

CURSO DE JAVASERVER FACES

EJERCICIO

COMPONENTE SELECT ITEM EN JSF



Experiencia y Conocimiento para tu vida

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

OBJETIVO DEL EJERCICIO

El objetivo de este ejercicio es poner en práctica el componente de Select Items de JSF. Además, crearemos la entidad Colonia para simular una lista que se obtiene de la base de datos. Este es el código previo a la consulta de la Base de Datos real. El resultado será similar a la siguiente figura:

Ubaldo

BolsaTrabajoV8

localhost:8080/BolsaTrab:

Nombre: Introduce tu nombre

Apellido: Introduce tu Apellido

Sueldo deseado: 0

Fecha de Nacimiento:

Código Postal: 11530

Colonia: Polanco

Ciudad: Ciudad de Mexico

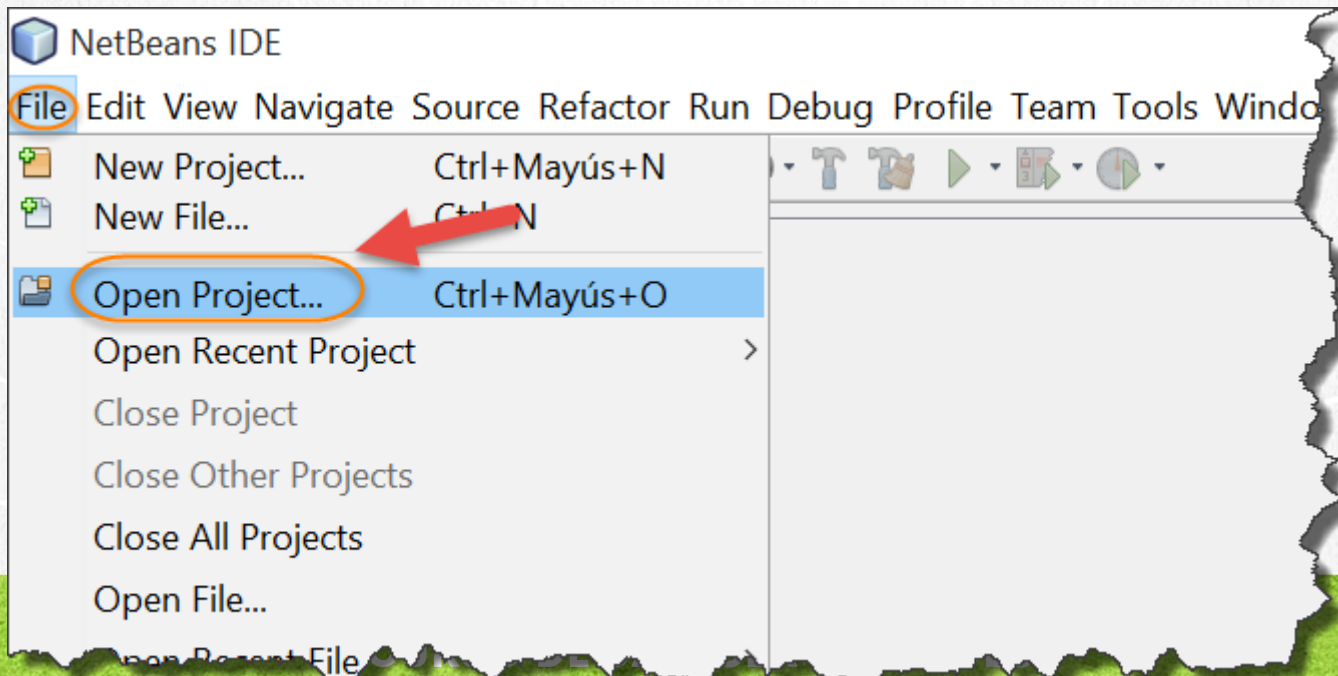
[Mostrar](#)

[Enviar](#)

Podemos probar con varios valores y el SelectItem cambiará de Manera Automática si se proporciona el valor correcto (Ej. 03810, 11530 ó 3100)

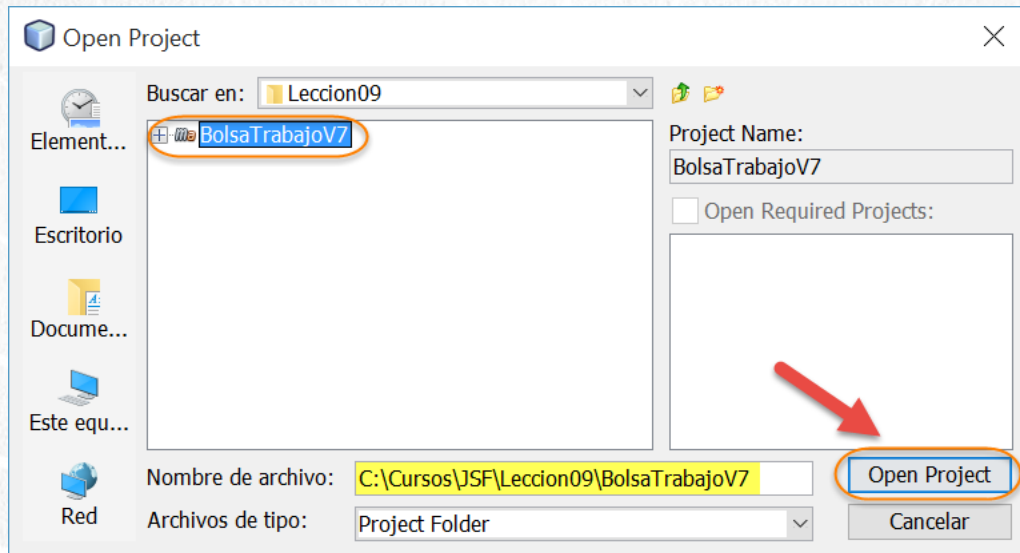
PASO 1. ABRIR EL PROYECTO

Abrimos el proyecto BolsaTrabajoV7, sólo en caso de que no lo tengamos ya abierto:



PASO 1. ABRIR EL PROYECTO

Abrimos el proyecto BolsaTrabajoV7, sólo en caso de que no lo tengamos ya abierto:

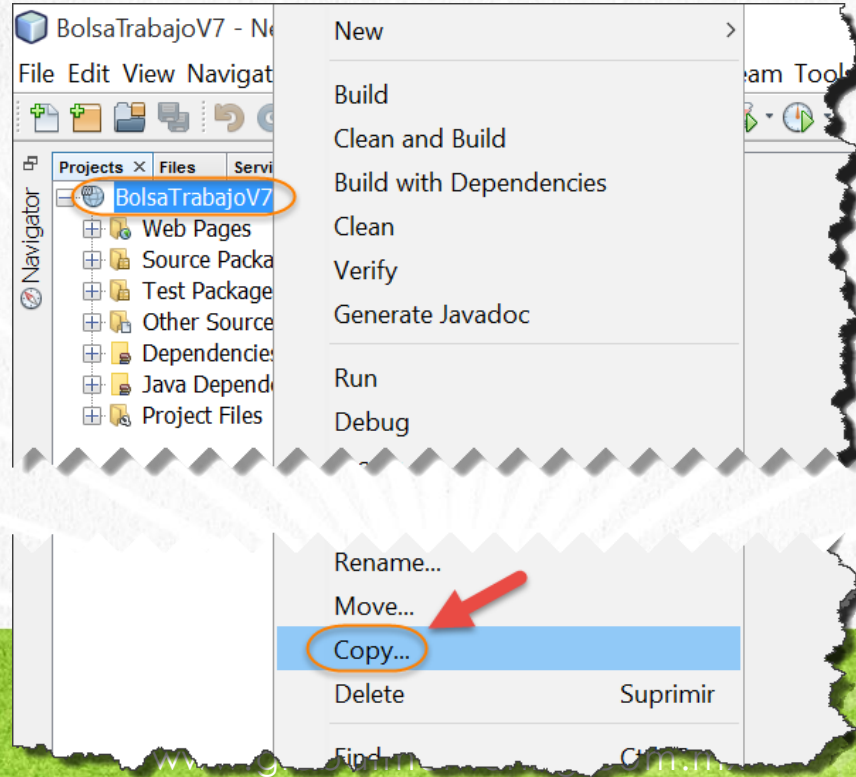


CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

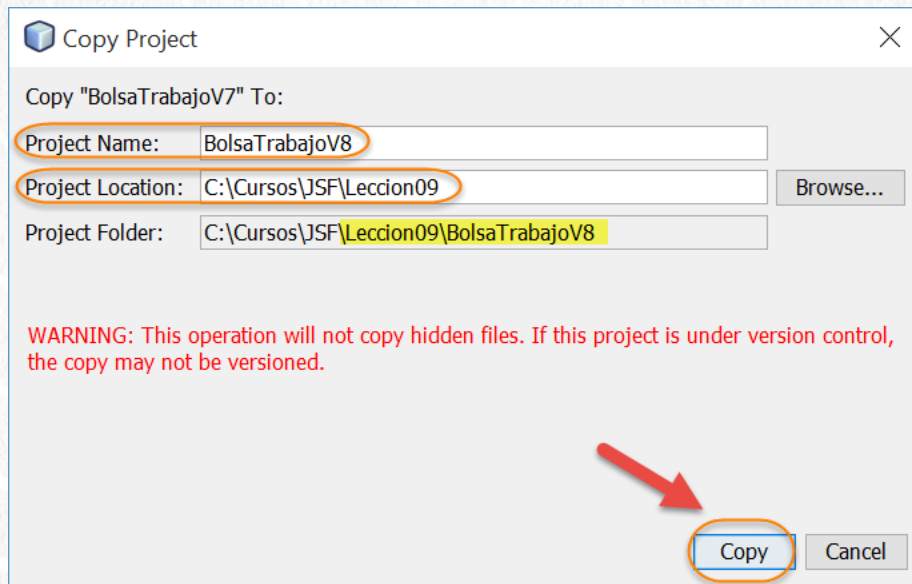
PASO 2. COPIAR EL PROYECTO

Copiamos el proyecto BolsaTrabajoV7:



PASO 2. COPIAR EL PROYECTO

Copiamos el proyecto BolsaTrabajoV7, lo renombramos a BolsaTrabajoV8 y lo depositamos en la nueva ruta mostrada:

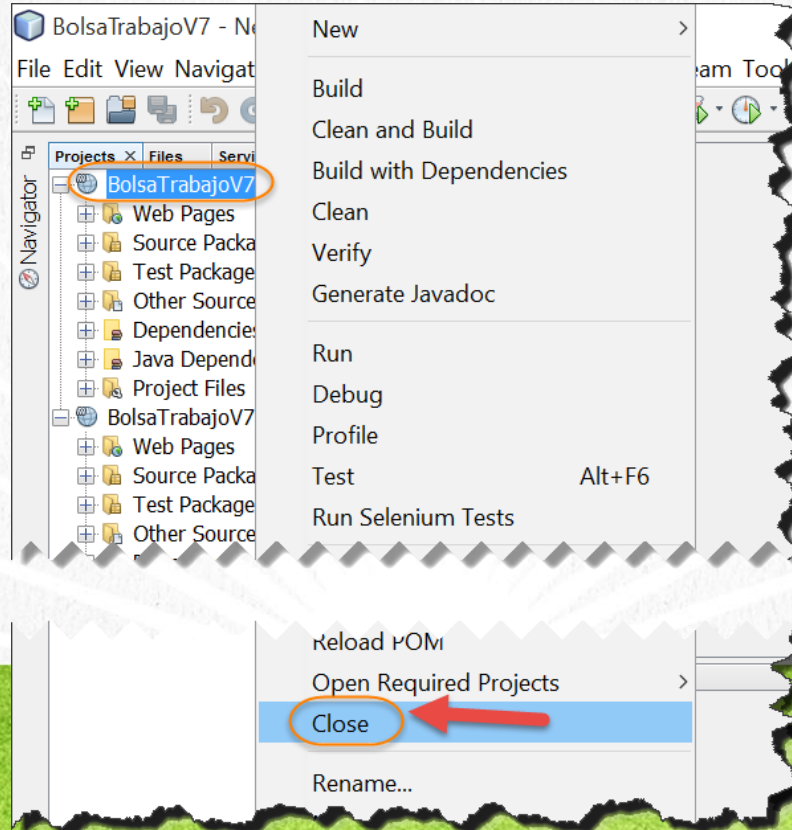


CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

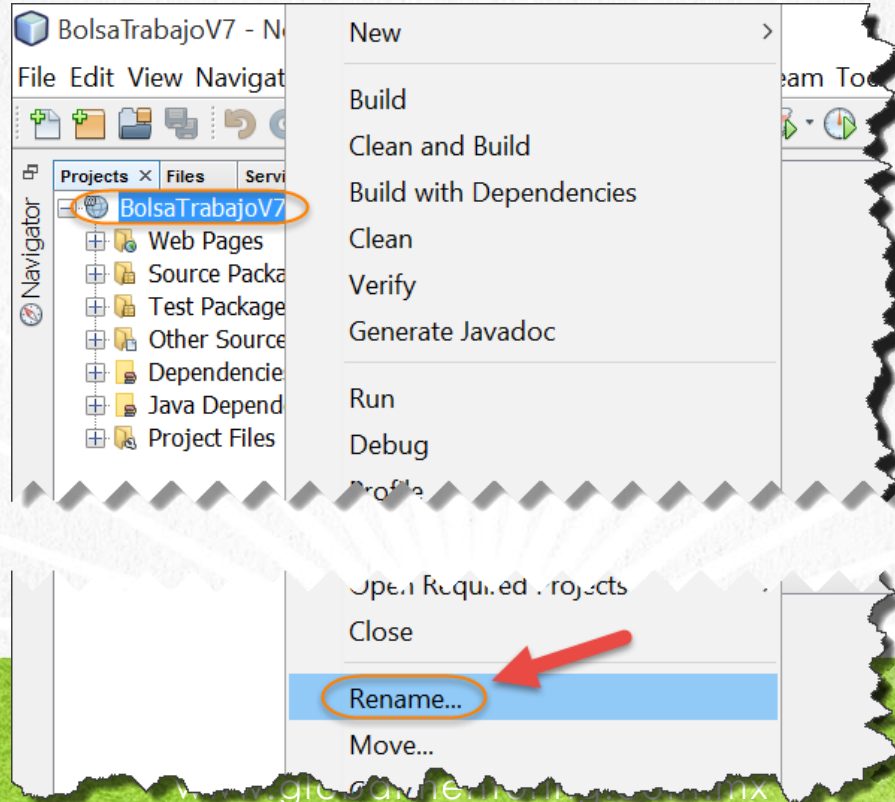
PASO 3. CERRAR PROYECTO

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



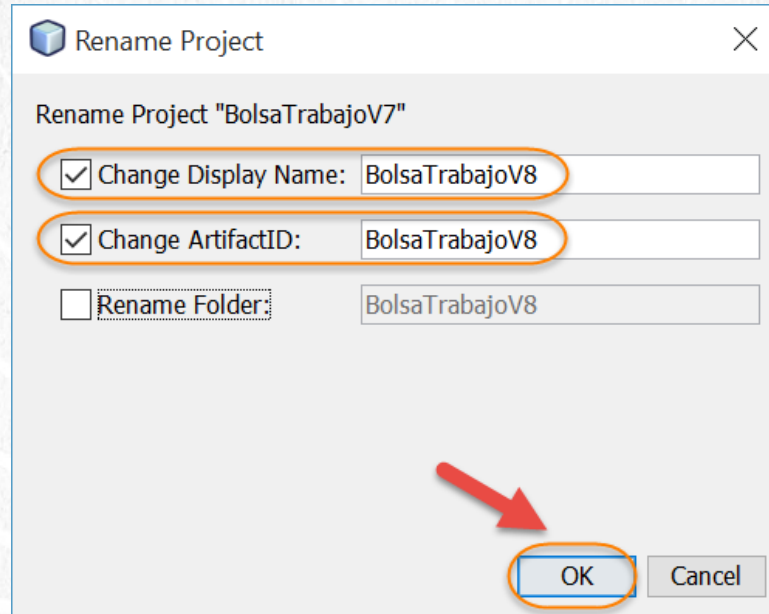
PASO 4. RENOMBRAR PROYECTO

Cambiamos el nombre del proyecto a BolsaTrabajoV7:



PASO 4. RENOMBRAR PROYECTO

Cambiamos el nombre del proyecto a BolsaTrabajoV8:

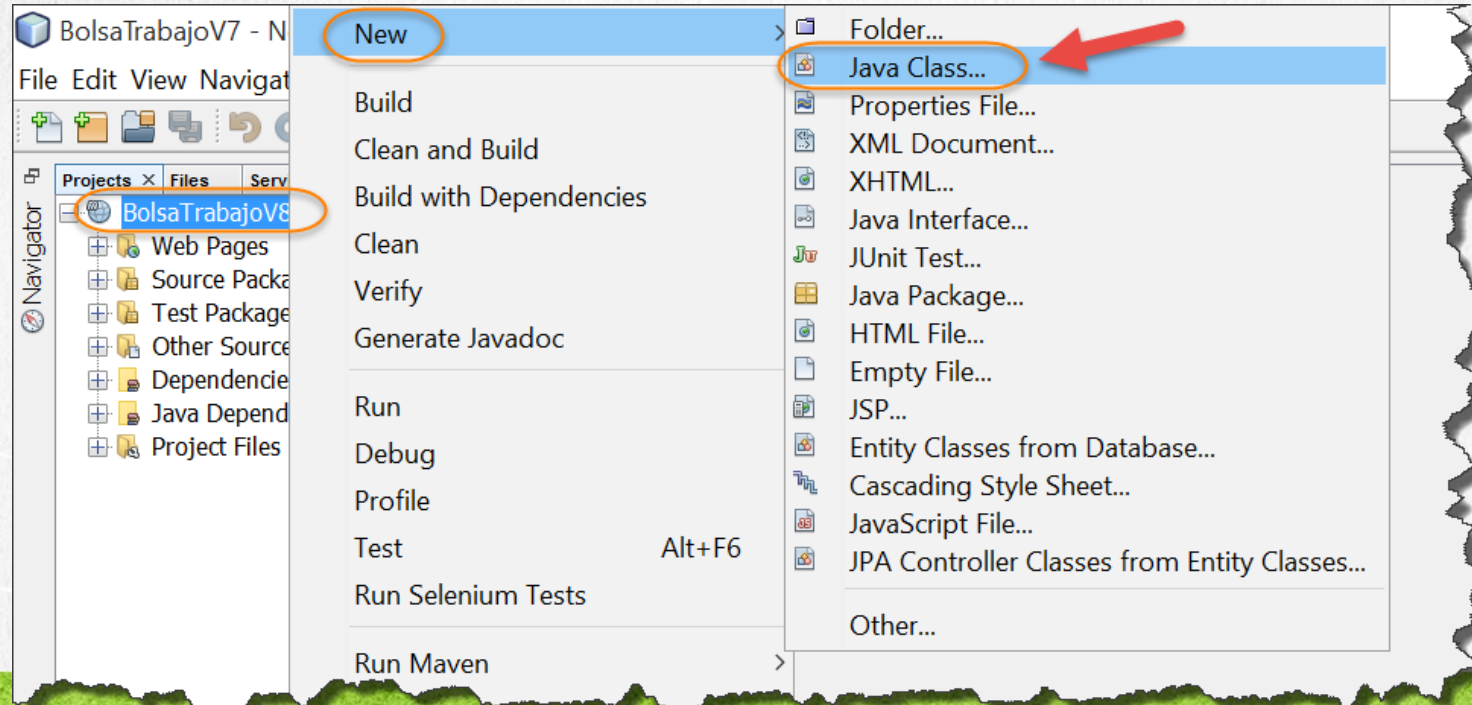


CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 5. CREAR UNA CLASE JAVA

Creamos la clase Colonia.java como sigue:



CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 5. CREAR UNA CLASE JAVA

Creamos la clase Colonia.java como sigue:

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back Next > **Finish** Cancel Help

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 6. MODIFICAMOS EL CÓDIGO

Archivo Colonia.java:

Dar click para ir al código

```
package beans.domain;

public class Colonia {

    private long coloniaId;
    private String nombreColonia;
    private long codigoPostal;

    public Colonia(long coloniaId, String nombreColonia, long codigoPostal) {
        this.coloniaId = coloniaId;
        this.nombreColonia = nombreColonia;
        this.codigoPostal = codigoPostal;
    }

    public long getColoniaId() {
        return coloniaId;
    }

    public void setColoniaId(long coloniaId) {
        this.coloniaId = coloniaId;
    }
}
```

PASO 6. MODIFICAMOS EL CÓDIGO

Archivo Colonia.java:

Dar click para ir al código

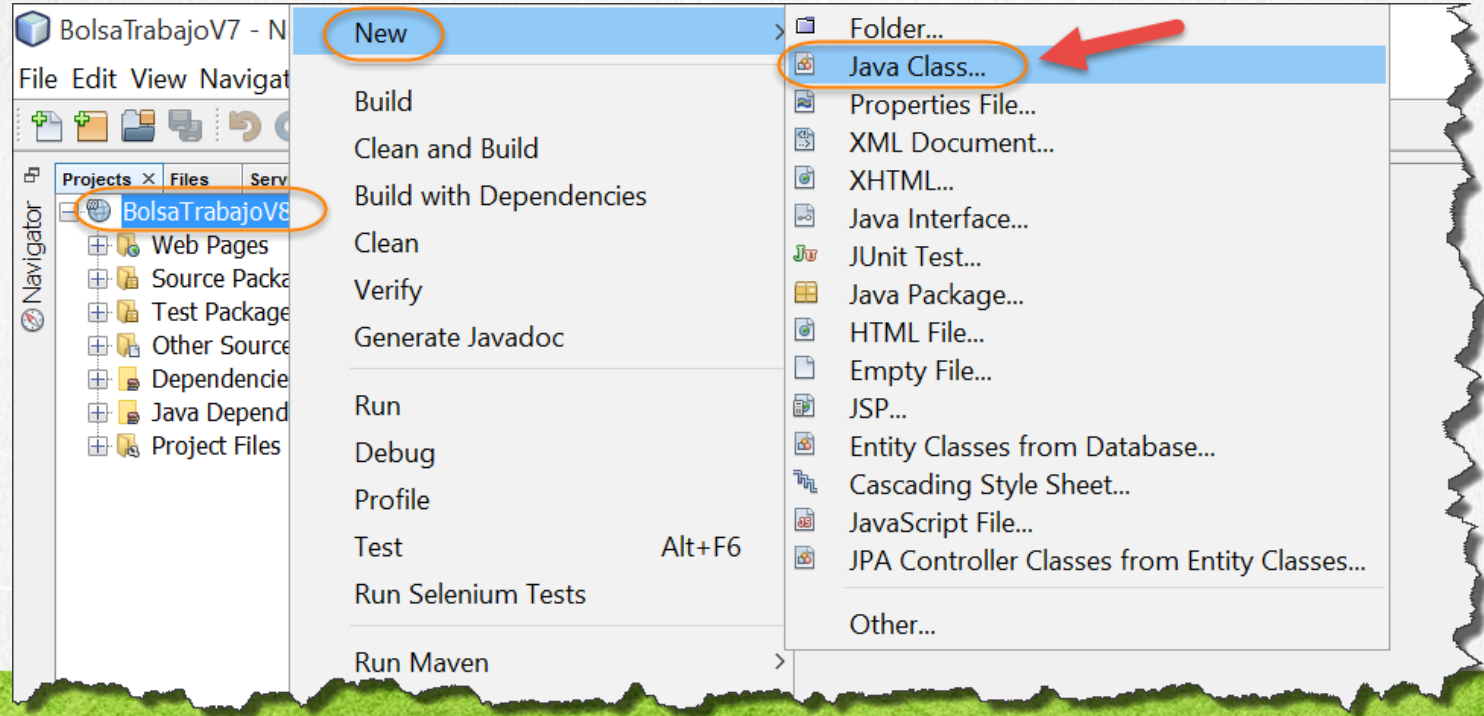
```
public String getNombreColonia() {  
    return nombreColonia;  
}  
  
public void setNombreColonia(String nombreColonia) {  
    this.nombreColonia = nombreColonia;  
}  
  
public long getCodigoPostal() {  
    return codigoPostal;  
}  
  
public void setCodigoPostal(long codigoPostal) {  
    this.codigoPostal = codigoPostal;  
}  
}
```

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 7. CREAR UNA CLASE JAVA

Creamos la clase ColoniaHelper como sigue:



CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 7. CREAR UNA CLASE JAVA

Creamos la clase ColoniaHelper como sigue:

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 8. MODIFICAMOS EL CÓDIGO

Archivo ColoniaHelper.java:

Dar click para ir al código

```
package beans.helper;

import beans.domain.Colonia;
import java.util.ArrayList;
import java.util.List;
import javax.inject.Named;
import javax.enterprise.context.RequestScoped;
import javax.faces.model.SelectItem;

@RequestScoped
@Named
public class ColoniaHelper {

    public List<Colonia> getColonias() {
        List<Colonia> colonias = new ArrayList<>();
        long coloniaId = 1;
        Colonia colonia = new Colonia(coloniaId++, "Nápoles", 3810); //no anteponer el cero, sino es un valor octal
        colonias.add(colonia);
        colonia = new Colonia(coloniaId++, "Polanco", 11530);
        colonias.add(colonia);
        colonia = new Colonia(coloniaId++, "Del Valle Centro", 3100);
        colonias.add(colonia);
        return colonias;
    }
}
```

PASO 8. MODIFICAMOS EL CÓDIGO

Archivo ColoniaHelper.java:

Dar click para ir al código

```
public long getColoniaIdPorNombre(String nombreColonia) {
    long coloniaId = 0;
    List<Colonia> colonias = getColonias();//Lista de colonias
    for (Colonia colonia : colonias) {
        if (colonia.getNombreColonia().equals(nombreColonia)) {
            coloniaId = colonia.getColoniaId();
            break;
        }
    }
    return coloniaId;
}

public long getColoniaIdPorCP(long codigoPostal) {
    long coloniaId = 0;
    List<Colonia> colonias = getColonias();
    for (Colonia colonia : colonias) {
        if (colonia.getCodigoPostal() == codigoPostal) {
            coloniaId = colonia.getColoniaId();
            break;
        }
    }
    return coloniaId;
}
```

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 8. MODIFICAMOS EL CÓDIGO

Archivo ColoniaHelper.java:

Dar click para ir al código

```
public List<SelectedItem> getSelectItems() {  
    List<SelectedItem> selectItems;  
    selectItems = new ArrayList<>();  
    List<Colonia> colonias = getColonias();  
    for (Colonia colonia : colonias) {  
        SelectedItem selectItem = new SelectedItem(colonia.getColoniaId(),  
            colonia.getNombreColonia());  
        selectItems.add(selectItem);  
    }  
    return selectItems;  
}
```

PASO 9. CREAR UNA CLASE JAVA

Modificar el archivo index.xhtml para utilizar las nuevas clases. Buscar las siguientes líneas:

```
<h:inputText id="codigoPostal" immediate="true"  
             onchange="this.form.submit()" required="true"  
             value="#{candidato.codigoPostal}"  
             valueChangeListener="#{vacanteForm.codigoPostalListener}" />
```

Remplazar por el siguiente código:

```
<h:inputText id="codigoPostal" immediate="true"  
             onchange="this.form.submit()" required="true"  
             value="#{candidato.codigoPostal}"  
             valueChangeListener="#{vacanteForm.codigoPostalListener}" >  
    <f:convertNumber integerOnly="true" pattern="00000"/>  
</h:inputText>
```

PASO 9. CREAR UNA CLASE JAVA

También buscar las siguientes líneas:

```
<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>
```

Remplazar por el siguiente código:

```
<h:selectOneMenu id="coloniaId" required="true" value="#{candidato.coloniaId}">
    <f:selectItem itemLabel="#{msgs['vacanteForm.seleccione']}" itemValue="" />
    <f:selectItems value="#{coloniaHelper.selectItems}" />
</h:selectOneMenu>
```


PASO 9. 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>BolsaTrabajoV8</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 9. 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}" >
    <f:convertNumber integerOnly="true" pattern="00000" />
  </h:inputText></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:selectItems value="#{coloniaHelper.selectItems}" />
    </h:selectOneMenu>
  </td>
  <td><h:message for="coloniaId" /></td>
</tr>
```

PASO 9. 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>
```


PASO 9. MODIFICAMOS EL CÓDIGO

Archivo index.xhtml:

Dar click para ir al código

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

PASO 10. MODIFICAR UNA CLASE JAVA

Modificar la clase VacanteForm.java para inyectar la dependencia del bean ColoniaHelper.java, :

```
@Inject  
private ColoniaHelper coloniaHelper;
```

Además inclumos el import respectivo de la clase ColoniaHelper.java y agregamos los métodos getter/setter del atributo coloniaHelper.



Experiencia y Conocimiento para tu vida

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 10. MODIFICAR UNA CLASE JAVA

Además, en la clase VacanteForm.java reemplazamos el código del método codigoPostalListener con el siguiente:

```
public void codigoPostalListener(ValueChangeEvent valueChangeEvent) {  
    FacesContext facesContext = FacesContext.getCurrentInstance();  
    UIViewRoot uiViewRoot = facesContext.getViewRoot();  
    Long newCodigoPostal = (Long) valueChangeEvent.getNewValue();  
    log.info("Nuevo codigo postal: " + newCodigoPostal);  
    UIInput ciudadInputText = (UIInput) uiViewRoot.findComponent("vacanteForm:ciudad");  
    String ciudad = "Ciudad de Mexico";  
    ciudadInputText.setValue(ciudad);  
    ciudadInputText.setSubmittedValue(ciudad);  
    UIInput coloniaInputText = (UIInput) uiViewRoot.findComponent("vacanteForm:coloniaId");  
    //Buscamos la colonia por id con ayuda del bean coloniaHelper  
    Long coloniaId = this.coloniaHelper.getColoniaIdPorCP(newCodigoPostal);  
    coloniaInputText.setValue(coloniaId);  
    coloniaInputText.setSubmittedValue(coloniaId);  
    //Enviamos la respuesta  
    facesContext.renderResponse();  
}
```


PASO 10. MODIFICAMOS EL CÓDIGO

Archivo VacanteForm.java:

Dar click para ir al código

```
package beans.backing;

import beans.helper.ColoniaHelper;
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 10. MODIFICAMOS EL CÓDIGO

Archivo VacanteForm.java:

Dar click para ir al código

```
@Inject
private ColoniaHelper coloniaHelper;

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 10. 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();
    Long newCodigoPostal = (Long) valueChangeEvent.getNewValue();
    log.info("Nuevo codigo postal: " + newCodigoPostal);
    UIInput ciudadInputText = (UIInput) uiViewRoot.findComponent("vacanteForm:ciudad");
    String ciudad = "Ciudad de Mexico";
    ciudadInputText.setValue(ciudad);
    ciudadInputText.setSubmittedValue(ciudad);
    UIInput coloniaInputText = (UIInput) uiViewRoot.findComponent("vacanteForm:coloniaId");
    //Buscamos la colonia por id con ayuda del bean coloniaHelper
    Long coloniaId = this.coloniaHelper.getColoniaIdPorCP(newCodigoPostal);
    coloniaInputText.setValue(coloniaId);
    coloniaInputText.setSubmittedValue(coloniaId);
    //Enviamos la respuesta
    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 10. MODIFICAMOS EL CÓDIGO

Archivo VacanteForm.java:

Dar click para ir al código

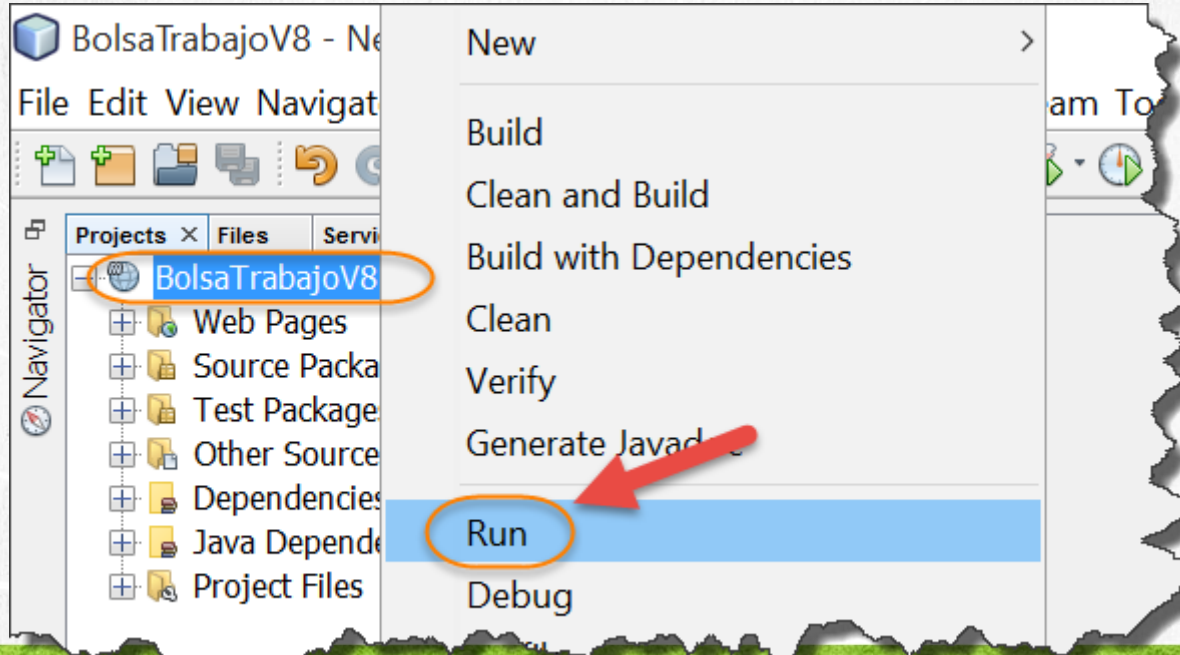
```
public boolean isComentarioEnviado() {  
    return comentarioEnviado;  
}  
  
public void setComentarioEnviado(boolean comentarioEnviado) {  
    this.comentarioEnviado = comentarioEnviado;  
}  
  
public ColoniaHelper getColoniaHelper() {  
    return coloniaHelper;  
}  
  
public void setColoniaHelper(ColoniaHelper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

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 11. EJECUCIÓN DEL PROYECTO

Ejecutamos el proyecto. Antes de ejecutar el proyecto se recomienda detener el servidor Java, hacer un clean& build del proyecto y ejecutar la aplicación como sigue:



CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 11. EJECUCIÓN DEL PROYECTO

Ejecutamos el proyecto. El resultado es como sigue:

Ubaldo

BolsaTrabajoV8

localhost:8080/BolsaTrab

Nombre: Introduce tu nombre

Apellido: Introduce tu Apellido

Sueldo deseado: 0

Fecha de Nacimiento:

Código Postal: 11530

Colonia: Polanco

Ciudad: Ciudad de Mexico

[Mostrar](#)

Enviar

Podemos probar con varios valores y el SelectItem cambiará de Manera Automática si se proporciona el valor correcto (Ej. 03810, 11530 ó 3100)

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

CONCLUSIÓN DEL EJERCICIO

Con este ejercicio hemos puesto las bases a nuestro proyecto para recuperar la lista de colonias de manera dinámica, la cual ya está en una clase separada, y el siguiente paso será obtener la lista de una base de datos.

Seguimos utilizando el componente SelectItem, y con ayuda de este componente ahora recuperamos la colección de datos desde código Java, y ya no desde la página index.xhtml.



Experiencia y Conocimiento para tu vida

CURSO DE JAVASERVER FACES

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