

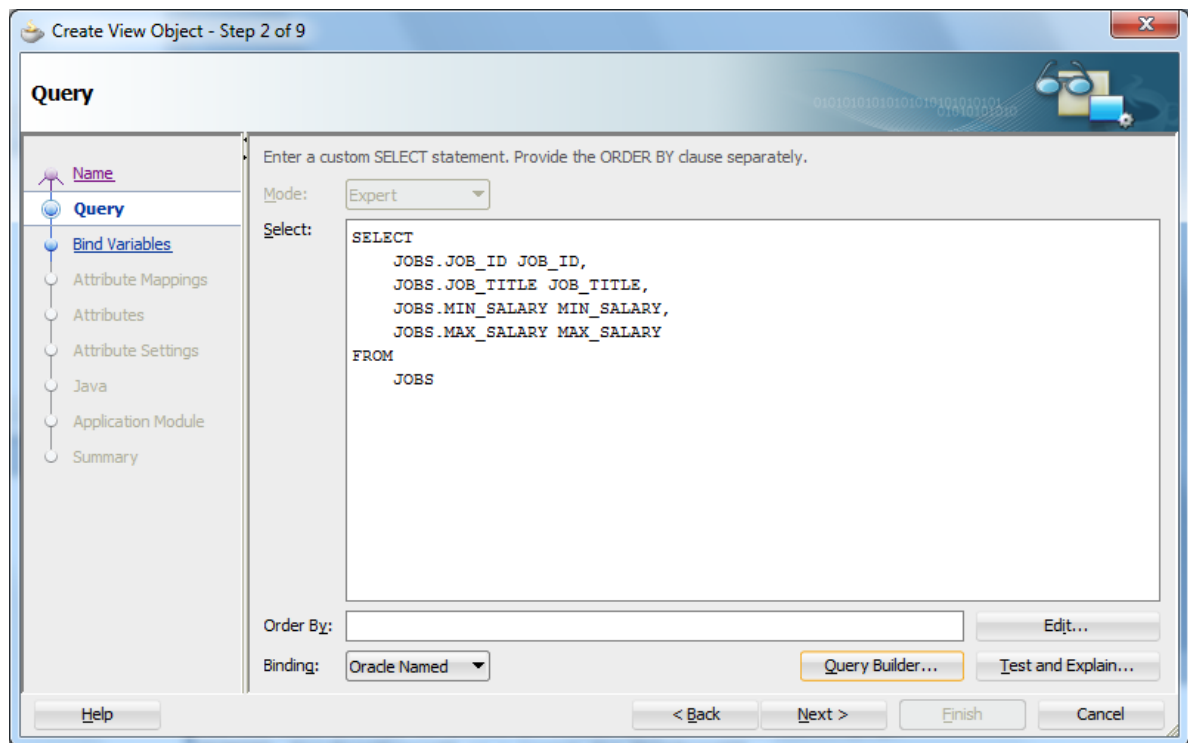
## Lab 9 – Página de Edición

**Objetivo:** Crear páginas con funcionalidad transaccional de Edición.

**Objetivo A:** Crear una página de edición de Empleados.

Paso 1: Prepare la capa Business Services para funcionar correctamente en la creación y actualización de registros en el entity Employees.

- i. Cree el view object “ListaJobs” de solo lectura que recupere los diferentes cargos (jobs).



Adicione este nuevo view object al application Module DepartamentosService.

- ii. Implemente la generación automática de la llave primaria en el entity Employees. Para esto emplee el dominio DBSequence.

- Cree la secuencia en la BD. (se hizo en el Laboratorio 3)
- Cree trigger “before insert” (se hizo en el Laboratorio 3)
- Asignar como tipo DBSequence al atributo EmployeeId del entity Employees.

ListaJobs.xml x DepartamentosServices.xml x Employees.xml x

General

**Attributes** Set Source Order

Entity attributes can be based upon columns in the schema object or can be based upon transient values.

Q Name + - X Override

Name	Type	Column Name	Column Type	Extends
EmployeeId	oracle.jbo.domain.DBS...	EMPLOYEE_ID	NUMBER(6, 0)	
FirstName	String	FIRST_NAME	VARCHAR2(20)	
LastName	String	LAST_NAME	VARCHAR2(25)	
Email	String	EMAIL	VARCHAR2(25)	
PhoneNumber	String	PHONE_NUMBER	VARCHAR2(20)	
HireDate	Timestamp	HIRE_DATE	DATE	
JobId	String	JOB_ID	VARCHAR2(10)	
Salary	BigDecimal	SALARY	NUMBER(8, 2)	
CommissionPct	BigDecimal	COMMISSION_PCT	NUMBER(2, 2)	
ManagerId	Integer	MANAGER_ID	NUMBER(6, 0)	
DepartmentId	Integer	DEPARTMENT_ID	NUMBER(4, 0)	

Details UI Hints Validation Rules Security Dependencies Custom Properties

Name :

Display Name:

Description:

Type :

Property Set:

Default Value

☒ Literal ☐ Expression ☐ SQL

Refresh Expression Value:

☐ Polymorphic Discriminator

Subtype Value:

Effective Date:

Updatable:

☒ Persistent ☐ Transient

☐ Mandatory ☒ Refresh on Insert

☒ Primary Key ☐ Refresh on Update

☒ Queryable ☐ Change Indicator

☒ Precision Rule

Track Change History:

Column Name:

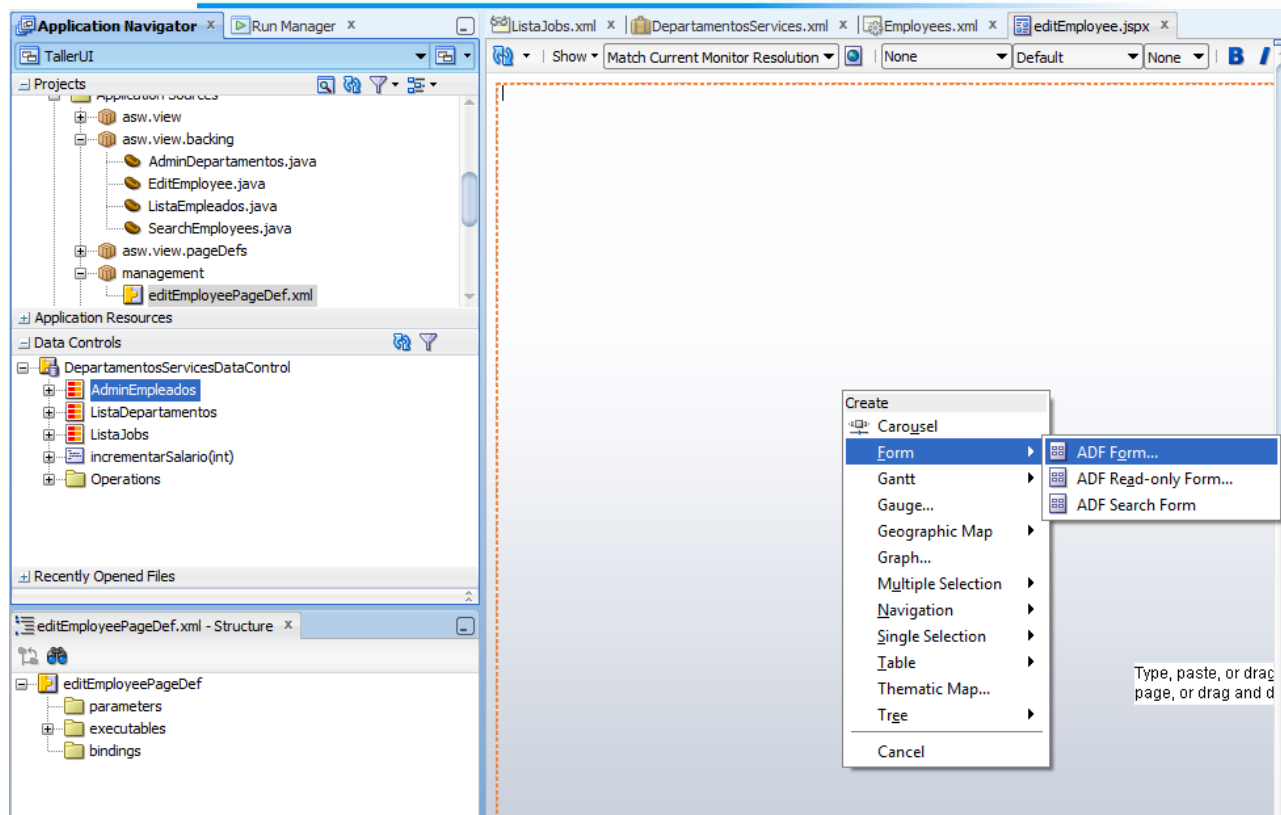
Column Type:

☐ Generate Unique Constraint

Sequence Name:

Paso 2: Edite la página Template.jsp y guárdelo bajo en nombre “editEmployee.jsp” en el directorio \public\_html\pages\management\ . Si el subdirectorio no existe, créelo.

Paso 3: Edite la página editEmployee.jsp y adicione en ella un componente “ADF Form” basado en la colección AdminEmpleados.



Siga el wizard, dejando los atributos por omisión.

Paso 4: Aplique cambios cosméticos a la forma:

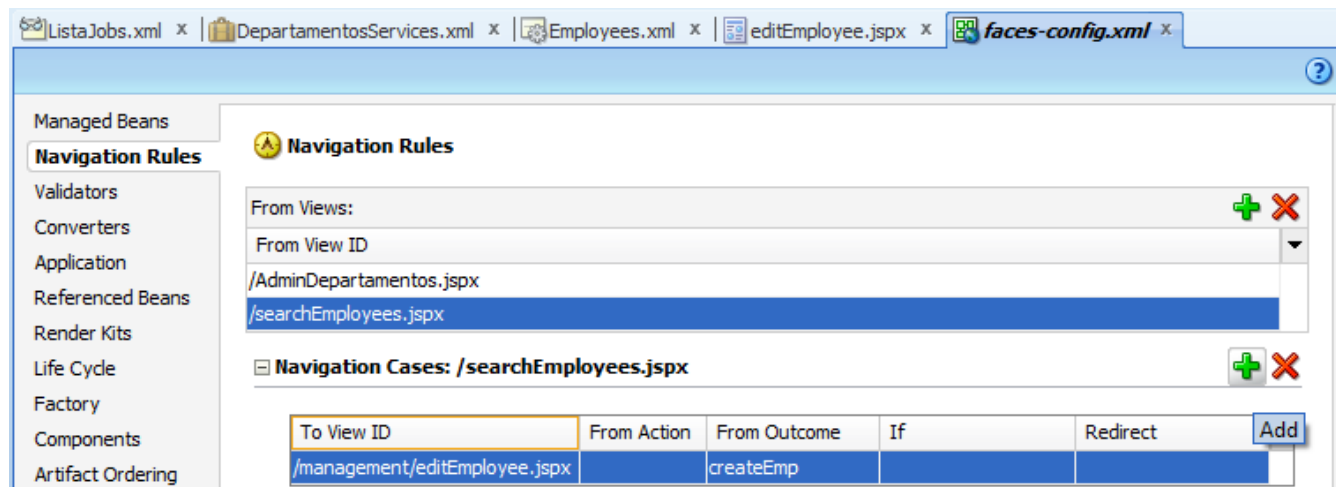
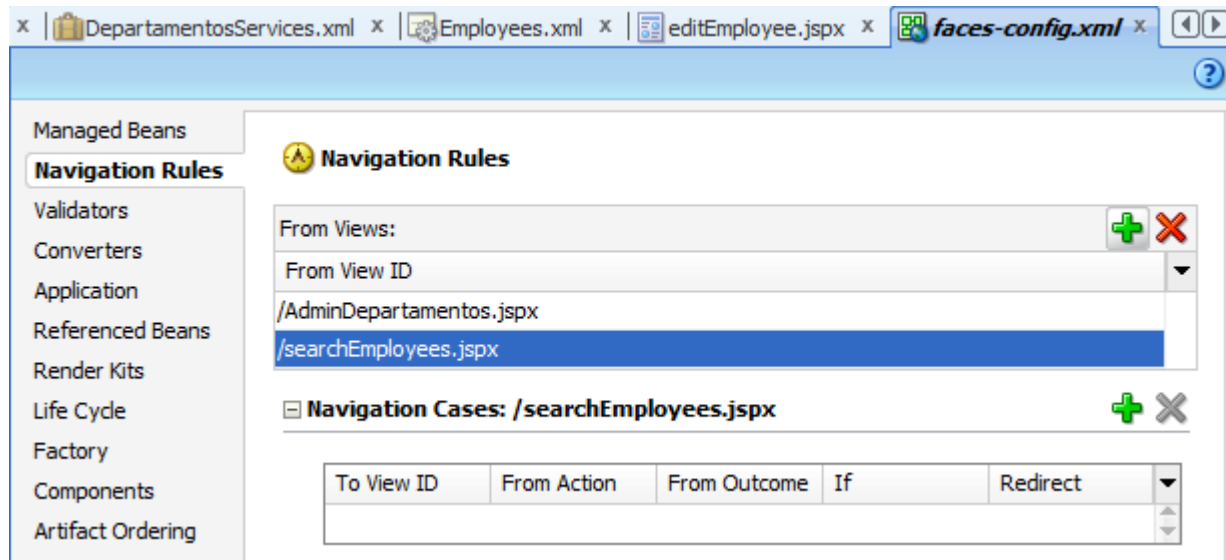
- i. Prenda la pestaña “Employees”
- ii. Presente los campos de la forma a dos columnas. Para esto, seleccione el componente af:panelForm y modifique las propiedades MaxColumns y Rows.
- iii. Esta forma es empleada tanto para CREAR como para EDITAR registros. De esta forma se espera mostrar el título adecuado: “Nuevo empleado” y “Edición empleado” respectivamente. Para esto seleccione el componente af:panelPage y asigne al atributo Title el valor:

```
##{bindings.EmployeeId.inputValue.value > 0 ? 'Edicion empleado' : 'Nuevo empleado'}
```

**Objetivo B:** Especificar la navegación entre la página de búsqueda (searchEmployees.jsp) y la página de edición (editEmployee.jsp).

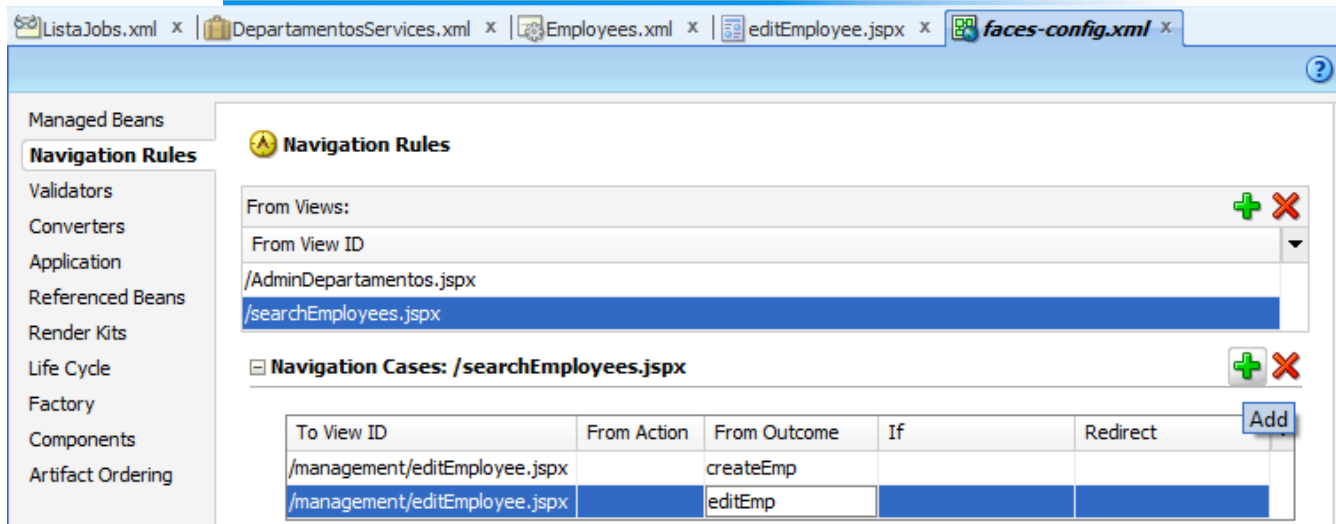
Paso 5: Cree la regla de navegación siguiente:

From View ID	/pages/searchEmployees.jsp
To View ID	/pages/management/editEmployee.jsp
Outcome	createEmp
Redirect	False



Paso 6: Bajo la misma regla de navegación (/pages/searchEmployees.jsp) cree un nuevo caso de navegación:

From View ID	/pages/searchEmployees.jsp
To View ID	/pages/management/editEmployee.jsp
Outcome	editEmp
Redirect	False

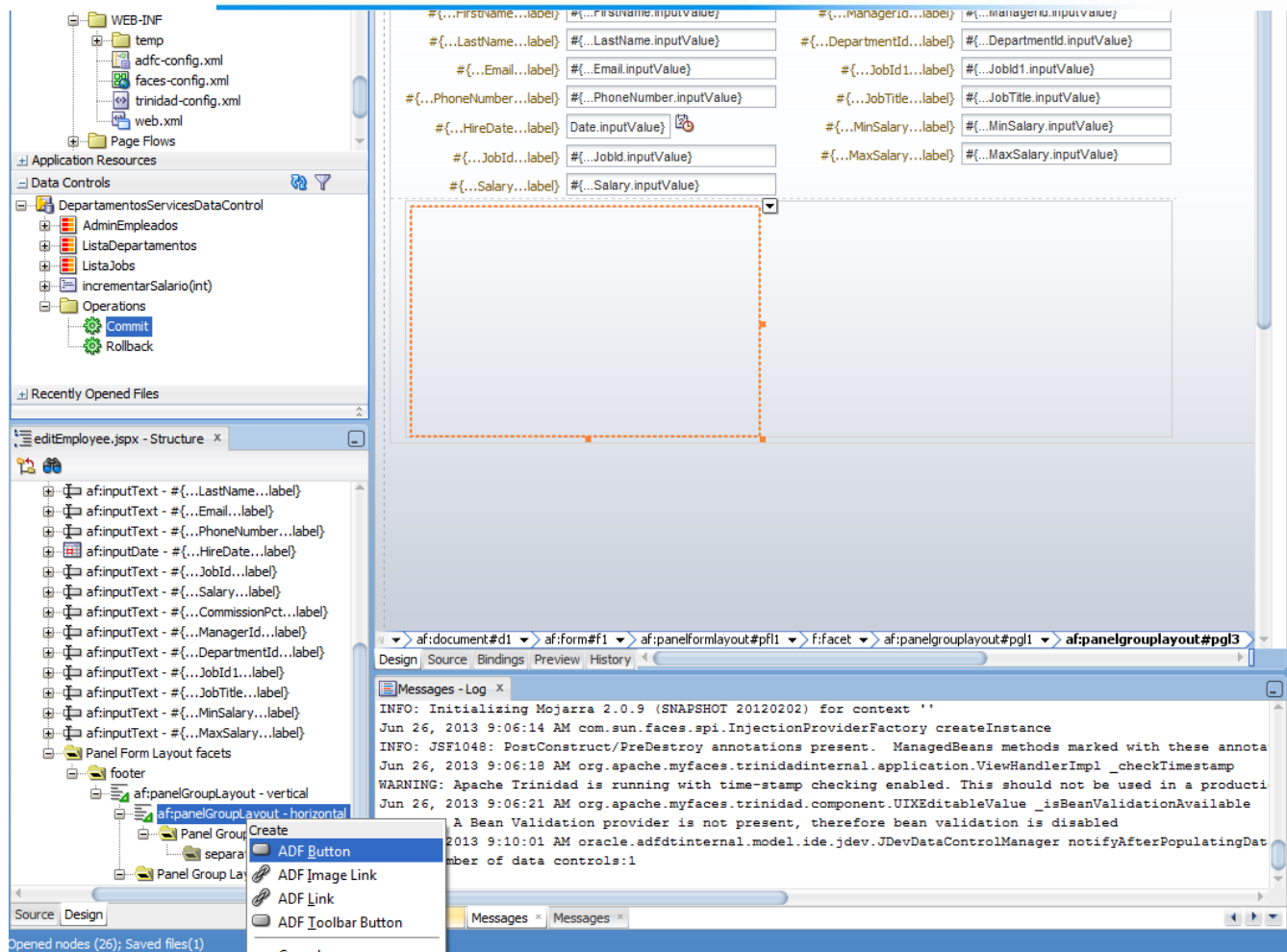


Paso 7: Cree la regla de navegación que permita retornar a la página de búsqueda desde la página de edición:

From View ID	/pages/management/editEmployee.jsp
To View ID	/pages/searchEmployees.jsp
Outcome	cancelEditEmp
Redirect	False

Paso 8: Adicione los botones de “Guardar” y “Cancelar” a la página editEmployee.jsp

- i. Adicione un “ADF Button” en el “footer facet” de la forma de edición. Este Command Button debe estar basado en la operación Commit del Data Control Palette. Cambie el atributo Text por “Guardar”.



Cambie los siguientes atributos del botón:

Text	Guardar
Disabled	False

- ii. Adicione un “ADF Button” en el “footer facet” de la forma de edición. Este Command Button debe estar basado en la operación Rollback del Data Contro Palette. Cambien los siguientes atributos del botón:

Text	Cancelar
Action	cancelEditEmp
Disabled	False

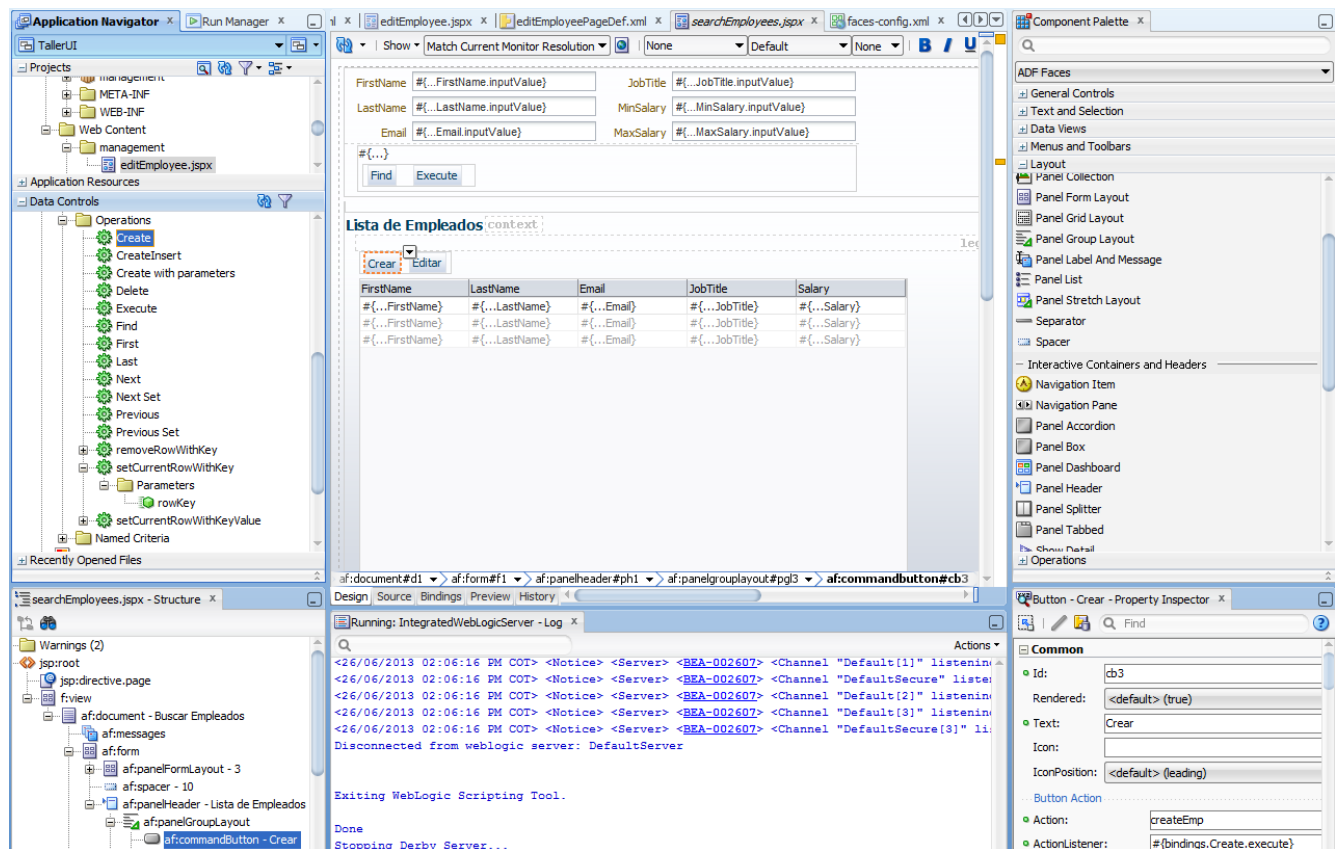
Paso 9: Ejecute la forma editEmployee.jsp.

Paso 10: Enlace la forma de búsqueda a la forma de edición a través de los botones de “Editar” y “Crear Empleado”.

- i. Edite la forma searchEmployees.jsp
- ii. En la tabla de empleados aparece el botón “Submit”. Seleccionelo y cambie los siguientes atributos:

Text	Editar
Action	editEmp

- iii. Adicione el nuevo botón “Crear Empleado” a partir de la operación “Create” de la colección AdminEmpleados. Deje caer este botón al lado del botón “Editar” que se encuentra en la tabla de empleados:



- iv. Seleccione el botón Create y cambie los siguientes atributos:

Text	Crear Empleado
Action	CreateEmp

- v. Pruebe la aplicación

**Objetivo C:** Mejorar la interfaz de usuario de la página de edición a través de listas de selección y listas de valores (LOV).

Paso 11: Cree una lista de valores para capturar el atributo jobId:

- Elimine el campo jobId de la forma.
- Adicione nuevamente el campo jobId, pero ahora como ADF Select one choice.

The screenshot displays the Oracle ADF IDE interface. On the left, the 'Application Navigator' shows the project structure, including 'TallerUI', 'Projects', 'Application Resources', and 'Data Controls'. The 'DepartamentosServicesDataControl' is expanded, showing attributes like 'CommissionPct', 'DepartmentId', 'Email', 'EmployeeId', 'FirstName', 'HireDate', 'JobId', 'JobId1', and 'JobTitle'. The 'JobId' attribute is selected. Below the navigator, the 'Structure' pane shows the 'af:panelFormLayout - 8' containing various input components. A context menu is open over the 'af:inputText - #{...ManagerId...label}' component, with the 'Create' option selected. The 'Create' submenu is open, showing 'List of Values', 'Single Selection', and 'Text'. The 'Single Selection' submenu is also open, showing 'ADF Select One Choice...', 'ADF Select One Listbox...', 'ADF Select One Radio...', and 'ADF Select Boolean Checkbox...'. The 'ADF Select One Choice...' option is selected. The main editor shows the 'editEmployee.jspx' file with the following bindings:

```

#{...EmployeeId...label} #{...EmployeeId.inputValue}
#{...FirstName...label} #{...FirstName.inputValue}
#{...LastName...label} #{...LastName.inputValue}
#{...Email...label} #{...Email.inputValue}
#{...PhoneNumber...label} #{...PhoneNumber.inputValue}
#{...HireDate...label} #{...HireDate.inputValue}
#{...Salary...label} #{...Salary.inputValue}
#{...CommissionPct...label} #{...CommissionPct.inputValue}

```

At the bottom of the form, there are 'Guardar' and 'Cancelar' buttons. The bottom status bar shows the time '3 02:06:16 PM COT' and the server 'DefaultServer'.



- iii. Especifique en el editor “List Binding Editor” los siguientes valores:

Configure list data and map its return values to the page's base data source. You can create a fixed list of values for the list, or populate values dynamically at runtime.

Base Data Source: DepartamentosServicesDataControl.AdminEmpleados Add...

The data source to be updated with the list value

☒ Dynamic List ☐ Fixed List ☐ Model Driven List

List Data Source: DepartamentosServicesDataControl.ListaJobs Add...

Data Mapping:

Data Value	List Attribute
JobId	JobId

Map one or more list attribute values to the data values you want to update.

List Items

Select the data source attribute that provides the list item display value.

Display Attribute: JobTitle

"No Selection" Item: Labeled Item (First of List) <No Selection> ...

☐ MRU MRU Count: 1

MRU Separator Value: ==== ...

Help OK Cancel

- iv. Pruebe la forma.

**Objetivo D:** Crear una forma de consulta sobre departamentos. Esta forma será empleada más adelante como LOV (List Of Value) del campo DepartmentID.

Cerrar

### Buscar departamento por:

Nombre del departamento

Buscar

### Resultado:

Select and	Seleccionar	
Seleccionar	DepartmentId	DepartmentName
<input checked="" type="radio"/>	20	Marketing
<input type="radio"/>	170	Manufacturing

Paso 12: Crear el ViewObject DepartmentsByName de solo lectura basado en el siguiente query:

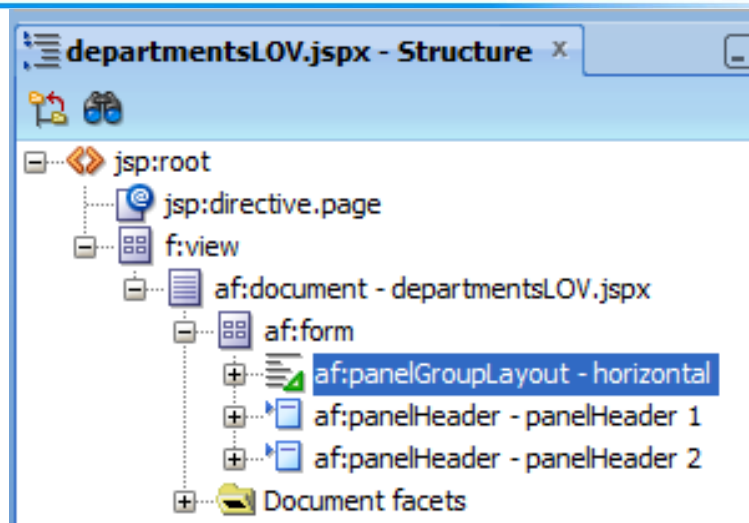
```
SELECT
  DEPARTMENTS.DEPARTMENT_ID DEPARTMENT_ID,
  DEPARTMENTS.DEPARTMENT_NAME DEPARTMENT_NAME
FROM
  DEPARTMENTS
WHERE UPPER(DEPARTMENTS.DEPARTMENT_NAME) LIKE
  CONCAT(UPPER(:cadenaBusqueda), '%')
```

- Verifique el haber declarado la variable bind cadenaBusqueda. Defina como label de esta variable el texto “Nombre del departamento”.
- Adicionar este nuevo viewObject al applicationModule DepartamentoServices.

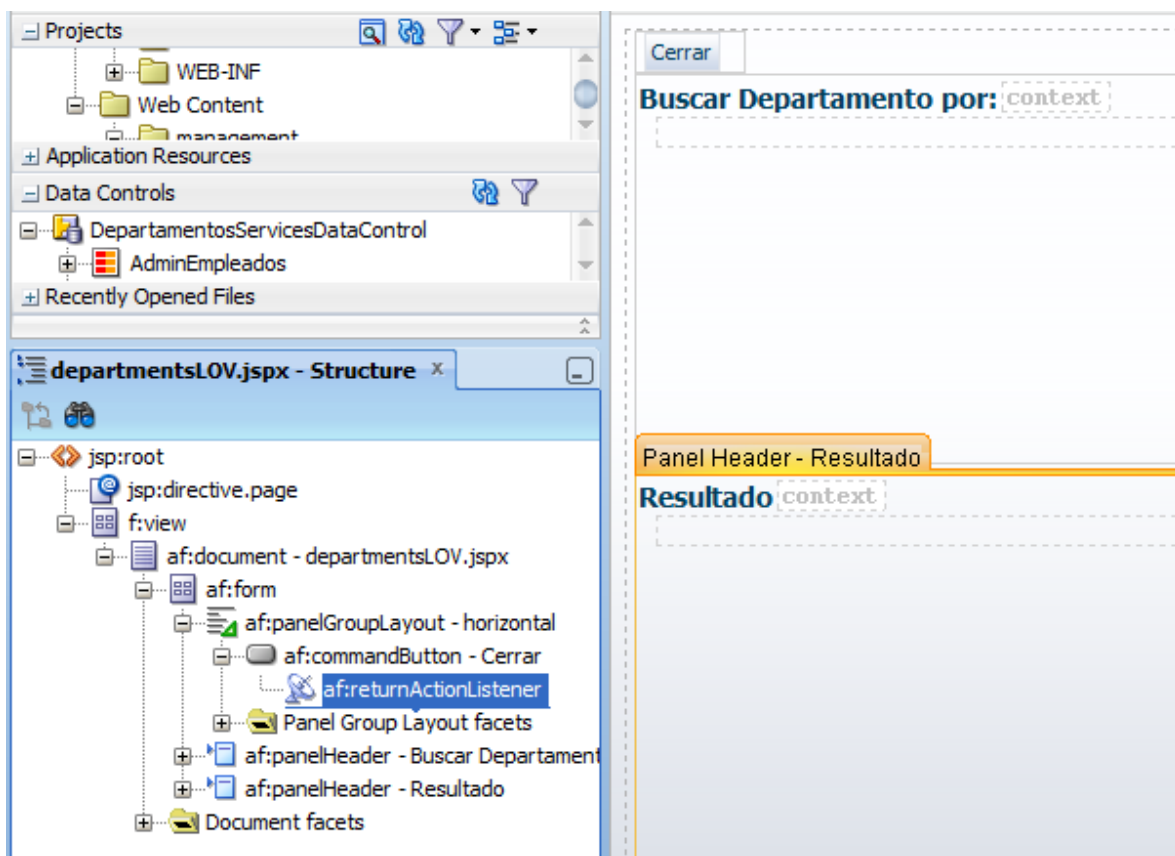
Paso 13: Crear una nueva página denominada deparmentsLOV.jsp en el subdirectorío pages.



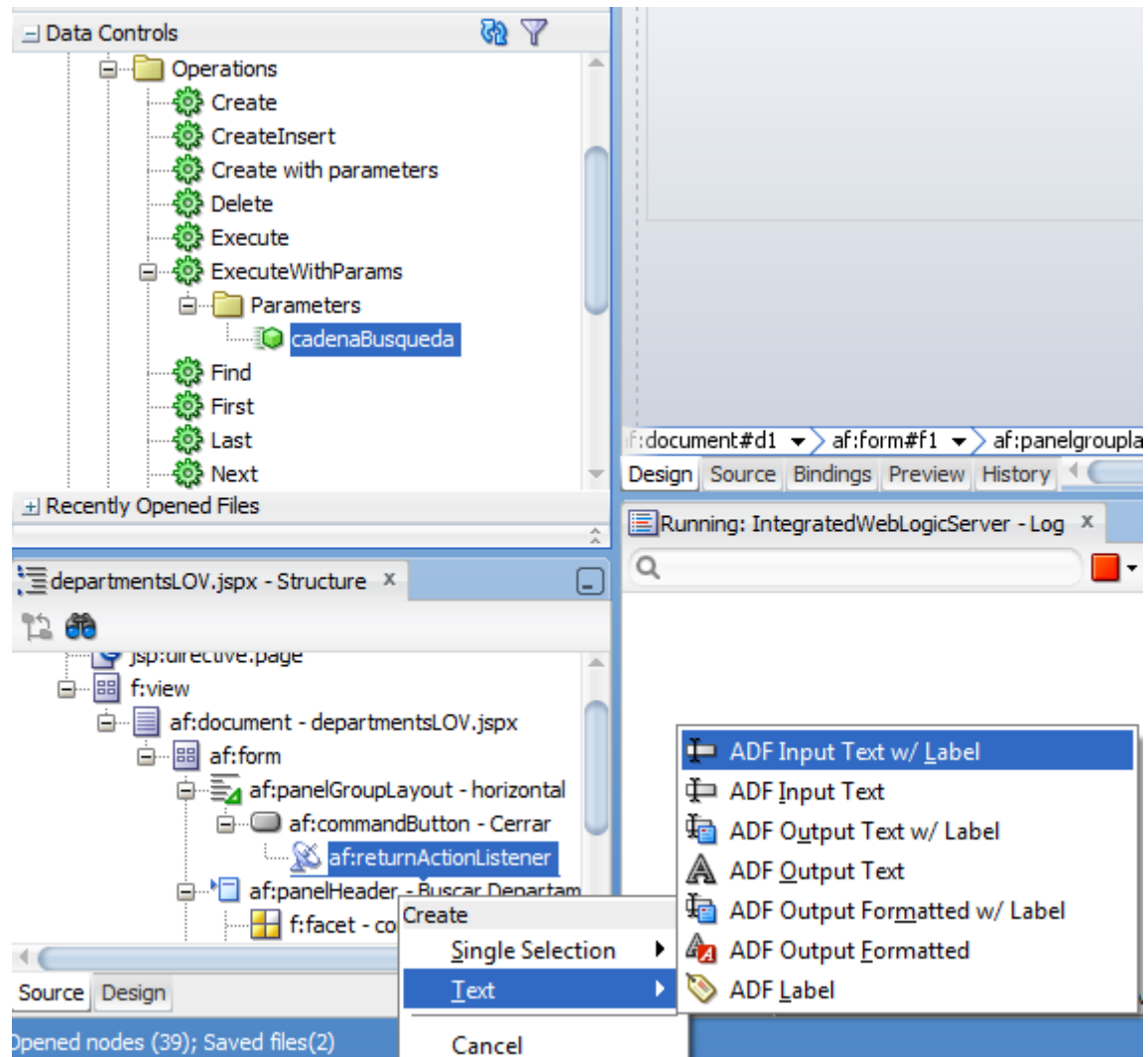
Paso 14. Cree el siguiente layout en la página deparmentsLOV:



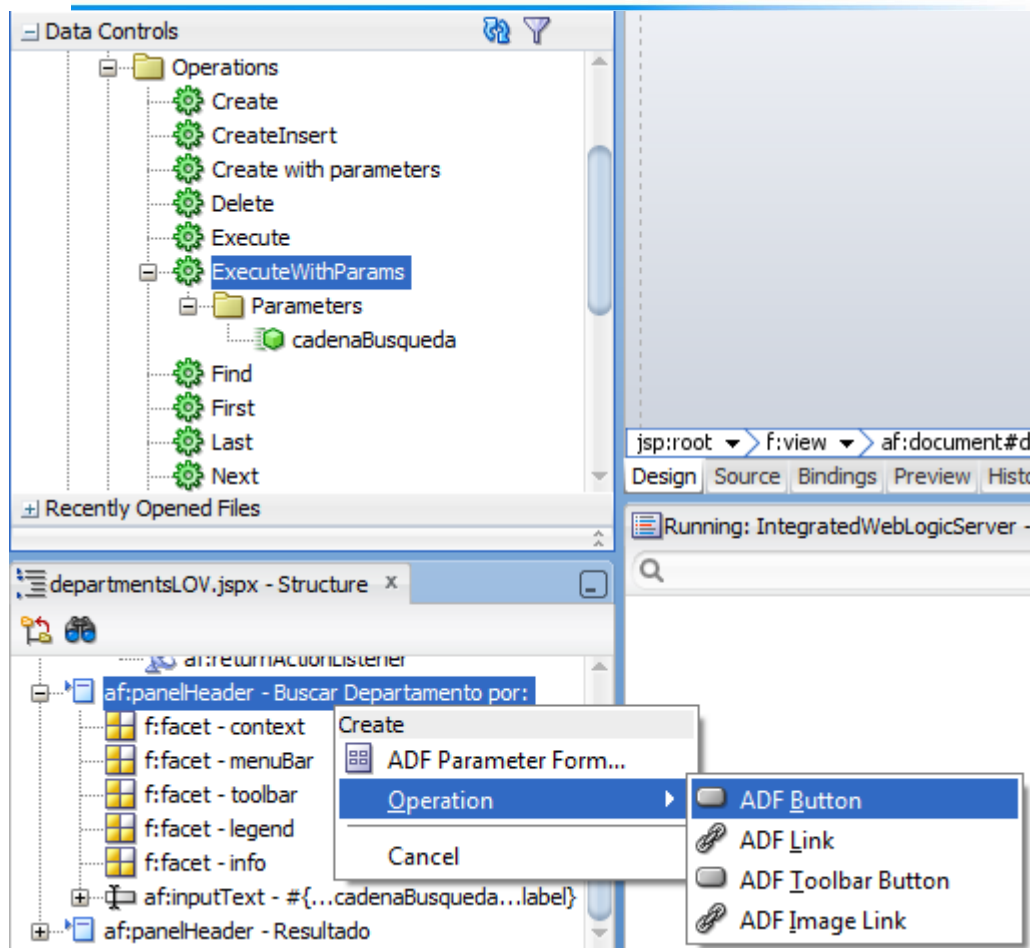
- Adicione un CommandButton (con el label Cerrar) sobre el panelHorizontal.
- Asigne a este botón el valor de true a la propiedad Immediate.
- Adicione al botón cerrar el componente af:returnActionListener.



Paso 15: Drag-and-Drop el parámetro cadenaBusqueda del action ExecuteWithParams en el panelHeader “Buscar departamento por”:



Paso 16: Adicione el botón “Buscar” a partir del action ExecuteWithParams:



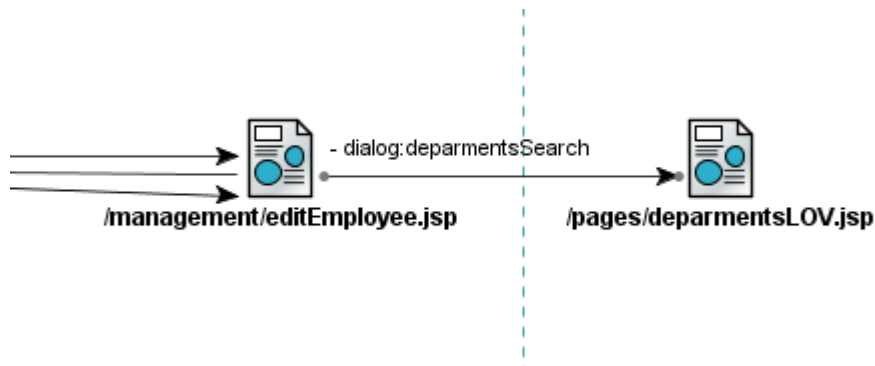
Paso 17: Adicione el panelHeader “Resultado” una ADF Table Read-Only basada en la colección DepartmentsByName. Seleccione Enable selection y Enable sorting.

- a. Cambie el label del botón Submit por Seleccionar.
- b. Drag-and-drop el componente af:ReturnActionListener sobre el botón seleccionar. Esto hace que la página retorne a la página que la invoco.

Paso 18: Pruebe la forma.

**Objetivo E:** El objetivo de este laboratorio es cambiar la forma de edición de empleados de forma tal que la captura del departamento al cual pertenece el empleado sea por medio de una lista de valores, donde el usuario final pueda seleccionar basado en el nombre del departamento y no el ID de este. Para este objetivo se hará uso de la página departmentsLOV.jsp.

Paso 19: Cree una nueva regla de navegación entre la página editEmployee.jsp y la página departmentsLOV.jsp. El nombre del outcome debe ser dialog:departmentsSearch.



Paso 20: Abra la página de edición de empleados (editEmployee.jsp).

Paso 21: Cambie el atributo DepartmentId de InputText a SelectInputText. Sobre el atributo DepartmentId, haga clic derecho y seleccione la opción **Convert**.

Paso 22: En la propiedad Action del atributo DepartmentId especifique dialog:departmentsSearch.

Paso 23: Abra la página departmentsLOV.jsp y seleccione en el Structure window el componente af:ReturnActionListener del botón Seleccionar. Modifique el atributo value a #{row.DepartmentId}. Observe que la variable row representa el registro que el usuario selecciono de la tabla.

Paso 24: Pruebe la forma editEmployee.jsp