

# **CURSO DE JAVASERVER FACES**

## **EJERCICIO**

### **BACKING MANAGED BEAN EN JSF**



Experiencia y Conocimiento para tu vida

**CURSO DE JAVASERVER FACES**

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

# OBJETIVO DEL EJERCICIO

Modificar el proyecto de holamundo-jsf para aplicar el concepto de Backing ManagedBean. El resultado final es como sigue:





# OBJETIVO DEL EJERCICIO

Introduciremos el concepto de Backing Managed Bean a nuestra aplicación.

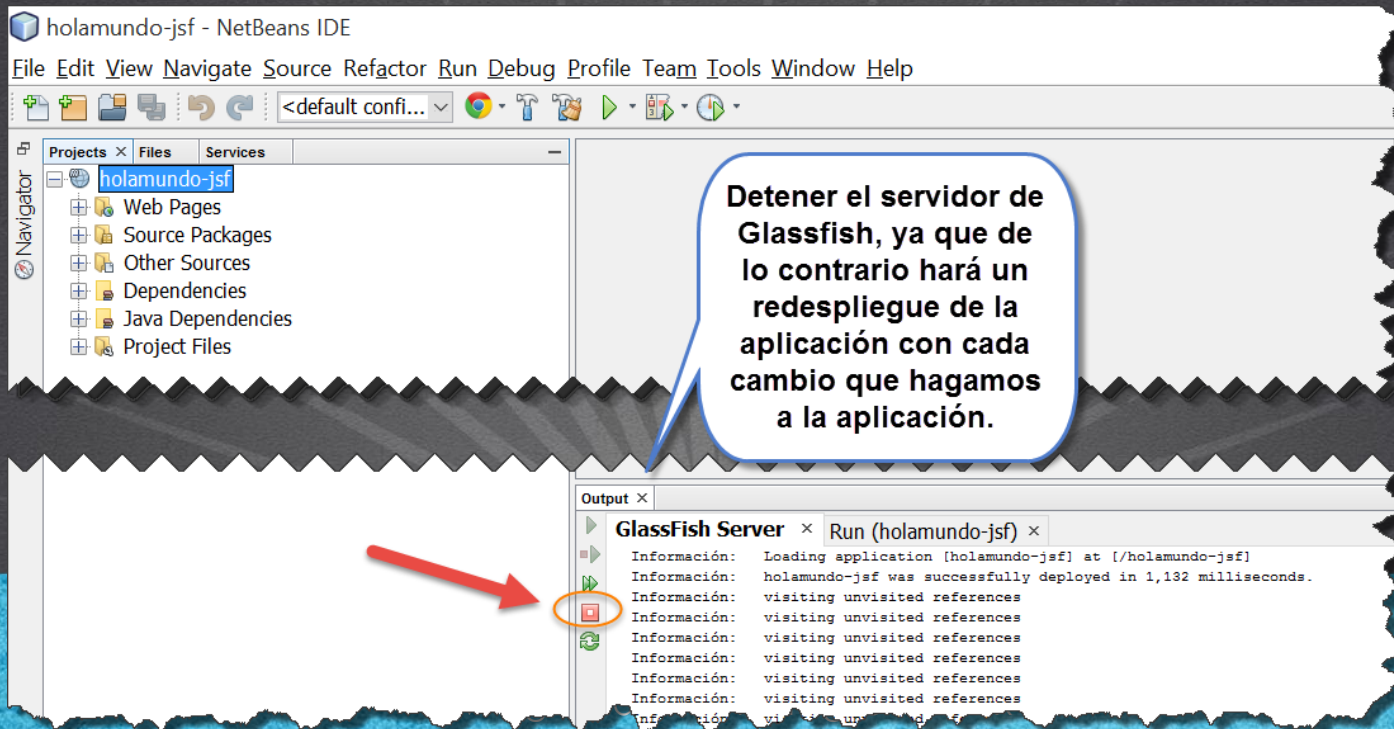
Utilizaremos el concepto de anotaciones para inyectar dependencias entre Managed Bean's.

Adicionalmente agregaremos algunas páginas JSF con las respectivas reglas de navegación.

Finalmente asociaremos funcionalidad al botón de Enviar para ejecutar el flujo de la aplicación.

# ANTES DE EMPEZAR

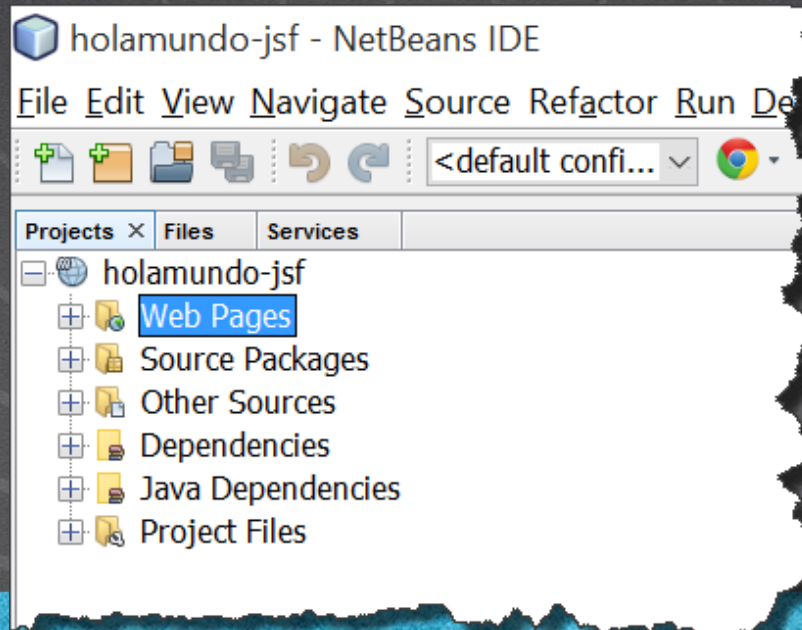
Continuamos trabajando con el proyecto de la lección anterior. Si está levantado el servidor de Glassfish, lo detenemos, ya que de lo contrario con cada cambio que hagamos se hará nuevamente un despliegue de la aplicación.





# PASO 1. ABRIR PROYECTO

Continuamos trabajando con el proyecto de la lección anterior, por lo que ya debe estar abierto:

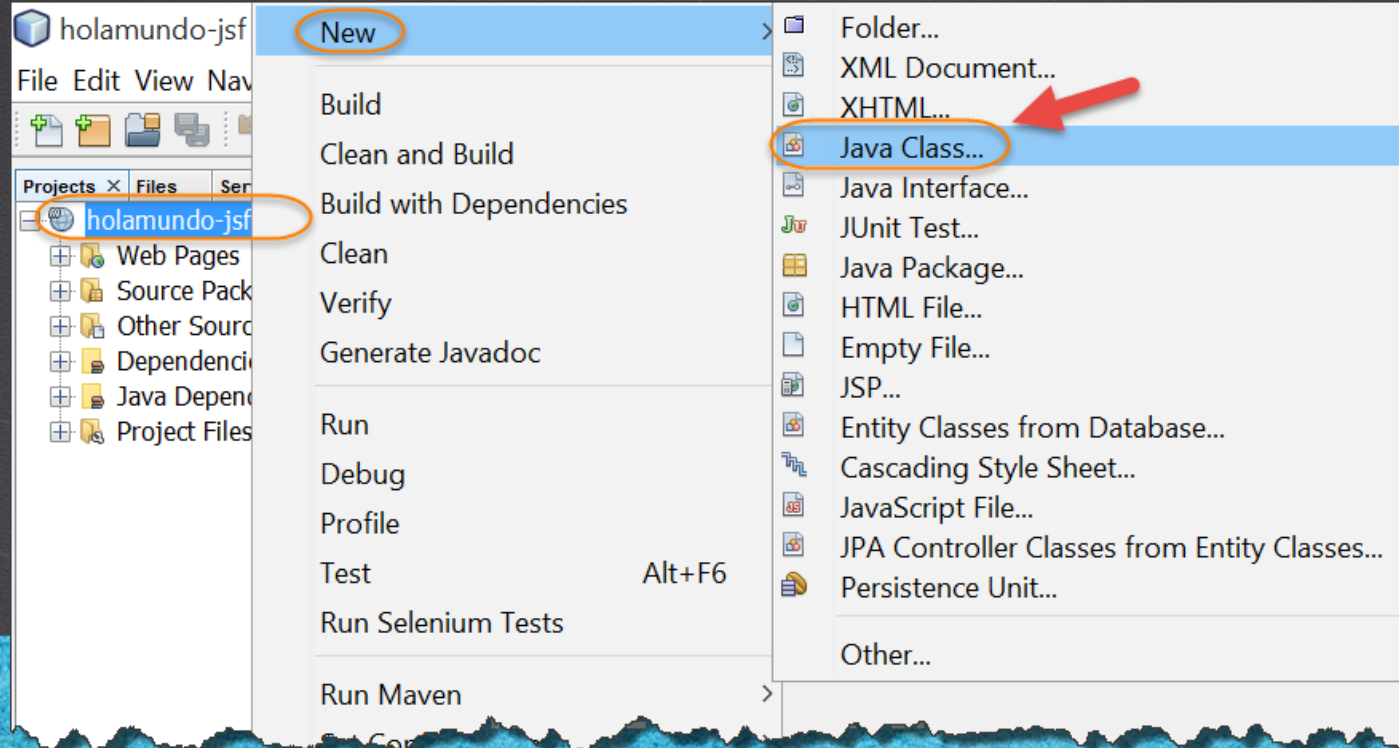


**CURSO DE JAVASERVER FACES**

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

# PASO 2. CREAR UNA CLASE JAVA

Creamos una clase Java que será un ManagedBean de JSF, llamada VacanteForm.java:



# PASO 2. CREAR UNA CLASE JAVA

Creamos una clase Java que será un ManagedBean de JSF, llamada VacanteForm.java:

**New Java Class**

**Steps**

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

**Name and Location**

Class Name: VacanteForm

Project: holamundo-jsf

Location: Source Packages

Package: beans.backing

Created File: sos\JSF\Leccion02\holamundo-jsf\src\main\java\beans\backing\VacanteForm.java

< Back Next > **Finish** Cancel Help

**CURSO DE JAVASERVER FACES**

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



# PASO 3. MODIFICAMOS EL CÓDIGO

## Archivo VacanteForm.java:

Dar click para ir al código

```
package beans.backing;

import beans.model.Candidato;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.ManagedProperty;
import javax.faces.bean.RequestScoped;

@ManagedBean
@RequestScoped
public class VacanteForm {

    @ManagedProperty(value="#{candidato}")
    private Candidato candidato;

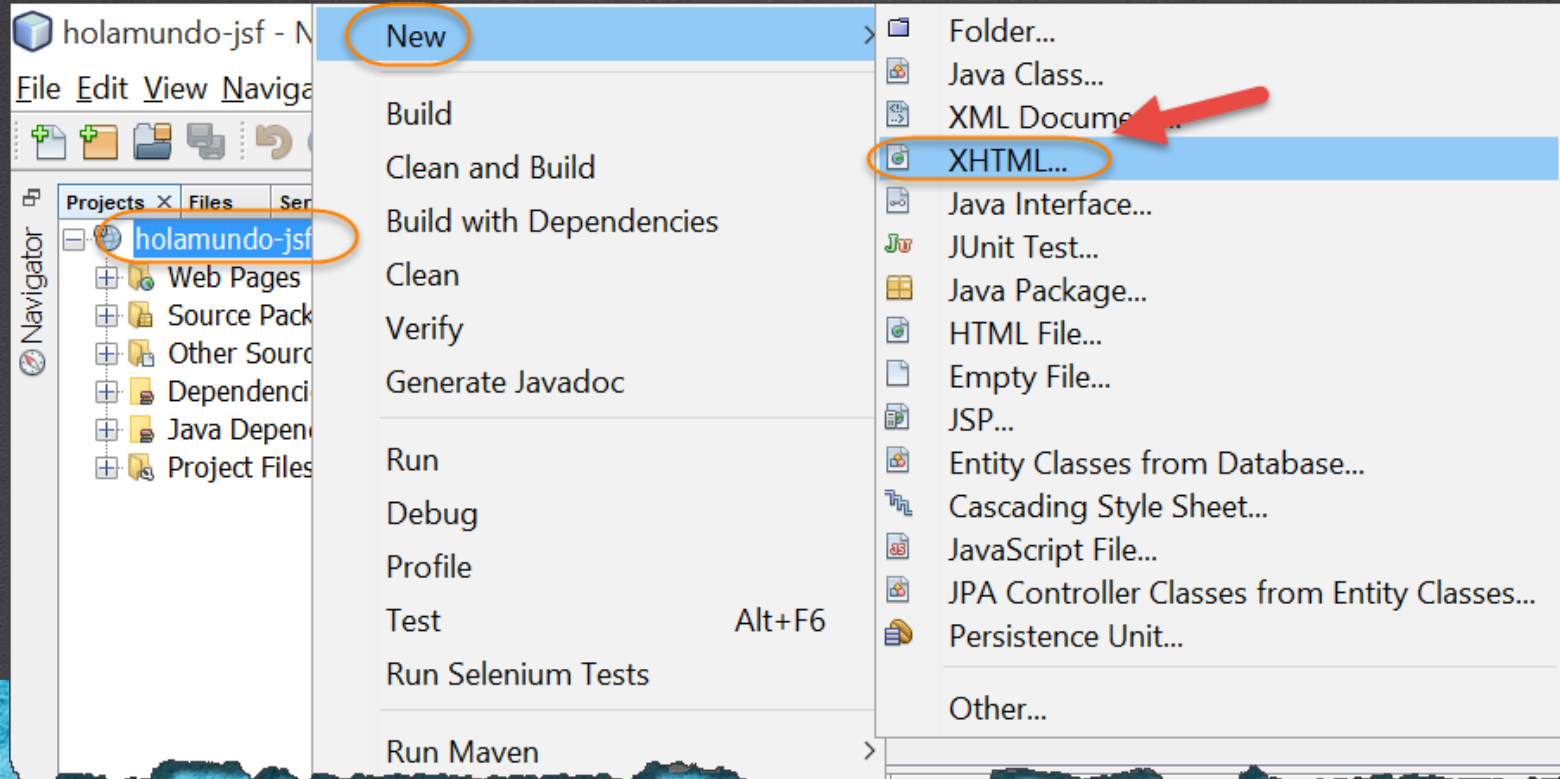
    public void setCandidato(Candidato candidato) {
        this.candidato = candidato;
    }

    public String enviar() {
        if (this.candidato.getNombre().equals("Juan")) {
            return "exito";
        } else {
            return "fallo";
        }
    }
}
```



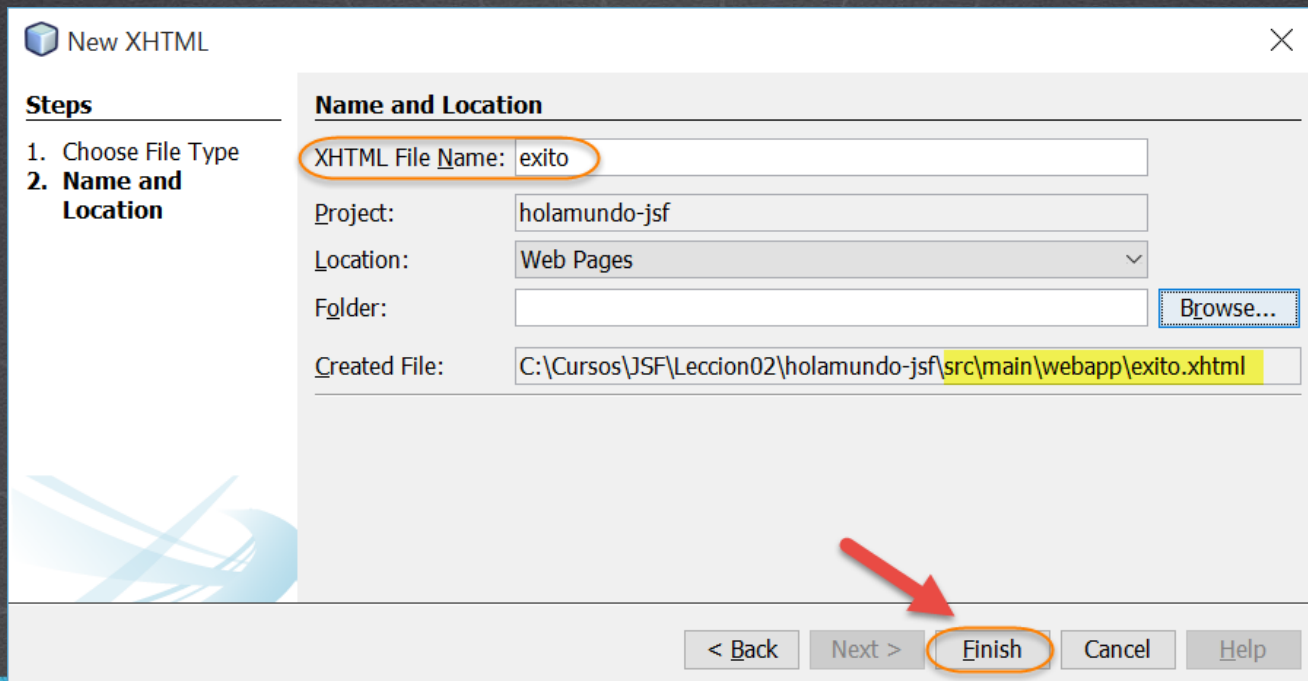
## PASO 4. AGREGAR UNA PÁGINA JSF PARA EL CASO DE "ÉXITO"

Creamos una pagina .xhtml para el caso de exito.



## PASO 4. AGREGAR UNA PÁGINA JSF PARA EL CASO DE "ÉXITO"

Creamos una pagina .xhtml para el caso de éxito.



New XHTML

**Steps**

1. Choose File Type
2. Name and Location

**Name and Location**

XHTML File Name:

Project:

Location:

Folder:

Created File:

**CURSO DE JAVASERVER FACES**

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

# PASO 5. MODIFICAMOS EL CÓDIGO

Archivo exito.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">
  <h:head>
    <title>Exito</title>
  </h:head>
  <h:body>
    <h:form>
      Gracias #{candidato.nombre} por aplicar a la vacante
      <br/>
      <h:commandLink action="index">Regresar</h:commandLink>
    </h:form>
  </h:body>
</html>
```

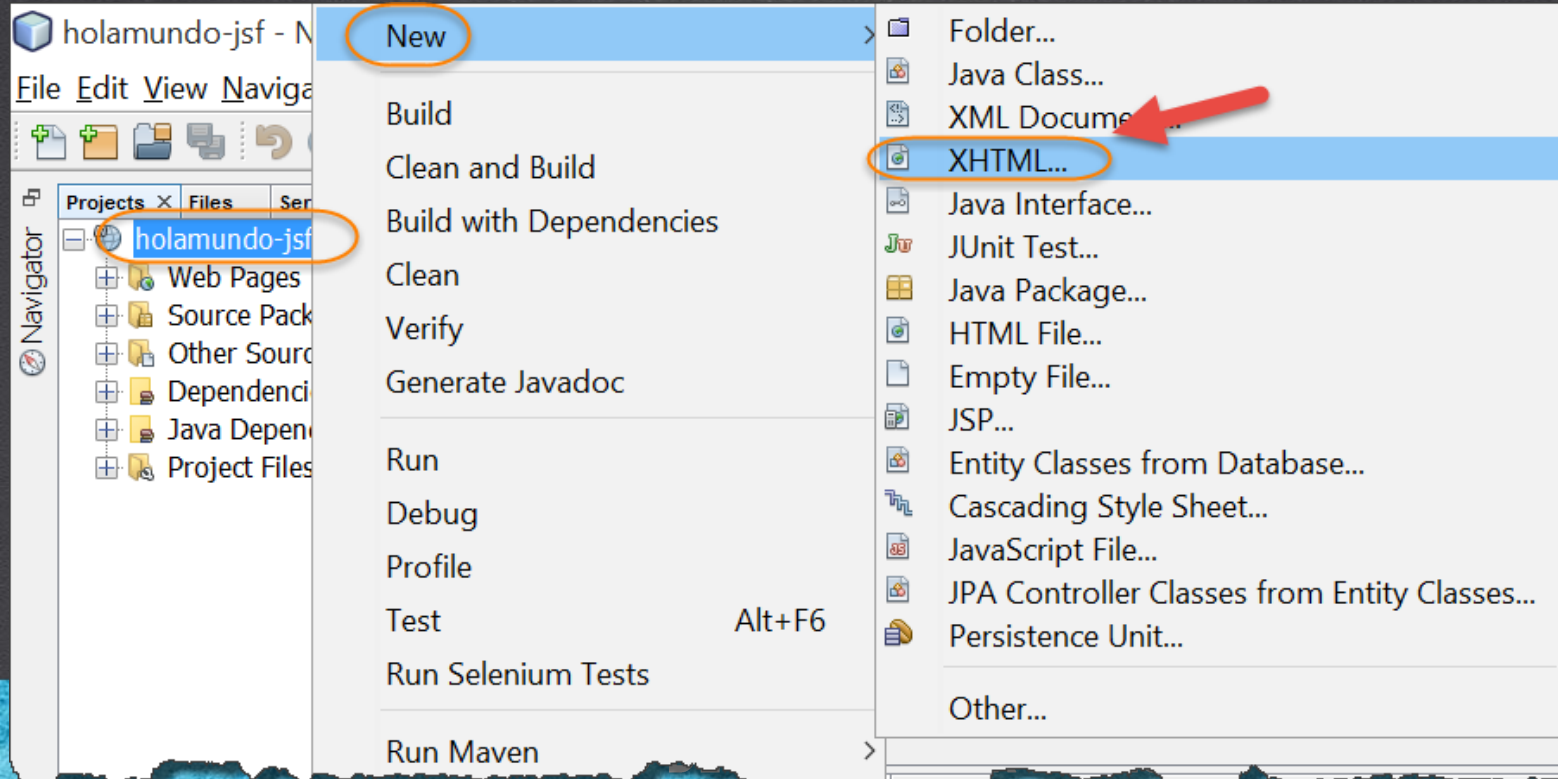
**CURSO DE JAVASERVER FACES**

www.globalmentoring.com.mx



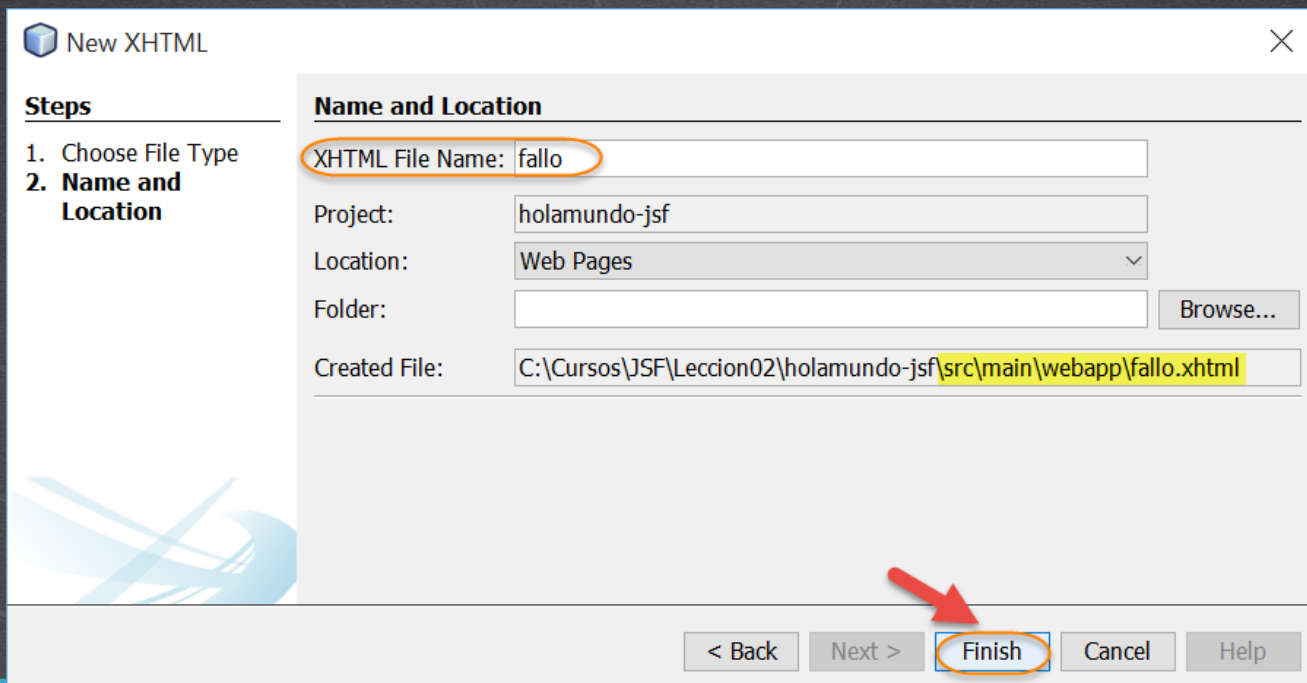
## PASO 6. AGREGAR UNA PÁGINA JSF PARA EL CASO DE "FALLO"

Creamos una pagina .xhtml para el caso de fallo.



## PASO 6. AGREGAR UNA PÁGINA JSF PARA EL CASO DE "FALLO"

Creamos una pagina .xhtml para el caso de fallo.



New XHTML

**Steps**

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

**Name and Location**

XHTML File Name: fallo

Project: holamundo-jsf

Location: Web Pages

Folder: Browse...

Created File: C:\Cursos\JSF\Leccion02\holamundo-jsf\src\main\webapp\fallo.xhtml

< Back Next > **Finish** Cancel Help

**CURSO DE JAVASERVER FACES**

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

# PASO 7. MODIFICAMOS EL CÓDIGO

## Archivo fallo.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">
  <h:head>
    <title>Fallo</title>
  </h:head>
  <h:body>
    <h:form>
      Lo sentimos #{candidato.nombre}, ya se han cubierto todas las vacantes.
      <br/>
      <h:commandLink action="index">Regresar</h:commandLink>
    </h:form>
  </h:body>
</html>
```

**CURSO DE JAVASERVER FACES**

www.globalmentoring.com.mx



## PASO 8. MODIFICAR LA PÁGINA INDEX.XHTML

- **Ubicar** en el contenido de la página index.xhtml, lo mostrado a continuación:

```
<h:commandButton value="Enviar" />
```

- **Sustituirlo** por el siguiente código:

```
<h:commandButton action="#{vacanteForm.enviar}" value="Enviar" />
```

# PASO 8. MODIFICAMOS EL CÓDIGO

## Archivo index.xhtml:

Dar click para ir al código

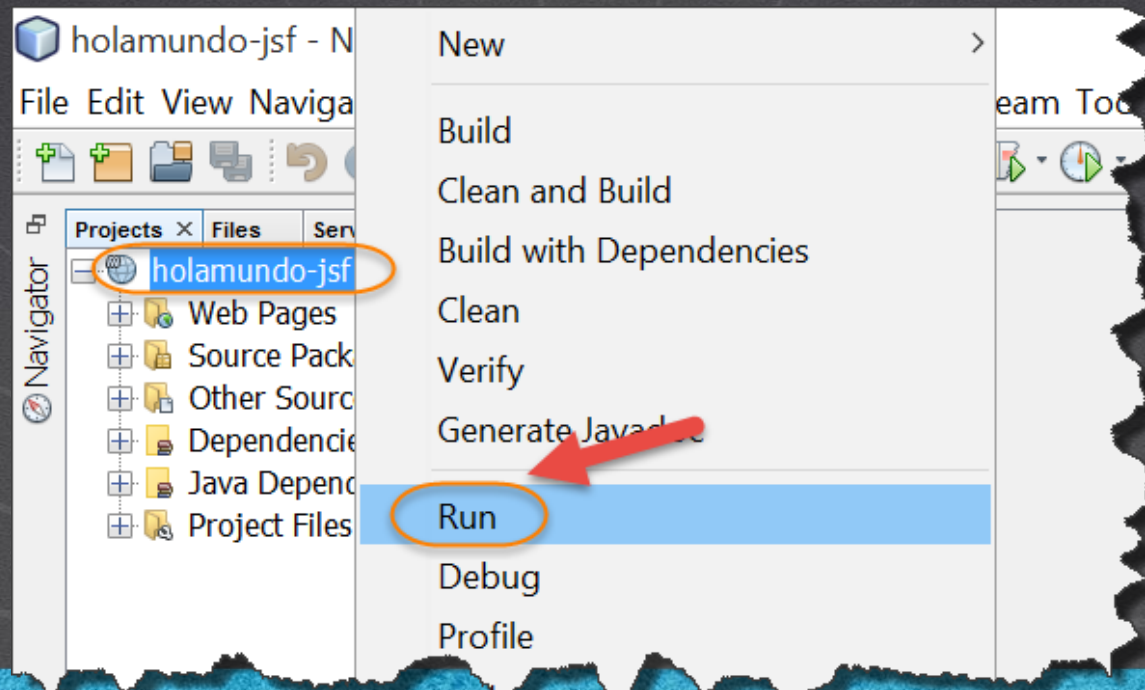
```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html">
  <h:head>
    <title>HolaMundo JSF</title>
  </h:head>
  <h:body>
    <h:form>
      <table>
        <tr>
          <td><h:outputLabel for="nombre" value="Nombre:" /></td>
          <td><h:inputText id="nombre" value="#{candidato.nombre}" /></td>
          <td><h:message for="nombre" /></td>
        </tr>
      </table>
      <h:commandButton action="#{vacanteForm.enviar}" value="Enviar" />
    </h:form>
  </h:body>
</html>
```

**CURSO DE JAVASERVER FACES**

www.globalmentoring.com.mx

# PASO 9. EJECUTAMOS EL PROYECTO

Ejecutamos el proyecto:



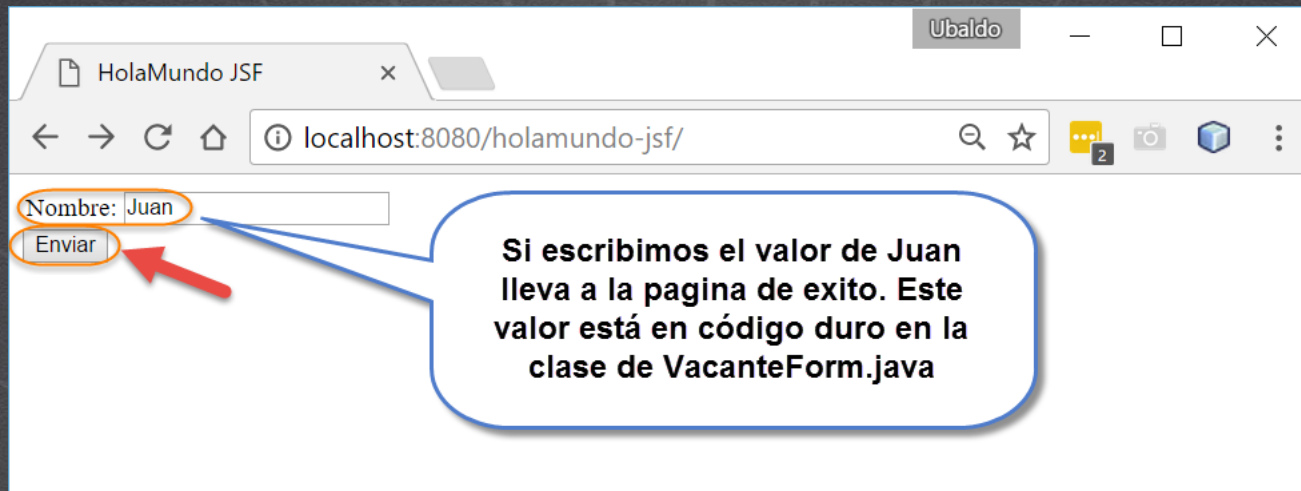
**CURSO DE JAVASERVER FACES**

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



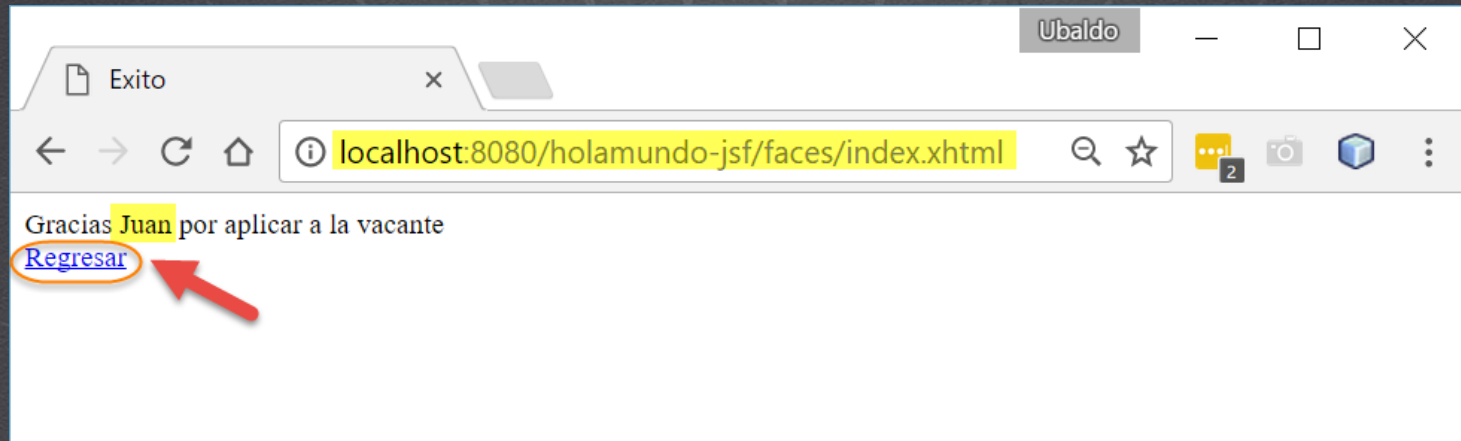
# PASO 9. EJECUTAMOS EL PROYECTO

Ejecutamos el proyecto. Probamos el caso de éxito, proporcionando el valor de Juan:



# PASO 9. EJECUTAMOS EL PROYECTO

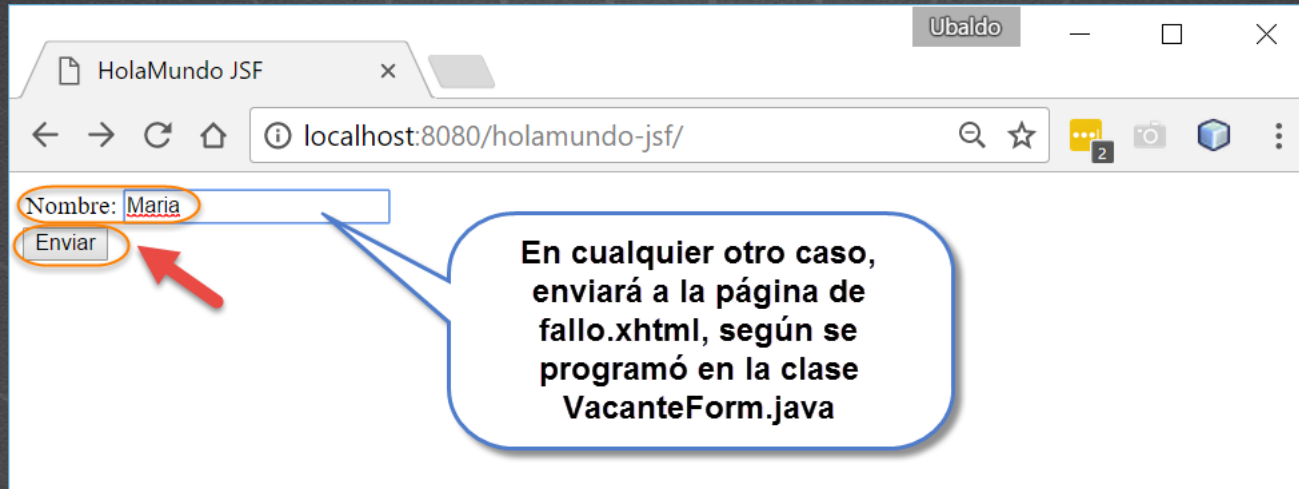
Ejecutamos el proyecto. Probamos el caso de éxito, proporcionando el valor de Juan:





# PASO 9. EJECUTAMOS EL PROYECTO

Ejecutamos el proyecto. Probamos el caso de fallo proporcionando un nombre distinto a Juan.





# PASO 9. EJECUTAMOS EL PROYECTO

Ejecutamos el proyecto. Probamos el caso de fallo proporcionando un nombre distinto a Juan.



# CONCLUSIÓN DEL EJERCICIO

- Con este ejercicio hemos agregado el concepto de Backing ManagedBean de JSF. Recordemos que este concepto es un Bean de JSF que sí tendrá lógica de navegación y alguna otra lógica de la aplicación respecto a la capa de presentación. A diferencia del ManagedBean que no lo utilizaremos para este objetivo, sino solo para asociar propiedades de una clase bean.
- Además agregamos dos páginas jsf para ejecutar dos casos, uno de éxito y otro de fallo. Si el nombre proporcionado es igual a “Juan”, entonces se muestra la página de éxito, de lo contrario se muestra la página de fallo.
- En este caso, estamos aplicando la navegación por default de JSF, sin necesidad de agregar los casos de navegación al archivo faces-config.xml, ya que el nombre de la página es igual al nombre que regresa el flujo de navegación.



**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)