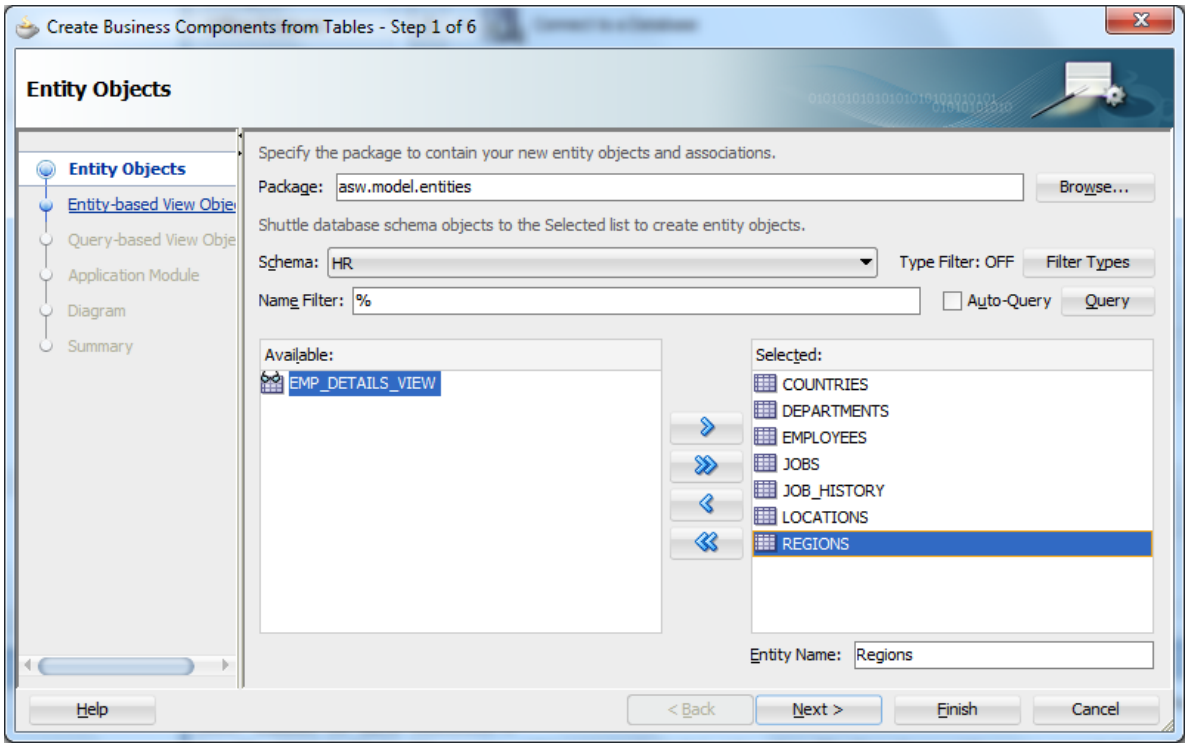
Lab 5 – Creación de Interfaces de Usuario

Objetivo: Emplear las herramientas de JDeveloper y ADF para generar interfaces de usuario de forma rápida y robusta, sin requerir código.

Objetivo A: Crear el Business Services base para el ejercicio.

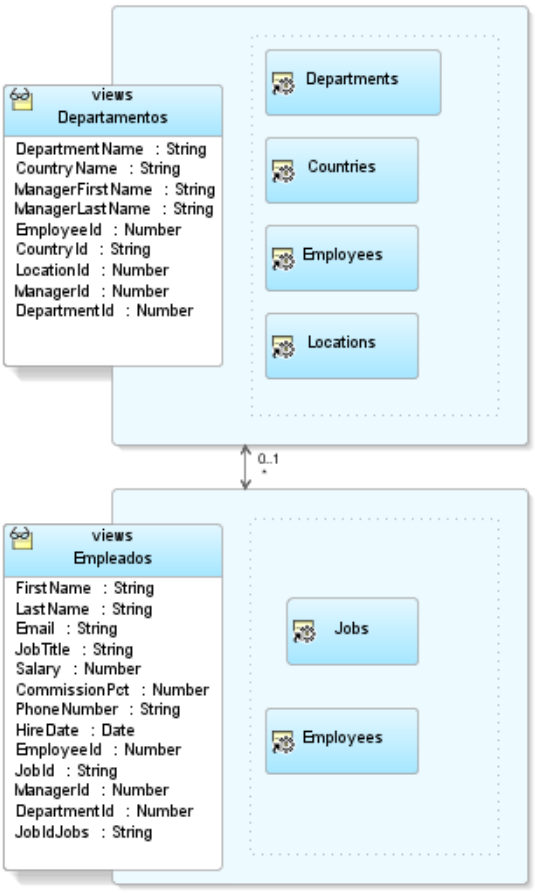
Paso 1: Crear una nueva aplicación denominada TallerUI.

Paso 2: Emplee el wizard "Business Components from tables" para crear un Entity por cada tabla del esquema HR en el paquete asw.model.entities. No cree views ni applicationModules.
El finalizar este wizard traslade las asociaciones generadas a asw.model.associations.

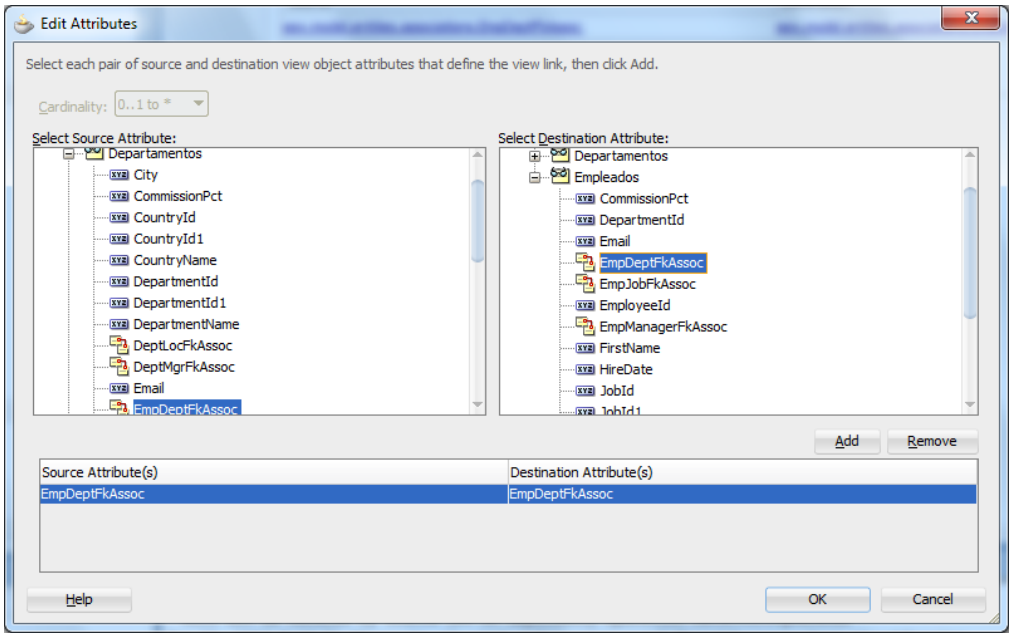


Paso 3: Cree la siguiente configuración de vistas:

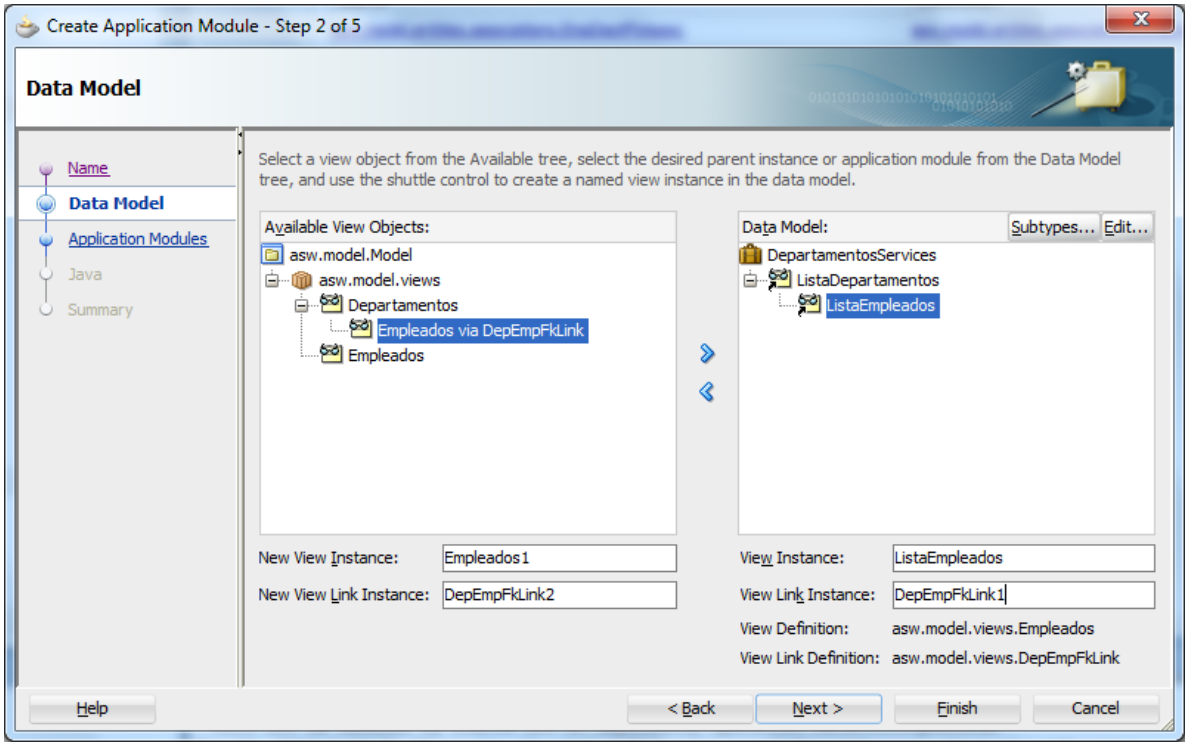
Departamentos
Empleados

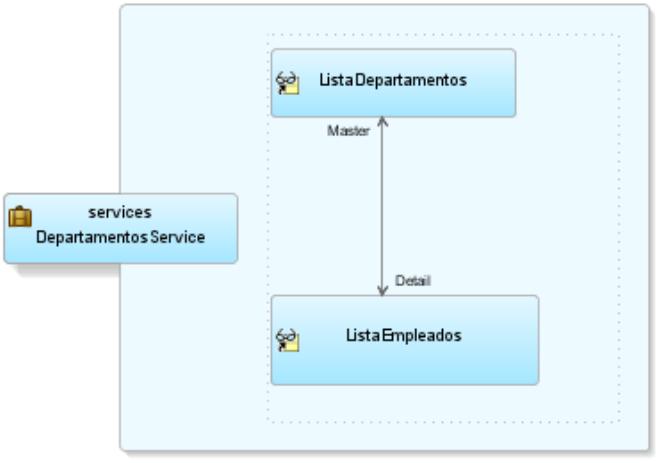


Cree el view link DepEmpFkLink que represente la relación maestro-detalle entre Departamentos y Empleados.



Paso 4: Cree el ApplicationModule DepartamentosServices con las instancias de vista ListaDepartamentos (basada en el viewObject Departamentos) y ListaEmpleados (basada en el viewObject empleados). En el applicationModule, estas dos instancias de vistas deben estar configuradas en una relación Maestro-Detalle.



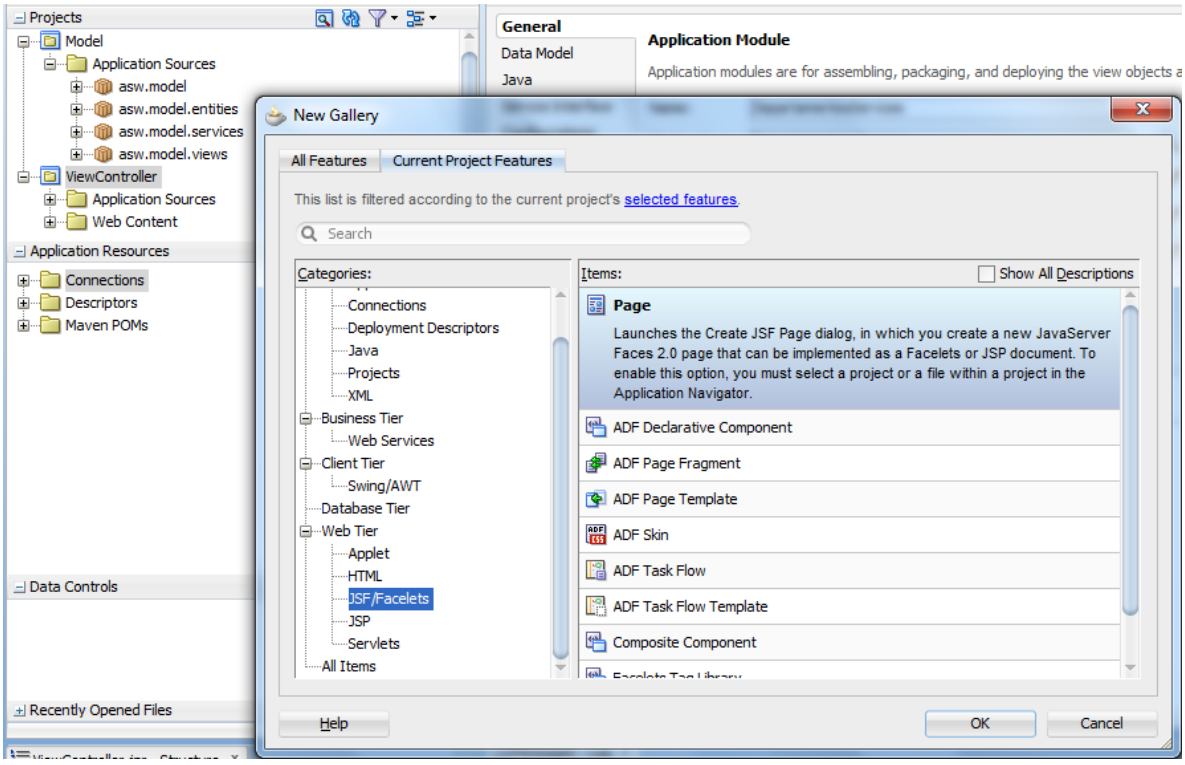


Objetivo B: Trabajar sobre el proyecto ViewController para crear diferentes páginas empleando el mecanismo de DragAndDrop desde la Data Control Palette.

Lab 5 – Creación de Interfaces de Usuario

Paso 5: Cree una página JSP.

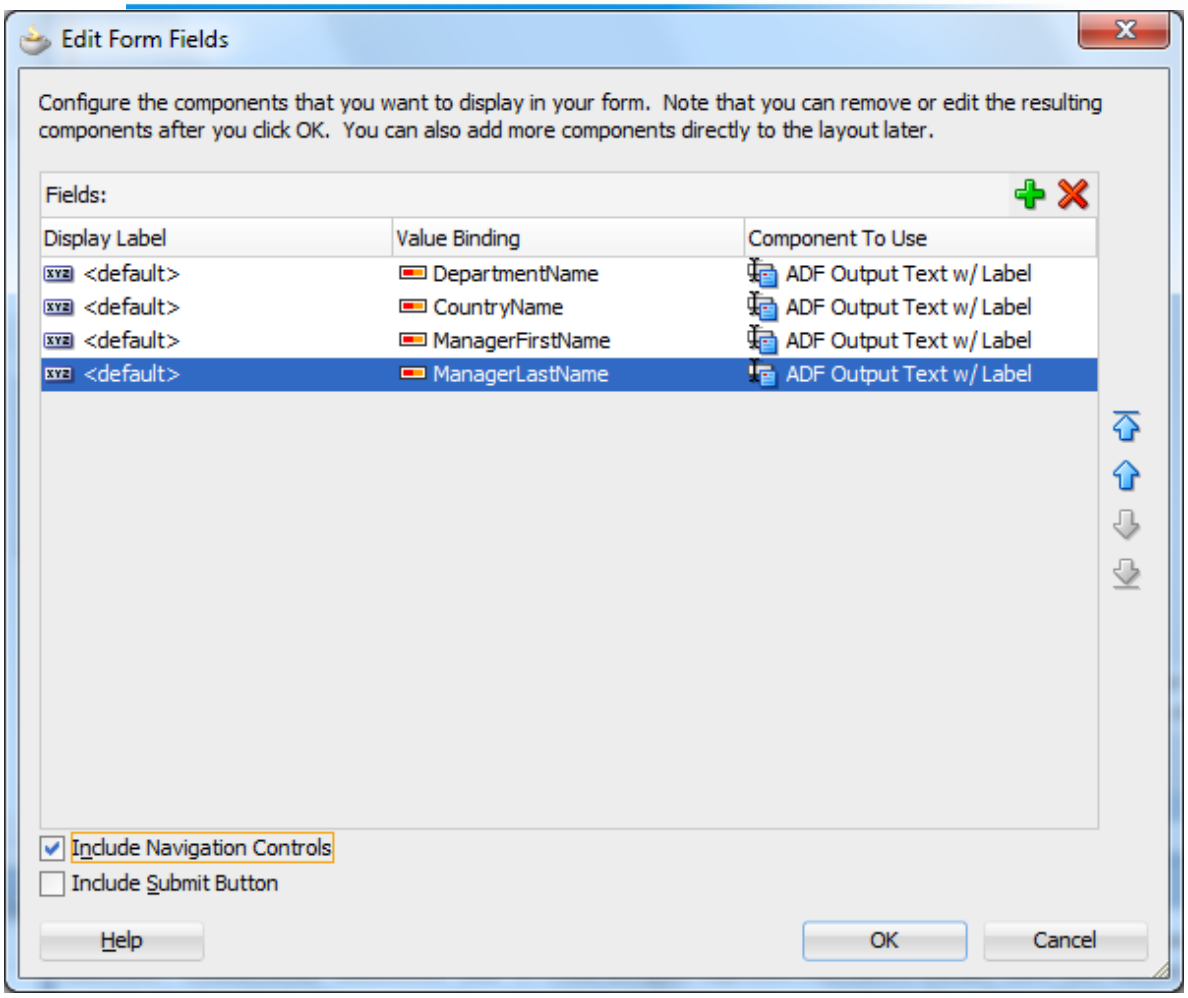
i. Invoque la creación de un nuevo artefacto estando en el proyecto ViewController.



ii. Siga el Wizart de creación de páginas JSP.

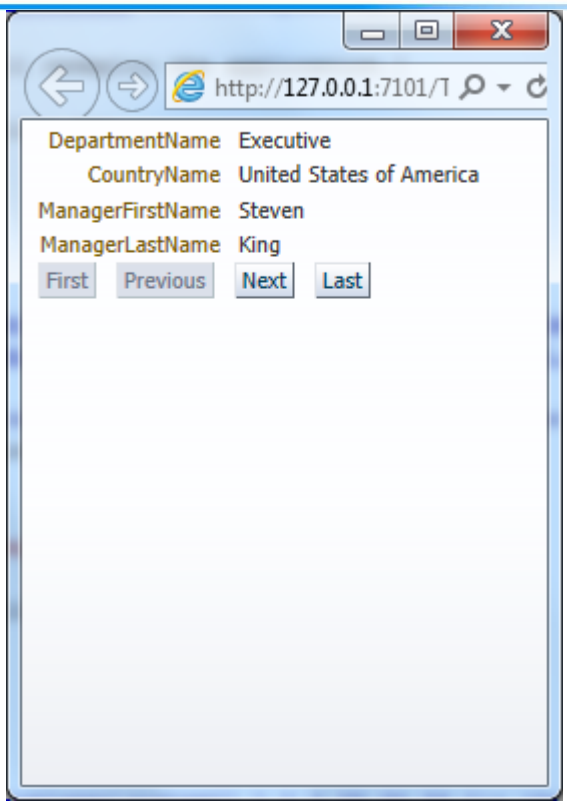
File Name	AdminDepartamentos
Directory Name	C:\adf\talleres\TallerUI\ViewController\public_html
Type	Page

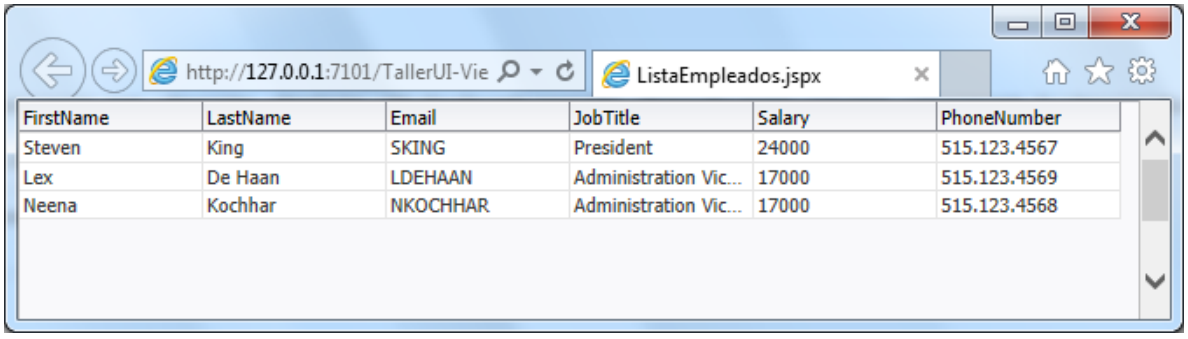
Paso 6: Inserte en la página AdminDepartamentos una forma de solo lectura (*ADF Read-only form…*), basada en la colección ListaDepartamentos que se encuentra como raíz del Data Control Palette. Los campos a ilustrar en la forma son: DepartmentName, ManagerFirstName, ManagerLastName y CountryName.



Paso 7: Cree la página JSP ListaEmpleados como una tabla de solo lectura (*ADF Read-only table*). Incluya las columnas FirstName, LastName, Email, JobTitle, Salary, PhoneNumber.

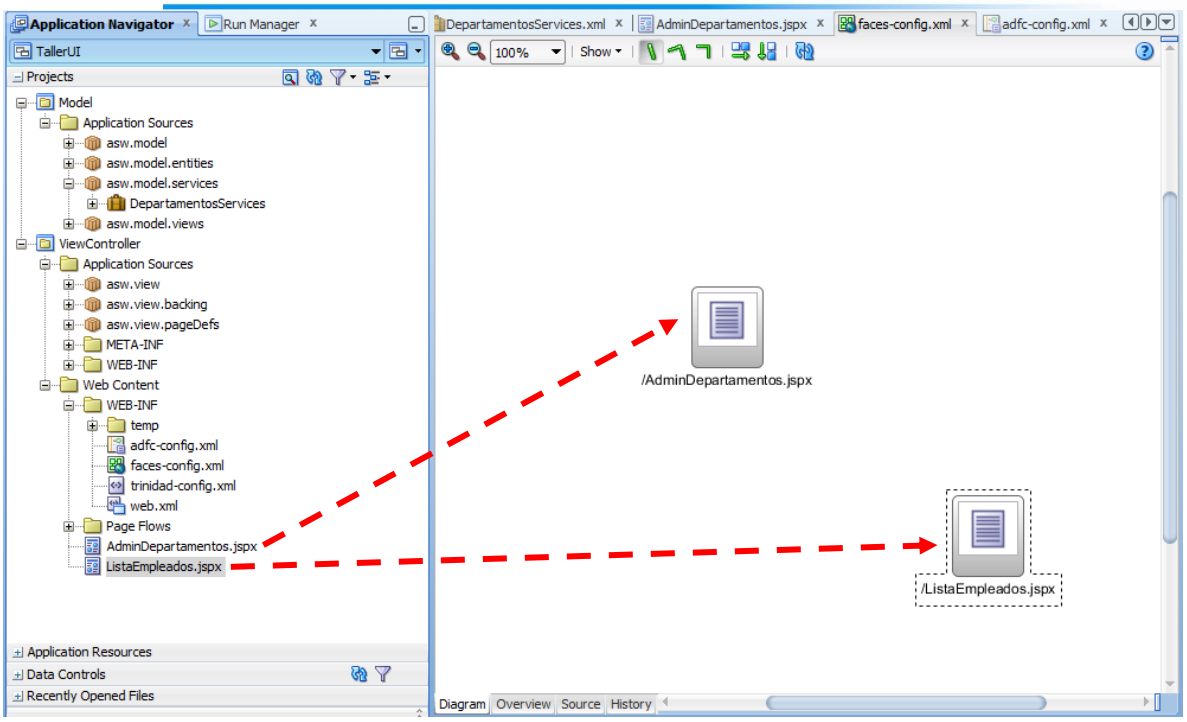
Paso 8: Probar las páginas creadas



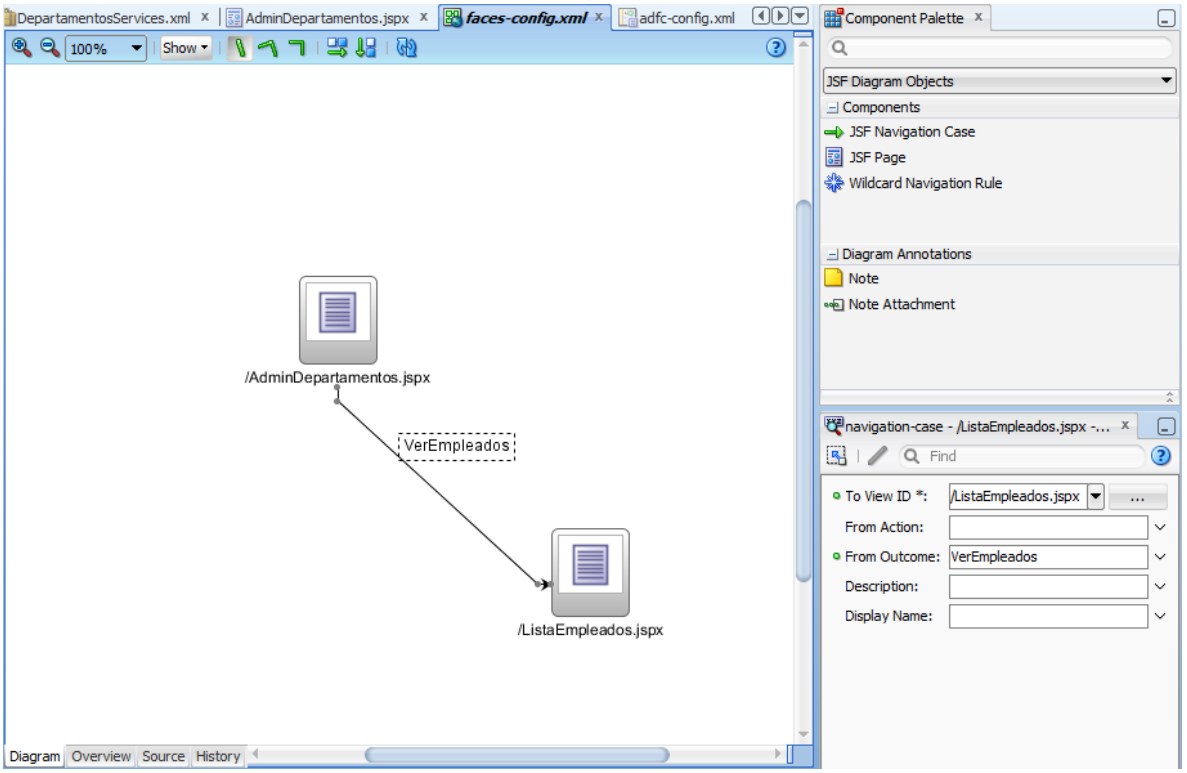


Paso 9: Abra el editor "JSF Navigation" (View controller\WebContent\WEB-INF\faces-config.xml).

Arrastre las páginas AdminDepartamentos.jsp y ListaEmpleados.jsp sobre el área de trabajo de faces-config.xml



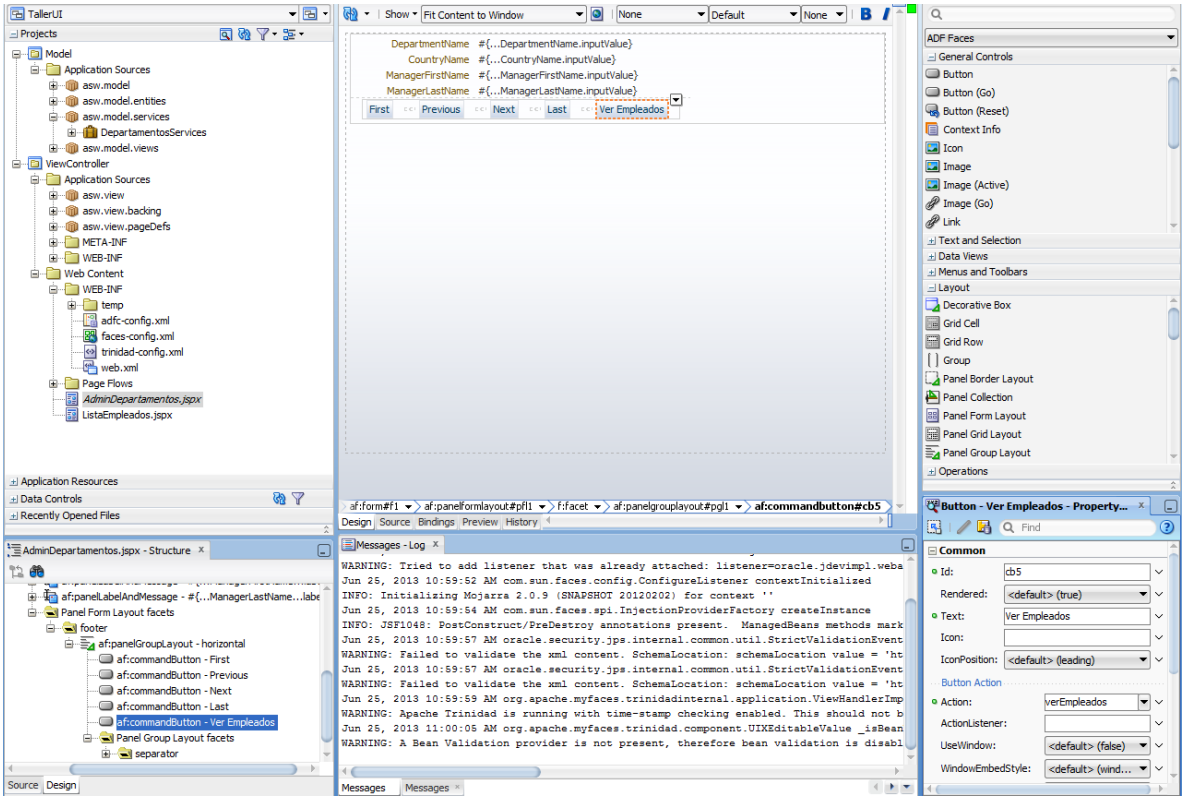
Paso 10: Cree un caso de navegación de la página AdminDepartamentos a ListaEmpleados. Cambie el outcome de "success" a "VerEmpleados".



Paso 11: Edite la página AdminDepartamentos y adicione un CommandButton con las siguientes propiedades:

Text	Lista de Empleados
Action	VerEmpleados

Observe que la lista de valores disponibles en Action, corresponde a los outcomes definidos en el "JSF Navigation".



Paso 12: Pruebe la aplicación y garantice que al listar los empleados, sean los del respectivo Departamento.

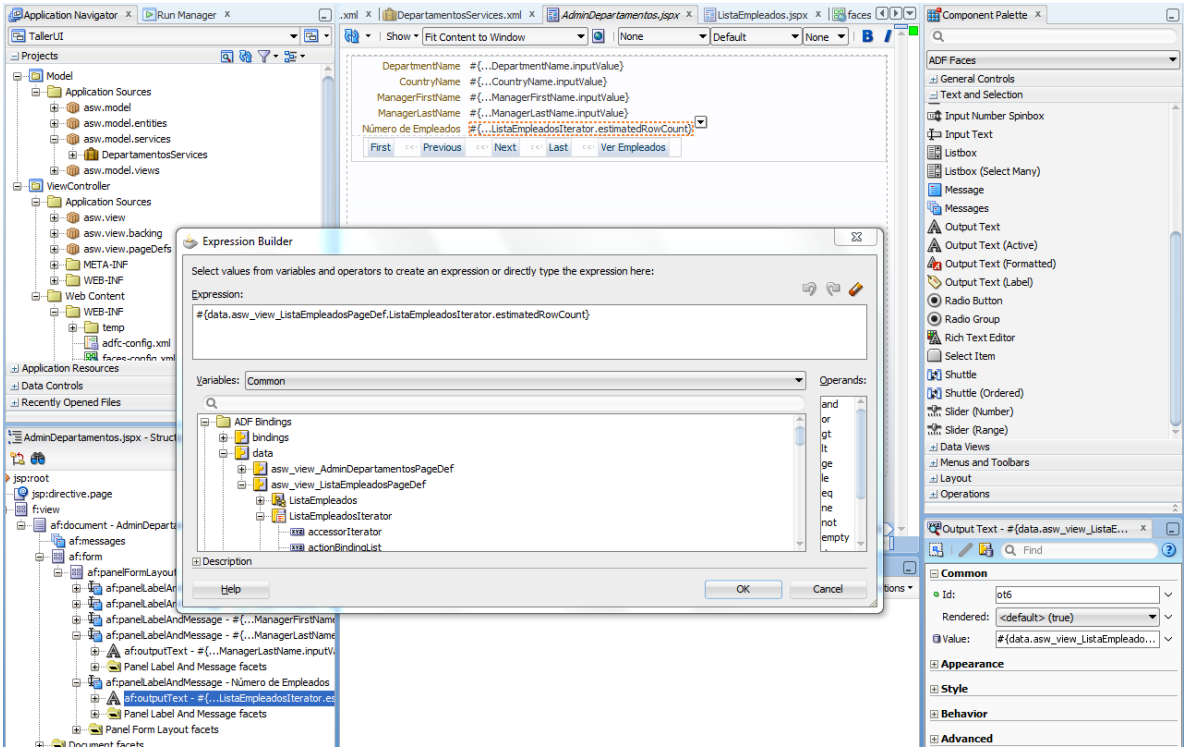
Objetivo C: Introducir aspectos adicionales a la forma, empleando Expression Language (EL) y bindings.

Paso 13: Perfeccione la forma:

1. Adicione a la forma un campo con el número de empleados que pertenecen al departamento.

a. Del Component Palette tome un componente de tipo OutputText y arrastrelo af:panelForm (en la ventana Structure).
b. Asigne los siguientes atributos al componente InputText:

Label	Número de empleados
Value	#{data.asw_view_ListaEmpleadosPageDef.ListaEmpleadosIterator.estimatedRowCount}



2. En caso que exista UN UNICO empleado asociado al departamento, no ilustre el botón "Lista de Empleados".
    a. Emplee el atributo Rendered del botón "Lista de Empleados" para introducir la expresión booleana:

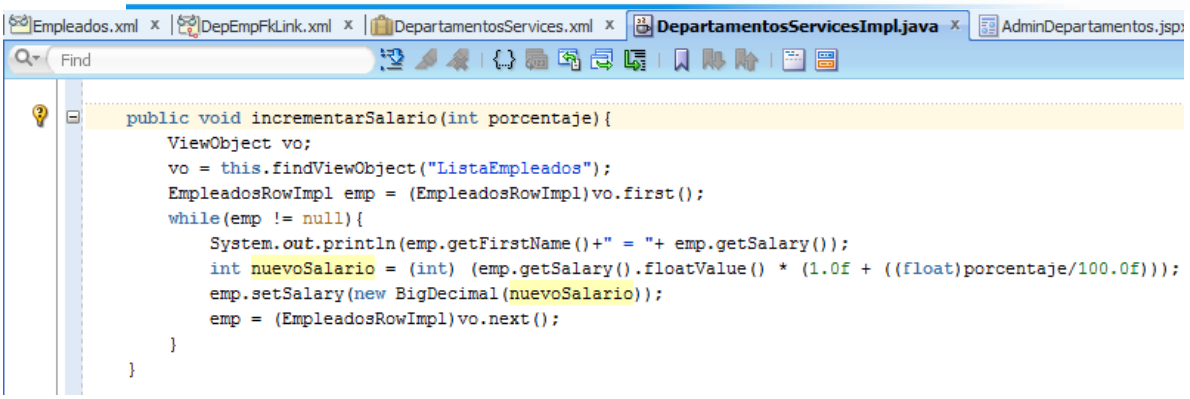
    #{data.asw_view_ListaEmpleadosPageDef.ListaEmpleadosIterator.estimatedRowCount >1}

    Esta expresión es VERDADERO cuando existen empleados asociados, diferentes al gerente (*Manager*).

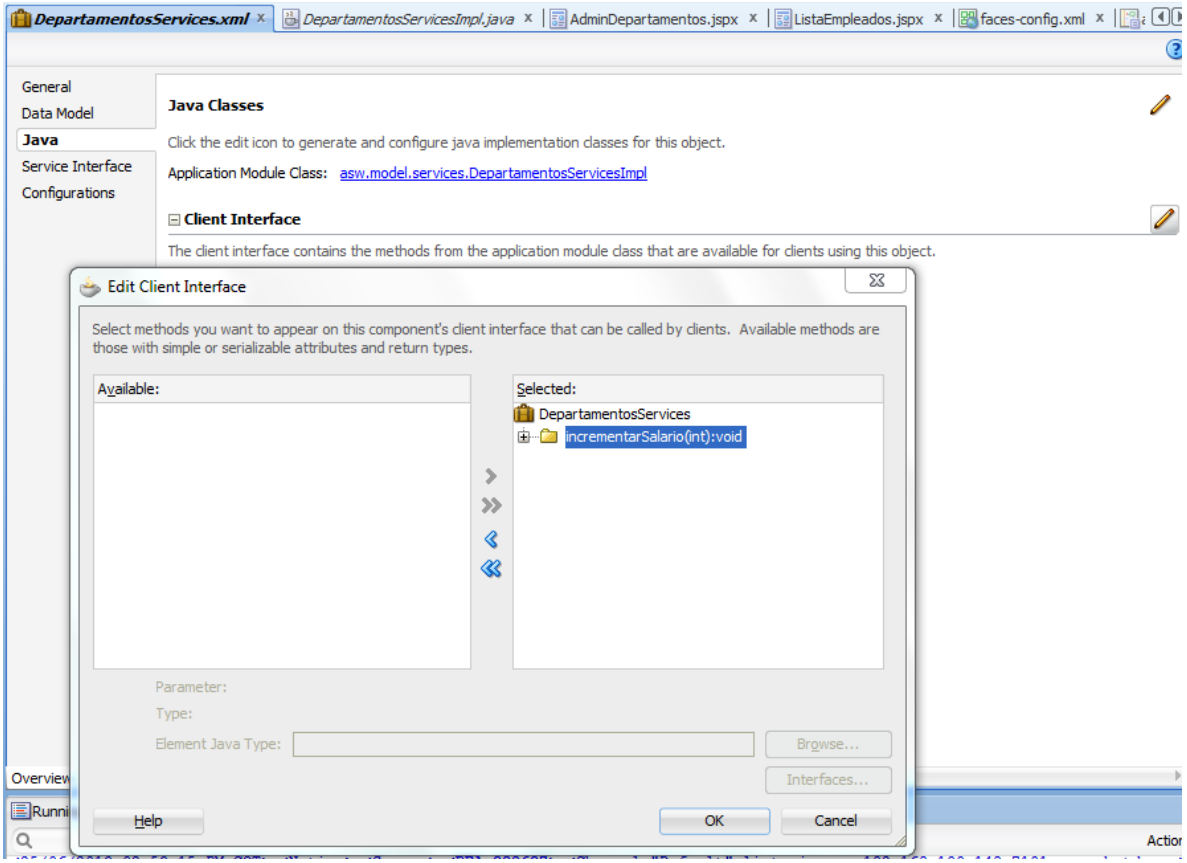
Objetivo D: Emplear binds a métodos, para realizar operaciones sobre el business services. El usuario final debe contar con la funcionalidad de incrementar los salarios de TODO un departamento, especificando el porcentaje de incremento (como un valor entre 1 y 100).

Paso 14: Adicionar el método incrementarSalario al applicationModule DepartamentosServices. Para esto debe editar su respectivo archivo java (DepartamentosServicesImpl.java).

Lab 5 – Creación de Interfaces de Usuario



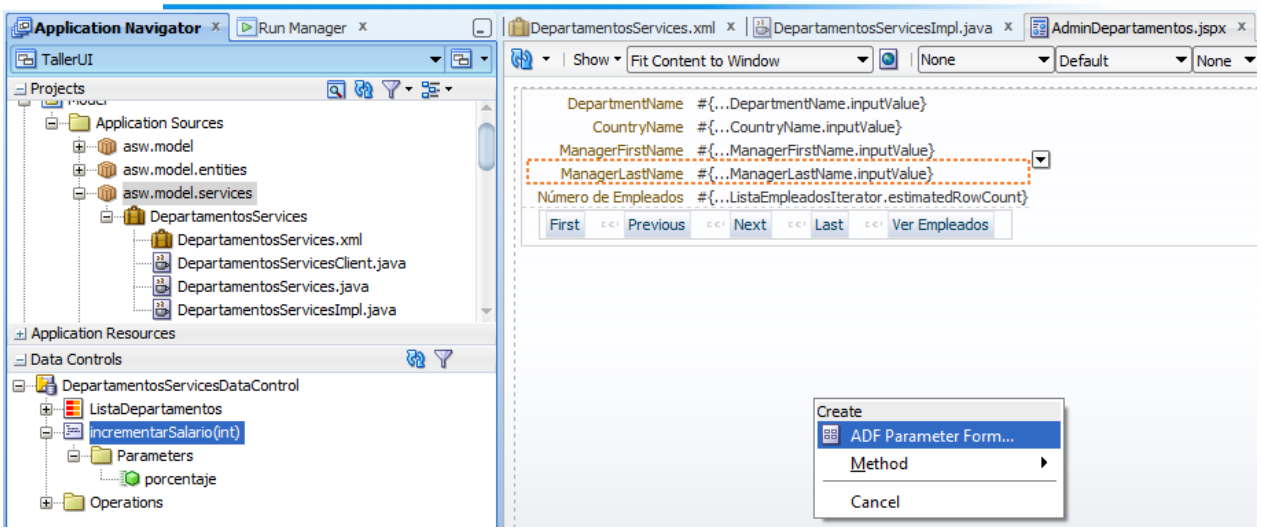
Paso 15: Publique el método incremetarSalario en la interface cliente.



Con este paso garantiza que el método esté disponible en el Data Control Palette.

Paso 16: Edite la página AdminDepartamentos.jsp y adicione un componente ADF Parameter Form a partir del método incrementarSalario(int).

Lab 5 – Creación de Interfaces de Usuario



Observe que el método se encuentre en el PageDefinition de la página (AdminDepartamentosPageDef.xml)

Paso 17: Pruebe la forma.

Paso 18: Cuestionario

a. Por qué el método incrementarSalario no recibe la identificación del Departamento sobre el cual se va a realizar el incremento ?
b. Explique el siguiente binding: *bindings.incrementarSalario_porcentaje*
c. Qué cambios hay que hacer para dejar el incremento como un valor constante ?
d. Qué cambios hay que hacer para que el porcentaje de incremento corresponda al número de empleados en dicho departamento ?

.