

CURSO DE JAVASERVER FACES

EJERCICIO

VALIDADORES EN JSF



Experiencia y Conocimiento para tu vida

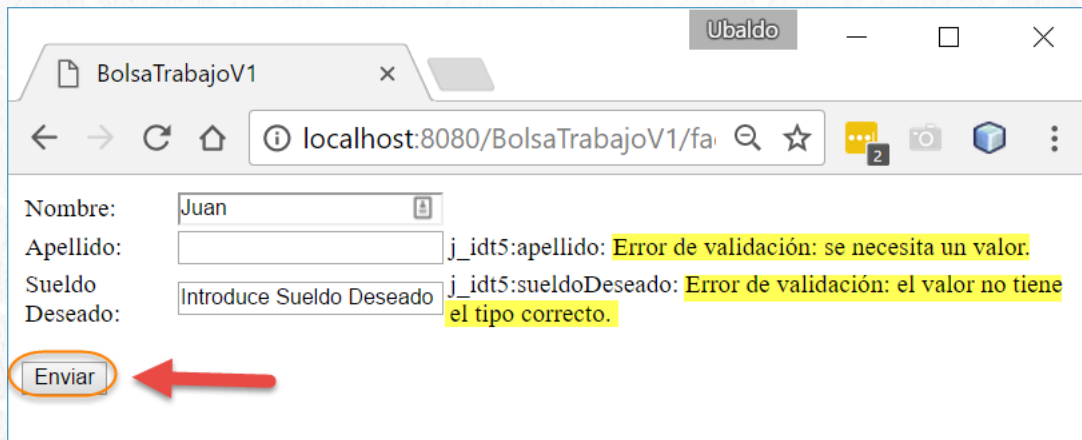
CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

OBJETIVO DEL EJERCICIO

En este ejercicio manejaremos el concepto de Validación en JSF.

Agregaremos el campo de apellido y sueldo deseado con el objetivo de hacerlos campos requeridos. Parte del resultado deberá ser similar al mostrado a continuación:



The screenshot shows a web browser window with the title 'BolsaTrabajoV1'. The address bar displays 'localhost:8080/BolsaTrabajoV1/fa'. The form contains the following fields and validation messages:

- Nombre:** Juan
- Apellido:** (empty field) j_idt5:apellido: Error de validación: se necesita un valor.
- Sueldo Deseado:** Introduce Sueldo Deseado j_idt5:sueldoDeseado: Error de validación: el valor no tiene el tipo correcto.

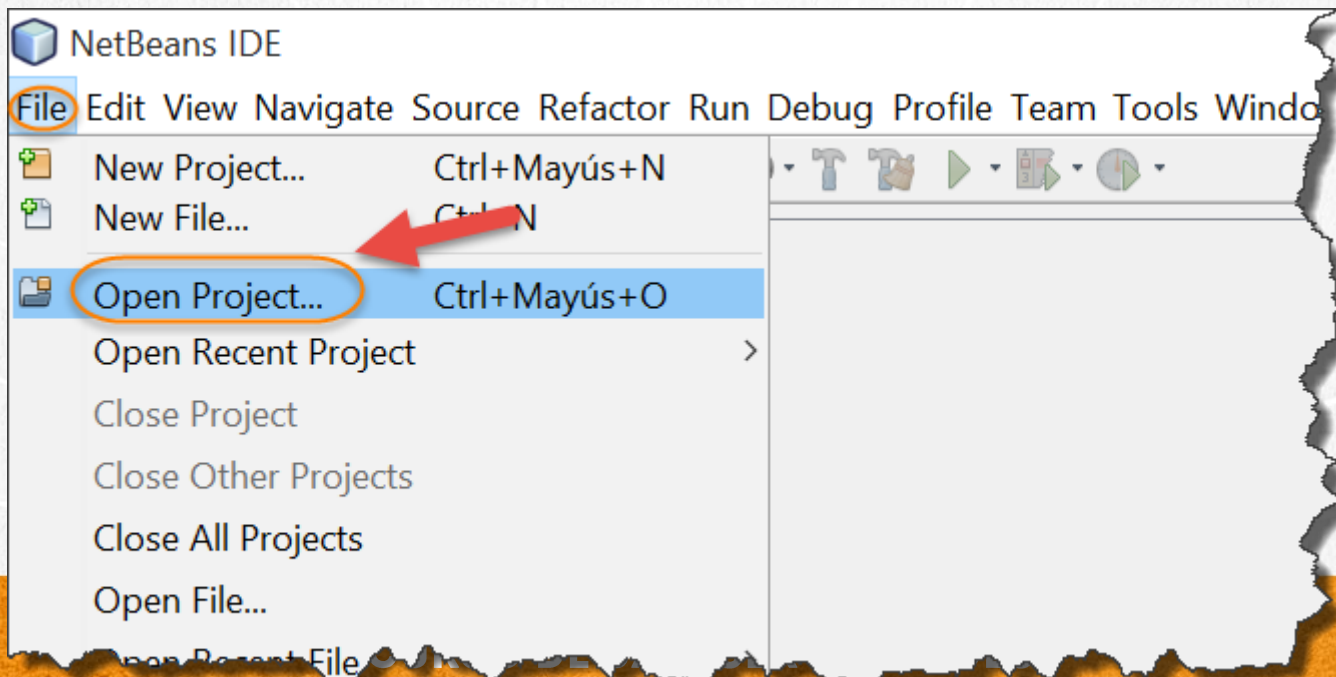
At the bottom left, there is an 'Enviar' button, which is circled in red with a red arrow pointing to it.

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

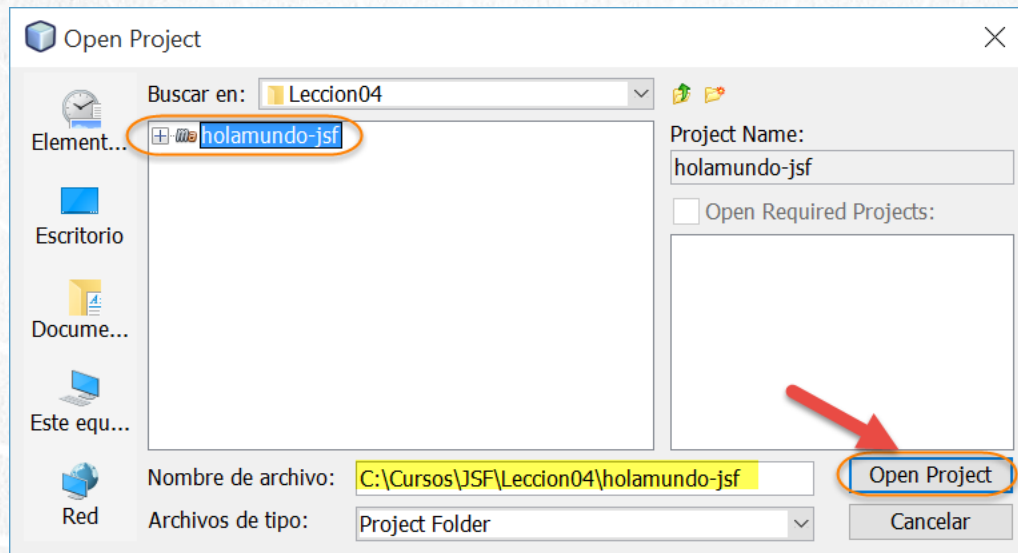
PASO 1. ABRIR EL PROYECTO

Abrimos el proyecto holamundo-jsf de la última lección sólo en caso de que no lo tengamos ya abierto:



PASO 1. ABRIR EL PROYECTO

Abrimos el proyecto holamundo-jsf de la última lección sólo en caso de que no lo tengamos ya abierto:

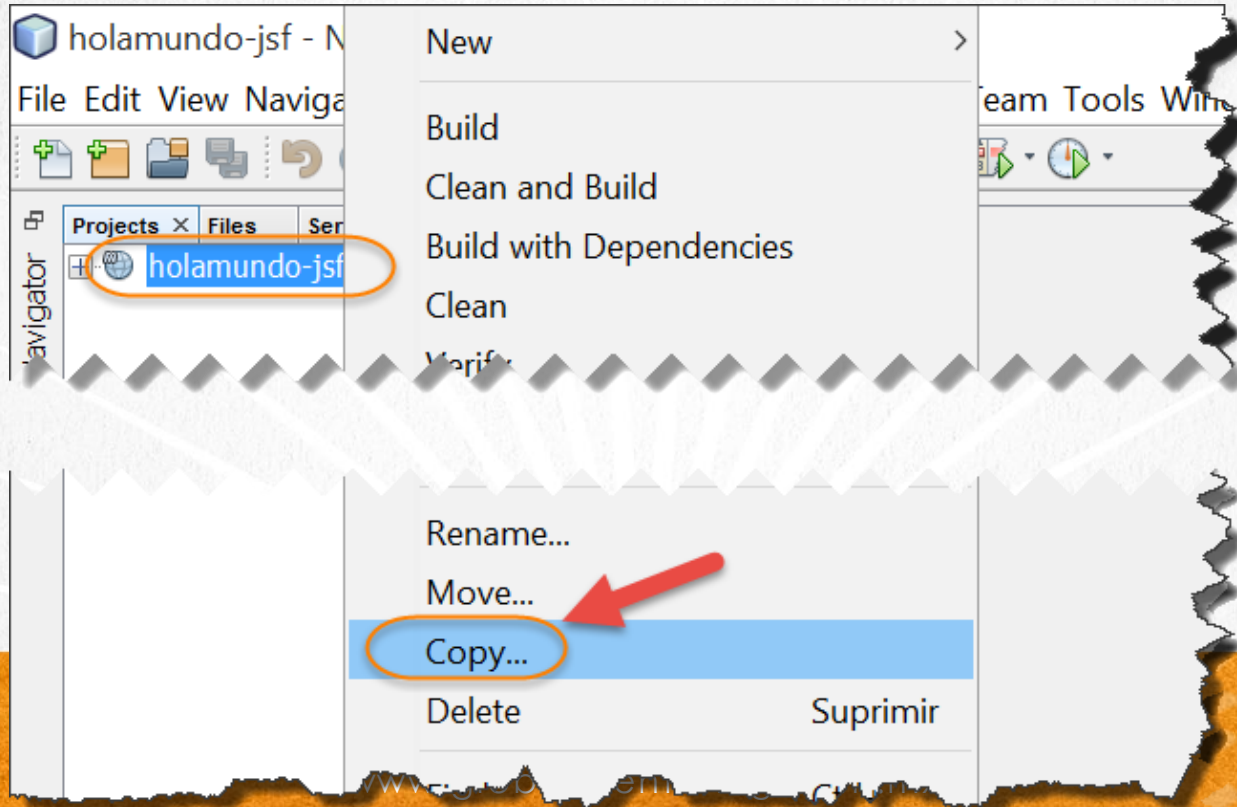


CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

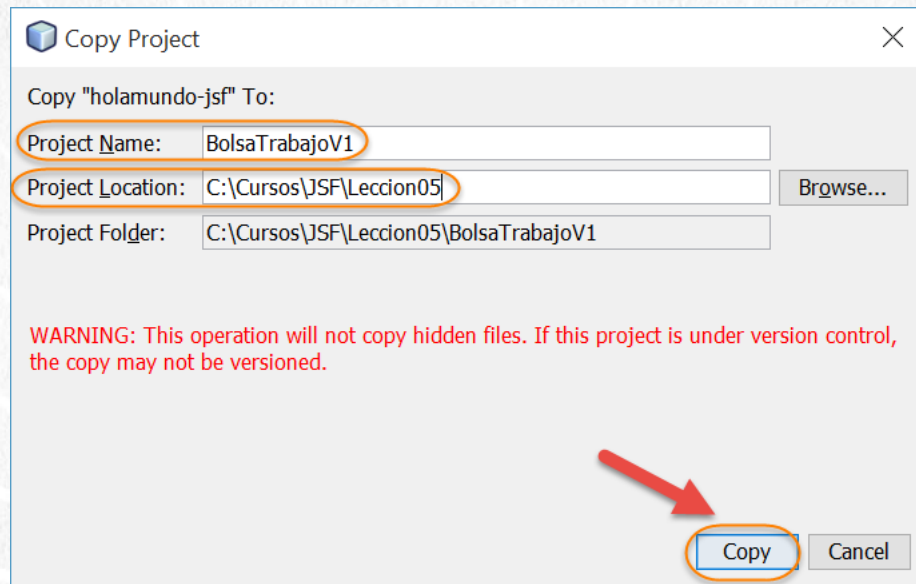
PASO 2. COPIAR EL PROYECTO

Copiamos el proyecto holamundo-jsf:



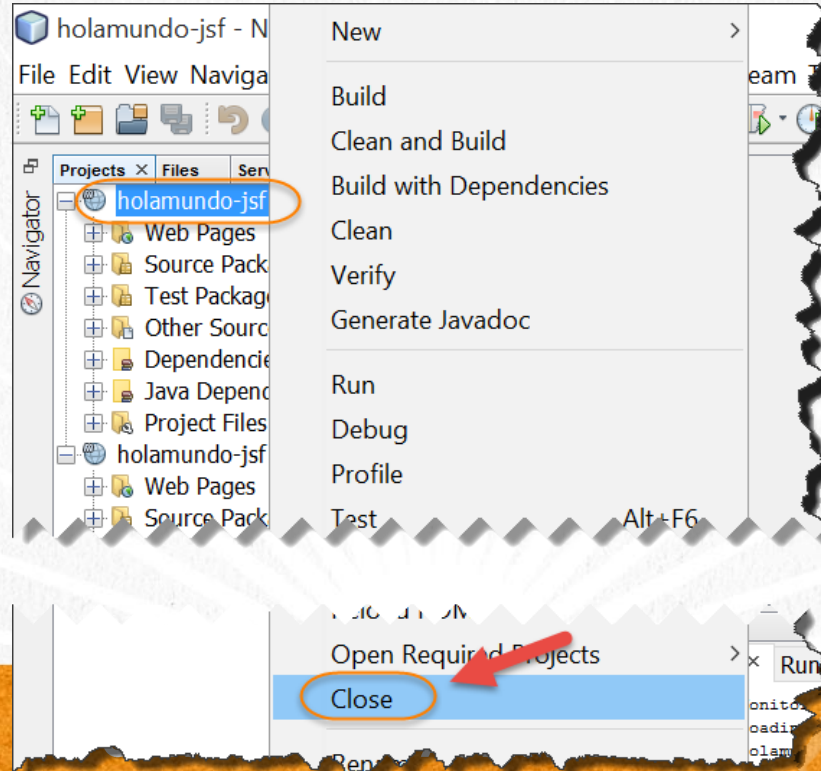
PASO 2. COPIAR EL PROYECTO

Copiamos el proyecto holamundo-jsf y lo renombramos a BolsaTrabajoV1:



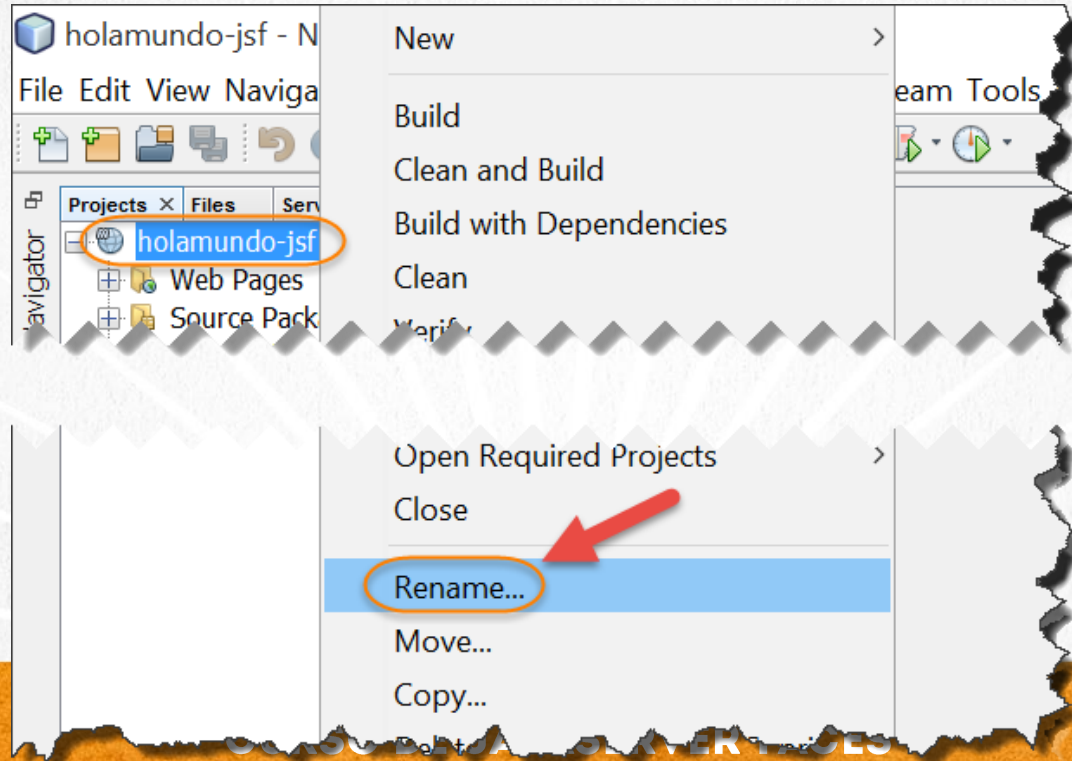
PASO 3. CERRAR PROYECTO

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



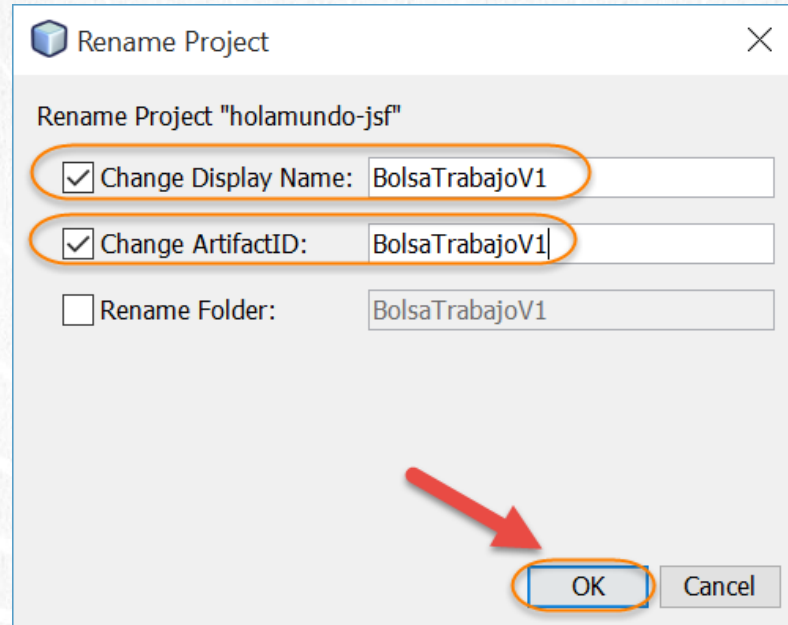
PASO 4. RENOMBRAR PROYECTO

Cambiamos el nombre del proyecto a BolsaTrabajoV1:



PASO 4. RENOMBRAR PROYECTO

Cambiamos el nombre del proyecto a BolsaTrabajoV1:



CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 5. MODIFICAR ARCHIVO JAVA

Modificamos el archivo Candidato.java para agregar el campo de apellido y el campo de sueldoDeseado, además crear sus métodos getters/setters.

Al final el archivo Candidato.java quedará como sigue:



Experiencia y Conocimiento para tu vida

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 5. MODIFICAMOS EL CÓDIGO

Archivo Candidato.java:

Clic para descargar código

```
package beans.model;

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 String sueldoDeseado = "Introduce Sueldo Deseado";

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

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


PASO 5. MODIFICAMOS EL CÓDIGO

Archivo Candidato.java:

Clic para descargar 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 String getSueldoDeseado() {  
    return sueldoDeseado;  
}  
  
public void setSueldoDeseado(String sueldoDeseado) {  
    this.sueldoDeseado = sueldoDeseado;  
    log.info("Modificando la propiedad sueldoDeseado:" + this.sueldoDeseado);  
}  
}
```

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 6. MODIFICAR EL ARCHIVO XHTML

Modificamos el archivo index.xhtml para validar el campo nombre:

En el archivo index.xhtml, sustituir:

```
<h:inputText id="nombre" value="#{candidato.nombre}" />
```

- Por el siguiente código:

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

- De esta manera validaremos que el campo nombre contenga información antes de enviar el formulario.

PASO 6. MODIFICAR EL ARCHIVO XHTML

Modificamos el archivo index.xhtml para agregar el campo de apellido:

- Pegar el siguiente código en index.xhtml generando un nuevo renglón <tr>:

```
<tr>
  <td><h:outputLabel for="apellido" value="Apellido:" /></td>
  <td><h:inputText id="apellido" required="true"
    value="#{candidato.apellido}" /></td>
  <td><h:message for="apellido" /></td>
</tr>
```


PASO 6. MODIFICAR EL ARCHIVO XHTML

Modificamos el archivo index.xhtml para agregar el campo de sueldoDeseado:

- Pegar el siguiente código en index.xhtml generando un nuevo renglón <tr>:

```
<tr>
  <td><h:outputLabel for="sueldoDeseado" value="Sueldo Deseado:" /></td>
  <td><h:inputText id="sueldoDeseado" required="true"
    value="#{candidato.sueldoDeseado}"
    <f:validateLongRange minimum="1000" maximum="50000"/>
  </h:inputText></td>
  <td><h:message for="sueldoDeseado" /></td>
</tr>
```

PASO 6. MODIFICAR EL ARCHIVO XHTML

Finalmente al hacer los cambios anteriores y algunos otros menores, como cambio del título de la página y agregar el taglib de core f al inicio del archivo index.xhtml.

```
xmlns:f=http://xmlns.jcp.org/jsf/core
```

Y agregamos el componente de manejo de mensajes globales en el formulario html:

```
<h:messages globalOnly="true" ></h:messages>
```

El archivo index.xhtml final queda como sigue:

PASO 7. MODIFICAMOS EL CÓDIGO

Archivo index.xhtml:

Clic para descargar 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>BolsaTrabajoV1</title>
  </h:head>
  <h:body>
    <h:form>

      <h:messages globalOnly="true" /></h:messages>

      <table border="0">
        <tbody>
          <tr>
            <td><h:outputLabel for="nombre" value="Nombre:" /></td>
            <td><h:inputText id="nombre" value="#{candidato.nombre}" required="true" /></td>
            <td><h:message for="nombre" /></td>
          </tr>
          <tr>
            <td><h:outputLabel for="apellido" value="Apellido:" /></td>
            <td><h:inputText id="apellido" value="#{candidato.apellido}" required="true" /></td>
            <td><h:message for="apellido" /></td>
          </tr>
        </tbody>
      </table>
    </h:body>
  </html>
```


PASO 7. MODIFICAMOS EL CÓDIGO

Archivo index.xhtml:

Clic para descargar código

```
<tr>
  <td><h:outputLabel for="sueldoDeseado" value="Sueldo Deseado:" /></td>
  <td><h:inputText id="sueldoDeseado" required="true"
    value="#{candidato.sueldoDeseado}">
    <f:validateLongRange minimum="1000" maximum="50000"/>
  </h:inputText></td>
  <td><h:message for="sueldoDeseado" /></td>
</tr>
</tbody>
</table>
<br/>
<h:commandButton action="#{vacanteForm.enviar}" value="Enviar"/>
</h:form>

</h:body>
</html>
```

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 8. MODIFICAR LA CLASE JAVA

Modificamos la clase VacanteForm.java para agregar validación pero ahora con código Java.

En caso de que se cumpla la condición especificada, mandamos un mensaje al usuario. Esto solo es un ejemplo, pero nosotros podemos agregar cualquier lógica de presentación a nuestro ManagedBean y así poder agregar las validaciones necesarias a nuestra capa de presentación con JSF.

Veamos como queda finalmente la clase VacanteForm.java

PASO 8. MODIFICAMOS EL CÓDIGO

Archivo VacanteForm.java:

Clic para descargar código

```
package beans.backing;

import beans.model.Candidato;
import javax.inject.Named;
import javax.enterprise.context.RequestScoped;
import javax.faces.application.FacesMessage;
import javax.faces.context.FacesContext;
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();

    @Inject
    private Candidato candidato;

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

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


PASO 8. MODIFICAMOS EL CÓDIGO

Archivo VacanteForm.java:

Clic para descargar código

```
//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());

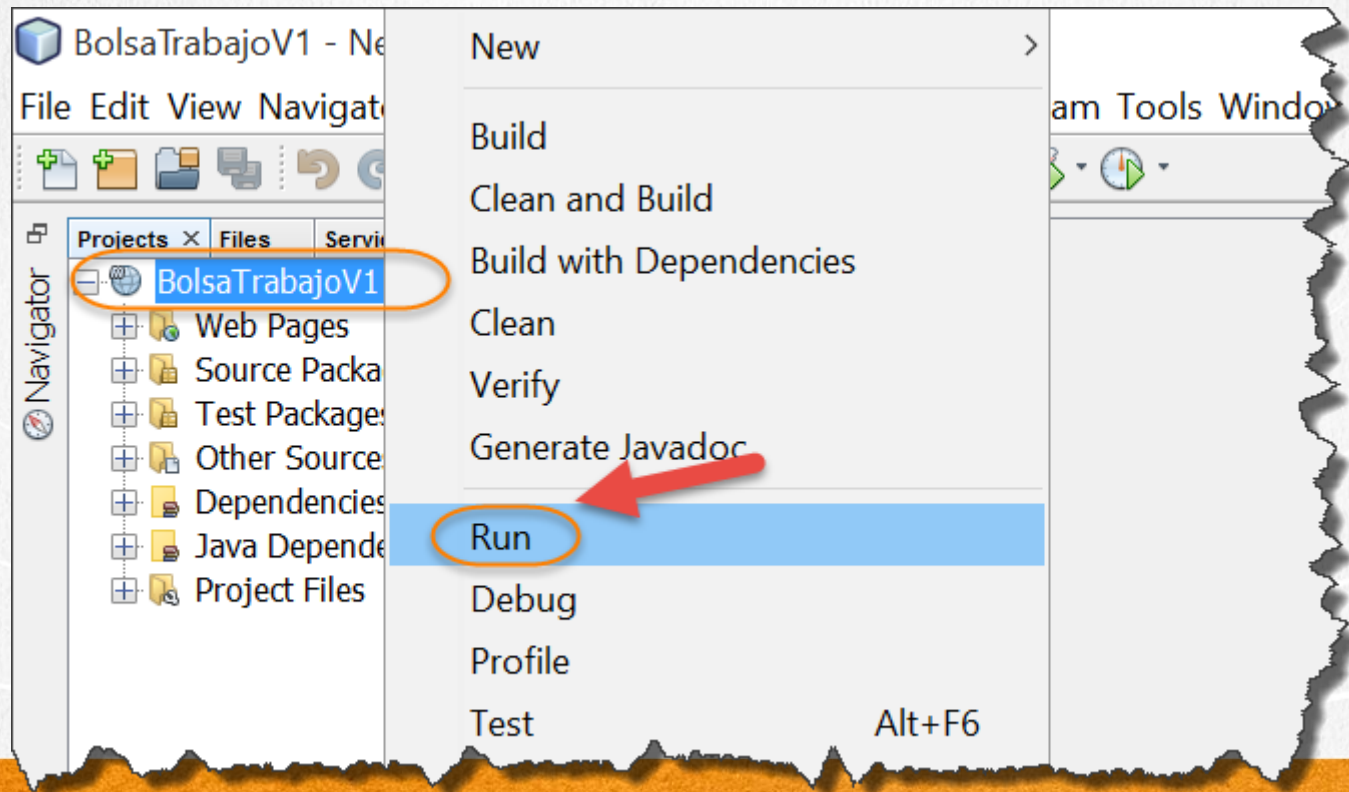
    if (this.candidato.getNombre().equals("Juan")) {

        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
    }
}
```

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 8. EJECUTAMOS EL PROYECTO

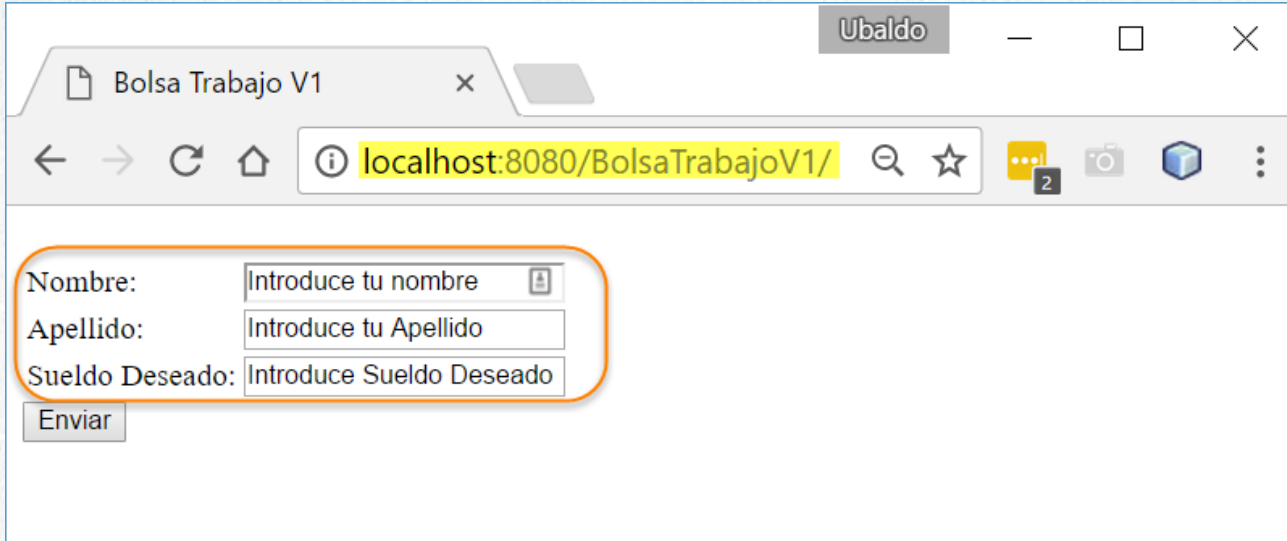


CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 8. EJECUTAMOS EL PROYECTO

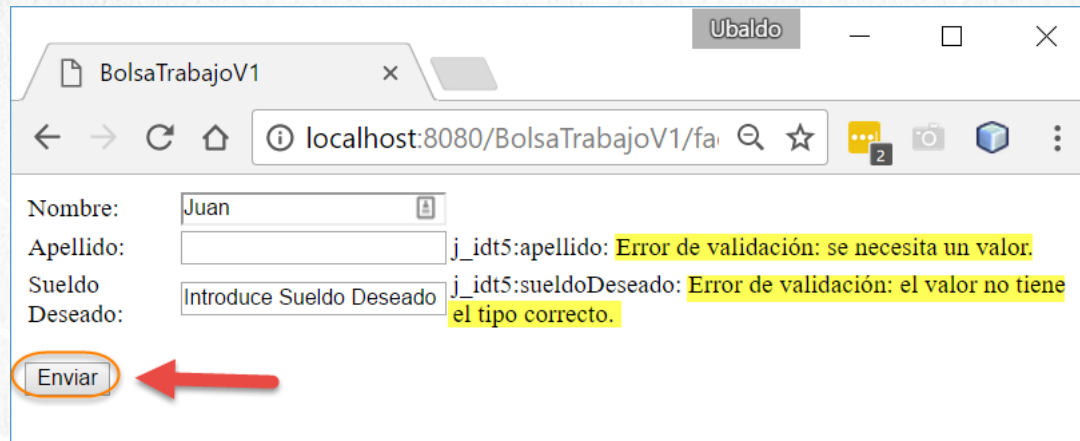
La salida del proyecto es como sigue:



A screenshot of a web browser window. The title bar shows the name 'Ubaldo'. The browser tab is labeled 'Bolsa Trabajo V1'. The address bar shows the URL 'localhost:8080/BolsaTrabajoV1/'. The main content area displays a form with three input fields: 'Nombre:' with placeholder text 'Introduce tu nombre', 'Apellido:' with placeholder text 'Introduce tu Apellido', and 'Sueldo Deseado:' with placeholder text 'Introduce Sueldo Deseado'. Each input field has a small icon on the right. Below the input fields is a button labeled 'Enviar'. The form is highlighted with an orange rounded rectangle.

PASO 8. EJECUTAMOS EL PROYECTO

Probamos con distintos valores:



The screenshot shows a web browser window with the title 'Ubaldo'. The address bar displays 'localhost:8080/BolsaTrabajoV1/fa'. The page content includes a form with the following fields and labels:

- Nombre: Juan
- Apellido: (empty)
- Sueldo Deseado: Introduce Sueldo Deseado

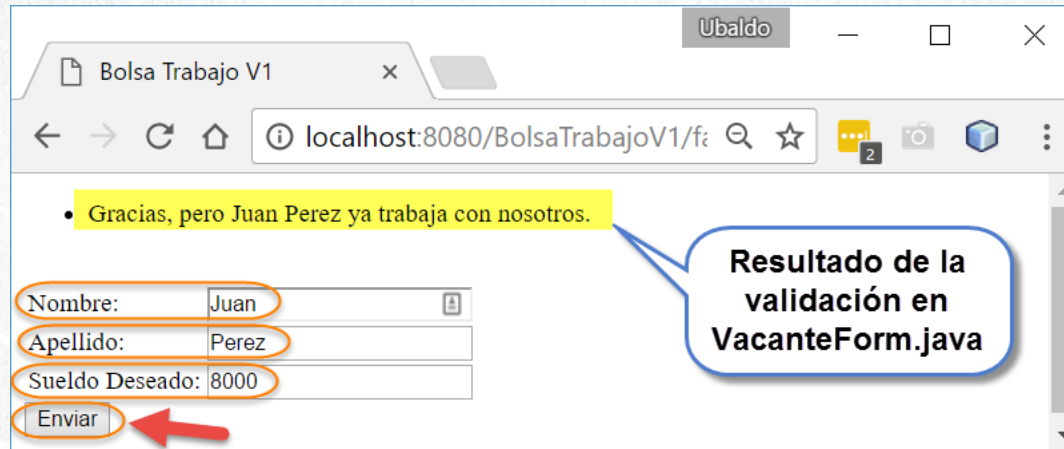
Validation error messages are displayed next to the 'Apellido' and 'Sueldo Deseado' fields:

- Next to 'Apellido': j_idt5:apellido: Error de validación: se necesita un valor.
- Next to 'Sueldo Deseado': j_idt5:sueldoDeseado: Error de validación: el valor no tiene el tipo correcto.

An 'Enviar' button is located at the bottom left of the form, highlighted with an orange circle. A red arrow points to this button.

PASO 8. EJECUTAMOS EL PROYECTO

Caso de validación en ManagedBean:



TAREAS EXTRAS

Se deja como tareas extra probar más casos con el formulario y verificar que se apliquen las validaciones correctamente.

También se puede hacer una revisión paso a paso aplicando el modo debug.

Por último, se debe revisar el log del servidor para ver que todo funciona según lo esperado y también veamos en qué momento se aplican las validaciones y en que fase del ciclo de vida de JSF.



Experiencia y Conocimiento para tu vida

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

CONCLUSIÓN DEL EJERCICIO

En esta lección estudiamos el tema de validadores en JSF.

Vimos varias formas de validar los campos de nuestro formulario JSF, además estudiamos como agregar validación desde un ManagedBean, y de esta forma no solo validar el formulario, sino también con lógica de presentación dentro de nuestro código Java con JSF.

Con esto ya tenemos las bases para manejar el tema de validadores en JSF.



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