

JSF 2.0 (Java Server Faces)

A decorative graphic consisting of a solid teal horizontal bar that spans the width of the slide. Below this bar, on the right side, there are several horizontal lines of varying lengths and colors, including teal and white, creating a layered, stepped effect.

Aportaciones de Jsf:

- Unión entre los cliente Web actuales y el modelo tradicional de componentes.
- **Separa la renderización del componente.**
- Desplazar la interacción del usuario con el programa al modelo tradicional de **eventos**.
- **JSF es la evolución estándar de Struts**
- Es un estándar claro y potente para poder hacer aplicaciones visuales mas potentes.

Arquitectura

- Renderización, árbol de componentes visuales, validadores, conversores y eventos y javabeans.
- Aporta un ciclo de vida claro y estándar.
- Permite configurar y definir externamente el flujo de pantallas, la navegación.
- Permite modificar o incorporar componentes básicos propios.

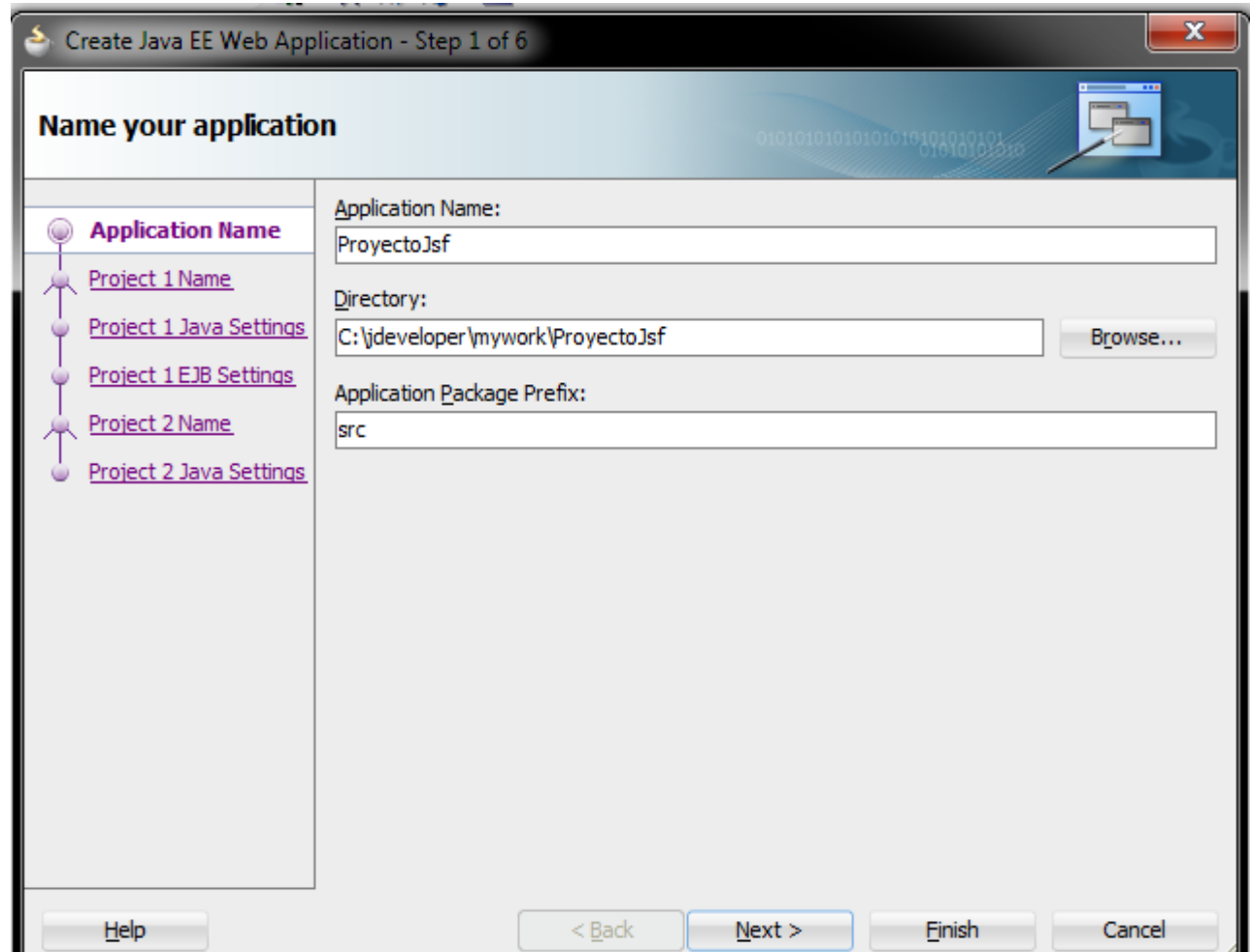
Faces-config.xml

- La etiqueta principal es `<faces-config>`, en donde podemos tener mas de un archivo de configuración y podemos gestionar:
 - Java beans.
 - Registrar validadores propios.
 - Registrar conversores propios.
 - Configurar reglas de navegación.
 - Registrar Renders.
 - Registrar componentes propios.

▫

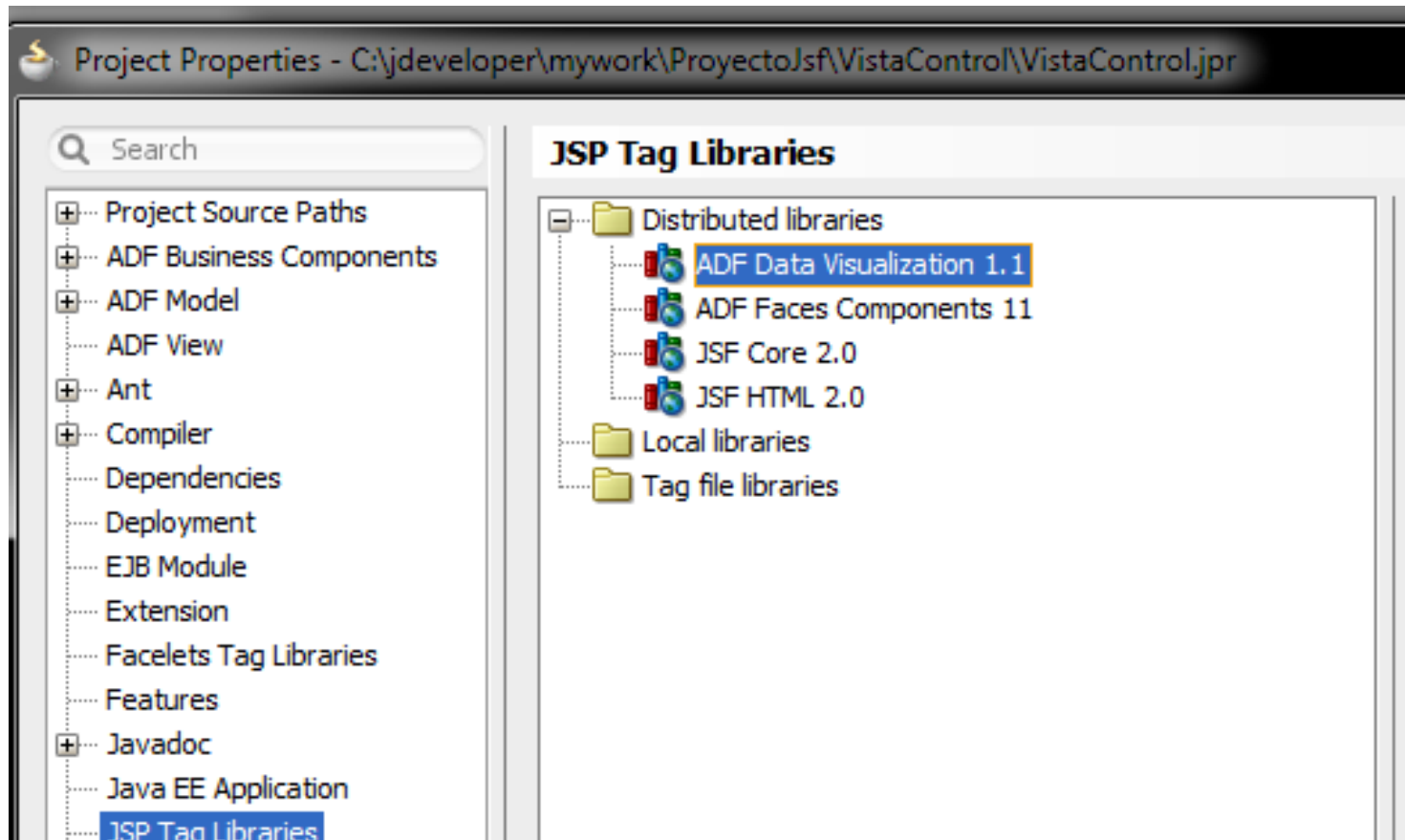
JSF en Jdeveloper

1. New Application
2. Java EE Web Application (estructura Modelo-Vista-Contrôlador)



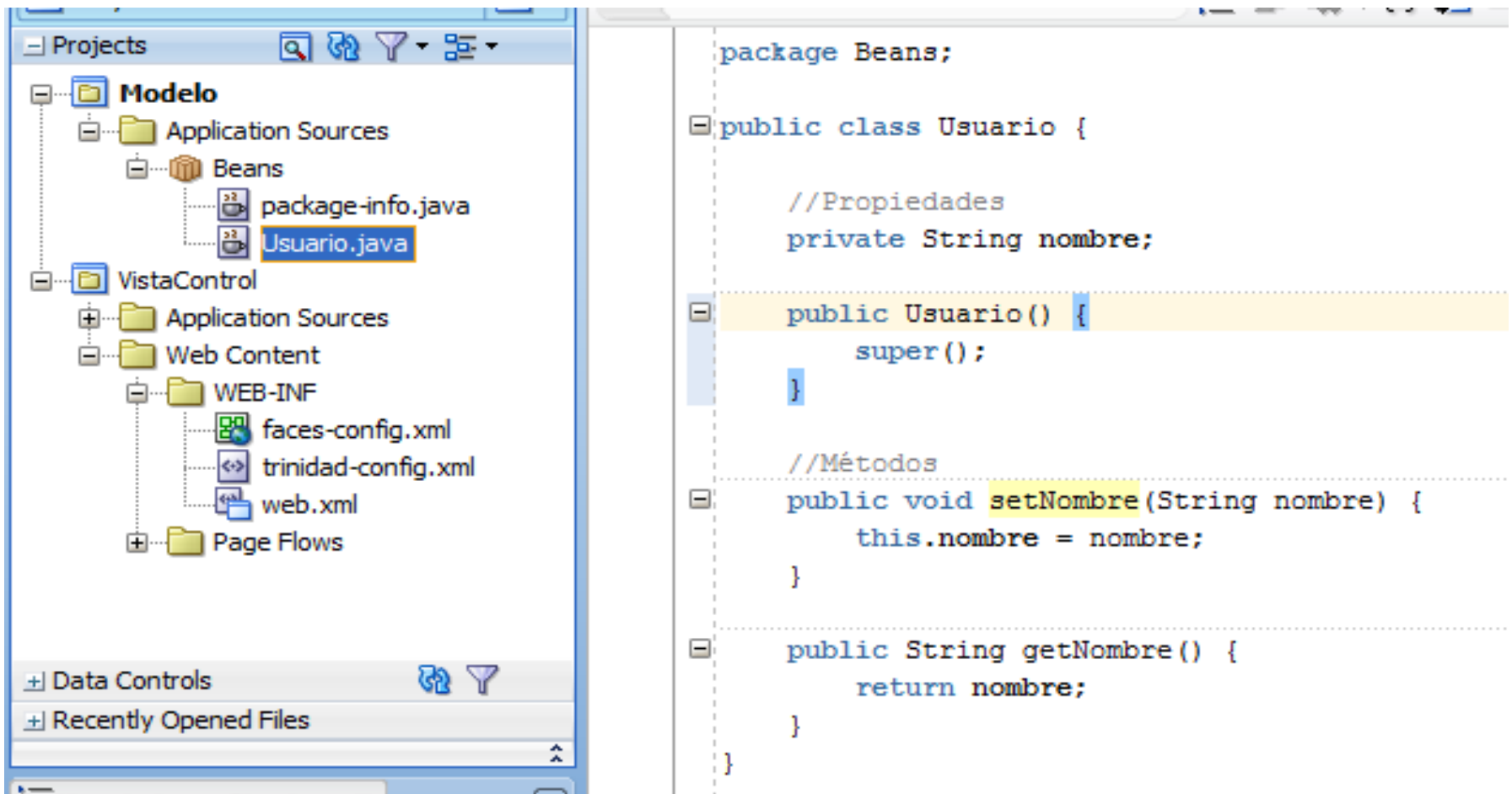
Incluyendo librerías al proyecto

- Clic derecho en el proyecto VistaControl->Project Properties->Jsp Tag Libraries->Add (ADF Data Visualization 1.1 y ADF Faces Components 11)



Agregando el Modelo

- Crear un Paquete en donde estarán los Java Beans
- Crear un bean para un Usuario, con sus propiedades y métodos necesarios.



The image shows a screenshot of an IDE with two panels. The left panel displays a project structure under 'Projects'. The 'Modelo' package is expanded, showing 'Application Sources' and 'Beans'. Under 'Beans', there are 'package-info.java' and 'Usuario.java'. The 'VistaControl' package is also visible, containing 'Application Sources', 'Web Content', and 'Page Flows'. The right panel shows the code for the 'Usuario' class in the 'Beans' package. The code includes a constructor and two methods: 'setNombre' and 'getNombre'.

```
package Beans;

public class Usuario {

    //Propiedades
    private String nombre;

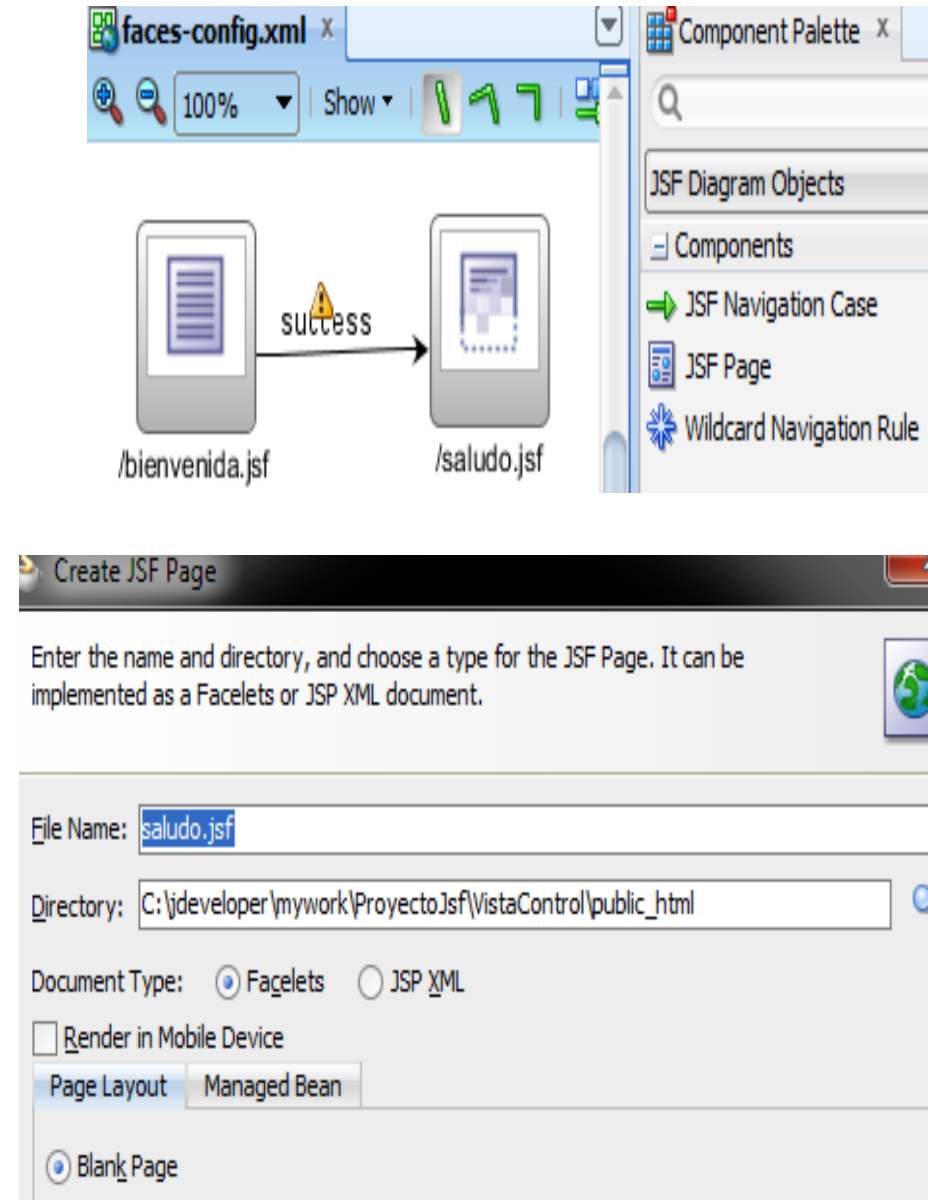
    public Usuario() {
        super();
    }

    //Métodos
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getNombre() {
        return nombre;
    }
}
```

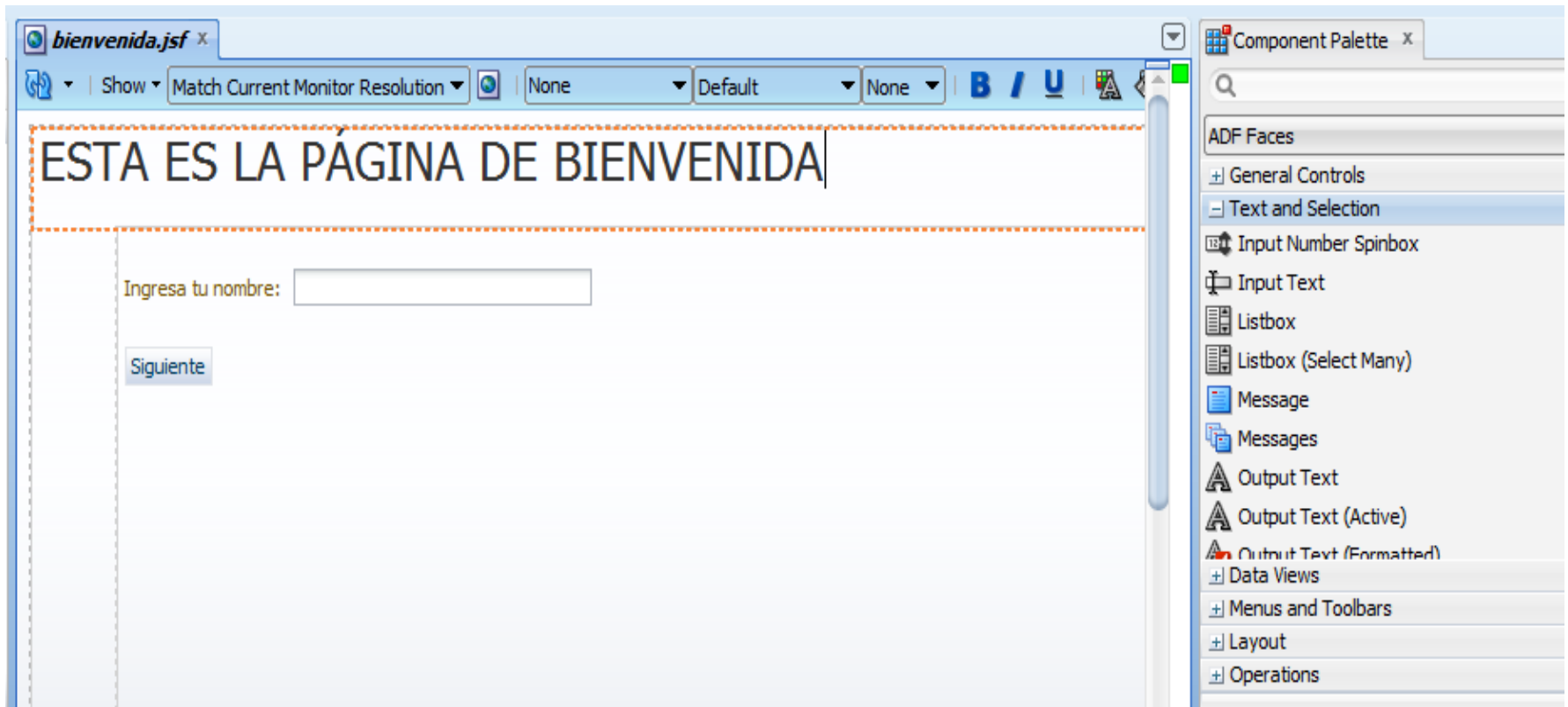
Creando la navegación

- Tendremos una página de bienvenida, en donde solicitamos el nombre al usuario y lo redirigimos a una página de saludo.
1. Abrir el archivo faces-config.xml
 2. Agregar dos JSF Pages para las respectivas vistas
 3. Agregar un JSF navigation case que vaya de la página de bienvenida a la de saludo
 4. Para crear los jsf basta con dar doble clic sobre las vistas



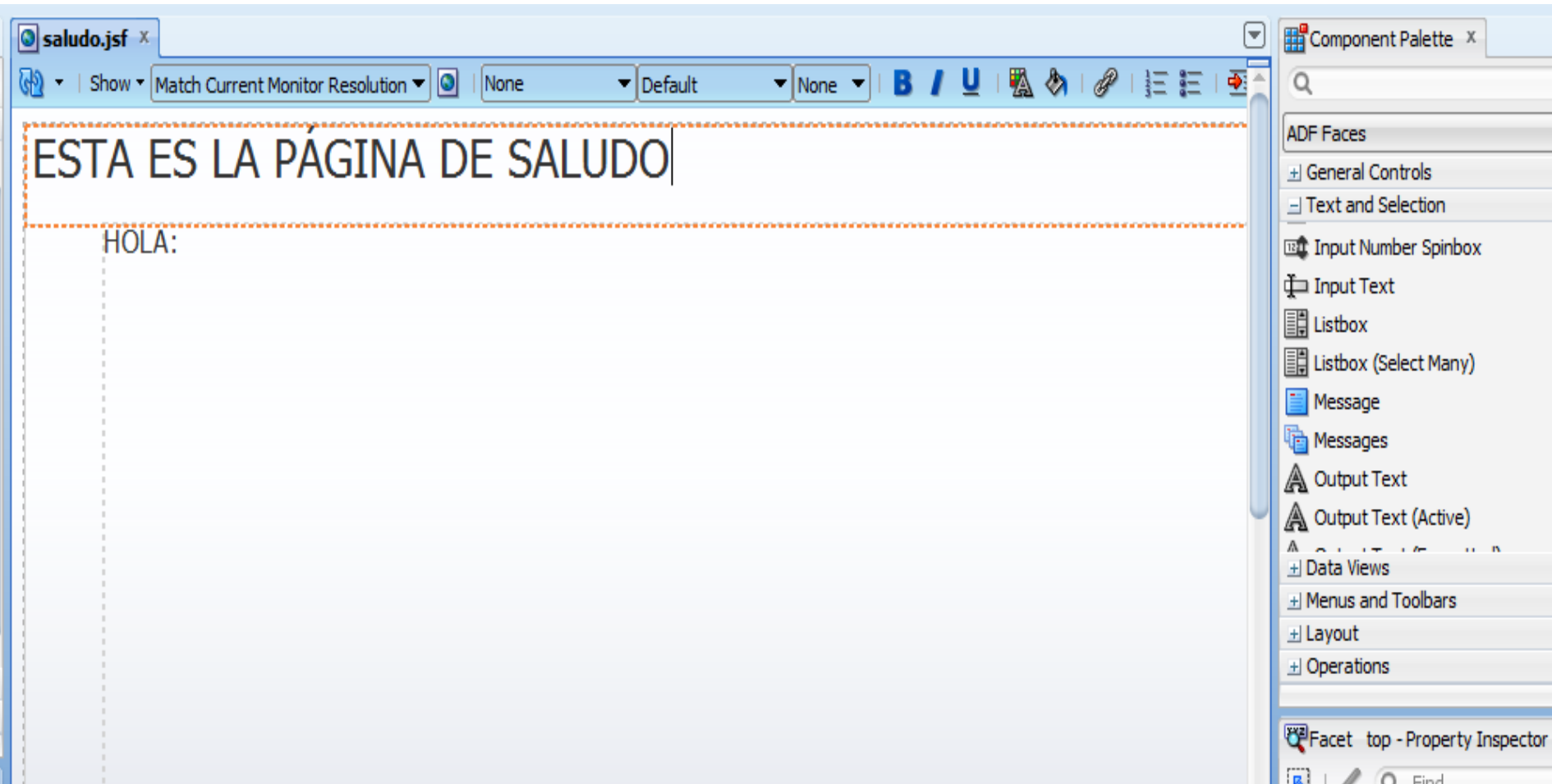
Creando las Vistas:

- Para la página de Bienvenida arrastraremos un `inputText` para solicitar el nombre del usuario y `Button` para redirigirlo a la siguiente página.



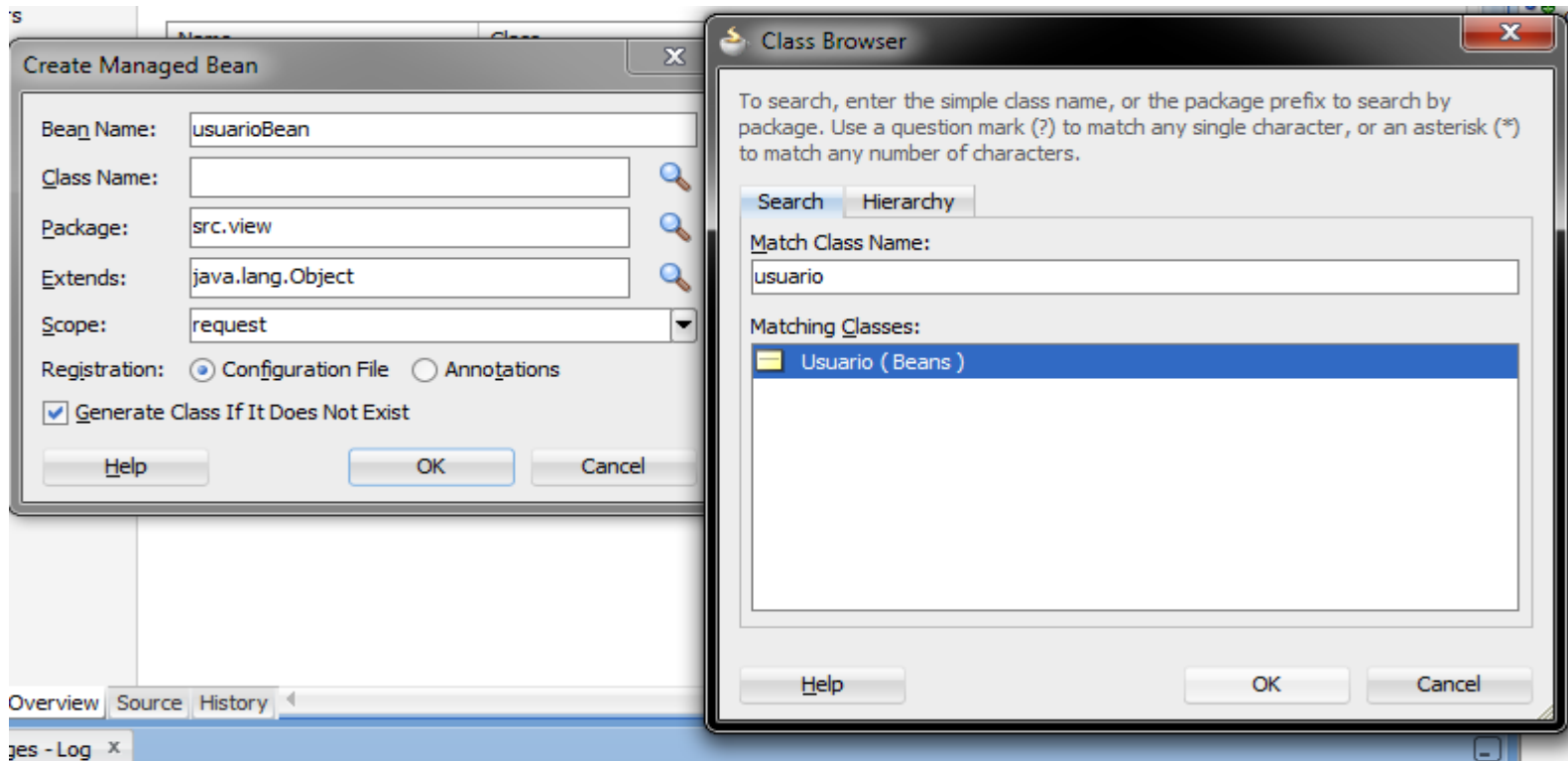
Creando las Vistas:

- Para la página de Saludo arrastraremos un `outputText` para mostrar el saludo mas el nombre del usuario



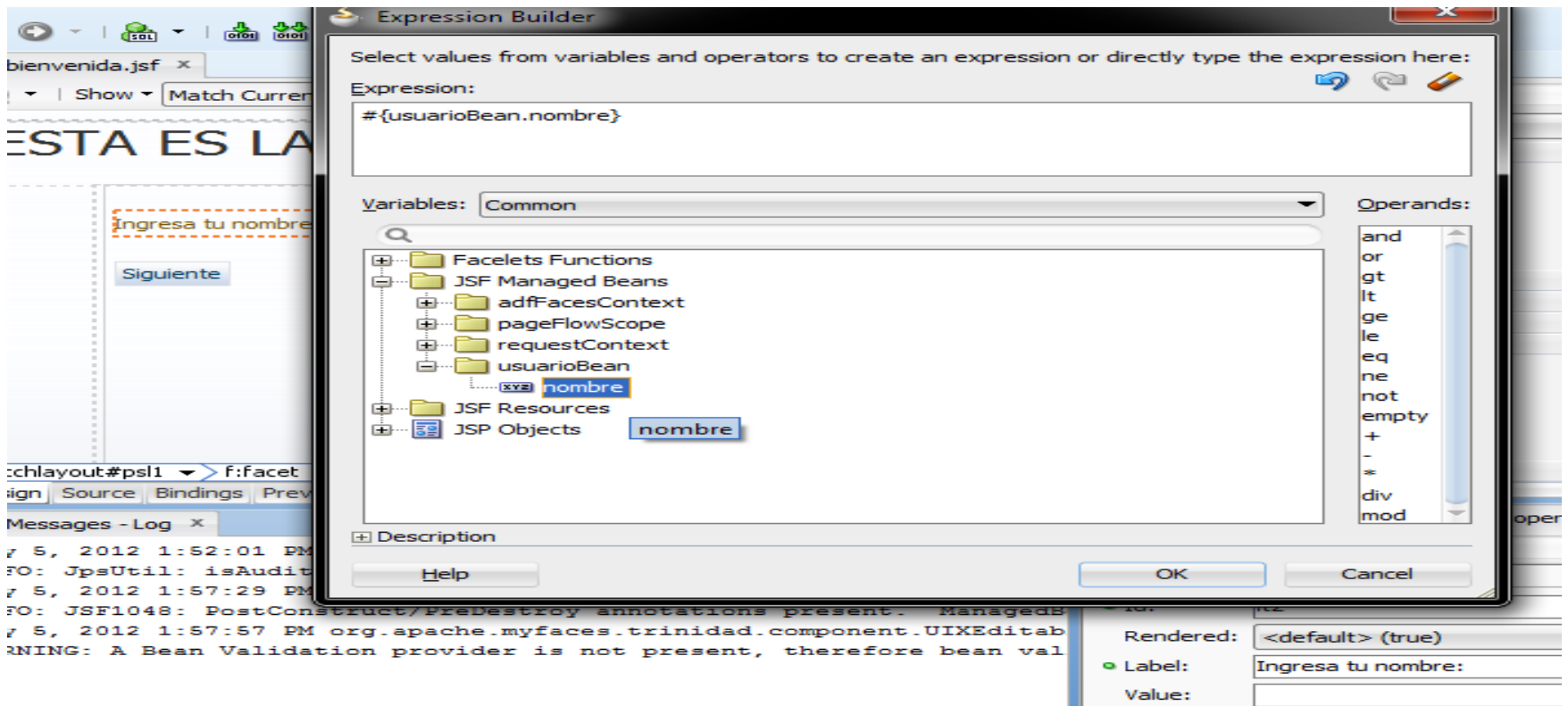
Registrar los Java Beans

1. Abrir el archivo faces-config.xml
2. Seleccionar la vista Overview
3. Agregar un Managed Bean
 - Bean Name: Nombre para el bean
 - Class Name: La clase java, en nuestro caso *usuario*
 - Scope: Session
 - Registration: Configuration File



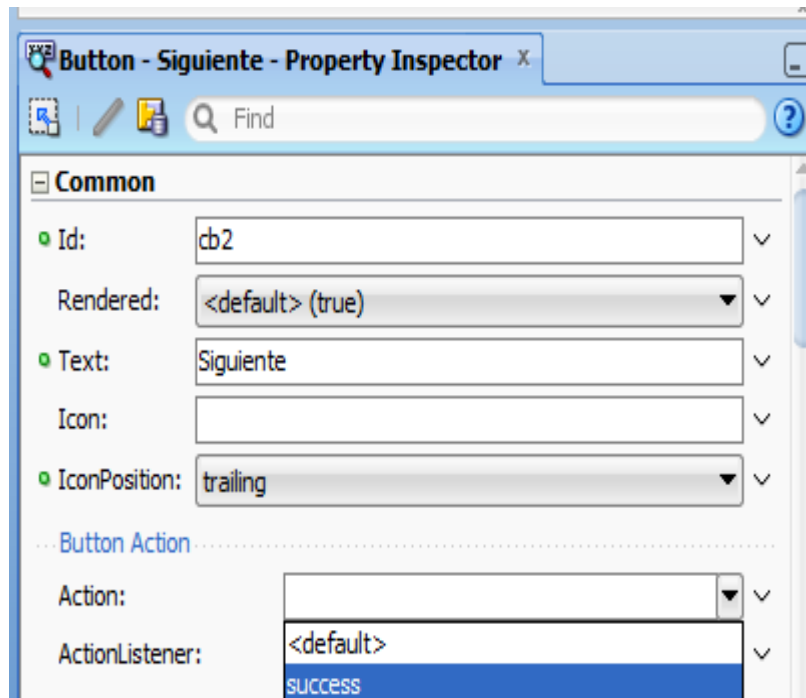
Relacionar las Vistas con el Bean

- En la página de Bienvenida
 1. Seleccionar el inputText
 2. En el inspector de propiedades
 3. En la propiedad *value*->Expression Builder
 4. Seleccionar la propiedad *nombre* de nuestro Bean

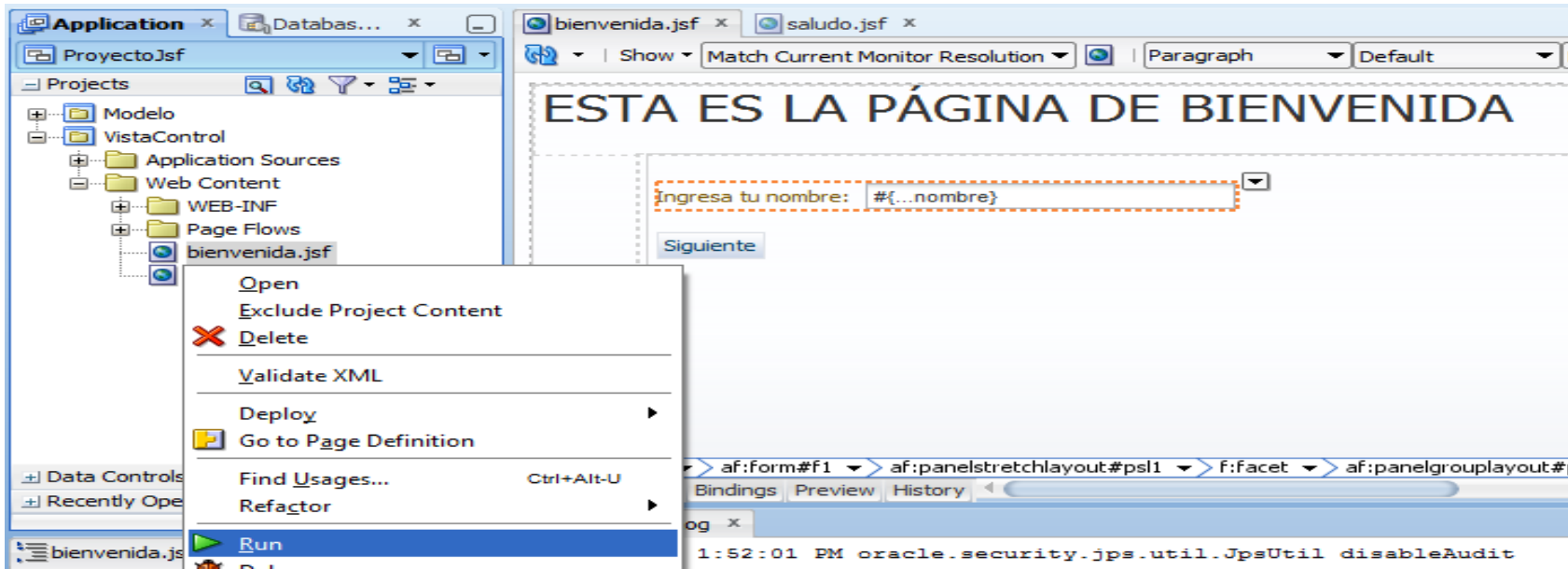


Relacionar las Vistas con el Bean

5. En la propiedad action del commandButton poner “success”
6. En la página de Saludo hacer lo mismo con el outputText
7. Ejecutamos la página de Bienvenida



Ejecutando la aplicación:



ESTA ES LA PÁGINA DE BIENVENIDA

Ingres tu nombre:

Siguiente

ESTA ES LA PÁGINA DE SALUDO

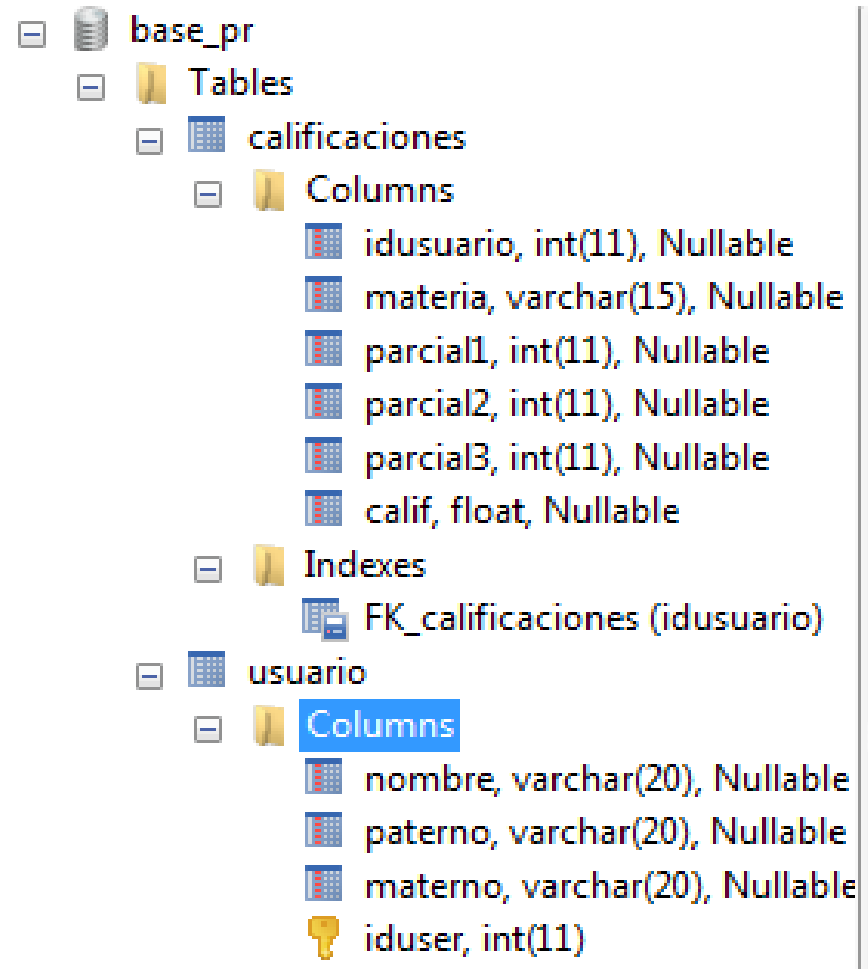
Hola Pepito

EJB (Enterprise JavaBeans)

- Son una arquitectura de componentes de servidor que simplifica el proceso de construcción de aplicaciones de componentes empresariales distribuidos en Java
- Con su utilización es posible escribir aplicaciones escalables, fiables y seguras sin escribir código de infraestructura

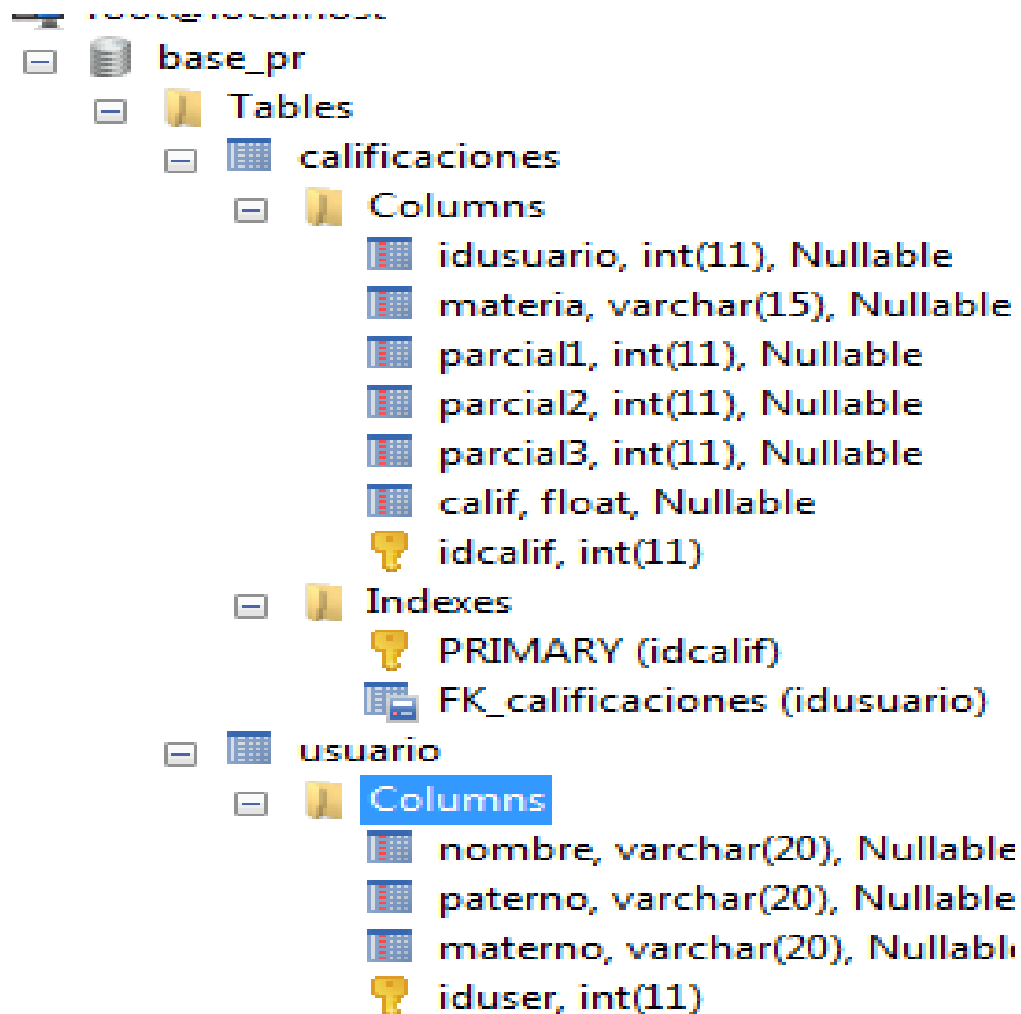
Creando una aplicación con EJB's

- Crear la base de datos, agregar una tabla usuario y calificaciones.
- Crear un usuario específico para esta conexión



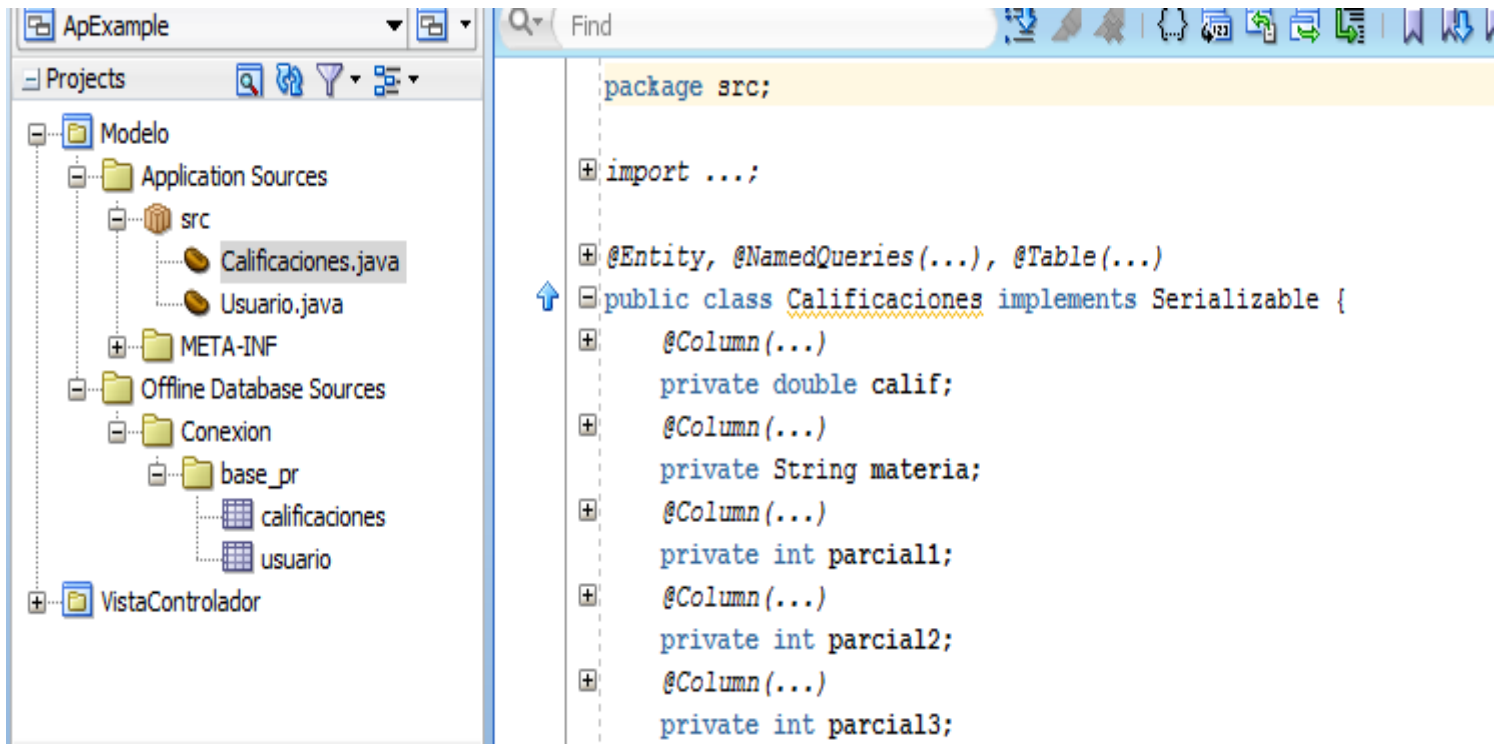
Creando una nueva conexión

- En Jdeveloper, seleccionar View-> DataBase->Database Navigator
- Clic derecho en IDE Connections->New Connection
- Llenar los datos y no olvidar ubicar el driver de MySql o SQL Server
- Arrastrar la conexión creada hasta nuestro proyecto



Generar las entidades desde las tablas

- Clic derecho sobre el proyecto Modelo->New->Entities from tables
- Seguir el asistente:
 - Type of connection: Online database connection
 - Select tables: seleccionar query y agregar las tablas necesarias
 - Finish



The screenshot shows an IDE interface. On the left, the 'Projects' view displays a tree structure for a project named 'ApExample'. Under 'Modelo', there is an 'Application Sources' folder containing a 'src' folder with 'Calificaciones.java' and 'Usuario.java'. Below that is a 'META-INF' folder, followed by 'Offline Database Sources' containing a 'Conexion' folder and a 'base_pr' folder with 'calificaciones' and 'usuario' tables. At the bottom is a 'VistaControlador' folder. On the right, the 'Find' view shows the content of 'Calificaciones.java'. The code is as follows:

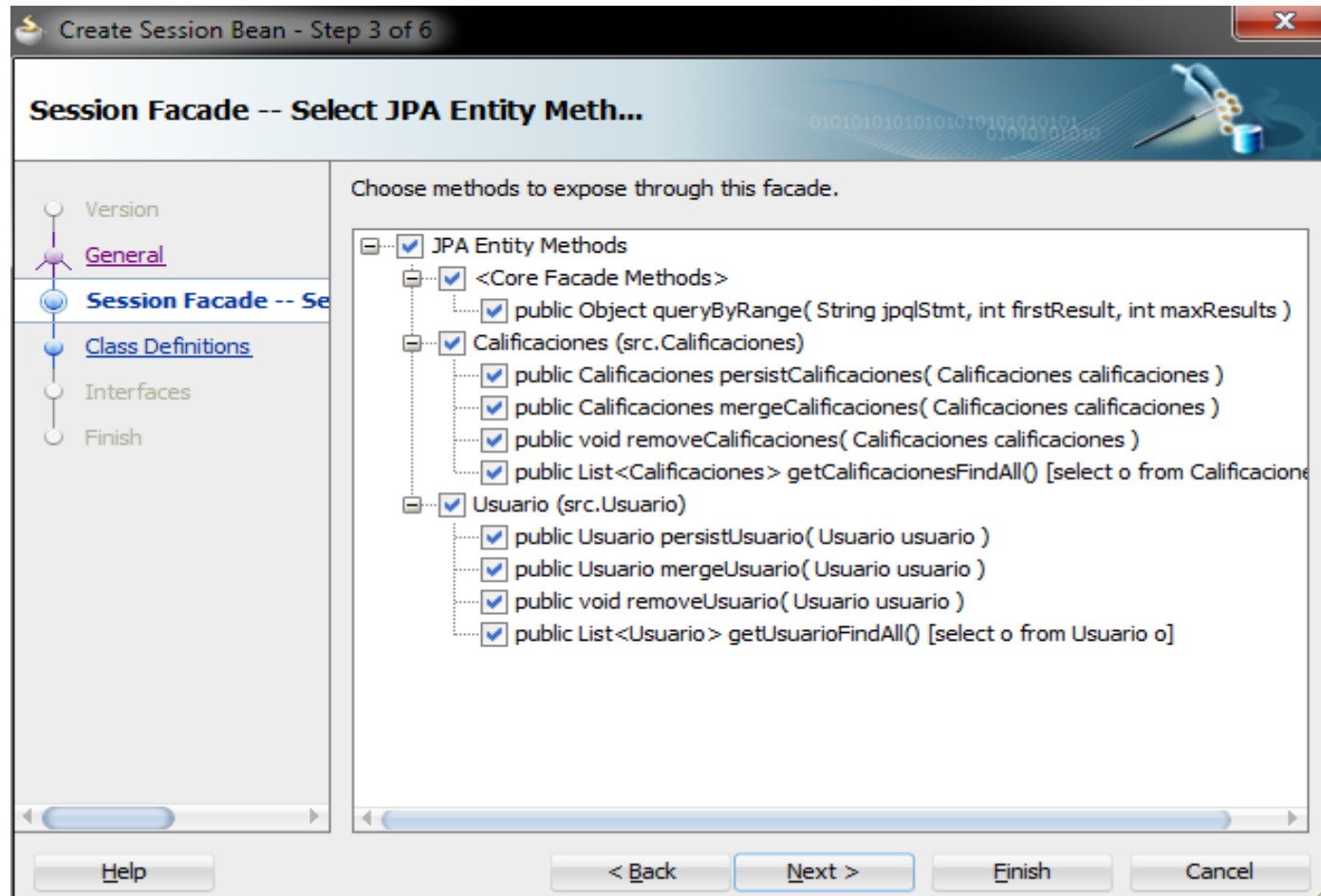
```
package src;

import ...;

@Entity, @NamedQueries(...), @Table(...)
public class Calificaciones implements Serializable {
    @Column(...)
    private double calif;
    @Column(...)
    private String materia;
    @Column(...)
    private int parcial1;
    @Column(...)
    private int parcial2;
    @Column(...)
    private int parcial3;
```

Crear un diagrama de EJB

- Clic derecho sobre el proyecto Modelo
- New-> EJB Diagram
- Arrastrar los beans (usuario y calificaciones) al diagrama
- Agregar un Session Bean



Application x Ru... x

ApExample

Projects

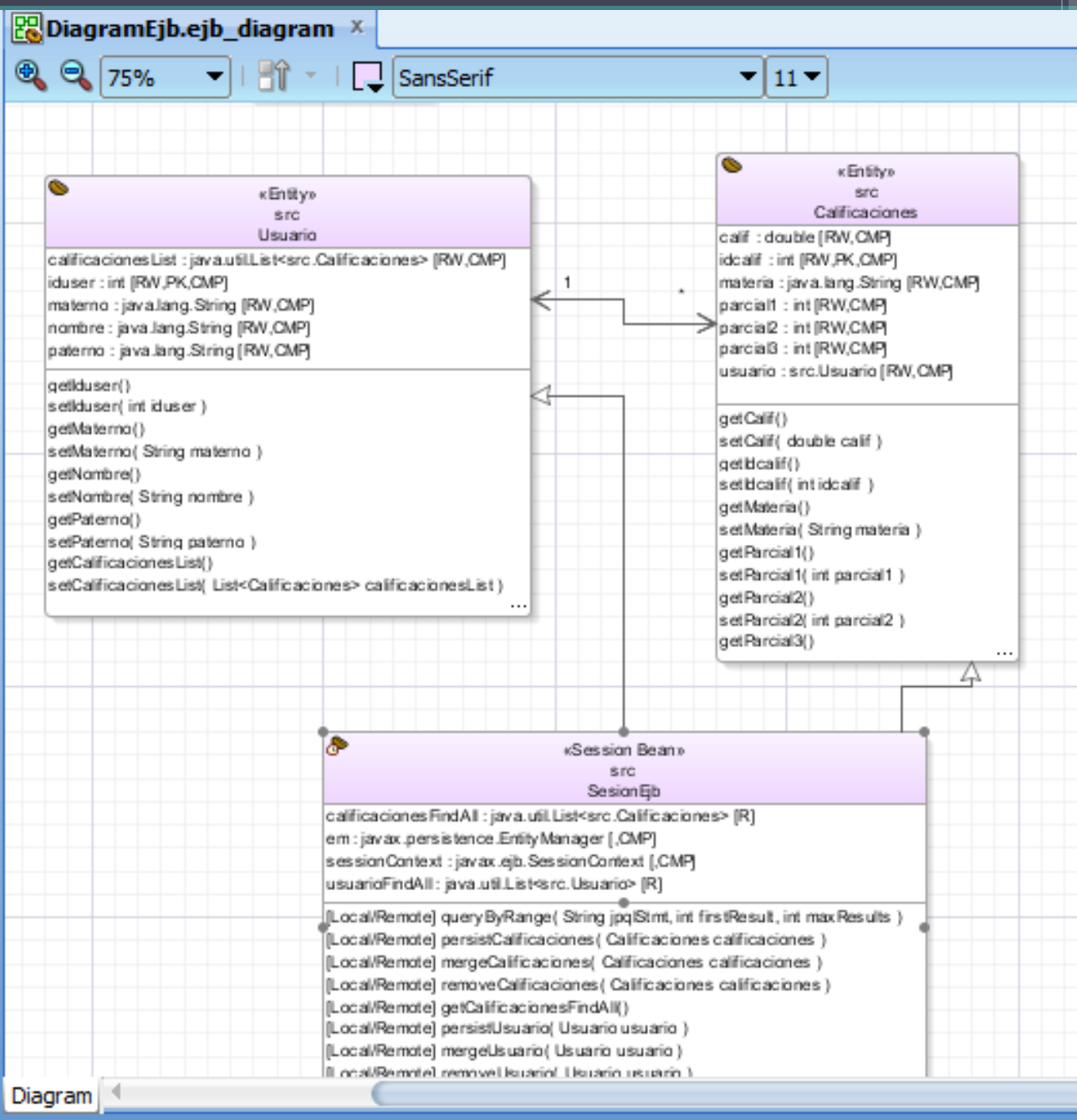
- Modelo
 - Application Sources
 - src
 - Calificaciones.java
 - DiagramEjb.ejb_diagram
 - SesionEjb.java
 - SesionEjbBean.java
 - SesionEjbLocal.java
 - Usuario.java
 - META-INF
 - adfm.xml
 - persistence.xml
 - weblogic-ejb-jar.xml
 - Offline Database Sources

Recently Opened Files

- DiagramEjb...
- Thumbnail

Calificaciones : Entity

Calificaciones-Usuario : *-1 EJB Rela



Agregando consultas propias

- Las consultas con nombre permiten definir consultas en tiempo de diseño y luego usarlos en tiempo de ejecución. El asistente crea un NamedQuery en la entidad Usuario. Esta consulta recupera todas las filas de la tabla Usuario.
- Abrir el bean de usuario
- Agregar el siguiente código:

```
@Entity
```

```
@NamedQueries({
```

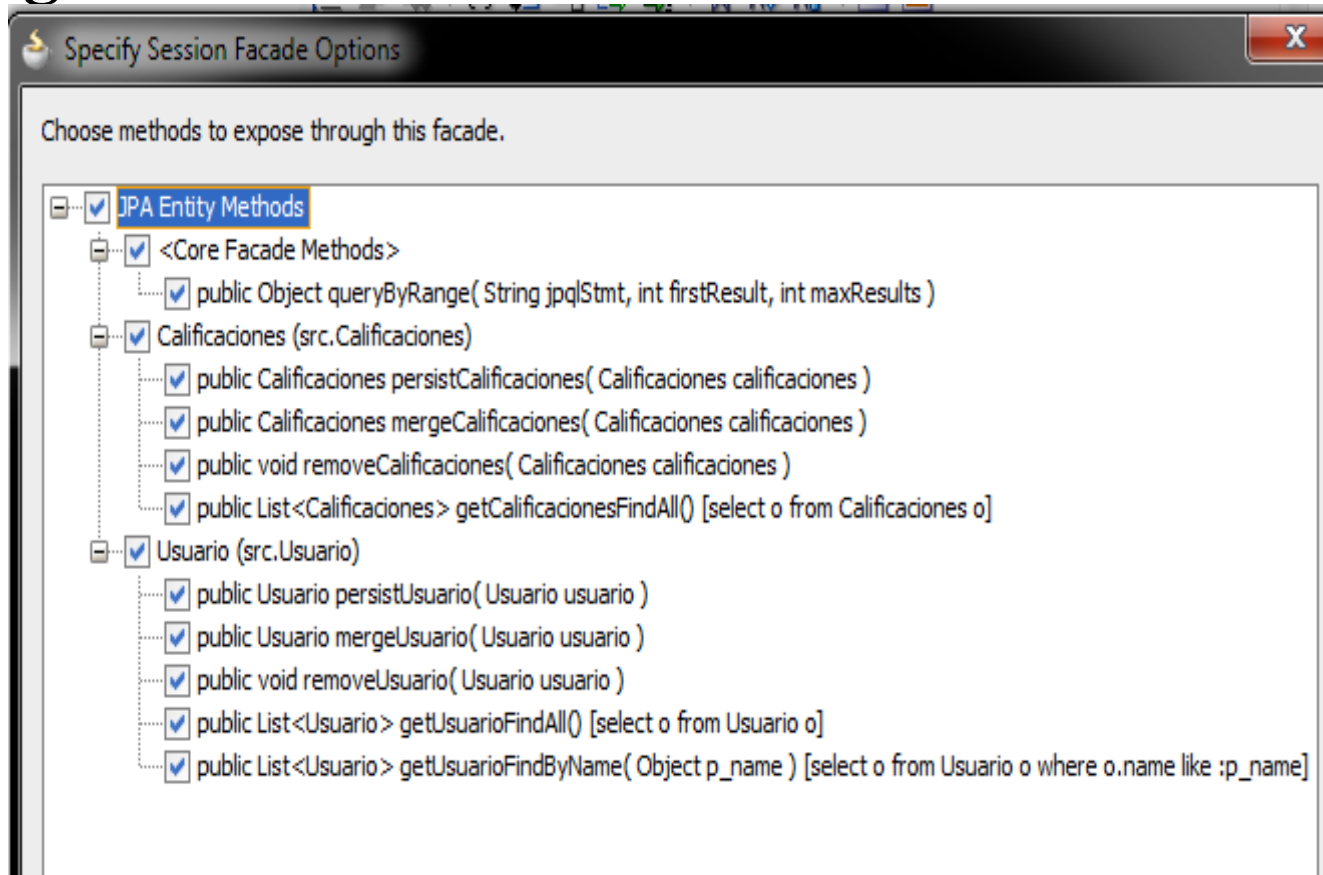
```
    @NamedQuery(name = "Usuario.findAll", query = "select o  
from Usuario o"),
```

```
    @NamedQuery(name = "Usuario.findByName", query =  
"select o from Usuario o where o.name like :p_name") })
```

- Compilar la clase Usuario.java.

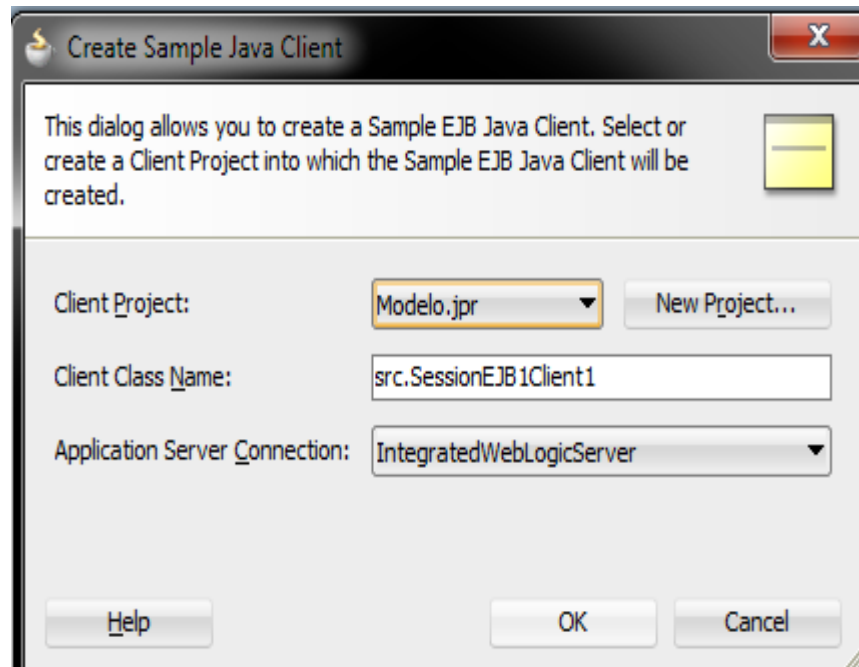
Agregando consultas propias

- Para añadir el nuevo método a la Sesión hacer clic derecho SesionEjbBean-> Edit Session Facade
- Agregar el nuevo método



Probando los Ejb's

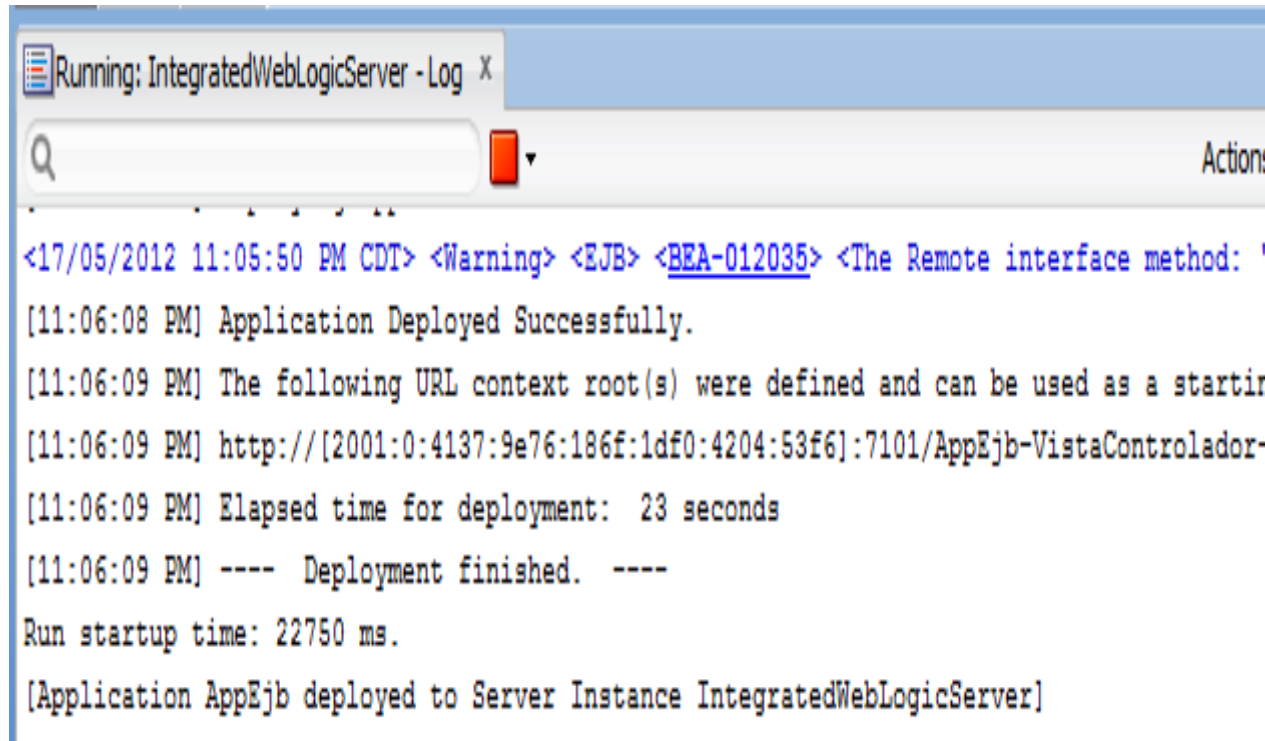
- Clic derecho sobre SesionEjbBean->New Sample Java Client.
- Corregir el error en el método getUsuarioFindByName, agregando el parámetro requerido.



```
}  
for (Usuario usuario :  
    (List<Usuario>)sessionEJB1.getUsuarioFindByName("P%") /* F:  
    printUsuario(usuario);  
}
```

Probando los Ejb's

- Ejecutar SesionEjbBean, esperar el WebLogicServer se ha iniciado.

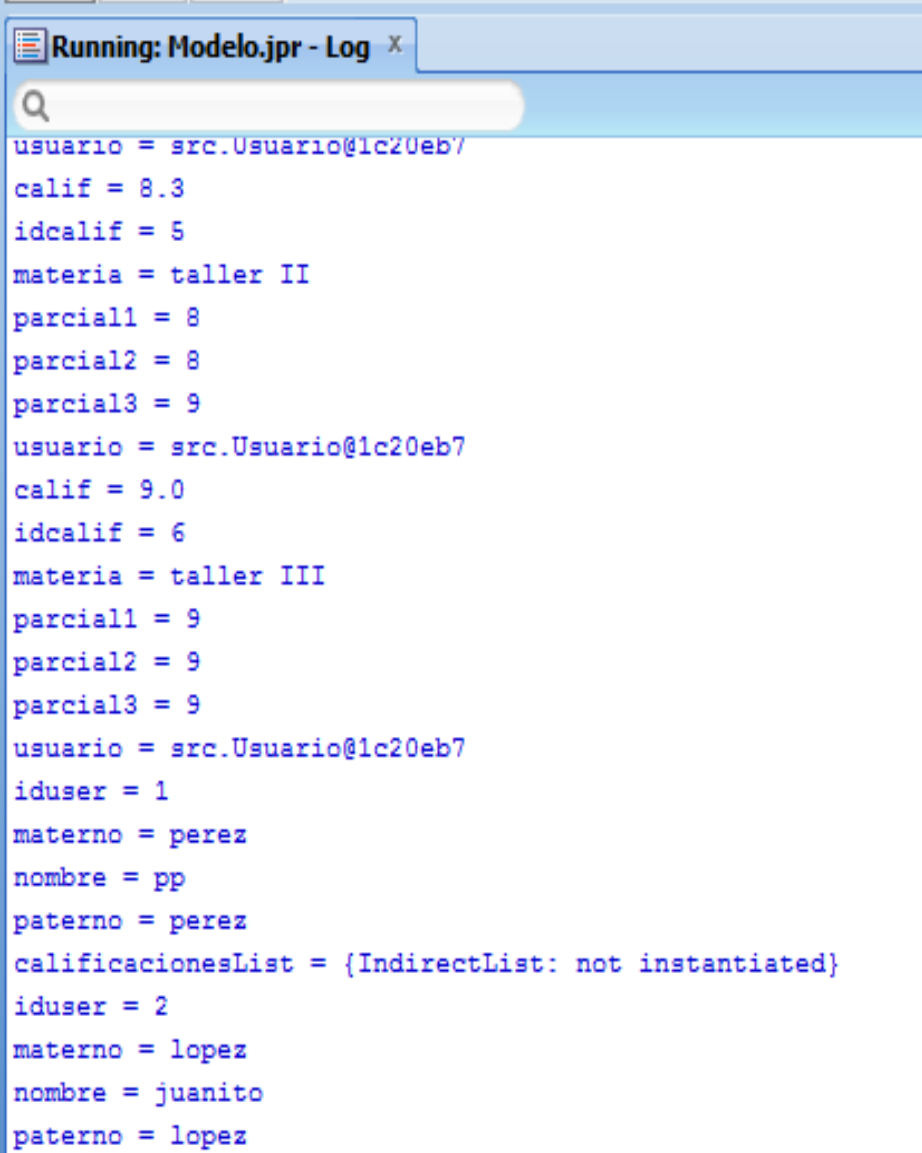


The screenshot shows a log window titled "Running: IntegratedWebLogicServer - Log". The log contains the following text:

```
<17/05/2012 11:05:50 PM CDT> <Warning> <EJB> <BEA-012035> <The Remote interface method: '  
[11:06:08 PM] Application Deployed Successfully.  
[11:06:09 PM] The following URL context root(s) were defined and can be used as a startin  
[11:06:09 PM] http://[2001:0:4137:9e76:186f:1df0:4204:53f6]:7101/AppEjb-VistaControlador-  
[11:06:09 PM] Elapsed time for deployment: 23 seconds  
[11:06:09 PM] ---- Deployment finished. ----  
Run startup time: 22750 ms.  
[Application AppEjb deployed to Server Instance IntegratedWebLogicServer]
```


Probando los Ejb's

- Ejecutar SesionEjebClient



```
Running: Modelo.jpr - Log x
usuario = src.Usuario@1c20eb7
calif = 8.3
idcalif = 5
materia = taller II
parcial1 = 8
parcial2 = 8
parcial3 = 9
usuario = src.Usuario@1c20eb7
calif = 9.0
idcalif = 6
materia = taller III
parcial1 = 9
parcial2 = 9
parcial3 = 9
usuario = src.Usuario@1c20eb7
iduser = 1
materno = perez
nombre = pp
paterno = perez
calificacionesList = {IndirectList: not instantiated}
iduser = 2
materno = lopez
nombre = juanito
paterno = lopez
```

Crear control de datos para la Sesión EJB

- Clic derecho sobre SesionEjbBean
- Create Data Control

The screenshot displays the Oracle ADF IDE interface. On the left, the 'Projects' pane shows a tree view of the application structure, including 'Modelo', 'Application Sources', 'src', and 'META-INF'. The 'src' folder contains several files, including 'SesionEjbLocal.java'. The main workspace shows the 'Data Control Registry' for 'SesionEjbLocal'. The registry lists various data controls, including 'calificacionesFindAll', 'usuarioFindAll', and several merge and persist operations. The 'Attributes' tab is selected, showing a table with columns for 'Name' and 'Type'.

Data Control Registry

This lists the Oracle ADF data controls published by this project, including a metadata summary. To edit the data control definition or [enable customization](#), select a Data controls node and click Edit.

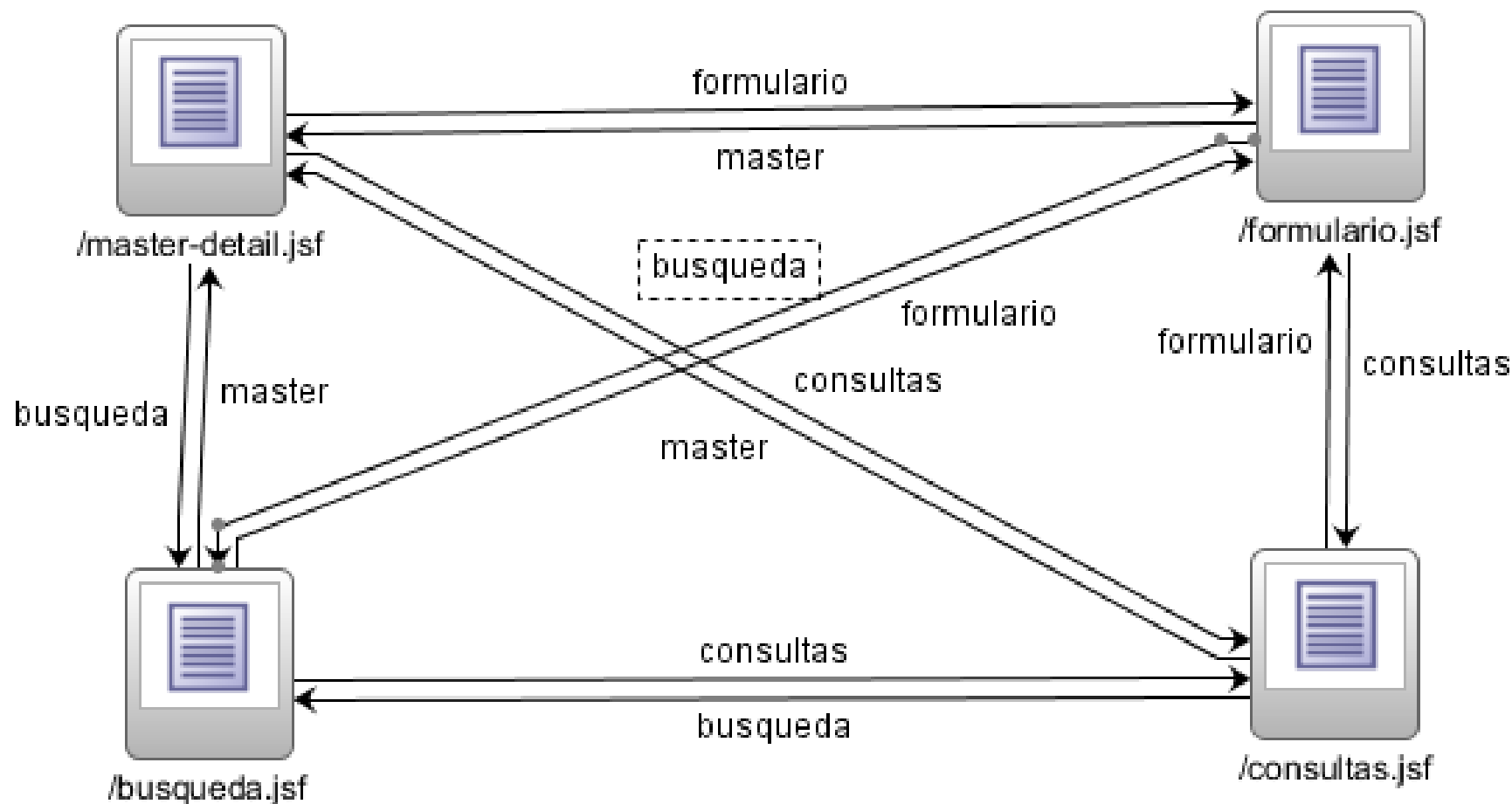
Data Controls

- calificacionesFindAll
- usuarioFindAll
- mergeCalificaciones(Calificaciones)
- mergeUsuario(Usuario)
- persistCalificaciones(Calificaciones)
- persistUsuario(Usuario)
- removeCalificaciones(Calificaciones)
- removeUsuario(Usuario)

Attributes		Accessors	Operations
Name	Type		

Creando la Vista:

- Diseñar el siguiente mapa de navegación



Creando las páginas Jsf:

- Doble clic sobre la vista
- Elegir Quick Start Layout->Browse->One colum, aplicar tema.

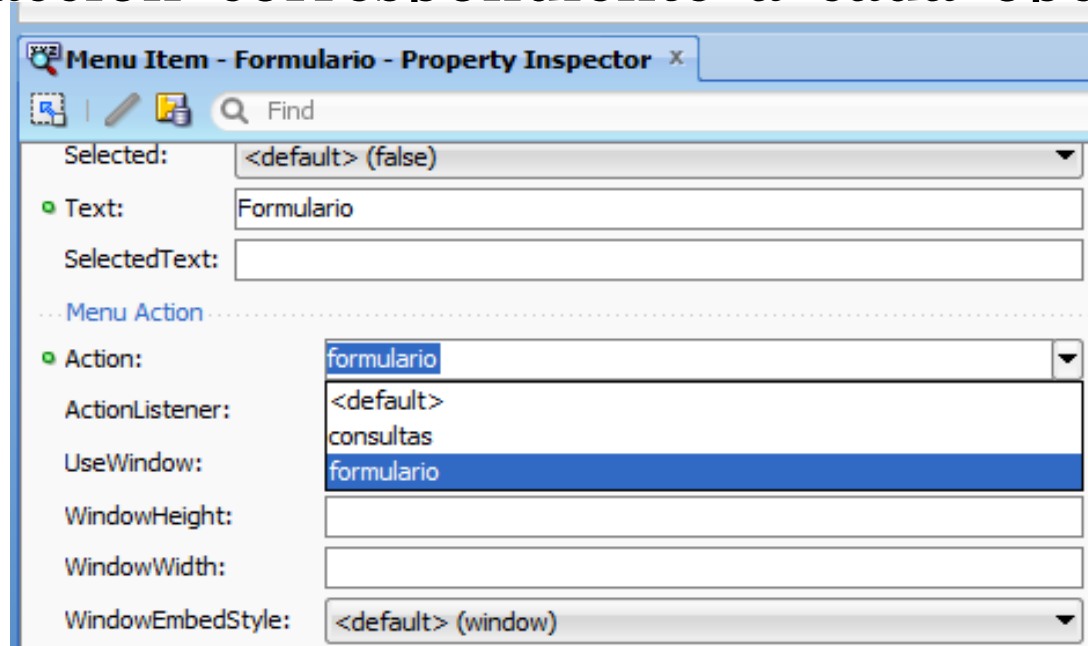
The screenshot shows the Oracle JDeveloper Component Gallery window. On the left, the 'File Name' is 'formulario.jsf' and the 'Directory' is 'C:\jdeveloper\mywork\ApExampl'. The 'Document Type' is set to 'Facelets'. Under 'Page Layout', 'Managed Bean' is selected. Under 'Quick Start Layout', 'One Column (Stretched)' is selected, and a 'Browse...' button is visible. The main area is divided into 'Categories', 'Types', and 'Layouts'. The 'Categories' section shows 'One Column' selected. The 'Types' section shows four layout options, with the first one (One Column) selected. The 'Layouts' section shows two layout options, with the first one (One Column) selected. The 'Options' section on the right has 'Apply Themes' checked and 'One Column (Stretched)' selected.

Diseñando página Maestro-Detalle

- Seleccionar el componente Panel Stretch Layout de la paleta de componentes y arrastrarlo a la página
- Agregar un Panel Splitter en el centro
- Agregar UsuarioFindAll dentro de First->Adf Read Only Form
- Quitar el id de usuario y seleccionar la casilla Include Navigation Controls
- Agregar un Panel Collection dentro del tag Second
- Seleccionar CalificacionesFindAll y agregarlo al panel->Adf Read-Only Table
- En el inspector de propiedades, establecer la propiedad Column Selection en single.
- Editar la propiedad PartialTrigger y agregar los botones del formulario de usuario

Diseñando página Maestro-Detalle

- Agregar el componente Menú Bar en el tag *top*
- Agregar un componente menú dentro el Menú Bar, editar la propiedad Detachable y ponerla en true
- En el panel de Estructura dar clic derecho sobre el Menú Bar ->MenuItem, para agregar las opciones
- Agregar la acción correspondiente a cada opción del menú



Opciones

nombre #{...nombre.inputValue}
paterno #{...paterno.inputValue}
materno #{...materno.inputValue}

First Previous Next Last

menus View Format toolbar [Icons] Freeze [Icon]

materia	parcial1	parcial2	parcial3	calif
#{...materia}	#{...parcial1}	#{...parcial2}	#{...parcial3}	#{...calif}
#{...materia}	#{...parcial1}	#{...parcial2}	#{...parcial3}	#{...calif}
#{...materia}	#{...parcial1}	#{...parcial2}	#{...parcial3}	#{...calif}

Diseñando página de Consultas

- Agregar el método `getUsuarioByName->ADF` Parameter Form.
- Expandir y agregar un `Usuario->Form->Adf` Form
- Incluir controles de navegación y el boton Submit
- Agregar el método `Merge` al `Submit` y configurarlo
- Abrir la definición de la página, modificar la propiedad `refresh` de método `GetUsuarioByName` a `Ifneeded`

Opciones ▾

`#{...p_name...label}`

Buscar

nombre

paterno

materno

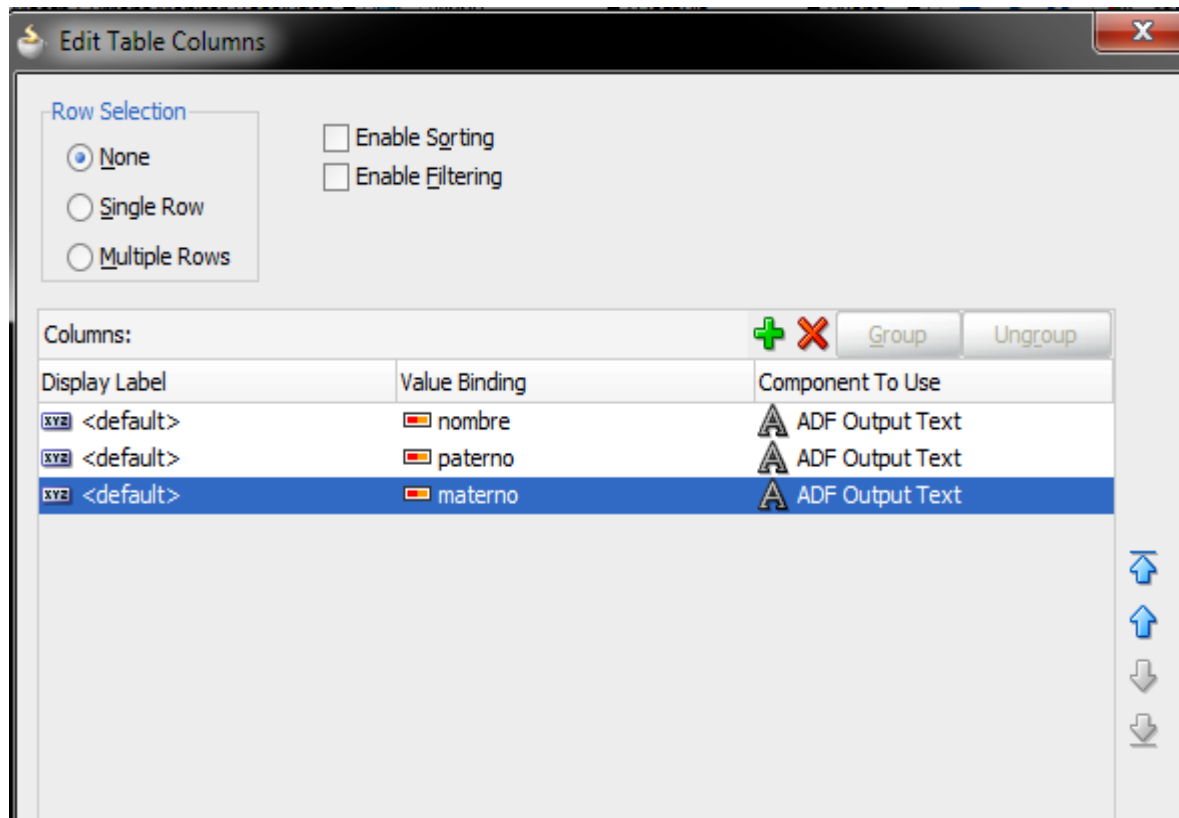
First << Previous << Next >> Last >>

Actualizar

start

Diseñando página de Búsqueda

- Expandir la opción UsuarioFindAll
- Expandir la opción Named Criteria
- Arrastar All Queriable Attributes->Query->Adf Query Panel with Table



Opciones ▾

Busqueda

Search

Basic Saved Se

Search fields added at runtime.

nombre	paterno	materno
# {...nombre}	# {...paterno}	# {...materno}
# {...nombre}	# {...paterno}	# {...materno}
# {...nombre}	# {...paterno}	# {...materno}

cart

Diseñando página Formulario

- Agregar como formulario `getUsuarioFindAll` y agregarle controles de navegación
- Agregar la operaciones `Crear` y `Eliminar`
- Agregar el método `PersistUsuario`

formulario.jsf x

Show Match Current Monitor Resolution

nombre

paterno

materno

Nuevo Guardar Eliminar First Previous Next Last



faces-config.xml x formulario.jsf x

Show Match Current Monitor Resolution [Globe icon] [Dropdown] [Dropdown] [Dropdown] [B icon]

Opciones ▾

nombre

paterno

materno

Ejecutando la aplicación:

Opciones ▾

nombre pepito
paterno perez
materno perez

First

Previous

Next

Last

Ver ▾ Formato ▾



Congelar

Separar

Ajustar

materia	parcial1	parcial2	parcial3	calif
taller I	8	9	8	8.33
taller II	9	9	9	9.0
taller III	7	8	8	7.6

- Página Maestro-Detalle

Ejecutando la aplicación:

Opciones ▾

Ingresa el nombre:

nombre

paterno

materno

- **Página Consultas**

Ejecutando la aplicación:

Busqueda

Search

Correspondencia Todo Cualquiera

iduser Igual que

materno Empieza por

nombre Empieza por p

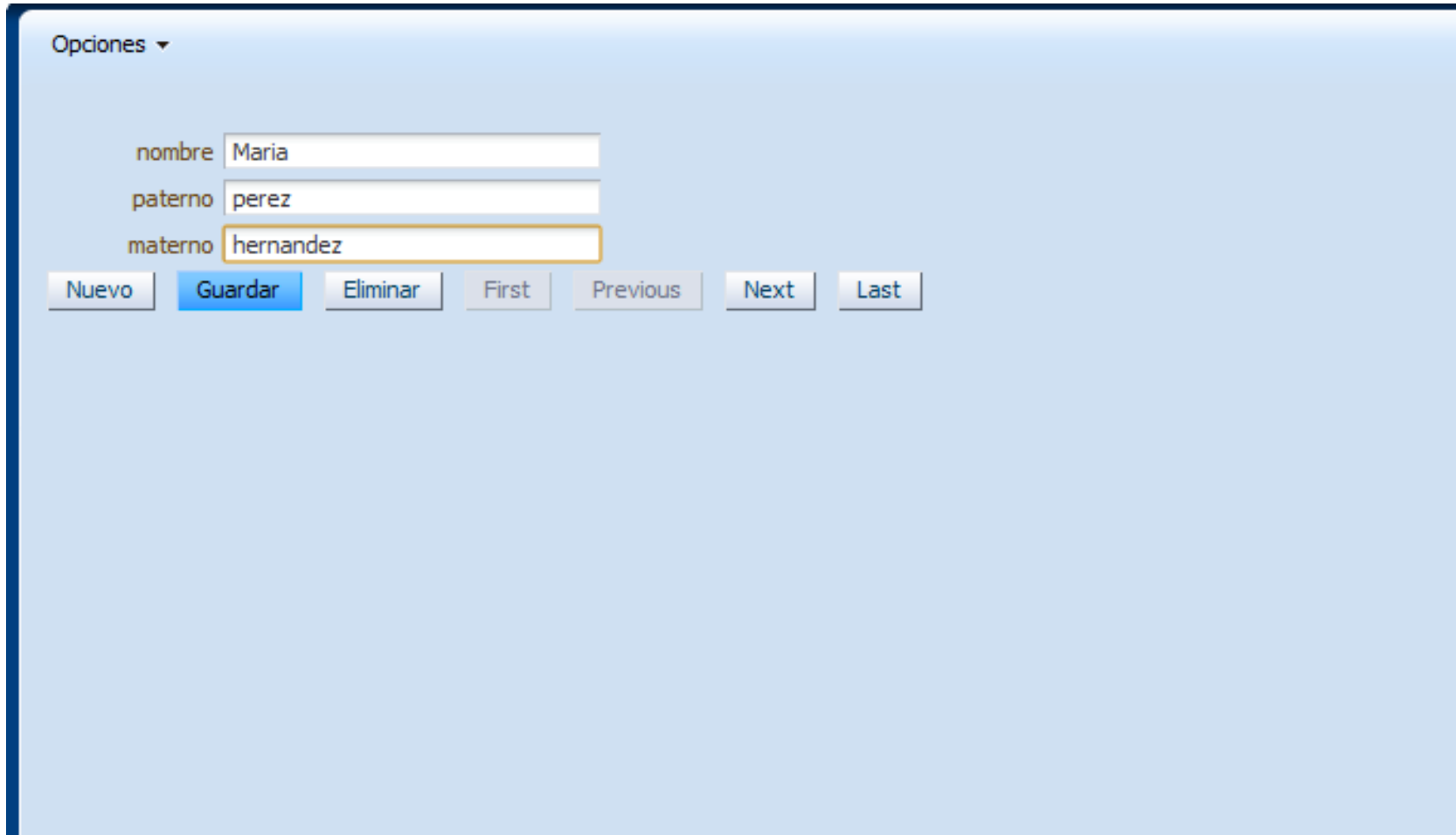
paterno Empieza por

Buscar Restablecer Guardar... Agre

nombre	paterno	materno
pp	perez	perez

- Página de Búsqueda

Ejecutando la aplicación:



Opciones ▾

nombre

paterno

materno

The screenshot shows a web application interface with a light blue background. At the top left, there is a dropdown menu labeled 'Opciones' with a downward arrow. Below it, there are three text input fields. The first is labeled 'nombre' and contains the text 'Maria'. The second is labeled 'paterno' and contains 'perez'. The third is labeled 'materno' and contains 'hernandez'. Below the input fields is a row of seven buttons: 'Nuevo', 'Guardar', 'Eliminar', 'First', 'Previous', 'Next', and 'Last'. The 'Guardar' button is highlighted in blue, indicating it is the active or selected option.

- Página de Formulario