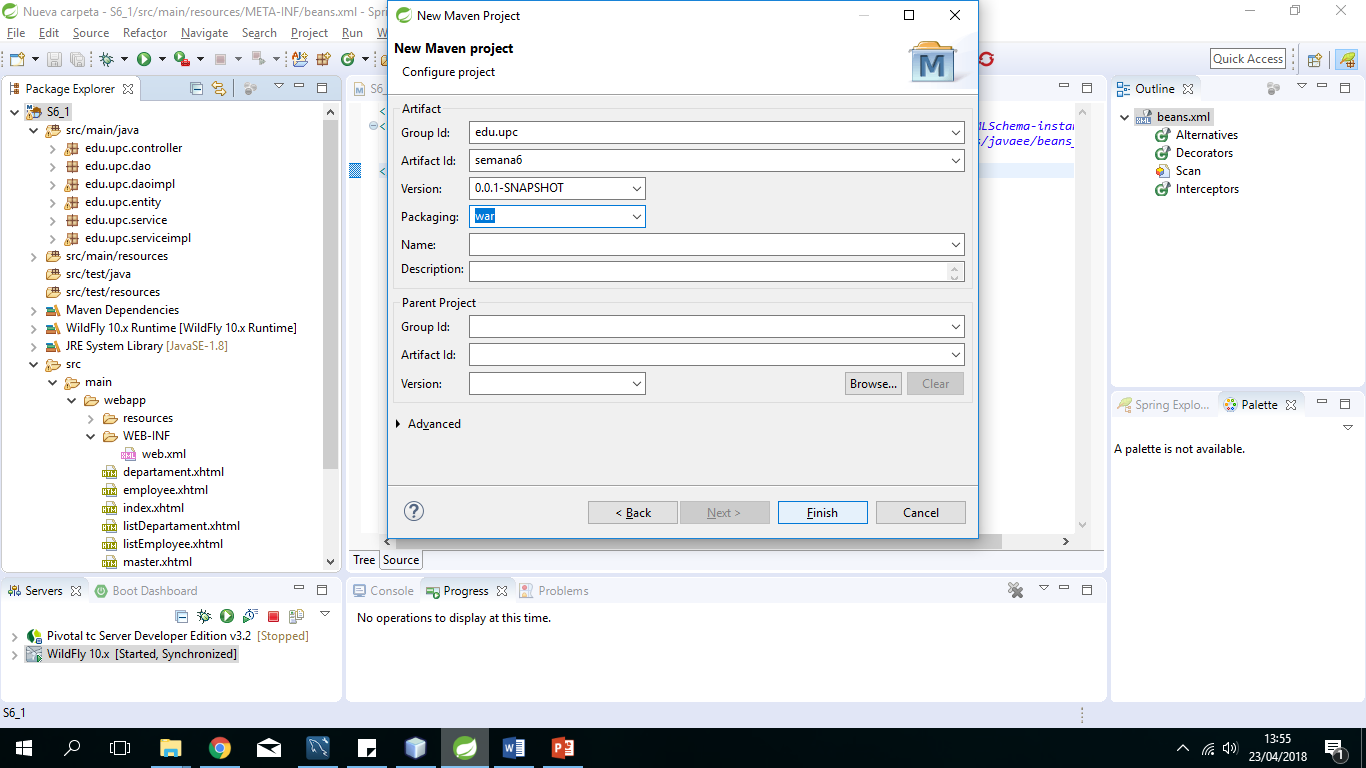
**EJEMPLO JPA-JSF-CDI-EJB**

Crear el proyecto



pom.xml: incluir la etiqueta build

|  |
| --- |
| <build>  <finalName>semana6</finalName>  <plugins>  <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-compiler-plugin</artifactId>  <version>3.1</version>  <configuration>  <source>1.8</source>  <target>1.8</target>  </configuration>  </plugin>  </plugins>  </build> |

Dependencias

|  |
| --- |
| <dependencies>  <dependency>  <groupId>javax</groupId>  <artifactId>javaee-api</artifactId>  <version>7.0</version>  <scope>provided</scope>  </dependency>  <!—API de JSF -->  <dependency>  <groupId>com.sun.faces</groupId>  <artifactId>jsf-api</artifactId>  <version>2.2.13</version>  <scope>provided</scope>  </dependency>  <!—Implementación de JSF -->  <dependency>  <groupId>com.sun.faces</groupId>  <artifactId>jsf-impl</artifactId>  <version>2.2.13</version>  <scope>provided</scope>  </dependency>  <!-- mysql-connector-java -->  <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  <version>5.1.38</version>  </dependency>  <!—Dependencia de primefaces -->  <dependency>  <groupId>org.primefaces</groupId>  <artifactId>primefaces</artifactId>  <version>6.0</version>  </dependency>  <!—Dependencia de omnifaces para la lista desplegable -->  <dependency>  <groupId>org.omnifaces</groupId>  <artifactId>omnifaces</artifactId>  <version>2.5.1</version>  </dependency>    <!-- Para acceder a los temas de primefaces -->  <dependency>  <groupId>org.primefaces.themes</groupId>  <artifactId>all-themes</artifactId>  <version>1.0.10</version>  </dependency>    </dependencies> |

Repositorio de los temas de primefaces

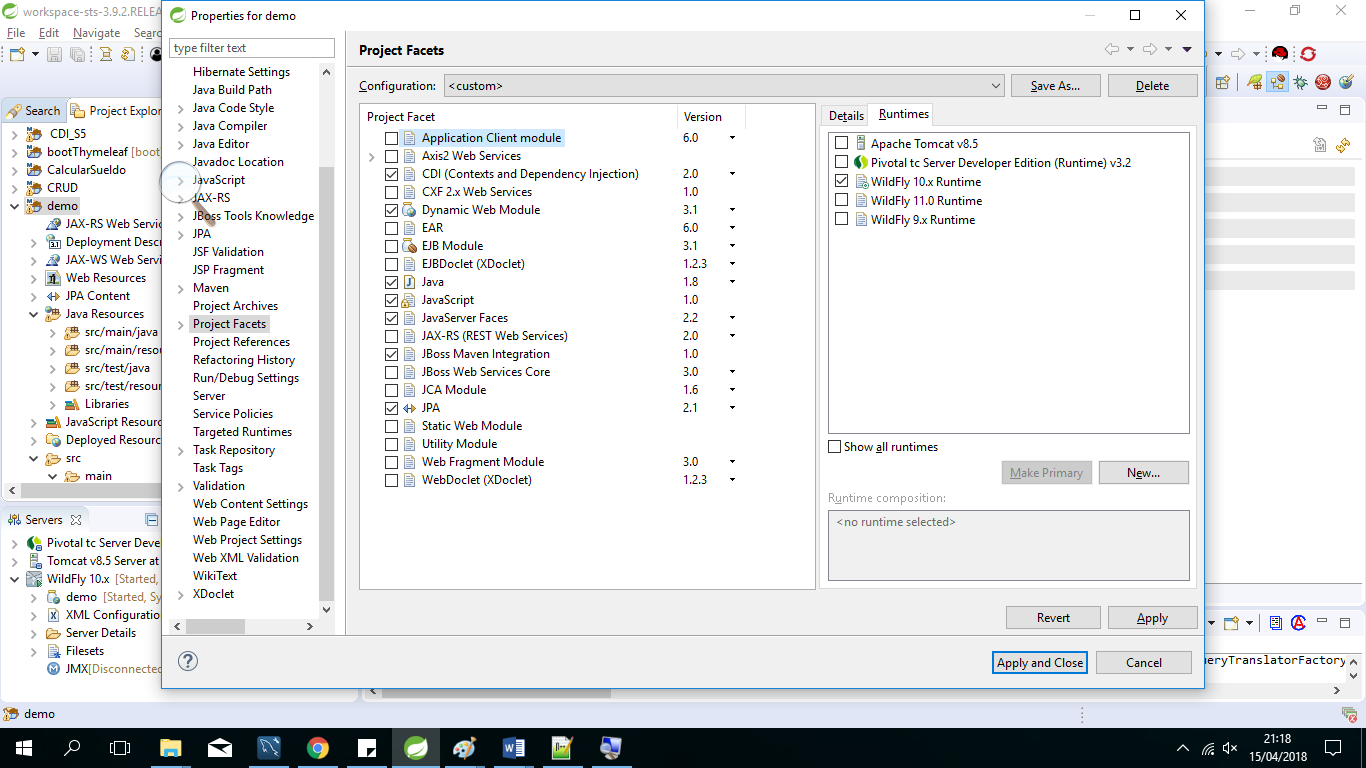
|  |
| --- |
| <repositories>  <repository>  <id>prime-repo</id>  <name>PrimeFaces Maven Repository</name>  <url>http://repository.primefaces.org</url>  <layout>default</layout>  </repository>  </repositories> |

-Generar el descriptor: web.xml, cambiar la versión a 3.1 y agregar

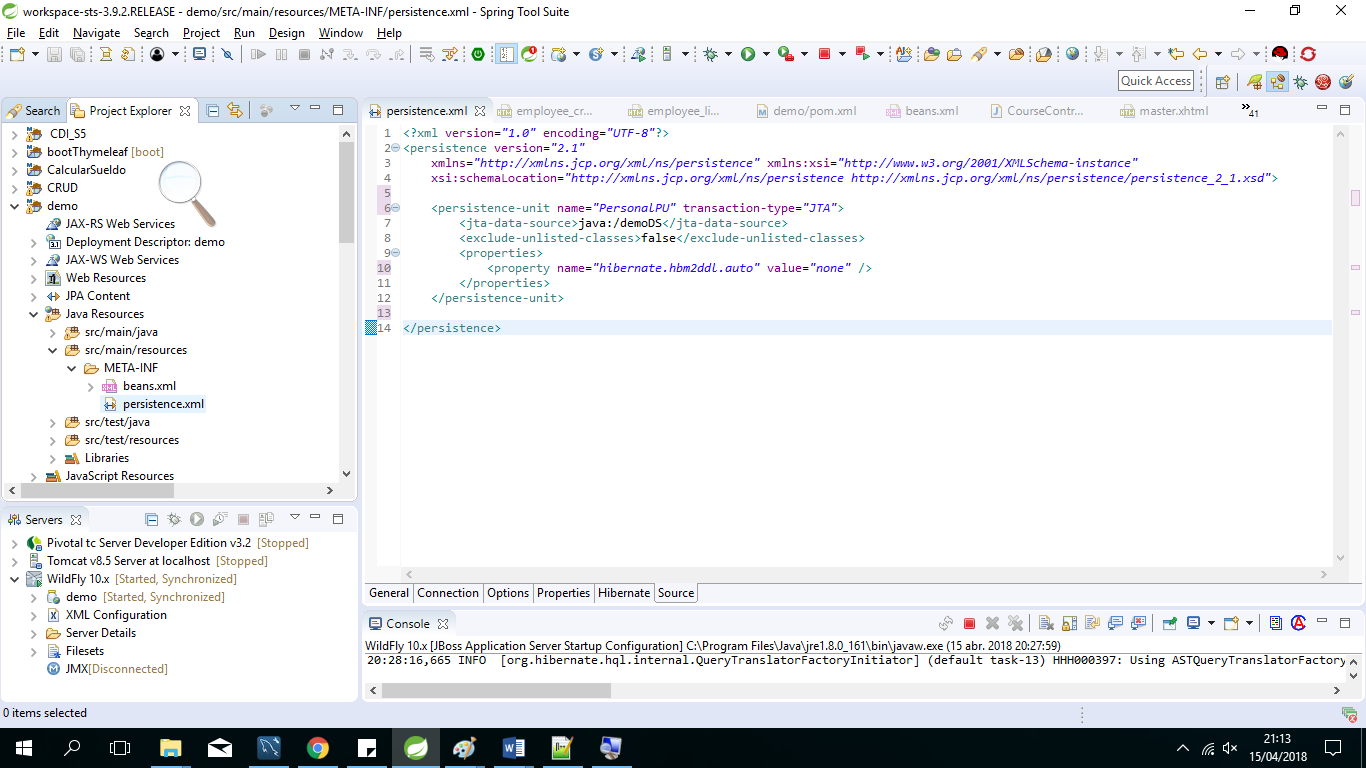
|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"*  xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"*  version=*"3.1"*>  <display-name>S6\_1</display-name>  <welcome-file-list>  <welcome-file>index.xhtml</welcome-file>  </welcome-file-list>  <!-- definimos una variable primefaces.THEME cuyo valor es bootstrap -->  <context-param>  <param-name>primefaces.THEME</param-name>  <param-value>bootstrap</param-value>  </context-param>  <!-- Define el nombre del Servlet y la -->  <servlet>  <servlet-name>Faces Servlet</servlet-name>  <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>  <load-on-startup>1</load-on-startup>  </servlet>  <!-- Añada correlaciones de servlets en el archivo WEB-INF/web.xml  para cada definición de servlet. La vía de acceso de servlet se añadirá  en la raíz de contexto de la aplicación web -->  <servlet-mapping>  <servlet-name>Faces Servlet</servlet-name>  <url-pattern>\*.xhtml</url-pattern>  </servlet-mapping>  </web-app> |

- Anticlick sobre el proyecto:

* Asignar el server
* seleccionar Project faces : chek a JPA y a CDI



Buscar la carpeta META-INF en el src/main/java y moverla a la carpeta src/main/resources, además traer el archivo beans.xml a esta carpeta ya que se genera en el WEB-INF



-En el archivo beans.xml , modificar el valor de la propiedad bean-discovery-mode=”all”

<beans xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

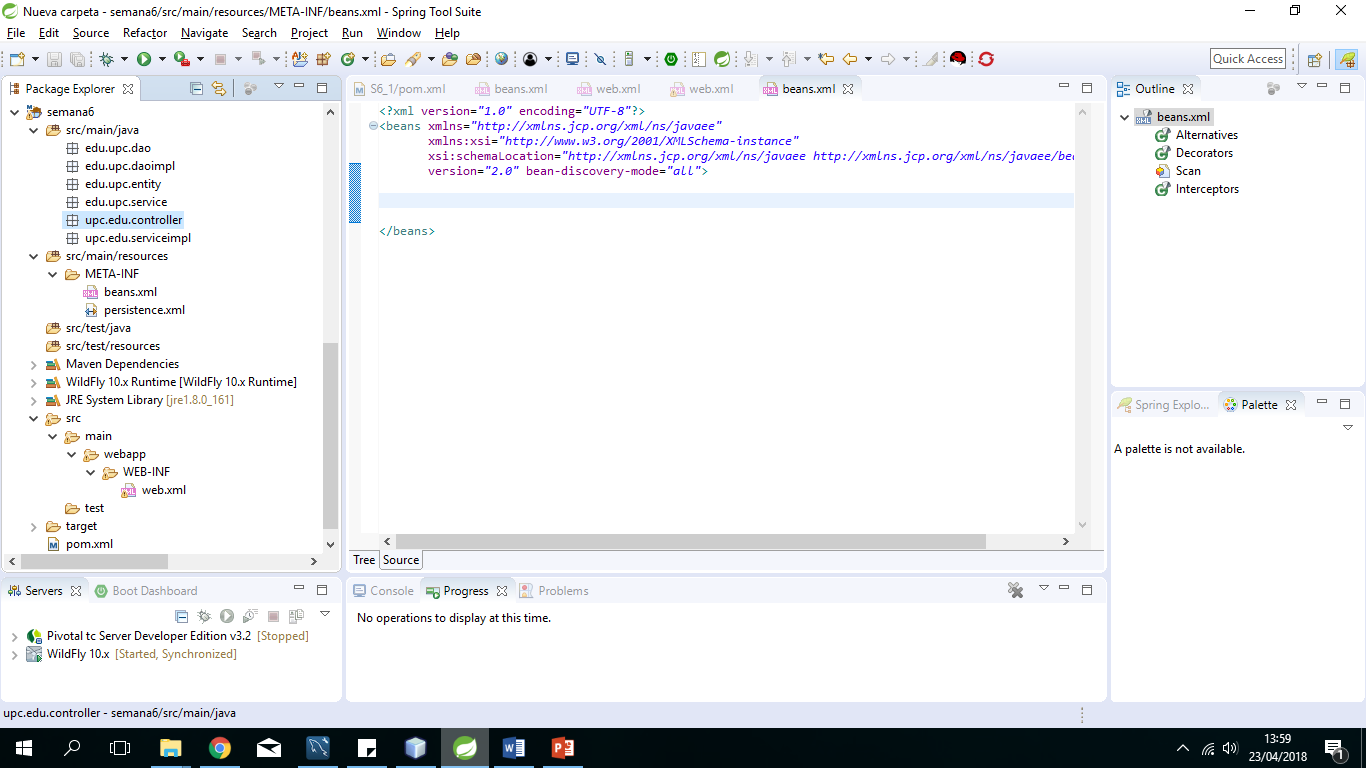
xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/beans\_2\_0.xsd"*

version=*"2.0"* bean-discovery-mode=*"all"*>

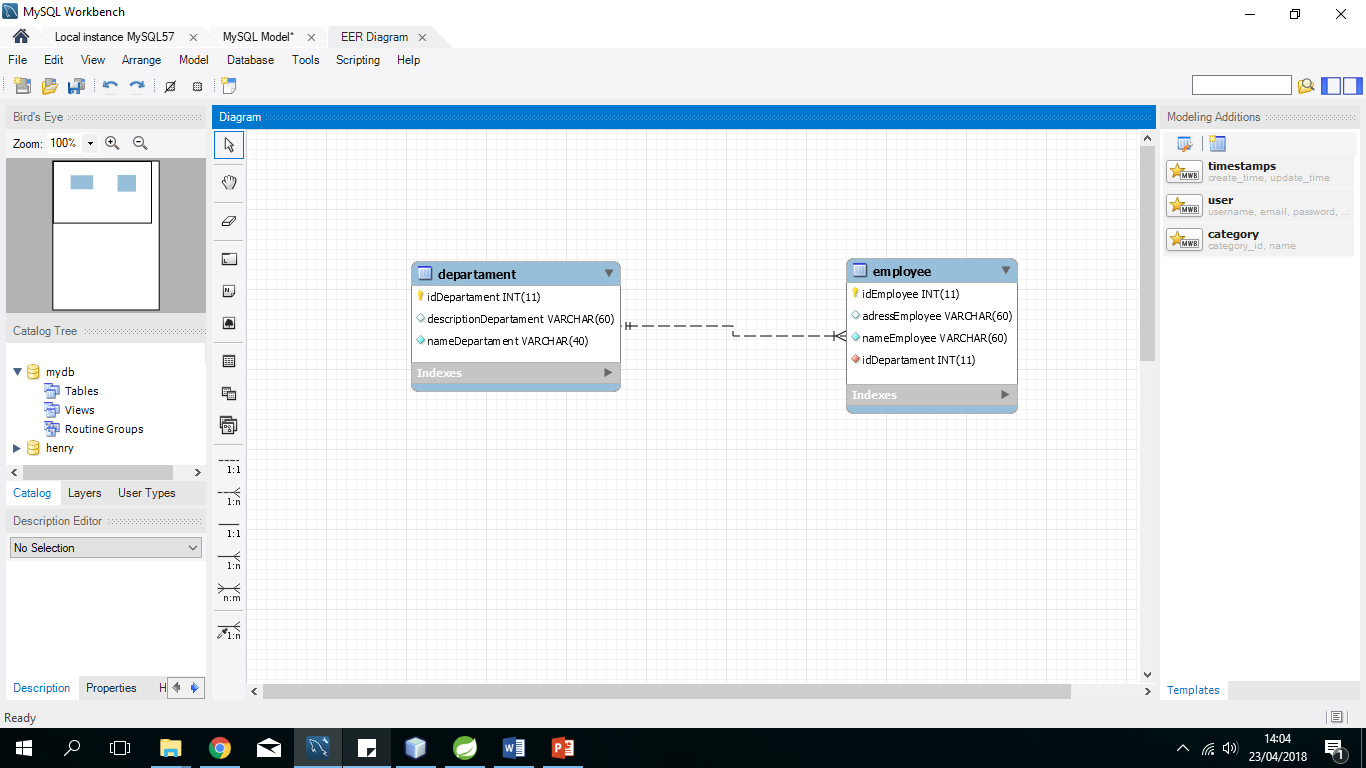
-El archivo persistence.xml, debemos agregar los datos del datasource y el modo de creación de tablas:

|  |
| --- |
| <persistence-unit name=*"semana6PU"* transaction-type=*"JTA"*>  <jta-data-source>java:/ejemploDS</jta-data-source>  <exclude-unlisted-classes>false</exclude-unlisted-classes>  <properties>  <property name=*"hibernate.hbm2ddl.auto"* value=*"create"* />  </properties>  </persistence-unit> |

-crear los paquetes utilizando ahora la capa de servicios:



* Crear clases en el paquete edu.upc.entity:



* Crear la clase Departament

Agregar el implements Serializable a cada clase en el paquete entity

Agregar la anotación:

@Entity

@Table(name = " Departament")

Luego, agregar los atributos y anotar:

@Id

@GeneratedValue(strategy=GenerationType.***IDENTITY***)

**private** **int** idDepartament;

@Column(name = "nameDepartament", length = 45, nullable = **false**)

**private** String nameDepartament;

Generar los get y set de la clase y hash and equals code

* crear la clase Employee

Agregar el implements Serializable a cada clase en el paquete entity

Agregar la anotación:

@Entity

@Table(name = " Employee")

Luego, agregar los atributos y anotar:

@Id

@GeneratedValue(strategy=GenerationType.***IDENTITY***)

**private** **int** idEmployee;

@Column(name="firstNameEmployee",nullable=**false**,length=60)

**private** String firstNameEmployee;

@Column(name="lastNameEmployee",nullable=**false**,length=60)

**private** String lastNameEmployee;

@ManyToOne

@JoinColumn(name = "idDepartament", nullable = **false**)

**private** Departament departament;

Generar los get y set de la clase

**-En el paquete dao**

Crear 3 interfaces

1. IDAO<T>.java, agregar el operador de superclase <T> y agregar los métodos

|  |
| --- |
| **package** edu.upc.dao;  **import** java.util.List;  **public** **interface** IDAO<T> {  **public** **void** insertar(T t);  **public** **void** actualizar(T t);  **public** **void** eliminar(**int** idT);  **public** List<T> listar();  **public** T listarId(**int** idT);  } |

1. IDepartamentDAO.java

Crear la interfaz y la anotamos con @Local e importar el paquete: **import** javax.ejb.Local;

Y agregamos los métodos propios de la interfaz

|  |
| --- |
| **package** edu.upc.dao;  **import** java.util.List;  **import** javax.ejb.Local;  **import** edu.upc.entity.Departament;  @Local  **public** **interface** IDepartamentDAO **extends** IDAO<Departament> {  **public** List<Departament> buscarNombre(Departament departament);  } |

1. IEmployeeDAO.java

Crear la interfaz y la anotamos con @Local e importar el paquete: **import** javax.ejb.Local;

Y agregamos los métodos propios de la interfaz

|  |
| --- |
| **package** edu.upc.dao;  **import** java.util.List;  **import** javax.ejb.Local;  **import** edu.upc.entity.Employee;  @Local  **public** **interface** IEmployeeDAO **extends** IDAO<Employee>{  **public** List<Employee> buscarNombre(Employee employee);  } |

**-En el paquete dao.impl,** crear las implementaciones de los métodos del dao

1. Crear DepartamentDAOImpl.java

-Agregar la anotación @stateless a la clase e importar el paquete: **import** javax.ejb.Stateless;

-Agregar: **implements** IDepartamentDAO, Serializable a la clase e importar los paquetes y la interface: **import** java.io.Serializable;

**-En la clase agregar el:**

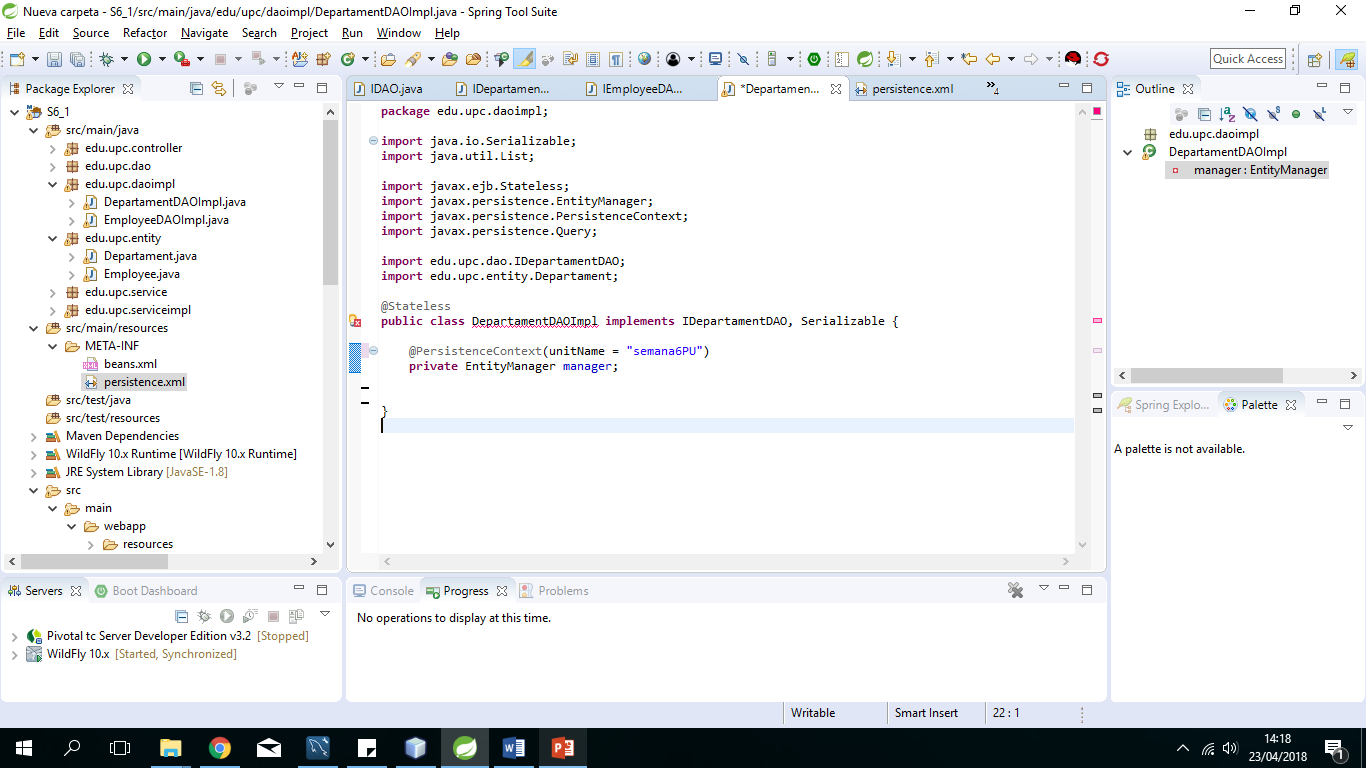
@PersistenceContext(unitName = "semana6PU")//agregar el nombre de la Unidad de persistencia.

-Agregar el: **private** EntityManager manager;

E importar los paquetes:

**import** javax.persistence.EntityManager;

**import** javax.persistence.PersistenceContext;



Sobre escribir e implementar los métodos en la clase:

|  |
| --- |
| @Override  **public** **void** insertar(Departament t) {  **try** {  manager.persist(t);  } **catch** (Exception e) {  e.getMessage();  }  }  @Override  **public** **void** actualizar(Departament t) {  **try** {  manager.merge(t);  } **catch** (Exception e) {  e.getMessage();  }  }  @Override  **public** **void** eliminar(**int** idT) {  **try** {  Departament dep = **new** Departament();  dep = manager.getReference(Departament.**class**, idT);  manager.remove(dep);  } **catch** (Exception e) {  e.getMessage();  }  }  @SuppressWarnings("unchecked")  @Override  **public** List<Departament> listar() {  List<Departament> lista = **null**;  **try** {  Query query = manager.createQuery("FROM Departament d");  lista = (List<Departament>) query.getResultList();  } **catch** (Exception e) {  e.getMessage();  }  **return** lista;  }  @SuppressWarnings("unchecked")  @Override  **public** Departament listarId(**int** idT) {  List<Departament> lista = **null**;  Departament dept =**new** Departament();  **try** {    Query query = manager.createQuery("FROM Departament d where d.idDepartament = ?1");  query.setParameter(1, idT);  lista = (List<Departament>) query.getResultList();  **if**(lista!=**null**&& !lista.isEmpty()) {  dept=lista.get(0);  }  } **catch** (Exception e) {  e.getMessage();  }    **return** dept;  }  @SuppressWarnings("unchecked")  @Override  **public** List<Departament> buscarNombre(Departament departament) {  List<Departament> lista = **null**;  **try** {  Query query = manager.createQuery("FROM Departament d where d.nameDepartament like ?1");  query.setParameter(1, "%" + departament.getNameDepartament() + "%");  lista = (List<Departament>) query.getResultList();  } **catch** (Exception e) {  e.getMessage();  }  **return** lista;  } |

1. Crear EmployeeDAOImpl.java

-Agregar la anotación @stateless a la clase e importar el paquete: **import** javax.ejb.Stateless;

-Agregar: **implements** IEmployeeDAO, Serializable a la clase e importar los paquetes y la interface: **import** java.io.Serializable;

**-En la clase agregar el:**

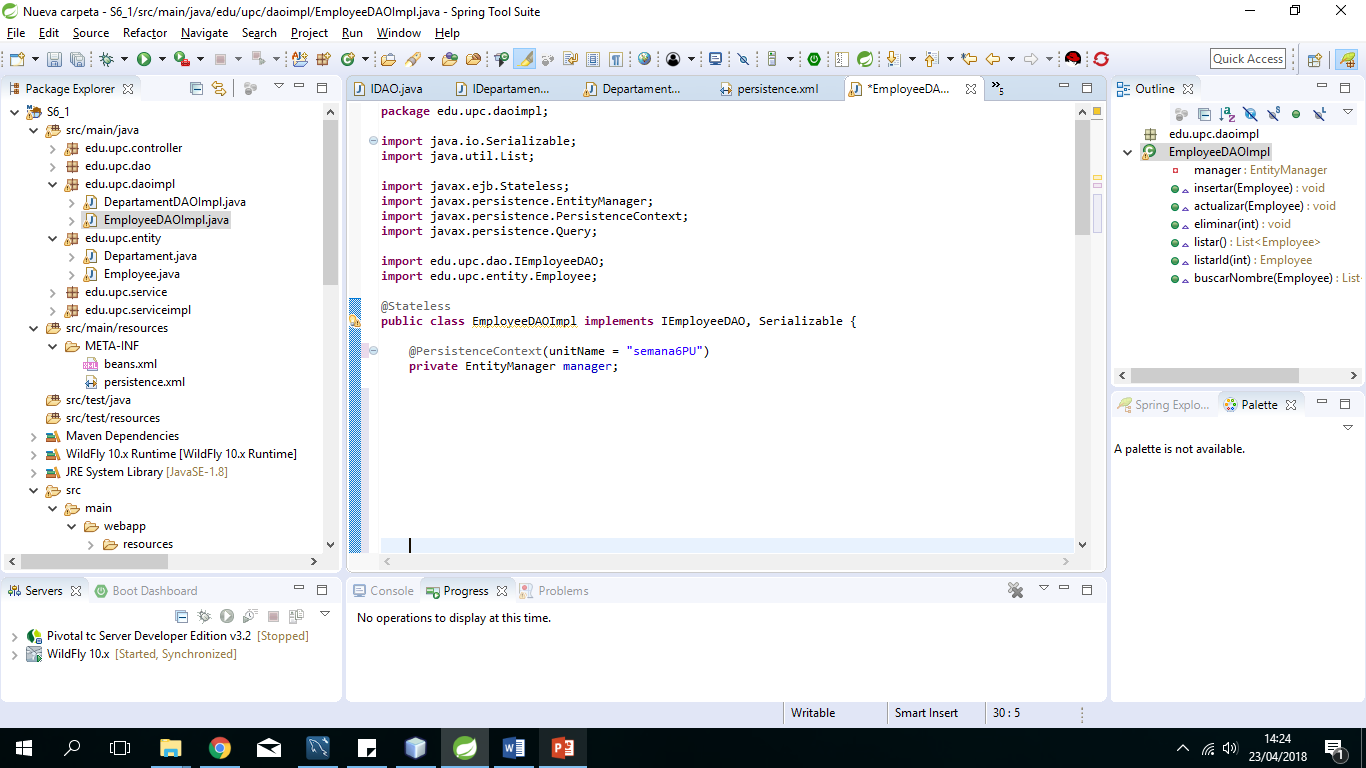
@PersistenceContext(unitName = "semana6PU")//agregar el nombre de la Unidad de persistencia.

-Agregar el: **private** EntityManager manager;

E importar los paquetes:

**import** javax.persistence.EntityManager;

**import** javax.persistence.PersistenceContext;



**-Agregar los métodos a implementar en la clase:**

|  |
| --- |
| @Override  **public** **void** insertar(Employee t) {  **try** {  manager.persist(t);  } **catch** (Exception e) {  e.getMessage();  }  }  @Override  **public** **void** actualizar(Employee t) {  **try** {  manager.merge(t);  } **catch** (Exception e) {  e.getMessage();  }  }  @Override  **public** **void** eliminar(**int** idT) {  **try** {  Employee empl = **new** Employee();  empl = manager.getReference(Employee.**class**, idT);  manager.remove(empl);  } **catch** (Exception e) {  e.getMessage();  }  }  @SuppressWarnings("unchecked")  @Override  **public** List<Employee> listar() {  List<Employee> lista = **null**;  **try** {  Query query = manager.createQuery("FROM Employee e");  lista = (List<Employee>) query.getResultList();  } **catch** (Exception e) {  e.getMessage();  }  **return** lista;  }  @SuppressWarnings("unchecked")  @Override  **public** Employee listarId(**int** idT) {  List<Employee> lista = **null**;  Employee em = **new** Employee();  **try** {  Query query = manager.createQuery("FROM Employee e where e.idEmployee=?1");  query.setParameter(1, idT);  lista = (List<Employee>) query.getResultList();  **if** (lista != **null** && !lista.isEmpty()) {  em = lista.get(0);  }  } **catch** (Exception e) {  e.getMessage();  }  **return** em;  }  @SuppressWarnings("unchecked")  @Override  **public** List<Employee> buscarNombre(Employee employee) {  List<Employee> lista = **null**;  **try** {  Query query = manager.createQuery("FROM Employee e where e.nameEmployee like ?1");  query.setParameter(1, "%" + employee.getNameEmployee() + "%");  lista=(List<Employee>) query.getResultList();  } **catch** (Exception e) {  e.getMessage();  }  **return** lista;  } |

**En el paquete service: creamos las interfaces**

**-**Crear la interfaz ISERVICE<T>.java

|  |
| --- |
| **package** edu.upc.service;  **import** java.util.List;  **public** **interface** ISERVICE<T> {    **public** **void** insertar(T t);  **public** **void** actualizar(T t);  **public** **void** eliminar(**int** idT);  **public** List<T> listar() **throws** Exception;  **public** T listarId(**int** idT) **throws** Exception;  } |

-Crear la interfaz IDepartamentSERVICE.java, extendemos de la ISERVICE (superclase)e implementamos los métodos propios de IDepartamentSERVICE

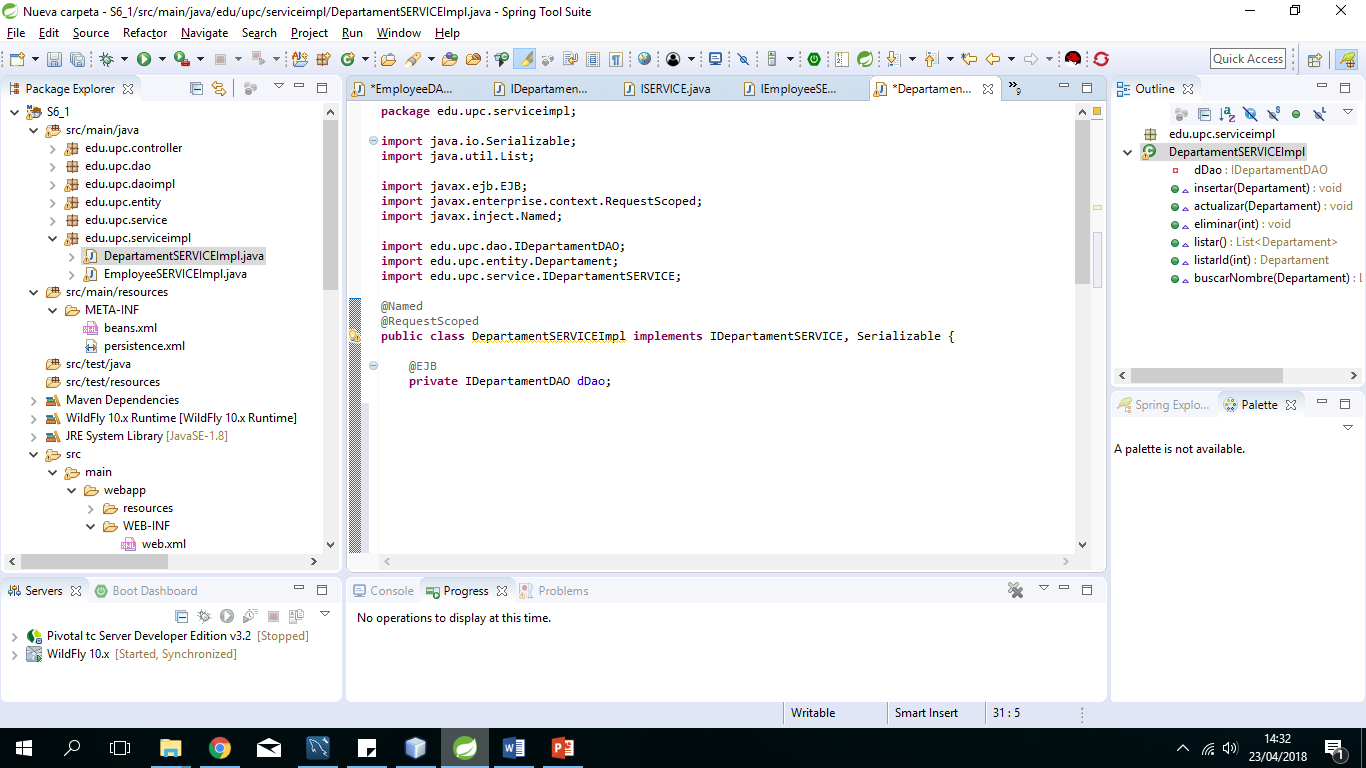
|  |
| --- |
| **package** edu.upc.service;  **import** java.util.List;  **import** edu.upc.entity.Departament;  **public** **interface** IDepartamentSERVICE **extends** ISERVICE<Departament> {  **public** List<Departament> buscarNombre(Departament departament) **throws** Exception;  } |

-Crear la interfaz IEmployeeSERVICE.java, extendemos de la ISERVICE (superclase)e implementamos los métodos propios de IEmployeeSERVICE

|  |
| --- |
| **package** edu.upc.service;  **import** java.util.List;  **import** edu.upc.entity.Employee;  **public** **interface** IEmployeeSERVICE **extends** ISERVICE<Employee> {  **public** List<Employee> buscarNombre(Employee employee) **throws** Exception;  } |

En el paquete serviceimpl: creamos clases para implementar las interfaces del service

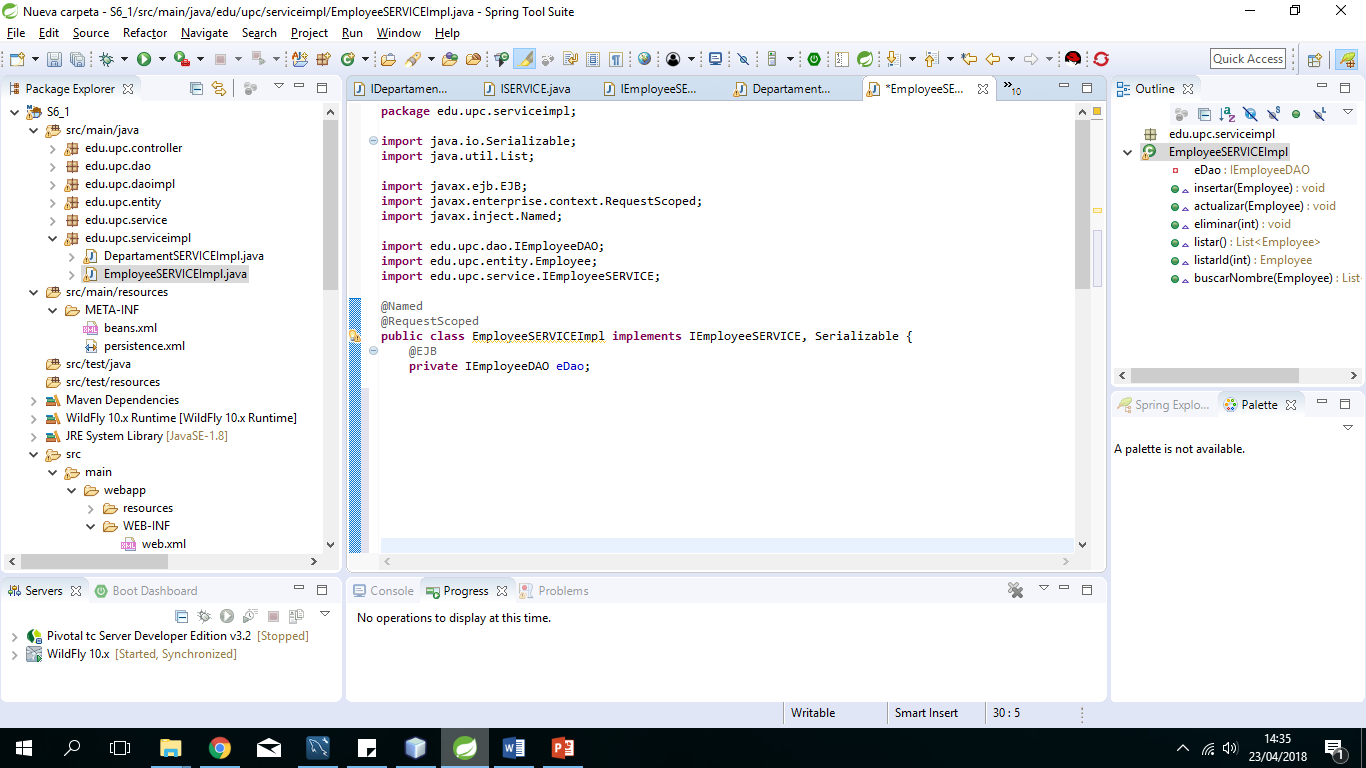
-Crear DepartamentSERVICEImpl.java, agregamos las anotaciones de CDI (@Named,@RequestScoped) e inyectamos con @EJB para tener acceso al dao. Además implementamos IDepartamentSERVICE y el Serializable



**Sobre escribimos e implementamos los métodos :**

|  |
| --- |
| @Override  **public** **void** insertar(Departament t) {  **try** {  dDao.insertar(t);  } **catch** (Exception e) {  e.getMessage();  }  }  @Override  **public** **void** actualizar(Departament t) {  **try** {  dDao.actualizar(t);  } **catch** (Exception e) {  e.getMessage();  }  }  @Override  **public** **void** eliminar(**int** idT) {  **try** {  dDao.eliminar(idT);  } **catch** (Exception e) {  e.getMessage();  }  }  @Override  **public** List<Departament> listar() **throws** Exception {  **return** dDao.listar();  }  @Override  **public** Departament listarId(**int** idT) **throws** Exception {  **return** dDao.listarId(idT);  }  @Override  **public** List<Departament> buscarNombre(Departament departament) **throws** Exception {  **return** dDao.buscarNombre(departament);  } |

-Crear EmployeeSERVICEImpl.java, agregamos las anotaciones de CDI(@Named,@RequestScoped) e inyectamos con @EJB para tener acceso al dao. Además implementamos IEmployeeSERVICE y el Serializable



**Sobre escribimos e inyectamos los métodos:**

|  |
| --- |
| @Override  **public** **void** insertar(Employee t) {  **try** {  eDao.insertar(t);  } **catch** (Exception e) {  e.getMessage();  }  }  @Override  **public** **void** actualizar(Employee t) {  **try** {  eDao.actualizar(t);  } **catch** (Exception e) {  e.getMessage();  }  }  @Override  **public** **void** eliminar(**int** idT) {  **try** {  eDao.eliminar(idT);  } **catch** (Exception e) {  e.getMessage();  }  }  @Override  **public** List<Employee> listar() **throws** Exception {  **return** eDao.listar();  }  @Override  **public** Employee listarId(**int** idT) **throws** Exception {  **return** eDao.listarId(idT);  }  @Override  **public** List<Employee> buscarNombre(Employee employee) **throws** Exception {  **return** eDao.buscarNombre(employee);  } |

En el paquete controller, creamos los controladores:

* Crear DepartamentController.java, y agregamos las anotaciones que permitan inyectar los beans(@Named,@SessionScoped) e importamos los paquetes

**import** javax.enterprise.context.SessionScoped;

**import** javax.inject.Named;

* Agregar implements Serializable e importamos el paquete: **import** java.io.Serializable;
* Agregamos las variables para manipular los datos:

**private** IDepartamentSERVICE departamentService;

**private** Departament departament;

**private** List<Departament> listaDepartamentos;

* **Agregamos la anotación @Inject para acceder a IDepartamentSERVICE**

@Inject

**private** IDepartamentSERVICE departamentService;

* Agregamos el constructor y la anotación de CDI @PostConstruct

@PostConstruct

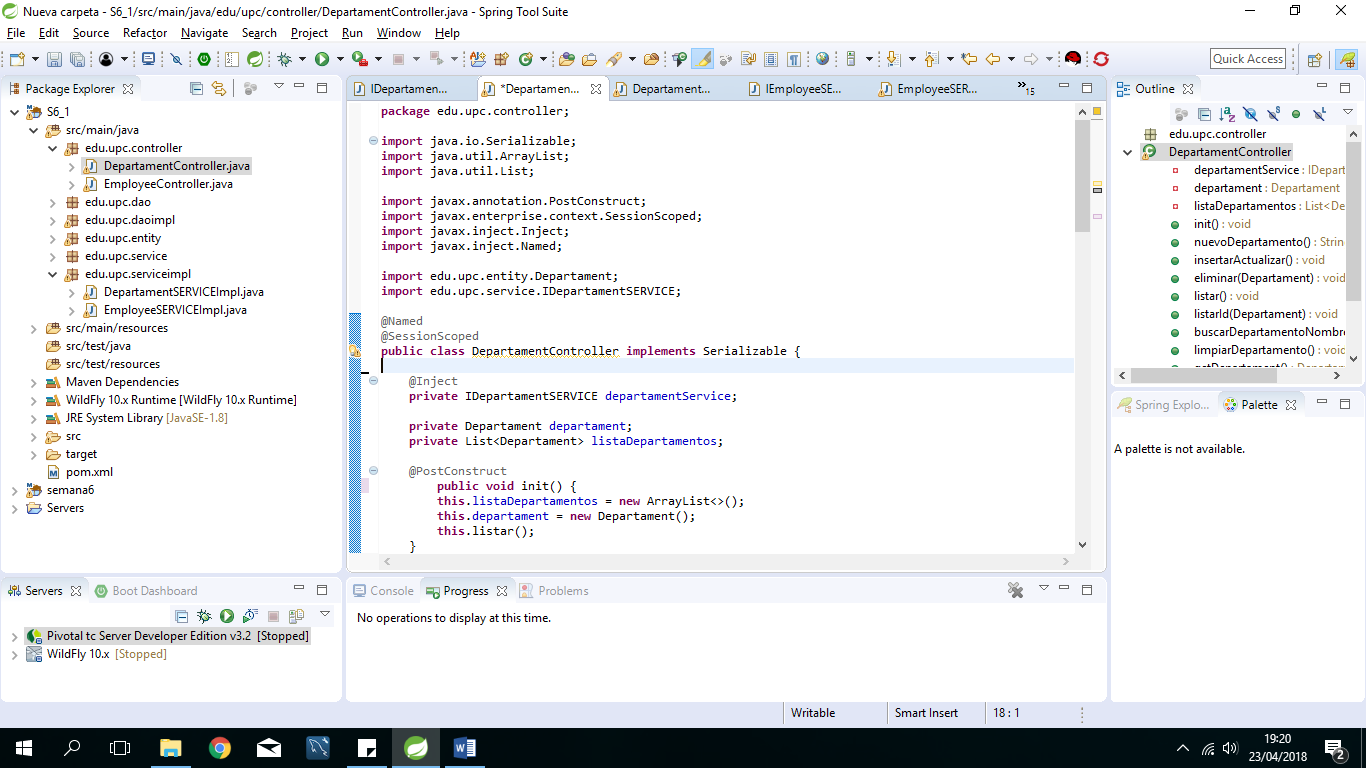
**public** **void** init() {

**this**.listaDepartamentos = **new** ArrayList<>();

**this**.departament = **new** Departament();

**this**.listar();

}



* Agregamos los métodos para traer los datos y enviarlos a los \*.xhtml

|  |
| --- |
| **public** String nuevoDepartamento() {  **this**.setDepartament(**new** Departament());  **return** "departament.xhtml";  }  **public** **void** insertarActualizar() {  **try** {  **if** (departament.getIdDepartament() > 0) {  departamentService.actualizar(departament);  limpiarDepartamento();  } **else** {  departamentService.insertar(departament);  limpiarDepartamento();  }  **this**.listar();  } **catch** (Exception e) {  e.getMessage();  }  }  **public** **void** eliminar(Departament departament) {  **try** {  departamentService.eliminar(departament.getIdDepartament());  listar();  } **catch** (Exception e) {  e.getMessage();  }  }  **public** **void** listar() {  **try** {  listaDepartamentos = departamentService.listar();  } **catch** (Exception e) {  e.getMessage();  }  }  **public** **void** listarId(Departament dept) {  **try** {  departament = departamentService.listarId(dept.getIdDepartament());  } **catch** (Exception e) {  e.getMessage();  }  }  **public** **void** buscarDepartamentoNombre() {  **try** {  **if** (departament.getNameDepartament().isEmpty()) {  **this**.listar();  } **else** {  listaDepartamentos =**this**.departamentService.buscarNombre(**this**.getDepartament());  }  } **catch** (Exception e) {  e.getMessage();  }  }  **public** **void** limpiarDepartamento() {  **this**.init();  } |

* Generamos los get y set de las variables declaradas no incluir la interface
* Crear EmployeeController.java, y agregamos las anotaciones que permitan inyectar los beans(@Named,@SessionScoped) e importamos los paquetes

**import** javax.enterprise.context.SessionScoped;

**import** javax.inject.Named;

* Agregar implements Serializable e importamos el paquete: **import** java.io.Serializable;
* Agregamos las variables para manipular los datos:

**private** IDepartamentSERVICE departamentService;

**private** IEmployeeSERVICE employeeService;

**private** Departament departament;;

**private** List<Departament> listaDepartamentos;

**private** Employee employee;

**private** List<Employee> listaEmpleados;

* Agregamos la anotación @Inject para acceder a IDepartamentSERVICE y al IEmployeeSERVICE:

@Inject

**private** IDepartamentSERVICE departamentService;

@Inject

**private** IEmployeeSERVICE employeeService;

* Agregamos el constructor y la anotación de CDI @PostConstruct

@PostConstruct

**public** **void** init() {

departament = **new** Departament();

employee = **new** Employee();

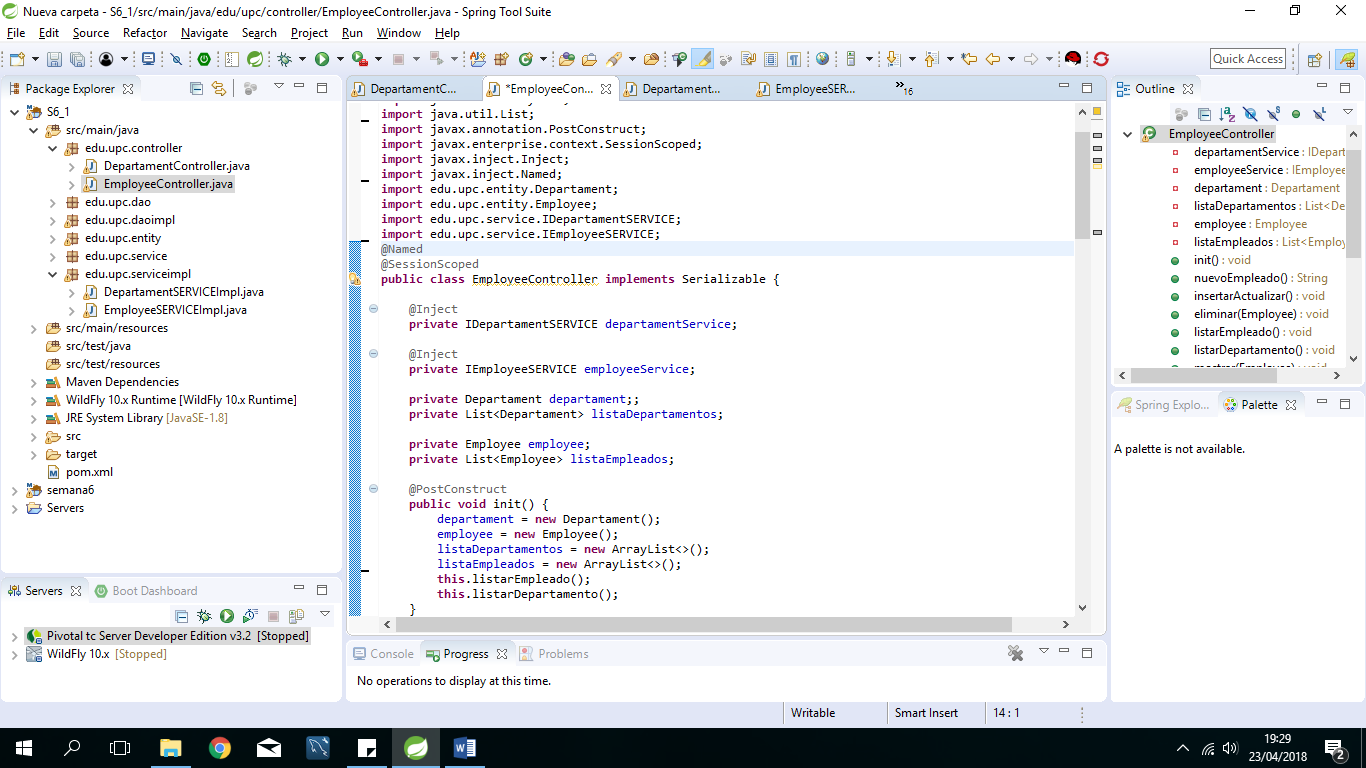
listaDepartamentos = **new** ArrayList<>();

listaEmpleados = **new** ArrayList<>();

**this**.listarEmpleado();

**this**.listarDepartamento();

}



* Agregamos los métodos para traer los datos y enviarlos a los \*.xhtml

|  |
| --- |
| **public** String nuevoEmpleado() {  **this**.setEmployee(**new** Employee());  **return** "employee.xhtml";  }  **public** **void** insertarActualizar() {  **try** {  **if** (employee.getIdEmployee() > 0) {  employeeService.actualizar(employee);  limpiarEmpleado();  } **else** {  employeeService.insertar(employee);  limpiarEmpleado();  }  **this**.listarEmpleado();  } **catch** (Exception e) {  e.getMessage();  }  }  **public** **void** eliminar(Employee employee) {  **try** {  employeeService.eliminar(employee.getIdEmployee());  listarEmpleado();  } **catch** (Exception e) {  e.getMessage();  }  }  **public** **void** listarEmpleado() {  **try** {  listaEmpleados = employeeService.listar();  } **catch** (Exception e) {  e.getMessage();  }  }  **public** **void** listarDepartamento() {  **try** {  listaDepartamentos = departamentService.listar();  } **catch** (Exception e) {  e.getMessage();  }  }  **public** **void** mostrar(Employee empl) {  **try** {  **this**.departament.setIdDepartament(empl.getDepartament().getIdDepartament());  **this**.departament.setNameDepartament(empl.getDepartament().getNameDepartament());  **this**.departament.setDescriptionDepartament(empl.getDepartament().getDescriptionDepartament());  **this**.employee.setIdEmployee(empl.getIdEmployee());  **this**.employee.setNameEmployee(empl.getNameEmployee());  **this**.employee.setAdressEmployee(empl.getAdressEmployee());  **this**.employee.setDepartament(departament);  } **catch** (Exception e) {  e.getStackTrace();  }  }  **public** **void** buscarEmployeeNombre() {  **try** {  **if** (employee.getNameEmployee().isEmpty()) {  **this**.listarEmpleado();  } **else** {  listaEmpleados = **this**.employeeService.buscarNombre(**this**.getEmployee());  }  } **catch** (Exception e) {  e.getMessage();  }  }  **public** **void** limpiarEmpleado() {  **this**.init();  } |

Generar los get y set de las variables, no incluir los service

* En la vista usaremos un tema de pirmefaces por lo que reutilizaremos una plantilla y un master
* Crear master.xhtml

|  |
| --- |
| <!DOCTYPE html>  <html lang=*"es"* class=*"no-js"* xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:f=*"http://java.sun.com/jsf/core"*  xmlns:h=*"http://java.sun.com/jsf/html"*  xmlns:ui=*"http://java.sun.com/jsf/facelets"*  xmlns:p=*"http://primefaces.org/ui"*>  <h:head>  <meta http-equiv=*"content-type"* content=*"text/html; charset=UTF-8"* />  <meta charset=*"utf-8"* />  <title><ui:insert name=*"title"*></ui:insert></title>  <meta name=*"generator"* content=*"Bootply"* />  <meta name=*"viewport"*  content=*"width=device-width, initial-scale=1, maximum-scale=1"* />  <link  href=*'https://fonts.googleapis.com/css?family=Roboto:400,300,500|Ubuntu+Condensed'*  rel=*'stylesheet'* type=*'text/css'* />  <h:outputStylesheet name=*"css/bootstrap.min.css"* />  <h:outputStylesheet name=*"css/main\_base.css"* />  <h:outputStylesheet name=*"css/font\_ico.css"* />  <ui:insert name=*"headCont"*></ui:insert>  </h:head>  <h:body>  <nav class=*"navbar navbar-default navbar-fixed-top base\_primary"*  role=*"banner"*>  <div class=*"container"*>  <div class=*"navbar-header"*>  <button class=*"navbar-toggle"* type=*"button"* data-toggle=*"collapse"*  data-target=*".navbar-collapse"*>  <span class=*"sr-only"*>Toggle navigation</span> <span  class=*"icon-bar"*></span> <span class=*"icon-bar"*></span> <span  class=*"icon-bar"*></span>  </button>  <a class=*"navbar-brand"*  style="display: *table*; padding-top: *5px*; padding-bottom: *5px*;"><img  src=*"resources/img/logo.png"* alt=*""* height=*"50"* /></a>  </div>  </div>  </nav>  <!--main-->  <div class=*"container base\_fondo\_fff"*  style="min-height: *600px*; border-radius: *4px*; margin-top: *35px*;">  <div class=*"row"*>  <!--left-->  <div class=*"col-md-3"* id=*"leftCol"*>  <ul class=*"nav nav-stacked"* id=*"sidebar"*>  <li id=*"sidebar-1"*><a href=*"panel.xhtml"*><i  class=*"icon-home"*></i> Inicio</a></li>  <li id=*"sidebar-2"*><a href=*"listDepartament.xhtml"*><i  class=*"icon-user"*></i> Departamento</a></li>  <li id=*"sidebar-2"*><a href=*"listEmployee.xhtml"*><i  class=*"icon-user"*></i> Empleado</a></li>  </ul>  </div>  <!--/left-->  <!--right-->  <div class=*"col-md-9"* style="padding-right: *30px*">  <h2 id=*"sec0"* class=*"title-style"*>  <ui:insert name=*"titleCont"*>Titulo</ui:insert>  </h2>  <!-- seccion para el contenido -->  <ui:insert name=*"content"*>  </ui:insert>  </div>  <!--/right-->  </div>  <!--/row-->  </div>  <!--/container-->  <ui:insert name=*"footerCont"*></ui:insert>  </h:body>  </html> |

* Crear panel.xhtml

|  |
| --- |
| <ui:composition template=*"master.xhtml"*  xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:f=*"http://java.sun.com/jsf/core"*  xmlns:h=*"http://java.sun.com/jsf/html"*  xmlns:p=*"http://primefaces.org/ui"*  xmlns:ui=*"http://java.sun.com/jsf/facelets"*>  <ui:define name=*"title"*>  .:Listado de Tabla :.  </ui:define>  <ui:define name=*"titleCont"*>  <span>Inicio </span>  </ui:define>  <ui:define name=*"content"*>  <h:form>  Bienvenido!!! a IO Visor Project  </h:form>  </ui:define>  <ui:define name=*"footerCont"*>    </ui:define>  </ui:composition> |

* Crear index.xhtml

|  |
| --- |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  <html xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:ui=*"http://java.sun.com/jsf/facelets"*  xmlns:f=*"http://java.sun.com/jsf/core"*  xmlns:h=*"http://java.sun.com/jsf/html"*>  <h:head></h:head>  <body>  <h:form>  <h:commandLink value=*"Ir"* action=*"panel"* />  </h:form>  </body>  </html> |

-Crear departament.xhtml

|  |
| --- |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  <html  xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:f=*"http://java.sun.com/jsf/core"*  xmlns:h=*"http://java.sun.com/jsf/html"*  xmlns:p=*"http://primefaces.org/ui"*  xmlns:ui=*"http://java.sun.com/jsf/facelets"*>    <ui:composition template=*"master.xhtml"*>  <ui:define name=*"title"*>  .:Registro de Departamento:.  </ui:define>  <ui:define name=*"titleCont"*>  <span>Registro de Departamento </span>  </ui:define>  <ui:define name=*"content"*>  <h:form>  <h:outputLabel for=*"idDept"* value=*"Id "*  styleClass=*"control-label"* />    <p:inputText autocomplete=*"off"* id=*"idDept"*  styleClass=*"form-control"*  value=*"#{departamentController.departament.idDepartament}"*  required=*"true"* maxlength=*"30"* />    <h:outputLabel for=*"nameDept"* value=*"Departamento: "*  styleClass=*"control-label"* />    <p:inputText autocomplete=*"off"* id=*"nameDept"*  styleClass=*"form-control"*  value=*"#{departamentController.departament.nameDepartament}"*  required=*"true"* maxlength=*"30"* />    <h:outputLabel for=*"desDept"* value=*"Descripción: "*  styleClass=*"control-label"* />  <p:inputText autocomplete=*"off"* id=*"desDept"*  styleClass=*"form-control"*  value=*"#{departamentController.departament.descriptionDepartament}"*  required=*"true"* maxlength=*"30"* />  <p:commandButton value=*"Grabar"*  actionListener=*"#{departamentController.insertarActualizar()}"*  styleClass=*"btn btn-primary "* update=*"@form"*  action=*"listDepartament?faces-redirect=true"* />  <p:commandButton value=*"Cancelar"*  action=*"listDepartament?faces-redirect=true"* immediate=*"true"*  styleClass=*"btn btn-default"* ajax=*"false"* />  <p:growl id=*"growl"* sticky=*"true"* showDetail=*"true"* life=*"3000"* />  </h:form>  </ui:define>  <ui:define name=*"footerCont"*>  </ui:define>  </ui:composition>  </html> |

-Crear listDepartament.xhtml

|  |
| --- |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  <html xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:ui=*"http://java.sun.com/jsf/facelets"*  xmlns:f=*"http://java.sun.com/jsf/core"*  xmlns:h=*"http://java.sun.com/jsf/html"*  xmlns:p=*"http://primefaces.org/ui"*>  <ui:composition template=*"master.xhtml"*>  <ui:define name=*"title"*>  .:Listado de Departamentos:.  </ui:define>  <ui:define name=*"titleCont"*>  <span>Listado de Departamentos </span>  </ui:define>  <ui:define name=*"content"*>  <h:form>  <div class=*"row"*>  <div class=*"col-sm-4"*>  <div class=*"form-group"*>  <h:outputLabel for=*"nombres"* value=*"Nombres:"*  styleClass=*"control-label"* />  <p:inputText autocomplete=*"off"* id=*"nombres"*  styleClass=*"form-control"*  value=*"#{departamentController.departament.nameDepartament}"*  maxlength=*"120"* />  </div>  </div>  <div class=*"col-sm-8 text-right"*>  <div class=*"form-group box-main-btn"*>  <span style="display: *block*; height: *30px*;"></span>  <p:commandButton value=*"Limpiar"* update=*"lstDepartament"*  actionListener=*"#{departamentController.limpiarDepartamento()}"*  styleClass=*"btn btn-default"* ajax=*"false"* />  <p:commandButton value=*"Buscar"* update=*"lstDepartament"*  actionListener=*"#{departamentController.buscarDepartamentoNombre()}"*  styleClass=*"btn btn-default btn-border"* ajax=*"true"* />  <p:commandButton value=*"Nuevo"*  action=*"#{departamentController.nuevoDepartamento()}"*  styleClass=*"btn btn-primary"* ajax=*"false"* />  </div>  </div>  </div>  <div class=*"row"*>  <div class=*"col-sm-12"*>  <div class=*"form-group table-responsive box-table-class"*>  <p:dataTable id=*"lstDepartament"* var=*"departament"*  value=*"#{departamentController.listaDepartamentos}"*  paginator=*"true"* paginatorPosition=*"top"*  rowsPerPageTemplate=*"10,20,30,40,50"* rows=*"10"*  styleClass=*"table table-bordered"*  emptyMessage=*"No existen registros"* rowIndexVar=*"item"*  style="margin-bottom:*0*">  <p:column headerText=*"Item"* style="width:*40px*;text-align:*center*">  <h:outputText value=*"#{item + 1}"* />  </p:column>  <p:column headerText=*"Id"*>  <h:outputText value=*"#{departament.idDepartament}"* />  </p:column>  <p:column headerText=*"Nombre"*>  <h:outputText value=*"#{departament.nameDepartament}"* />  </p:column>  <p:column headerText=*"Descripción"*>  <h:outputText value=*"#{departament.descriptionDepartament}"* />  </p:column>  <p:column headerText=*"Operación"*>  <h:commandLink action=*"listDepartament"* value=*"Eliminar"*  actionListener=*"#{departamentController.eliminar(departament)}"* />  </p:column>  <p:column headerText=*"Operación"*>  <h:commandLink action=*"departament"* value=*"Actualizar"*  actionListener=*"#{departamentController.listarId(departament)}"* />  </p:column>  </p:dataTable>  </div>  </div>  </div>  </h:form>  </ui:define>  <ui:define name=*"footerCont"*>  </ui:define>  </ui:composition>  </html> |

-Crear employee.xhtml

|  |
| --- |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  <html xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:f=*"http://java.sun.com/jsf/core"*  xmlns:h=*"http://java.sun.com/jsf/html"*  xmlns:p=*"http://primefaces.org/ui"*  xmlns:ui=*"http://java.sun.com/jsf/facelets"*>  <ui:composition template=*"master.xhtml"*>  <ui:define name=*"title"*>  .:Registro de Empleado:.  </ui:define>  <ui:define name=*"titleCont"*>  <span>Registro de Empleado </span>  </ui:define>  <ui:define name=*"content"*>  <h:form>  <div class=*"row"*>  <div class=*"col-sm-4"*>  <div class=*"form-group"*>  <h:outputLabel for=*"id"* value=*"Id"* styleClass=*"control-label"* />  <p:inputText autocomplete=*"off"* id=*"id"* styleClass=*"form-control"*  value=*"#{employeeController.employee.idEmployee}"* required=*"true"*  maxlength=*"120"* />  <h:outputLabel for=*"nombre"* value=*"Nombre:"*  styleClass=*"control-label"* />  <p:inputText autocomplete=*"off"* id=*"nombre"*  styleClass=*"form-control"*  value=*"#{employeeController.employee.nameEmployee}"*  required=*"true"* maxlength=*"120"* />  </div>  </div>  </div>  <div class=*"row"*>  <div class=*"col-sm-4"*>  <div class=*"form-group"*>  <h:outputLabel for=*"adress"* value=*"Dirección"*  styleClass=*"control-label"* />  <p:inputText autocomplete=*"off"* id=*"adress"*  styleClass=*"form-control"*  value=*"#{employeeController.employee.adressEmployee}"*  required=*"true"* maxlength=*"120"* />  </div>  </div>  </div>  <div class=*"row"*>  <div class=*"col-sm-4"*>  <div class=*"form-group"*>  <p:outputLabel for=*"cboDepartament"* value=*"Departamento"* />  <p:selectOneMenu id=*"cboDepartament"*  value=*"#{employeeController.employee.departament}"*  converter=*"omnifaces.SelectItemsConverter"*>  <f:selectItems  value=*"#{employeeController.listaDepartamentos}"*  var=*"d"*  itemLabel=*"#{d.nameDepartament}"*  itemValue=*"#{d}"* />  </p:selectOneMenu>  </div>  </div>  </div>  <div class=*"row"*>  <div class=*"col-sm-2 text-right"*>  <div class=*"form-group box-main-btn"*>  <p:commandButton value=*"Grabar"*  actionListener=*"#{employeeController.insertarActualizar()}"*  styleClass=*"btn btn-primary "* update=*""*  action=*"listEmployee.xhtml"* />  <p:commandButton value=*"Cancelar"* action=*"listEmployee.xhtml"*  immediate=*"true"* styleClass=*"btn btn-default"* ajax=*"false"* />  </div>  </div>  </div>  </h:form>  </ui:define>  <ui:define name=*"footerCont"*>  </ui:define>  </ui:composition>  </html> |

* Crear listEmployee.xhtml

|  |
| --- |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  <html xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:ui=*"http://java.sun.com/jsf/facelets"*  xmlns:f=*"http://java.sun.com/jsf/core"*  xmlns:h=*"http://java.sun.com/jsf/html"*  xmlns:p=*"http://primefaces.org/ui"*>  <ui:composition template=*"master.xhtml"*>  <ui:define name=*"title"*>  .:Listado de Empleados:.  </ui:define>  <ui:define name=*"titleCont"*>  <span>Listado de Empleados </span>  </ui:define>  <ui:define name=*"content"*>  <h:form>  <div class=*"row"*>  <div class=*"col-sm-4"*>  <div class=*"form-group"*>  <h:outputLabel for=*"nombres"* value=*"Nombres:"*  styleClass=*"control-label"* />  <p:inputText autocomplete=*"off"* id=*"nombres"*  styleClass=*"form-control"*  value=*"#{employeeController.employee.nameEmployee}"*  maxlength=*"120"* />  </div>  </div>  <div class=*"col-sm-8 text-right"*>  <div class=*"form-group box-main-btn"*>  <span style="display: *block*; height: *30px*;"></span>  <p:commandButton value=*"Limpiar"* update=*"lstEmployees"*  actionListener=*"#{employeeController.limpiarEmpleado()}"*  styleClass=*"btn btn-default"* ajax=*"false"* />  <p:commandButton value=*"Buscar"* update=*"lstEmployees"*  actionListener=*"#{employeeController.buscarEmployeeNombre()}"*  styleClass=*"btn btn-default btn-border"* ajax=*"true"* />  <p:commandButton value=*"Nuevo"*  action=*"#{employeeController.nuevoEmpleado()}"*  styleClass=*"btn btn-primary"* ajax=*"false"* />  </div>  </div>  </div>  <div class=*"row"*>  <div class=*"col-sm-12"*>  <div class=*"form-group table-responsive box-table-class"*>  <p:dataTable id=*"lstEmployees"* var=*"employee"*  value=*"#{employeeController.listaEmpleados}"* paginator=*"true"*  paginatorPosition=*"top"* rowsPerPageTemplate=*"10,20,30,40,50"*  rows=*"10"* styleClass=*"table table-bordered"*  emptyMessage=*"No existen registros"* rowIndexVar=*"item"*  style="margin-bottom:*0*">  <p:column headerText=*"Item"* style="width:*40px*;text-align:*center*">  <h:outputText value=*"#{item + 1}"* />  </p:column>  <p:column headerText=*"Id"*>  <h:outputText value=*"#{employee.idEmployee}"* />  </p:column>  <p:column headerText=*"Nombre"*>  <h:outputText value=*"#{employee.nameEmployee}"* />  </p:column>  <p:column headerText=*"Dirección"*>  <h:outputText value=*"#{employee.adressEmployee}"* />  </p:column>  <p:column headerText=*"Departamento"*>  <h:outputText value=*"#{employee.departament.nameDepartament}"* />  </p:column>  <p:column headerText=*"Operación"*>  <h:commandLink action=*"listEmployee"* value=*"Eliminar"*  actionListener=*"#{employeeController.eliminar(employee)}"* />  </p:column>  <p:column headerText=*"Operación"*>  <h:commandLink action=*"employee"* value=*"Actualizar"*  actionListener=*"#{employeeController.mostrar(employee)}"* />  </p:column>  </p:dataTable>  </div>  </div>  </div>  </h:form>  </ui:define>  <ui:define name=*"footerCont"*>  </ui:define>  </ui:composition>  </html> |