId(int idEvento) {
    Evento evento = new Evento();
    HashMap<String, Object> parametrosEntrada = new HashMap<>();
    parametrosEntrada.put("_id_evento", idEvento);
    rs = DBManager.getInstance().ejecutarProcedimientoLectura("OBTENER_EVENTO_X_ID",
parametrosEntrada);
    try{
        if(rs.next()){
            evento.setIdEvento(rs.getInt("id_evento"));
            evento.setClasificacion(rs.getString("id_clasificacion").charAt(0));
            evento.setCostoRealizacion(rs.getDouble("costo_realizacion"));
            evento.setTipoEvento(TipoEvento.valueOf(rs.getString("tipo_evento")));
            evento.setNombre(rs.getString("nombre_evento"));
            evento.setDescripcion(rs.getString("descripcion"));
            evento.setBannerPromocional(rs.getBytes("banner_promocional"));
            evento.setPermiteGrabacion(rs.getBoolean("permite_grabacion"));
            evento.setPermiteReingreso(rs.getBoolean("permite_reingreso"));
            evento.setFechaRealizacion(rs.getDate("fecha_realizacion"));
            Productora productora = new Productora();
            productora.setIdProductora(rs.getInt("id_productora"));
            productora.setNombre(rs.getString("nombre_productora"));
            evento.setProductora(productora);
            evento.setActivo(true);
        }
    }catch(SQLException ex){
        System.out.println("Error leyendo datos: " + ex.getMessage());
    }
    return evento;
}
```

Utilice el siguiente código en la clase **EventoWS**

```
@WebMethod(operationName = "obtenerEventoPorId")
public Evento obtenerEventoPorId(@WebParam(name = "idEvento") int idEvento) {
    Evento evento = null;
    try{
        daoEvento = new EventoMySQL();
        evento = daoEvento.obtenerPorId(idEvento);
    }catch(Exception ex){
        System.out.println(ex.getMessage());
    }
    return evento;
}
```

Utilice el siguiente código en la clase **RegistrarEvento.aspx.cs**

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack) {
        ddlProductora.DataSource = daoProductora.listarTodasProductoras();
        ddlProductora.DataTextField = "nombre";
        ddlProductora.DataValueField = "idProductora";
        ddlProductora.DataBind();
    }

    string idEvento = Request.QueryString["idEvento"];
    string accion = Request.QueryString["accion"];
    if (accion != null && accion == "ver" && idEvento != null)
    {
        lblTitulo.Text = "Visualizar Evento";
        evento evento = daoEvento.obtenerEventoPorId(Int32.Parse(idEvento));
        string base64String = Convert.ToBase64String(evento.bannerPromocional);
        string imageUrl = "data:image/jpeg;base64," + base64String;
        imgBannerPromocional.ImageUrl = imageUrl;
        dtpFechaRealizacion.Value = evento.fechaRealizacion.ToString("yyyy-MM-dd");
        txtDescripcion.Value = evento.descripcion;
        txtNombreEvento.Text = evento.nombre;
        txtCostoRealizacion.Text = evento.costoRealizacion.ToString("N2");
        if (evento.clasificacion.Equals('A') ? rbAdultos.Checked = true :
        (evento.clasificacion.Equals('J') ? rbJovenes.Checked = true : (evento.clasificacion.Equals('N')
        ? rbNinhos.Checked = true : rbTodos.Checked = true))) ;
        ddlProductora.SelectedValue = evento.productora.idProductora.ToString();
        if (evento.tipoEvento == tipoEvento.OBRA_TEATRAL ? rbObraTeatral.Checked = true :
        rbConcierto.Checked = true) ;
        txtIDEvento.Text = evento.idEvento.ToString();
        cbReingreso.Checked = evento.permiteReingreso;
        cbGrabacion.Checked = evento.permiteGrabacion;
        Deshabilitar_Componentes();
    }
    else
    {
        lblTitulo.Text = "Registrar Evento";
        Cargar_Foto(sender, e);
    }
}
```

La revisión será progresiva.

23 de octubre del 2024