

# CURSO DE JAVASERVER FACES

## EJERCICIO

### JSF Y CDI (CONTEXT AND DEPENDENCY INYECTION)



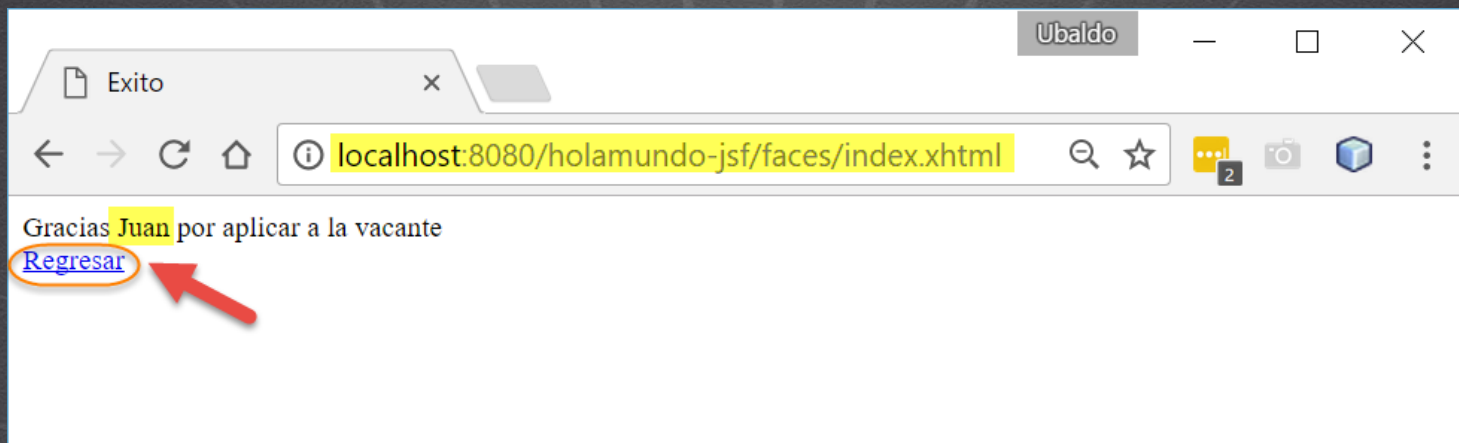
Experiencia y Conocimiento para tu vida

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 CDI al utilizar los ManagedBean. El resultado final es como sigue:





# OBJETIVO DEL EJERCICIO

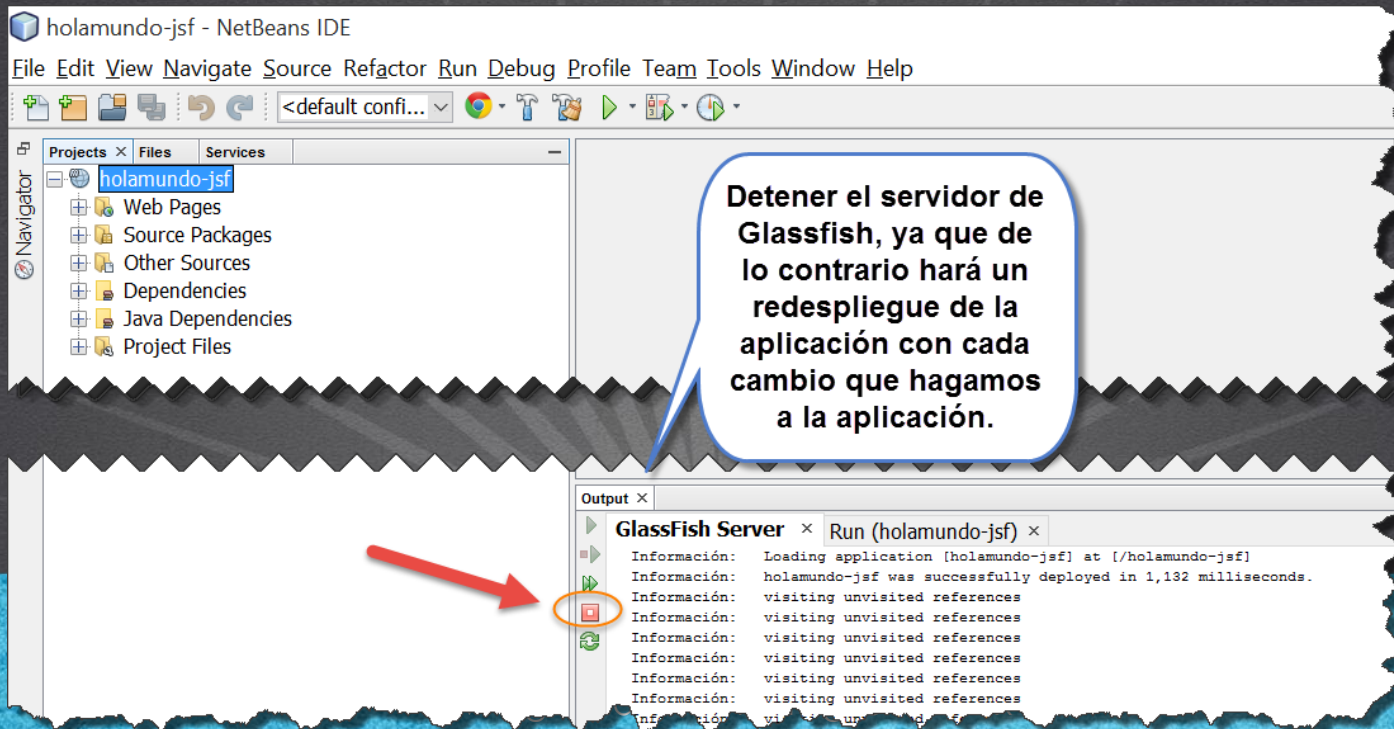
Introduciremos el concepto de CDI (Context and Dependency Injection) utilizando las anotaciones de Java EE, en lugar de las anotaciones de JSF.

Las anotaciones de CDI de Java EE son más sofisticadas que las inyección de JSF, sin embargo se requiere de un un servidor full de Java, como Glassfish o WildFly (antes Jboss).

Sin embargo, si nuestra aplicación la desplegamos en un servidor Web de Java como Tomcat o Jetty, es decir, solo haremos uso de Servlets y no de otro tipo de componentes de Java EE se recomienda utilizar las anotaciones de JSF como estudiamos en los ejercicios anteriores.

# ANTES DE EMPEZAR

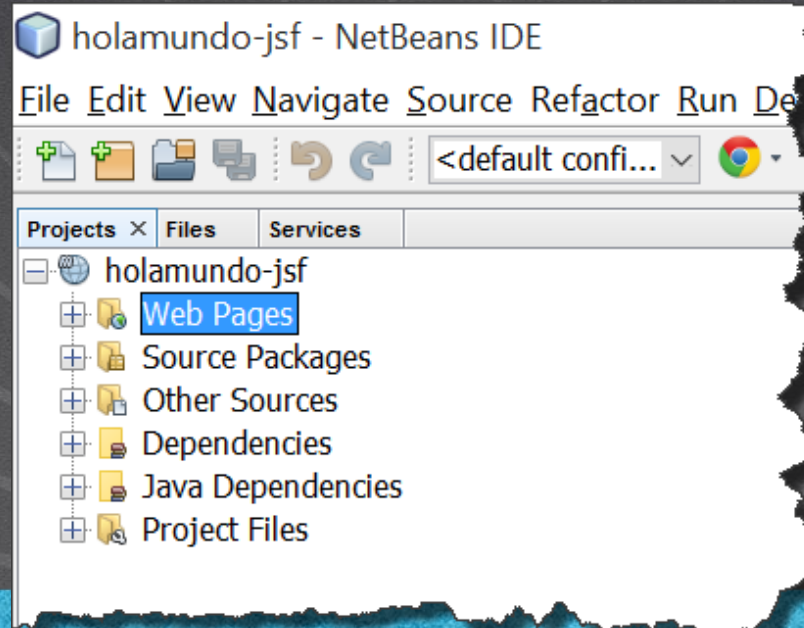
Continuamos trabajando con el proyecto de la lección anterior. Si está levantado el servidor de Glassfish, lo detenemos, ya que de lo contrario con cada cambio que hagamos se hará nuevamente un despliegue de la aplicación.





# 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)

# MODIFICAMOS LOS BEANS DE JSF

Vamos a modificar ahora los beans de JSF para que en lugar de utilizar la anotación `@ManagedBean` del paquete `import javax.faces.bean`, ahora vamos a utilizar la anotación `@Named` del paquete `javax.inject`.

También en lugar de utilizar la anotación `@ManagedProperty(value="#{candidato}")`, la vamos a sustituir simplemente por `@Inject` y ya no tenemos necesidad de especificar el nombre del bean a inyectar, ya que CDI buscará automáticamente inyectar el bean por tipo de dato, además de que no se requiere obligatoriamente agregar los métodos getters o setters para que se pueda inyectar la propiedad respectiva.

Por último, el manejo de alcances (scopes) también los utilizaremos con CDI en lugar de JSF, por lo que el alcance de `RequestScope` o cualquier otro los utilizaremos del paquete `javax.enterprise.context`, en lugar de `javax.faces.bean` que solo funciona para JSF y no para más componentes de Java EE.



# PASO 2. MODIFICAMOS EL CÓDIGO

## Archivo Candidato.java:

Dar click para ir al código

```
package beans.model;

import javax.inject.Named;
import javax.enterprise.context.RequestScoped;

@RequestScoped
@Named
public class Candidato {

    private String nombre = "Introduce tu nombre";

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

}
```

# PASO 3. MODIFICAMOS EL CÓDIGO

## Archivo VacanteForm.java:

Dar click para ir al código

```
package beans.backing;

import beans.model.Candidato;
import javax.inject.Named;
import javax.enterprise.context.RequestScoped;
import javax.inject.Inject;

@RequestScoped
@Named
public class VacanteForm {

    @Inject
    private Candidato candidato;

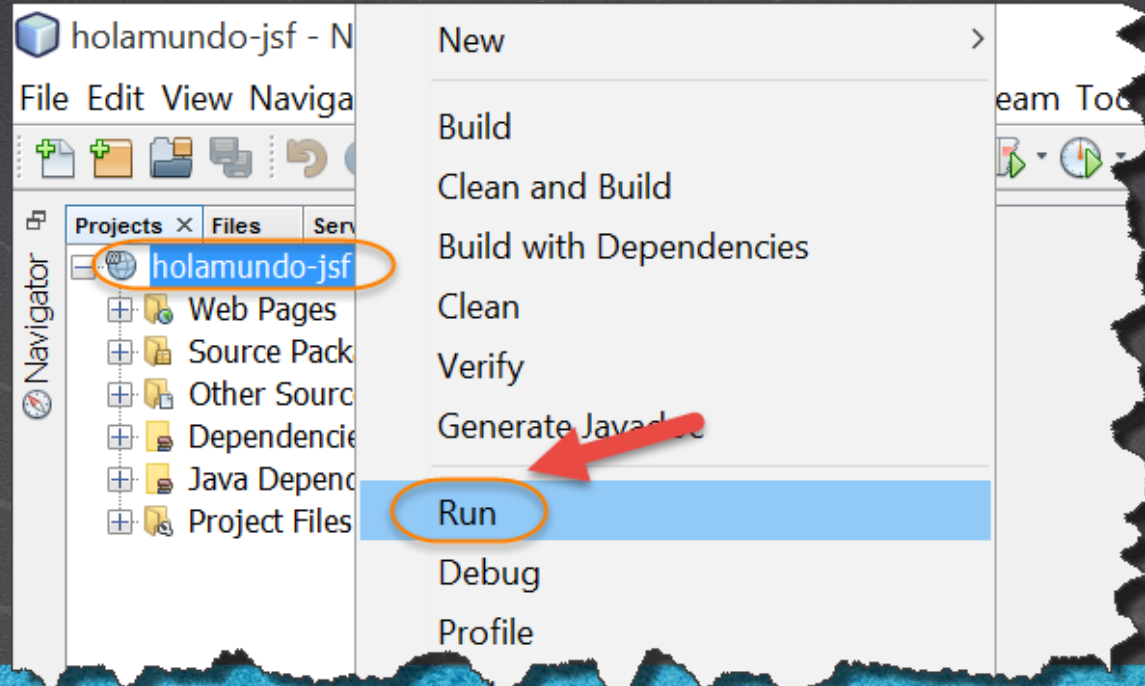
    public void setCandidato(Candidato candidato) {
        this.candidato = candidato;
    }

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



# PASO 4. EJECUTAMOS EL PROYECTO

Ejecutamos el proyecto:

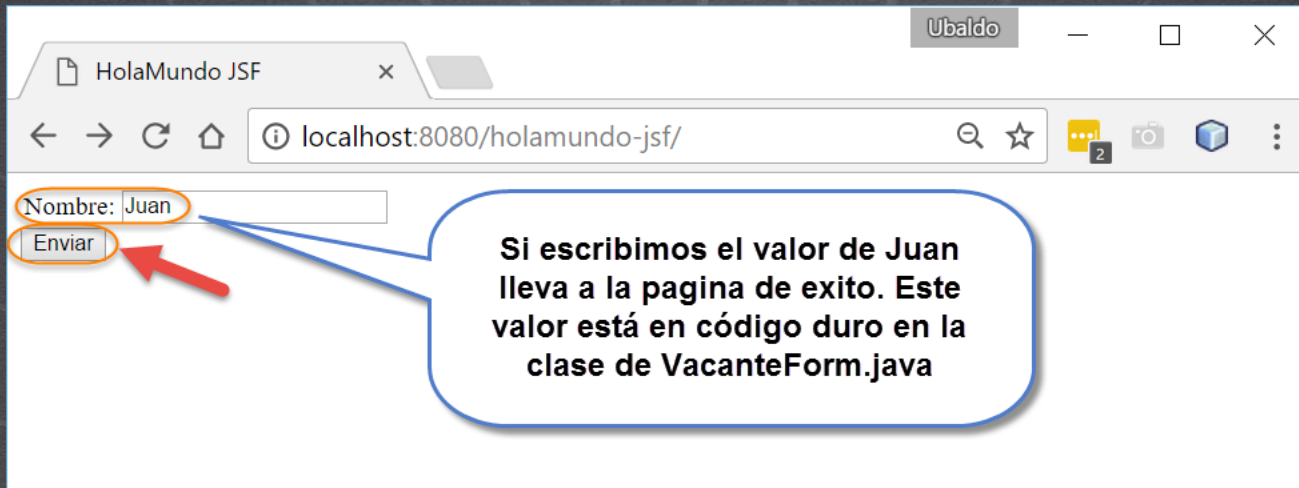


**CURSO DE JAVASERVER FACES**

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

## PASO 4. EJECUTAMOS EL PROYECTO

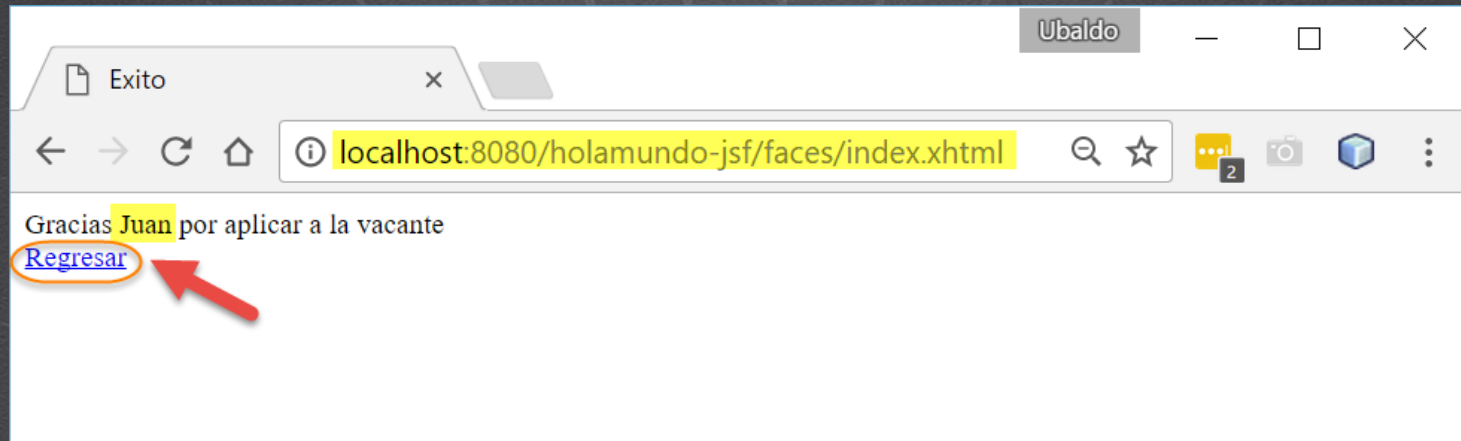
Ejecutamos el proyecto. El resultado debe ser el mismo ya que solamente cambiamos el tipo de anotación de `@ManagedBean` a `@Inject` en nuestros beans de JSF:





# PASO 4. EJECUTAMOS EL PROYECTO

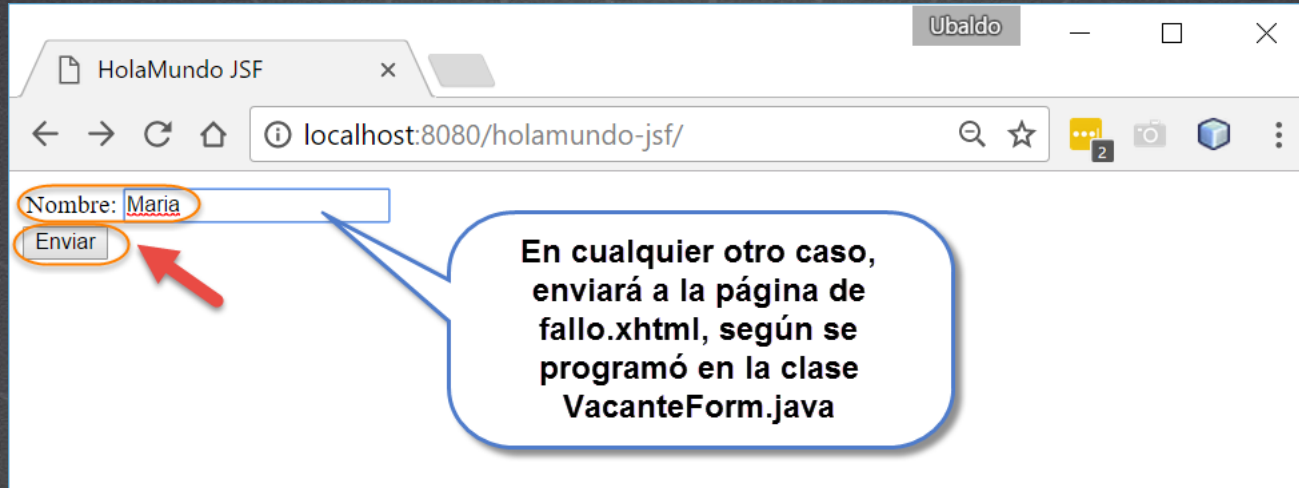
Ejecutamos el proyecto. Probamos el caso de éxito, proporcionando el valor de Juan:





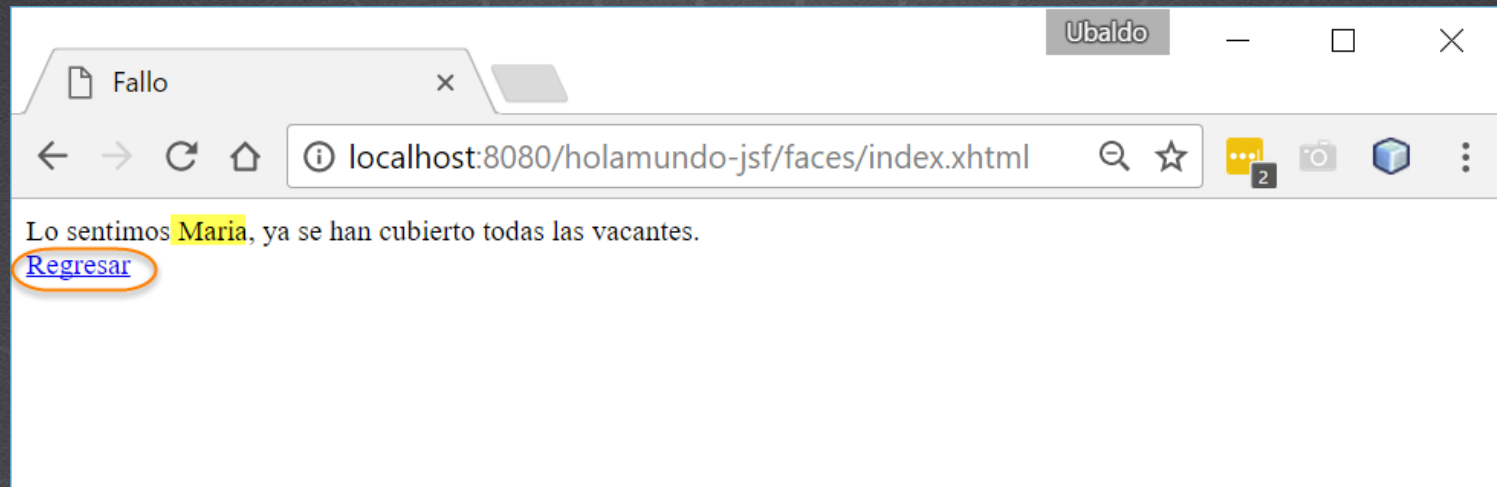
# PASO 4. EJECUTAMOS EL PROYECTO

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



# PASO 4. EJECUTAMOS EL PROYECTO

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





# CONCLUSIÓN DEL EJERCICIO

- Con este ejercicio hemos utilizado la configuración de CDI para JSF. Como comentamos esto es recomendable si vamos a utilizar un servidor full de Java como Glassfish, WildFly o cualquier otro servidor Java EE.
- Este es el tipo de configuración que utilizaremos a lo largo del curso ya que vamos a utilizar Glassfish como servidor de aplicaciones.
- El archivo beans.xml ya no es necesario agregarlo en la versión Java EE 7. Pero si se está utilizando la versión Java EE 6 se debe crear un archivo beans.xml dentro de la carpeta de WEB-INF para que los beans de JSF sean reconocidos:
- <http://icursos.net/cursos/JSF/Leccion02/holamundo-jsf-cdi/beans.txt>
- Con esto concluimos el tema de ManagedBeans en JSF.



**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)