

CURSO DE JAVASERVER FACES

EJERCICIO

CONVERTIDORES EN JSF



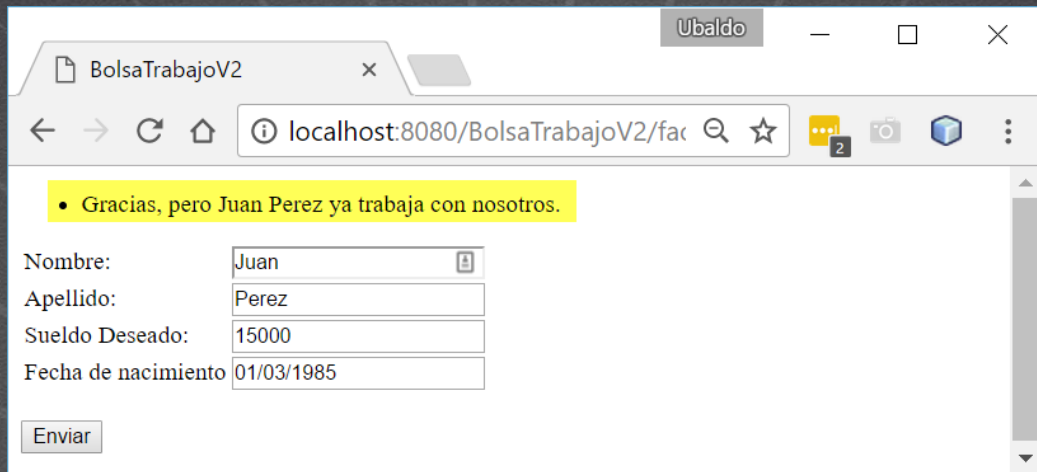
Experiencia y Conocimiento para tu vida

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

OBJETIVO DEL EJERCICIO

Modificar el proyecto JSF para poner en práctica el manejo de Convertidores en JSF. Al finalizar deberemos observar lo siguiente:



Ubaldo

BolsaTrabajoV2

localhost:8080/BolsaTrabajoV2/fac

- Gracias, pero Juan Perez ya trabaja con nosotros.

Nombre:

Apellido:

Sueldo Deseado:

Fecha de nacimiento:

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

OBJETIVO DEL EJERCICIO

Cambiar el tipo de dato de Sueldo deseado de String a Integer para que la conversión implícita se lleve a cabo.

Agregar un campo para la fecha de nacimiento y el uso de un convertidor de JSF para convertir a java.util.Date.



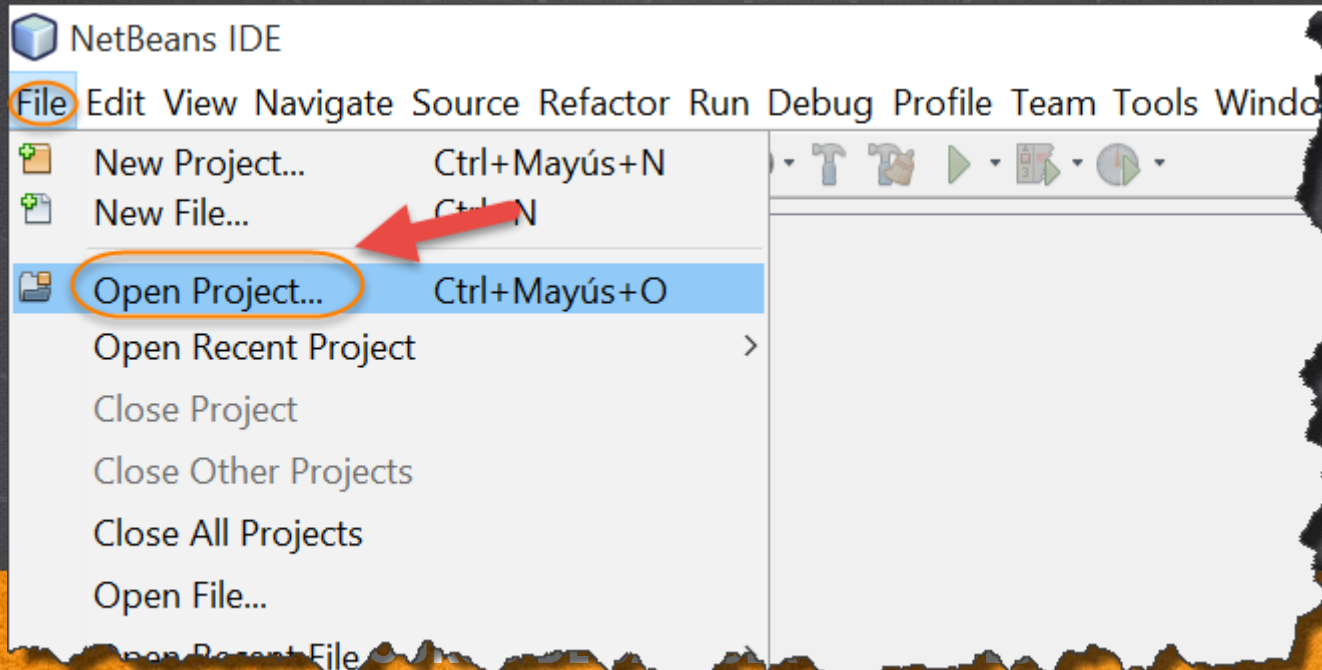
Experiencia y Conocimiento para tu vida

CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

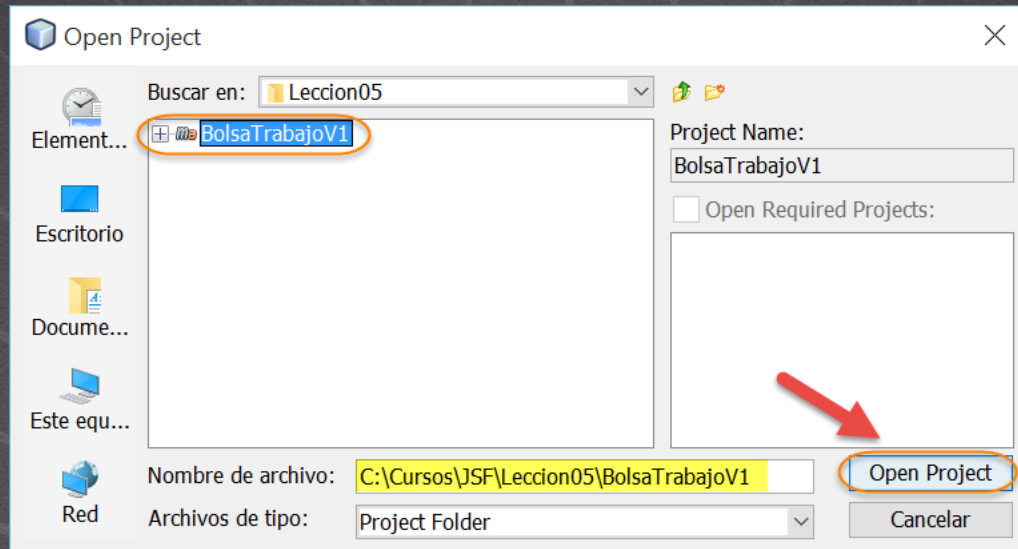
PASO 1. ABRIR EL PROYECTO

Abrimos el proyecto BolsaTrabajoV1 de la última lección sólo en caso de que no lo tengamos ya abierto:



PASO 1. ABRIR EL PROYECTO

Abrimos el proyecto BolsaTrabajoV1 de la última lección sólo en caso de que no lo tengamos ya abierto:

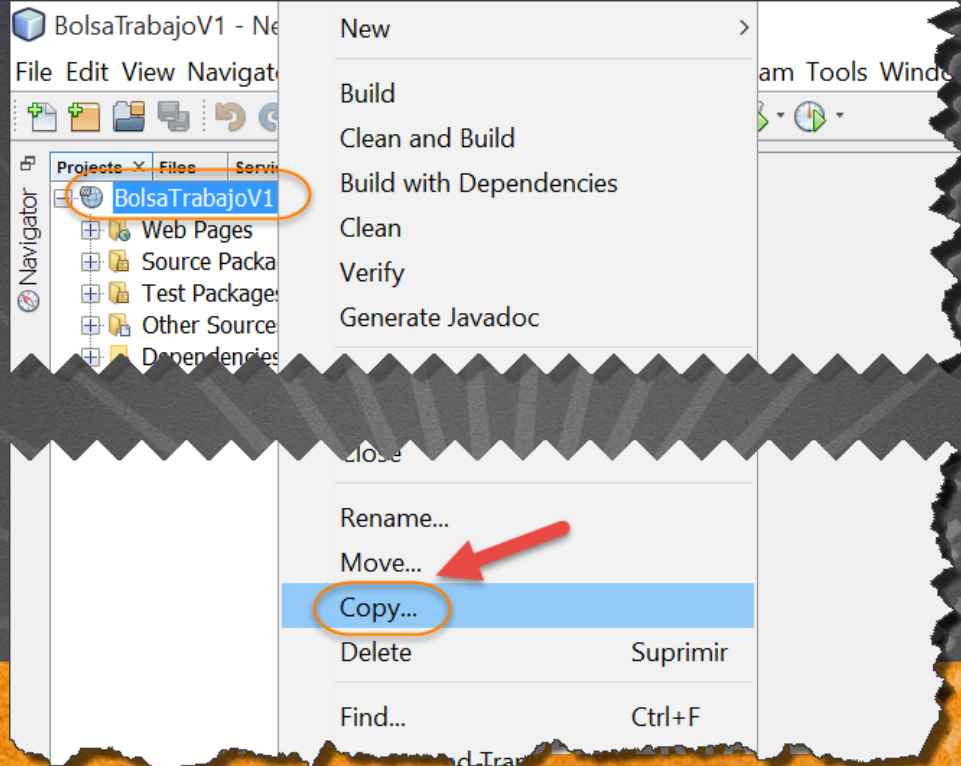


CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

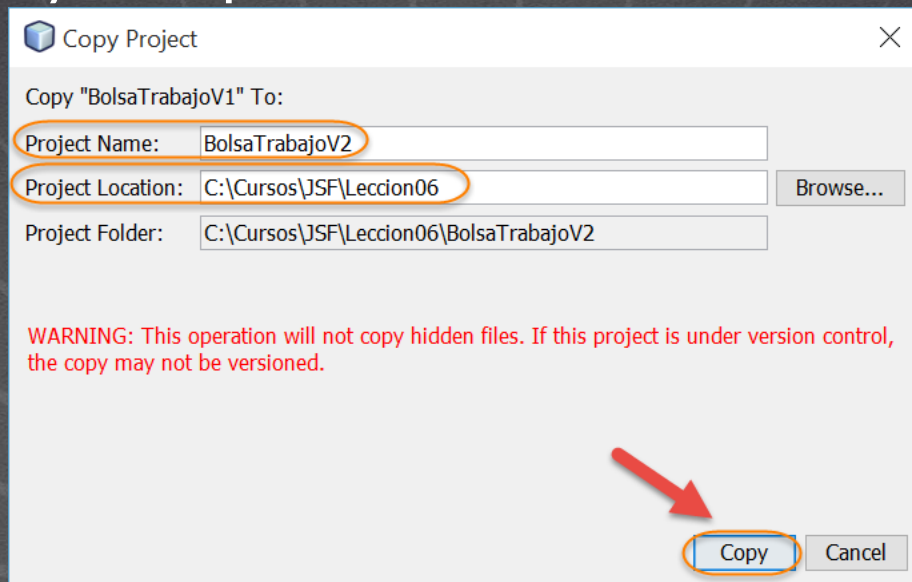
PASO 2. COPIAR EL PROYECTO

Copiamos el proyecto BolsaTrabajoV1:



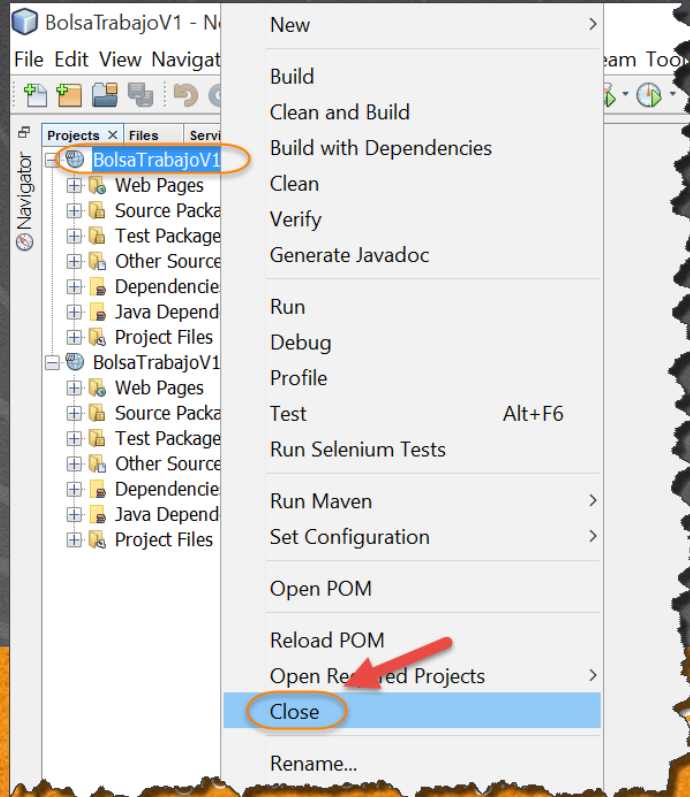
PASO 2. COPIAR EL PROYECTO

Copiamos el proyecto BolsaTrabajoV1 y lo renombramos a BolsaTrabajoV2 y lo depositamos en la nueva ruta mostrada:



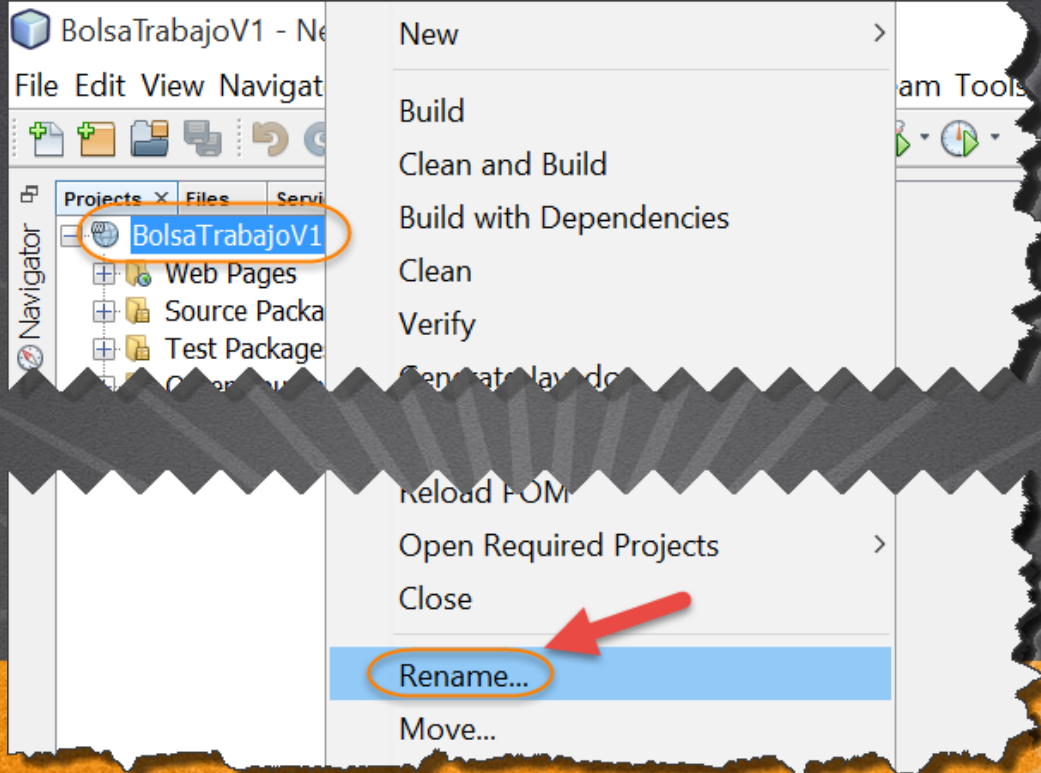
PASO 3. CERRAR PROYECTO

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



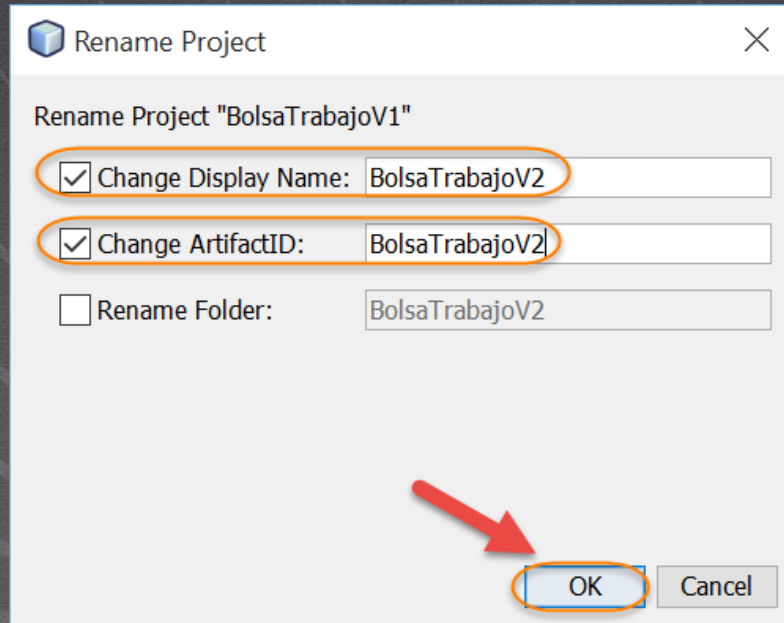
PASO 4. RENOMBRAR PROYECTO

Cambiamos el nombre del proyecto a BolsaTrabajoV2:



PASO 4. RENOMBRAR PROYECTO

Cambiamos el nombre del proyecto a BolsaTrabajoV2:



CURSO DE JAVASERVER FACES

www.globalmentoring.com.mx

PASO 5. MODICAR UNA CLASE JAVA

En la clase Candidato.java, cambiar el tipo de dato del campo "sueldoDeseado" de String a int. La variable debe quedar así:

```
private int sueldoDeseado;
```

Los métodos set/get tiene que ser cambiados de la misma manera. Quedando el código como se muestra:

```
public int getSueloDeseado() {  
    return sueldoDeseado;  
}  
  
public void setSueloDeseado(int sueldoDeseado) {  
    this.sueloDeseado = sueldoDeseado;  
}
```

PASO 5. MODICAR UNA CLASE JAVA

En la clase Candidato.java, pegar el siguiente código:

```
private Date fechaNacimiento;

public Date getFechaNacimiento() {
    return fechaNacimiento;
}

public void setFechaNacimiento(Date fechaNacimiento) {
    this.fechaNacimiento = fechaNacimiento;
}
```

Agregar al principio de la clase Candidato, el import siguiente:

```
import java.util.Date;
```


PASO 5. MODIFICAMOS EL CÓDIGO

Archivo Candidato.java:

Clic para descargar 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;

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

    public String getNombre() {
        return nombre;
    }

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

PASO 5. MODIFICAMOS EL CÓDIGO

Archivo Candidato.java:

Clic para descargar código

```
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 6. MODIFICAR EL ARCHIVO XHTML

En el archivo index.xhtml, agregar un renglón más <tr>, después del sueldo deseado:

```
<tr>
    <td><h:outputLabel for="fechaNacimiento" value="Fecha de
        nacimiento" /></td>
    <td><h:inputText id="fechaNacimiento" required="true"
        value="#{candidato.fechaNacimiento}">
        <f:convertDateTime pattern="dd/MM/yyyy" />
    </h:inputText></td>
    <td><h:message for="fechaNacimiento" /></td>
</tr>
```

PASO 6. 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>BolsaTrabajoV2</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:form>
  </h:body>
</html>
```

PASO 6. MODIFICAMOS EL CÓDIGO

Archivo index.xhtml:

Clic para descargar código

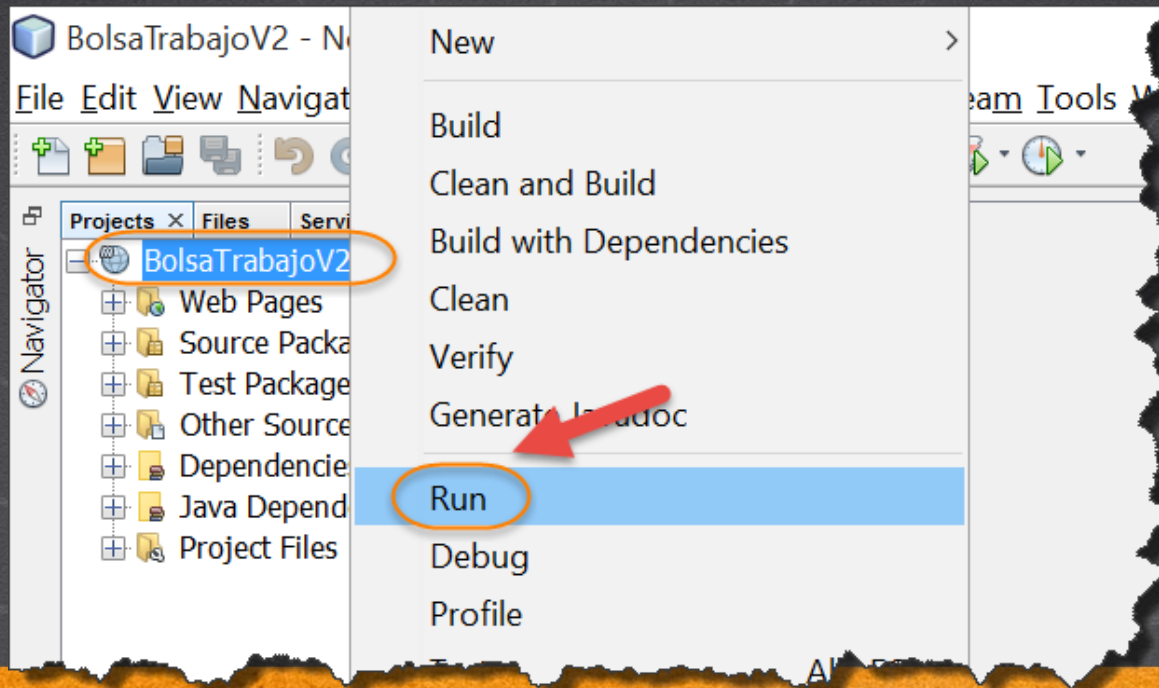
```
<tr>
  <td><h:outputLabel for="sueldoDeseado" value="Sueldo Deseado:" /></td>
  <td><h:inputText id="sueldoDeseado" required="true"
    value="#{candidato.sueldoDeseado}">
    <f:validateLongRange minimum="1000" maximum="50000" />
  </h:inputText></td>
  <td><h:message for="sueldoDeseado" /></td>
</tr>
<tr>
  <td><h:outputLabel for="fechaNacimiento" value="Fecha de nacimiento" /></td>
  <td><h:inputText id="fechaNacimiento" required="true"
    value="#{candidato.fechaNacimiento}">
    <f:convertDateTime pattern="dd/MM/yyyy" />
  </h:inputText></td>
  <td><h:message for="fechaNacimiento" /></td>
</tr>

</tbody>
</table>
<br/>
<h:commandButton action="#{vacanteForm.enviar}" value="Enviar"/>
</h:form>

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


PASO 7. EJECUTAMOS EL PROYECTO

Ejecutamos la aplicación:



CURSO DE JAVASERVER FACES

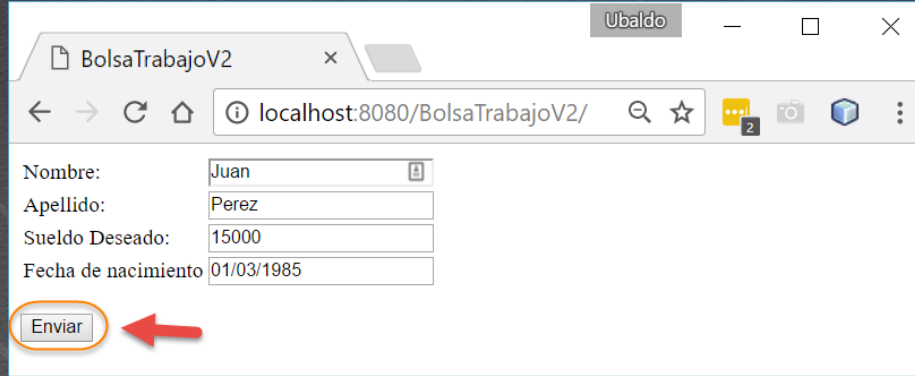
www.globalmentoring.com.mx

PASO 7. EJECUTAMOS EL PROYECTO

- Ejecutar la aplicación BolsaTrabajoV2
- Introducir los siguientes datos y haga clic en Enviar solicitud:
 - Nombre: Juan
 - Apellido: Perez
 - Sueldo Deseado: 15000
 - Fecha de Nacimiento: 01/03/1985
- El patrón de la fecha de nacimiento que se especificó es:
dd / mm / aaaa (día / mes / año)

PASO 18. EJECUTAMOS EL PROYECTO

Ejecutamos la aplicación. El resultado es similar al siguiente:



Ubaldo

BolsaTrabajoV2

localhost:8080/BolsaTrabajoV2/

Nombre: Juan

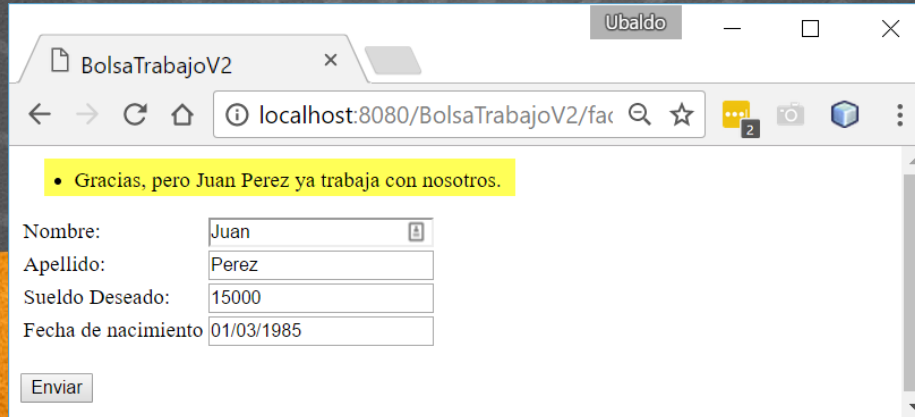
Apellido: Perez

Sueldo Deseado: 15000

Fecha de nacimiento: 01/03/1985

Enviar

A red arrow points to the 'Enviar' button.



Ubaldo

BolsaTrabajoV2

localhost:8080/BolsaTrabajoV2/fac

- Gracias, pero Juan Perez ya trabaja con nosotros.

Nombre: Juan

Apellido: Perez

Sueldo Deseado: 15000

Fecha de nacimiento: 01/03/1985

Enviar

A large blue curved arrow points from the first screenshot to this one.

TAREAS EXTRAS

Se deja como tareas extra probar más casos con el formulario y verificar que se apliquen las validaciones y conversiones 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 considerando las fases del ciclo de vida de JSF.

CONCLUSIÓN DEL EJERCICIO

- Con este ejercicio hemos puesto en práctica el concepto Convertidores en JSF.
- Probamos con distintos tipos de datos para observar cómo funciona este concepto en JSF.
- Con esto ya tenemos las bases para aplicar convertidores 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