

**CURSO DE JAVASERVER FACES**

# **EJERCICIO**

## **MANEJO DE COMPONENTES EN JSF**



Experiencia y Conocimiento para tu vida

**CURSO DE JAVASERVER FACES**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# OBJETIVO DEL EJERCICIO

El objetivo de este ejercicio es poner en práctica algunos de los Componentes de los disponibles en JSF. Convertiremos el campo de Colonia de un `h:inputText` a `h:selectOneMenu`. Los campos a mostrar estarán en código duro por el momento, posteriormente los obtendremos de una base de datos. El resultado será similar al siguiente:

Ubaldo

BolsaTrabajoV7

localhost:8080/BolsaTrabajoV7/faces/index.xhtml

Nombre: Introduce tu nombre

Apellido: Introduce tu Apellido

Sueldo deseado: 0

Fecha de Nacimiento:

Codigo Postal: 03810

Colonia: Nápoles

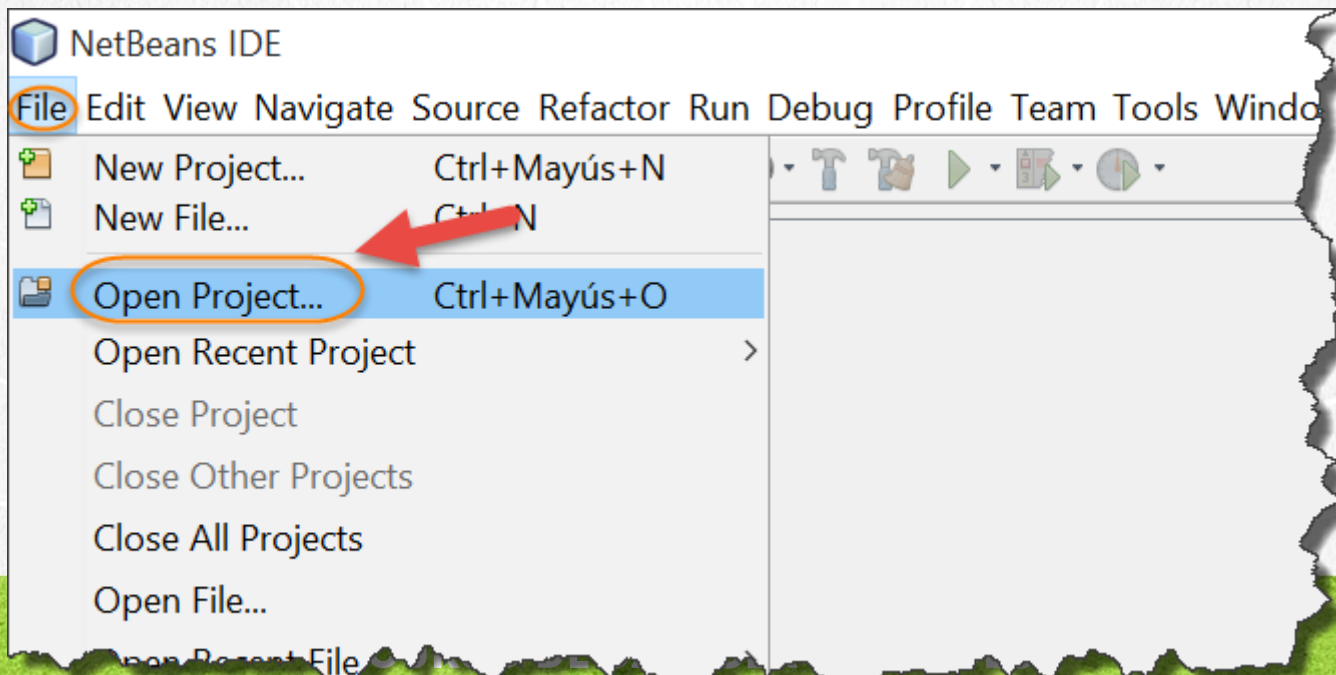
Ciudad: Ciudad de Mexico

[Mostrar](#)

Se selecciona la opción 1 del elemento SelectItem de Manera Automática

# PASO 1. ABRIR EL PROYECTO

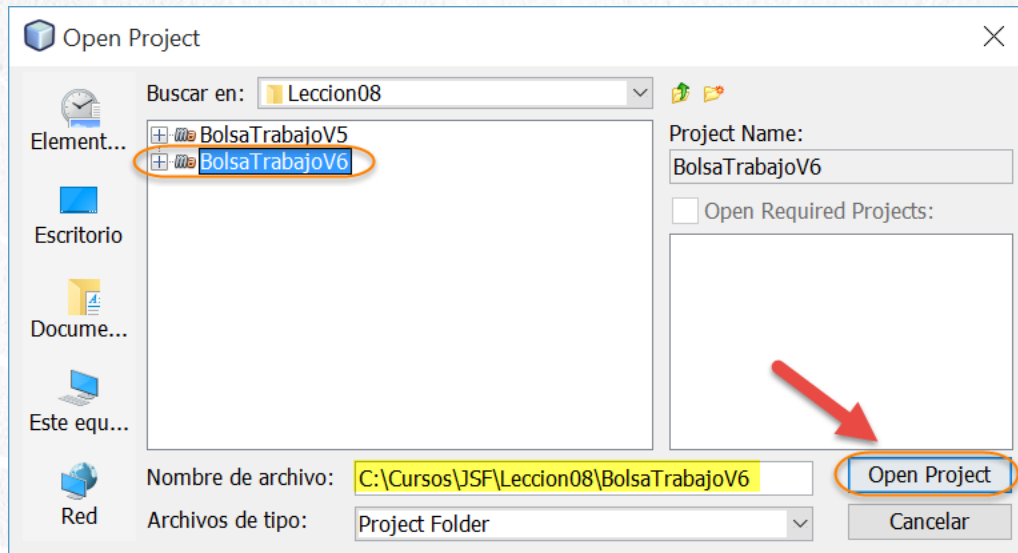
Abrimos el proyecto BolsaTrabajoV6, sólo en caso de que no lo tengamos ya abierto:





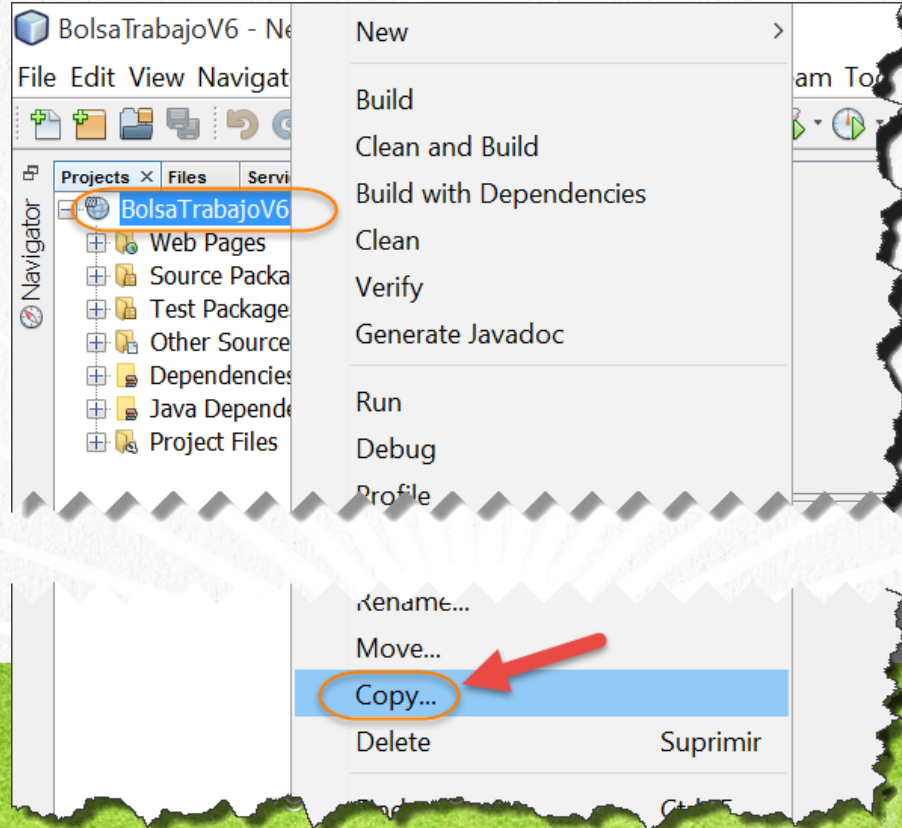
# PASO 1. ABRIR EL PROYECTO

Abrimos el proyecto BolsaTrabajoV6, sólo en caso de que no lo tengamos ya abierto:



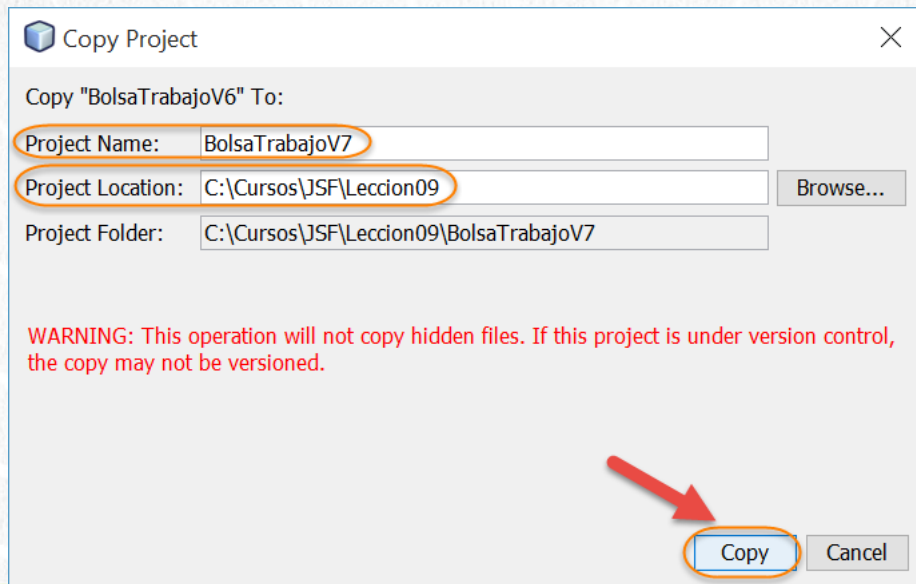
# PASO 2. COPIAR EL PROYECTO

Copiamos el proyecto BolsaTrabajoV6:



## PASO 2. COPIAR EL PROYECTO

Copiamos el proyecto BolsaTrabajoV6, lo renombramos a BolsaTrabajoV7 y lo depositamos en la nueva ruta mostrada:



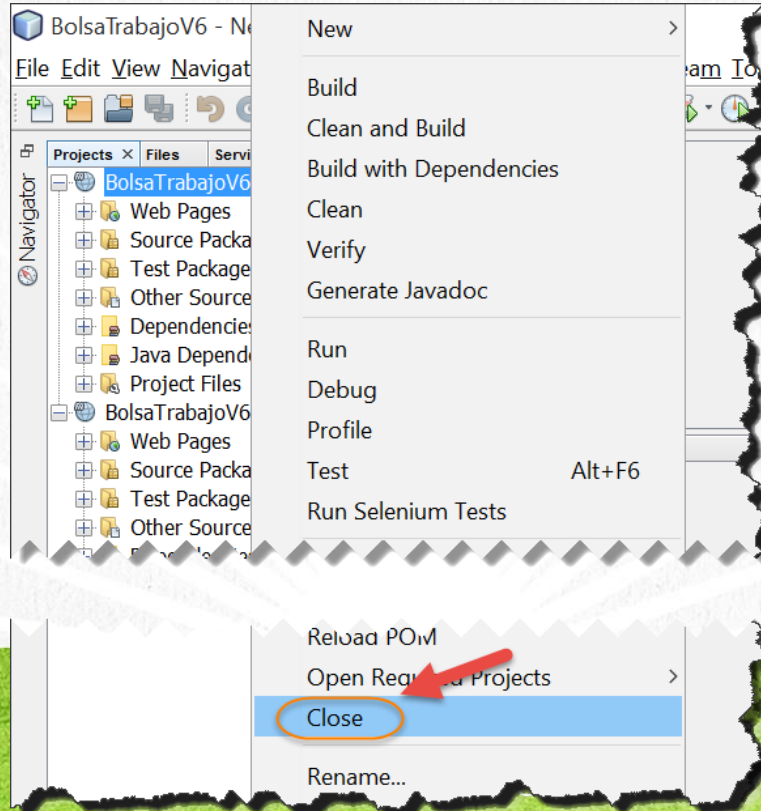
**CURSO DE JAVASERVER FACES**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)



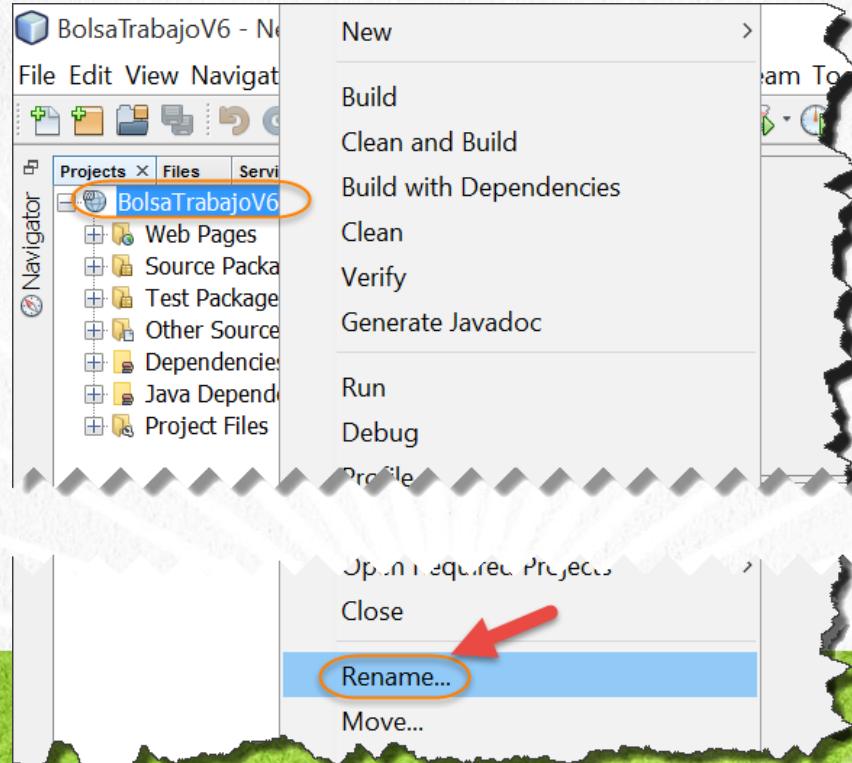
# PASO 3. CERRAR PROYECTO

Cerramos el proyecto anterior, y dejamos solo el nuevo abierto:



# PASO 4. RENOMBRAR PROYECTO

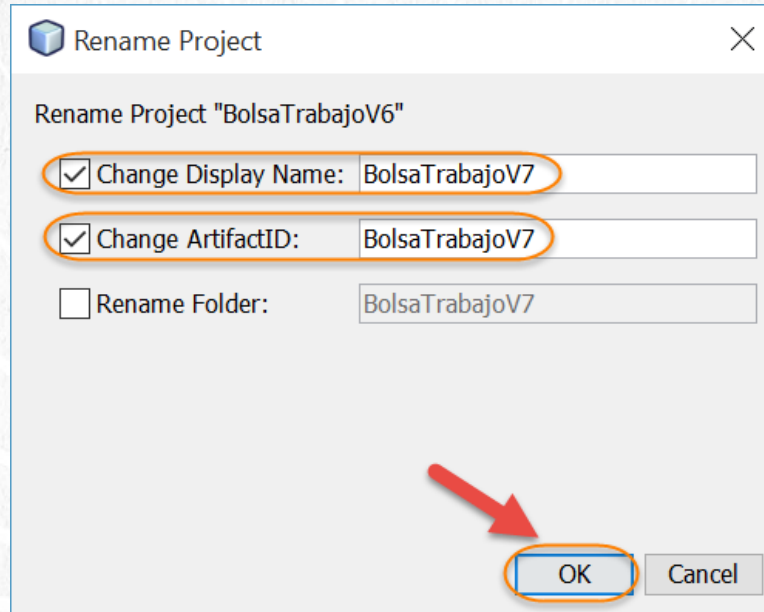
Cambiamos el nombre del proyecto a BolsaTrabajoV6:





# PASO 4. RENOMBRAR PROYECTO

Cambiamos el nombre del proyecto a BolsaTrabajoV7:



The screenshot shows a 'Rename Project' dialog box with the title 'Rename Project' and a close button (X). The main text reads 'Rename Project "BolsaTrabajoV6"'. There are three input fields, each with a checkbox and a text box:

- ☒ Change Display Name: BolsaTrabajoV7
- ☒ Change ArtifactID: BolsaTrabajoV7
- ☐ Rename Folder: BolsaTrabajoV7

At the bottom right, there are two buttons: 'OK' and 'Cancel'. A red arrow points to the 'OK' button, which is also circled in orange.

**CURSO DE JAVASERVER FACES**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

## PASO 5. MODIFICAR UNA CLASE JAVA

Abrir la clase Candidato.java y realizar las siguientes acciones:

```
Eliminar la propiedad colonia  
Eliminar el método getColonia ()  
Eliminar el método setColonia ()
```

Agregar la propiedad siguiente:

```
private long coloniaId;
```

Generar los métodos getter y setter para esta nueva propiedad

# PASO 5. MODIFICAMOS EL CÓDIGO

## Archivo Candidato.java:

Dar click para ir al código

```
package beans.model;

import java.util.Date;
import javax.inject.Named;
import javax.enterprise.context.RequestScoped;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;

@RequestScoped
@Named
public class Candidato {

    Logger log = LogManager.getRootLogger();
    private String nombre = "Introduce tu nombre";
    private String apellido = "Introduce tu Apellido";
    private int sueldoDeseado;
    private Date fechaNacimiento;
    private String codigoPostal;
    private String ciudad;
    private String comentario = "Escribe tu comentario";
    private long coloniaId;

    public Candidato() {
        log.info("Creando el objeto Candidato");
    }

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



# PASO 5. MODIFICAMOS EL CÓDIGO

## Archivo Candidato.java:

Dar click para ir al código

```
public void setNombre(String nombre) {
    this.nombre = nombre;
    log.info("Modificando la propiedad nombre:" + this.nombre);
}

public String getApellido() {
    return apellido;
}

public void setApellido(String apellido) {
    this.apellido = apellido;
    log.info("Modificando la propiedad apellido:" + this.apellido);
}

public int getSueldoDeseado() {
    return sueldoDeseado;
}

public void setSueldoDeseado(int sueldoDeseado) {
    this.sueldoDeseado = sueldoDeseado;
    log.info("Modificando la propiedad sueldoDeseado:" + this.sueldoDeseado);
}

public Date getFechaNacimiento() {
    return fechaNacimiento;
}

public void setFechaNacimiento(Date fechaNacimiento) {
    this.fechaNacimiento = fechaNacimiento;
    log.info("Modificando la propiedad fechaNacimiento:" + this.fechaNacimiento);
}
```

# PASO 5. MODIFICAMOS EL CÓDIGO

## Archivo Candidato.java:

Dar click para ir al código

```
public String getCodigoPostal() {  
    return codigoPostal;  
}  
  
public void setCodigoPostal(String codigoPostal) {  
    this.codigoPostal = codigoPostal;  
}  
  
public String getCiudad() {  
    return ciudad;  
}  
  
public void setCiudad(String ciudad) {  
    this.ciudad = ciudad;  
}  
  
public String getComentario() {  
    return comentario;  
}  
  
public void setComentario(String comentario) {  
    this.comentario = comentario;  
}  
  
public long getColoniaId() {  
    return coloniaId;  
}  
  
public void setColoniaId(long coloniaId) {  
    this.coloniaId = coloniaId;  
}  
}
```

# PASO 6. MODIFICAR UN ARCHIVO PROPERTIES

Abrir el archivo mensajes.properties y realizar los siguientes cambios:

Eliminar el siguiente campo:

```
vacanteForm.colonia= Colonia
```

Agregar los nuevos campos

```
vacanteForm.coloniaId= Colonia  
vacanteForm.seleccione= -- Seleccione --
```

Realizar los mismos cambios en el archivo mensajes\_en.properties en el idioma correspondiente.



# PASO 6. MODIFICAMOS EL CÓDIGO

Archivo mensajes.properties:

Dar click para ir al código

```
vacanteForm.nombre=Nombre  
vacanteForm.apellido=Apellido  
vacanteForm.sueldoDeseado=Sueldo deseado  
vacanteForm.nacimiento=Fecha de Nacimiento  
vacanteForm.codigoPostal=Código Postal  
vacanteForm.ciudad=Ciudad  
vacanteForm.enviar=Enviar  
vacanteForm.comentario=Comentario  
vacanteForm.mostrar=Mostrar  
vacanteForm.ocultar=Ocultar  
vacanteForm.coloniaId= Colonia  
vacanteForm.seleccione=--Seleccione--
```

# PASO 7. MODIFICAMOS EL CÓDIGO

Archivo mensajes\_en.properties:

Dar click para ir al código

```
vacanteForm.nombre=Name  
vacanteForm.apellido=Last Name  
vacanteForm.sueldoDeseado=Salary desired  
vacanteForm.nacimiento=Date of birth  
vacanteForm.codigoPostal=Zip Code  
vacanteForm.colonia=Suburb  
vacanteForm.ciudad=City  
vacanteForm.enviar=Submit  
vacanteForm.comentario=Annotation  
vacanteForm.mostrar=Show  
vacanteForm.ocultar=Hide  
vacanteForm.coloniaId=Suburb  
vacanteForm.seleccione=--Select--
```

# PASO 8. MODIFICAR UN ARCHIVO XHTML

Modificar el archivo index.xhtml . Vamos a realizar varios cambios:

1) Eliminar la siguiente etiqueta:

```
<td><h:inputText id="colonia" required="true" value="#{candidato.colonia}" /></td>
```

2) Inserte el siguiente código en lugar del que hemos removido:

```
<td>  
    <h:selectOneMenu id="coloniaId" required="true" value="#{candidato.coloniaId}">  
        <f:selectItem itemLabel="#{msgs['vacanteForm.seleccione']}" itemValue="" />  
        <f:selectItem itemLabel="Nápoles" itemValue="1" />  
        <f:selectItem itemLabel="Polanco" itemValue="2" />  
        <f:selectItem itemLabel="Del Valle Centro" itemValue="3" />  
    </h:selectOneMenu>  
</td>
```



## PASO 8. MODIFICAR UN ARCHIVO XHTML

Buscar h:outputLabel para el campo de coloniaId y realice los siguientes cambios:

Cambiar el h:ouputLabel de:

```
<h:outputLabel for="colonia" value="#{msgs['vacanteForm.colonia']}"
```

Por el siguiente:

```
<h:outputLabel for="coloniaId" value="#{msgs['vacanteForm.coloniaId']}" />
```

# PASO 8. MODIFICAR UN ARCHIVO XHTML

Cambiar el h:message de:

```
<h:message for="colonia" />
```

Por el siguiente:

```
<h:message for="coloniaId" />
```

El archivo index.xhtml debe quedar como sigue:

# PASO 8. MODIFICAMOS EL CÓDIGO

## Archivo index.xhtml:

Dar click para ir al código

```
<?xml version='1.0' encoding='UTF-8' ?>
<!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:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core">
  <h:head>
    <title>BolsaTrabajoV7</title>
  </h:head>
  <h:body>
    <h:form id="vacanteForm">
      <h:messages globalOnly="true" ></h:messages>
      <table>
        <tr>
          <td><h:outputLabel for="nombre" value="#{msgs['vacanteForm.nombre']}" /></td>
          <td><h:inputText id="nombre" required="true" value="#{candidato.nombre}" /> </td>
          <td><h:message for="nombre" /></td>
        </tr>
        <tr>
          <td><h:outputLabel for="apellido" value="#{msgs['vacanteForm.apellido']}" /></td>
          <td><h:inputText id="apellido" required="true" value="#{candidato.apellido}" /></td>
          <td><h:message for="apellido" /></td>
        </tr>
        <tr>
          <td><h:outputLabel for="sueldoDeseado" value="#{msgs['vacanteForm.sueldoDeseado']}" /></td>
          <td><h:inputText id="sueldoDeseado" required="true"
                    value="#{candidato.sueldoDeseado}"
                    <f:validateLongRange minimum="10000" maximum="50000" />
          </h:inputText></td>
          <td><h:message for="sueldoDeseado" /></td>
        </tr>
      </table>
    </h:form>
  </h:body>
</html>
```



# PASO 8. MODIFICAMOS EL CÓDIGO

Archivo index.xhtml:

Dar click para ir al código

```
<tr>
  <td><h:outputLabel for="fechaNacimiento" value="#{msgs['vacanteForm.nacimiento']}" /></td>
  <td><h:inputText id="fechaNacimiento" required="true"
    value="#{candidato.fechaNacimiento}"
    <f:convertDateTime pattern="MM/dd/yyyy" />
  </h:inputText></td>
  <td><h:message for="fechaNacimiento" /></td>
</tr>
<tr>
  <td><h:outputLabel for="codigoPostal" value="#{msgs['vacanteForm.codigoPostal']}" /></td>
  <td><h:inputText id="codigoPostal" immediate="true"
    onchange="this.form.submit()" required="true"
    value="#{candidato.codigoPostal}"
    valueChangeListener="#{vacanteForm.codigoPostalListener}" /></td>
  <td><h:message for="codigoPostal" /></td>
</tr>
<tr>
  <td><h:outputLabel for="coloniaId" value="#{msgs['vacanteForm.coloniaId']}" /></td>
  <td>
    <h:selectOneMenu id="coloniaId" required="true" value="#{candidato.coloniaId}">
      <f:selectItem itemLabel="#{msgs['vacanteForm.seleccione']}" itemValue="" />
      <f:selectItem itemLabel="Nápoles" itemValue="1" />
      <f:selectItem itemLabel="Polanco" itemValue="2" />
      <f:selectItem itemLabel="Del Valle Centro" itemValue="3" />
    </h:selectOneMenu>
  </td>
  <td><h:message for="coloniaId" /></td>
</tr>
```

# PASO 8. MODIFICAMOS EL CÓDIGO

Archivo index.xhtml:

Dar click para ir al código

```
<tr>
  <td><h:outputLabel for="ciudad" value="#{msgs['vacanteForm.ciudad']}" /></td>
  <td><h:inputText id="ciudad" required="true"
    value="#{candidato.ciudad}" /></td>
  <td><h:message for="ciudad" /></td>
</tr>
</table>
<table>
  <tr>
    <td>
      <h:commandLink actionListener="#{vacanteForm.ocultarComentario}"
        immediate="true" rendered="#{!vacanteForm.comentarioEnviado}"
        value="#{msgs['vacanteForm.mostrar']}" />
      <h:commandLink actionListener="#{vacanteForm.ocultarComentario}"
        immediate="true" rendered="#{vacanteForm.comentarioEnviado}"
        value="#{msgs['vacanteForm.ocultar']}" />
    </td>
  </tr>
</table>
<h:panelGroup rendered="#{vacanteForm.comentarioEnviado}">
  <table>
    <tr>
      <td><h:outputLabel for="comentario" value="#{msgs['vacanteForm.comentario']}" /></td>
    </tr>
    <tr>
      <td><h:inputTextarea id="comentario" cols="50" rows="10" value="#{candidato.comentario}" /></td>
    </tr>
  </table>
</h:panelGroup>
```

# PASO 8. MODIFICAMOS EL CÓDIGO

Archivo index.xhtml:

Dar click para ir al código

```
<h:commandButton action="#{vacanteForm.enviar}"  
                 value="#{msgs['vacanteForm.enviar']}" />  
  
</h:form>  
</h:body>  
</html>
```

**CURSO DE JAVASERVER FACES**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)



# PASO 9. MODIFICAR UN ARCHIVO JAVA

Modificamos el archivo VacanteForm.java.

Sustituir el código del método codigoPostalListener ():

```
UIInput coloniaInputText = (UIInput) uiViewRoot.findComponent("vacanteForm:colonia");  
String colonia = "Nápoles";  
coloniaInputText.setValue(colonia);  
coloniaInputText.setSubmittedValue(colonia);
```

Por el siguiente:

```
UIInput coloniaInputText = (UIInput)  
uiViewRoot.findComponent("vacanteForm:coloniaId");  
Long coloniaId = 1L;  
coloniaInputText.setValue(coloniaId);  
coloniaInputText.setSubmittedValue(Long.toString(coloniaId));
```

# PASO 9. MODIFICAMOS EL CÓDIGO

Archivo VacanteForm.java:

Dar click para ir al código

```
package beans.backing;

import beans.helper.FacesContextHelper;
import beans.model.Candidato;
import javax.inject.Named;
import javax.enterprise.context.RequestScoped;
import javax.faces.application.FacesMessage;
import javax.faces.component.UIInput;
import javax.faces.component.UIViewRoot;
import javax.faces.context.FacesContext;
import javax.faces.event.ActionEvent;
import javax.faces.event.ValueChangeEvent;
import javax.inject.Inject;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
```

```
@RequestScoped
@Named
public class VacanteForm {

    Logger log = LogManager.getRootLogger();

    private boolean comentarioEnviado = false;

    @Inject
    private Candidato candidato;
```

# PASO 9. MODIFICAMOS EL CÓDIGO

## Archivo VacanteForm.java:

Dar click para ir al código

```
public VacanteForm() {
    log.info("Creando objeto VacanteForm");
}

public void setCandidato(Candidato candidato) {
    this.candidato = candidato;
}

//Metodo de flujo de control
public String enviar() {

    System.out.println("enviar() Nombre=" + this.candidato.getNombre());
    System.out.println("enviar() Apellido=" + this.candidato.getApellido());
    System.out.println("enviar() Sueldo deseado" + this.candidato.getSueldoDeseado());

    if (this.candidato.getNombre().equals("Juan")) {

        if (this.candidato.getApellido().equals("Perez")) {
            String msg = "Gracias, pero Juan Perez ya trabaja con nosotros.";
            FacesMessage facesMessage = new FacesMessage(FacesMessage.SEVERITY_ERROR, msg, msg);
            FacesContext facesContext = FacesContext.getCurrentInstance();
            String clientId = null; //Este es un mensaje global
            facesContext.addMessage(clientId, facesMessage);
            return "index";
        }
        return "exito";//exito.xhtml
    } else {
        return "fallo"; //fallo.xhtml
    }
}
```



# PASO 9. MODIFICAMOS EL CÓDIGO

## Archivo VacanteForm.java:

Dar click para ir al código

```
//Metodo de tipo Value Change Listener
public void codigoPostalListener(ValueChangeEvent valueChangeEvent) {
    FacesContext facesContext = FacesContext.getCurrentInstance();
    UIViewRoot uiViewRoot = facesContext.getViewRoot();
    String newCodigoPostal = (String) valueChangeEvent.getNewValue();
    if ("03810".equals(newCodigoPostal)) {
        log.info("Modificamos los valores de colonia y ciudad dinamicamente con ValueChangeListener");
        //Utilizamos el nombre del form de index.xhtml para encontrar el componente
        UIInput coloniaInputText = (UIInput) uiViewRoot.findComponent("vacanteForm:coloniaId");
        Long coloniaId = 1L;
        coloniaInputText.setValue(coloniaId);
        coloniaInputText.setSubmittedValue(Long.toString(coloniaId));

        UIInput ciudadInputText = (UIInput) uiViewRoot.findComponent("vacanteForm:ciudad");
        String ciudad = "Ciudad de Mexico";
        ciudadInputText.setValue(ciudad);
        ciudadInputText.setSubmittedValue(ciudad);

        facesContext.renderResponse();
    }
}

public void ocultarComentario(ActionEvent actionEvent) {
    this.comentarioEnviado = !this.comentarioEnviado;
    log.info("Mostrando/ocultando el comentario");
    FacesContext facesContext = FacesContext.getCurrentInstance();
    FacesContextHelper.limpiarImmediateFacesMessages(facesContext);
}
```

# PASO 9. MODIFICAMOS EL CÓDIGO

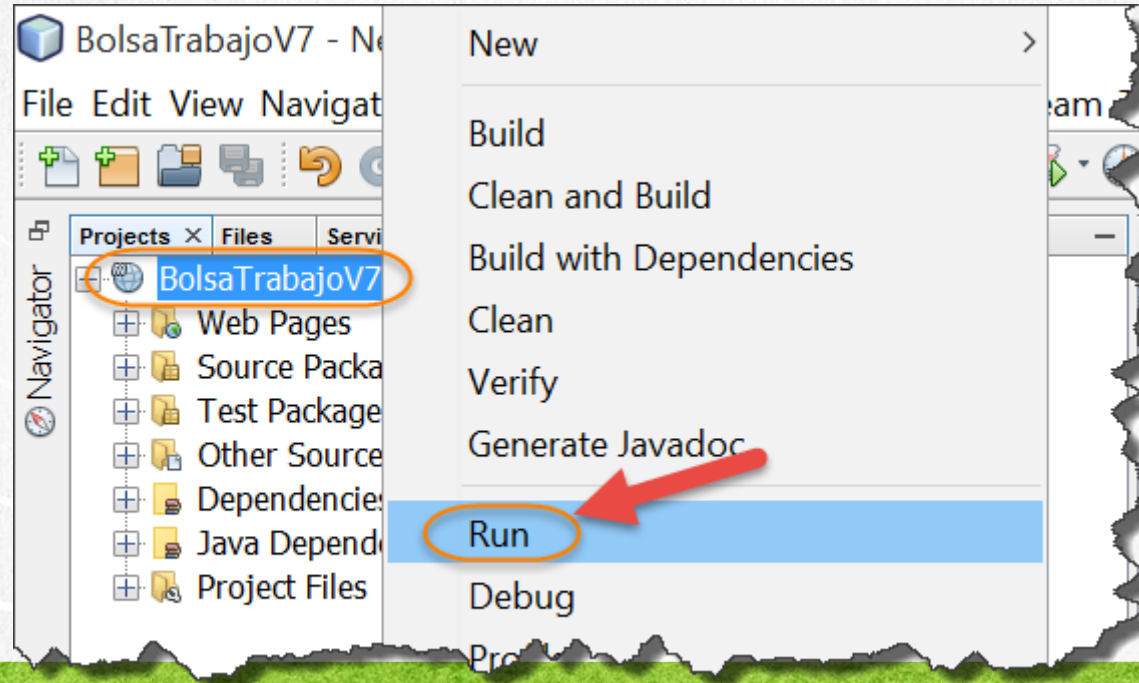
Archivo VacanteForm.java:

Dar click para ir al código

```
public boolean isComentarioEnviado() {  
    return comentarioEnviado;  
}  
  
public void setComentarioEnviado(boolean comentarioEnviado) {  
    this.comentarioEnviado = comentarioEnviado;  
}  
}
```

# PASO 10. EJECUCIÓN DEL PROYECTO

Ejecutamos el proyecto:



**CURSO DE JAVASERVER FACES**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)



# PASO 10. EJECUCIÓN DEL PROYECTO

Ejecutamos el proyecto:

Ubaldo

BolsaTrabajoV7

localhost:8080/BolsaTrabajoV7/faces/index.xhtml

Nombre: Introduce tu nombre

Apellido: Introduce tu Apellido

Sueldo deseado: 0

Fecha de Nacimiento:

Codigo Postal: 03810

Colonia: Nápoles

Ciudad: Ciudad de Mexico

[Mostrar](#)

Se selecciona la opción 1 del elemento SelectItem de Manera Automática

**CURSO DE JAVASERVER FACES**

www.globalmentoring.com.mx

# CONCLUSIÓN DEL EJERCICIO

Con este ejercicio pusimos en práctica el concepto de componentes en JSF.

Utilizamos el componente SelectItem, el cual será uno de los componentes que más utilizaremos en nuestros formularios JSF.

Una de las grandes ventajas de usar componentes JSF, es que muchos de los componentes son igual de simples de aplicar o utilizar una vez teniendo las bases del manejo de componentes JSF.



Experiencia y Conocimiento para tu vida

**CURSO DE JAVASERVER FACES**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

**CURSO ONLINE**

# **JAVASERVER FACES (JSF)**

---

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida



**CURSO DE JAVASERVER FACES**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)