# 0110510-jsf-logo.tiff

**Ejercicios Básicos**

## Ejercicio #1 Inicio de un proyecto con JSF

**Objetivo**: Crear un proyecto Maven que pueda ser desplegado en un servidor de aplicaciones (Jboss preferentemente) que implemente la tecnología JSF.

El resultado será similar al siguiente:

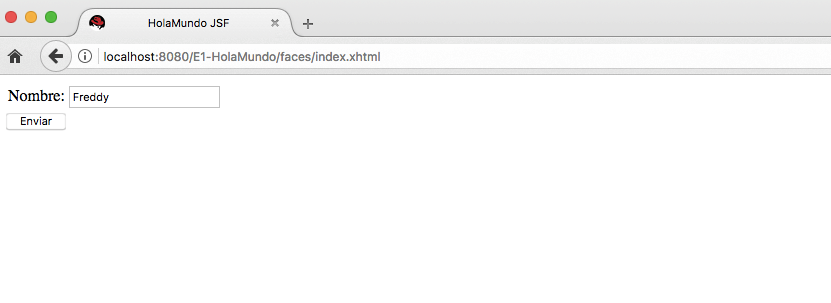


Ilustración 1 - Resultado esperado del ejercicio 1

Pasos a seguir:

1. Creación de un proyecto Maven con ayuda del IDE NetBeans como se muestra a continuación:

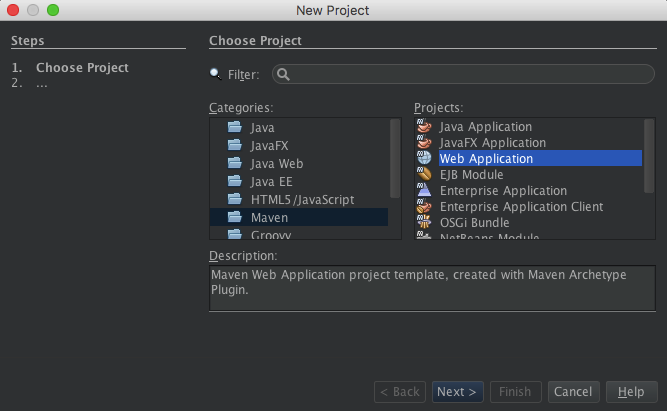


Ilustración 2 - Creación del proyecto con el arquetipo Maven Archetype Plugin

*Nota: este arquetipo ha sido utilizado anteriormente y también puede implementarse por medio de otros IDE´s como Eclipse o por medio de la terminal utilizando los comandos de Maven vistos al inicio del curso.*

1. Configurar el proyecto Maven proporcionando Nombre, ArtifactId, GroupId y la versión

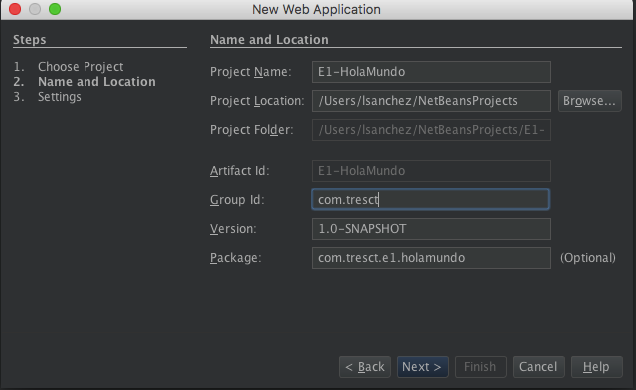


Ilustración 3 - Configuración de proyecto Maven

1. Elija el servidor de aplicaciones web de su preferencia:

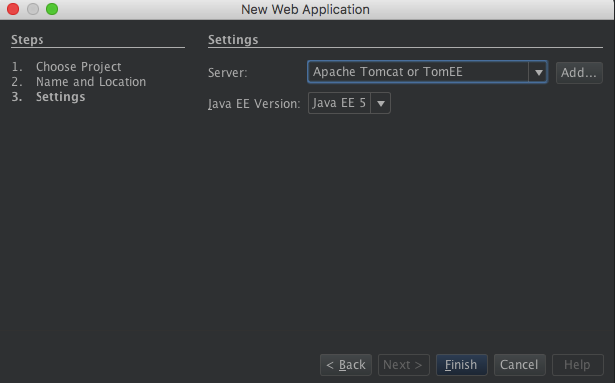


Ilustración 4 - Elección de servidor web

Para los ejercicios que se presentan en el curso utilizaré el servidor Jboss manejado desde la terminal del sistema. Los ejercicios pueden desplegarse en Jboss, GlasFish, Apache Tomcat, etc.

1. Revisar la arquitectura creada por la herramienta Maven:

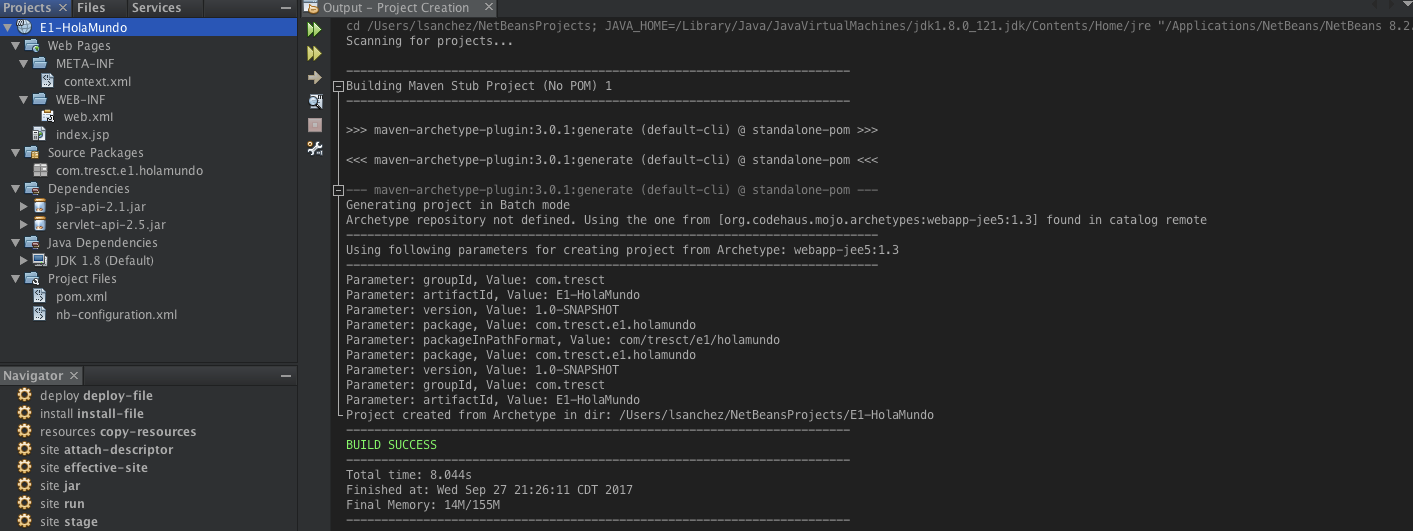


Ilustración 5 - Arquitectura Web creada por Maven

1. Abrir el archivo ***pom.xml*** y agregar las siguientes dependencias:

* ***jsf-impl.jar***
* ***jsf-api.jar***
* ***javax.servlet.jar***
* ***javax.servlet.jsp***

Obteniendo el siguiente resultado:



Ilustración 6 - Dependencias en pom.xml

Nota: Podemos agregar las dependencias de dos formas:

* Escribiendo el código en el pom.xml realizando búsquedas en MvnRepository:

<https://mvnrepository.com/artifact/com.sun.faces/jsf-api/2.2.14>

* Clic derecho sobre el código del pom.xml -> Insert code -> Dependency

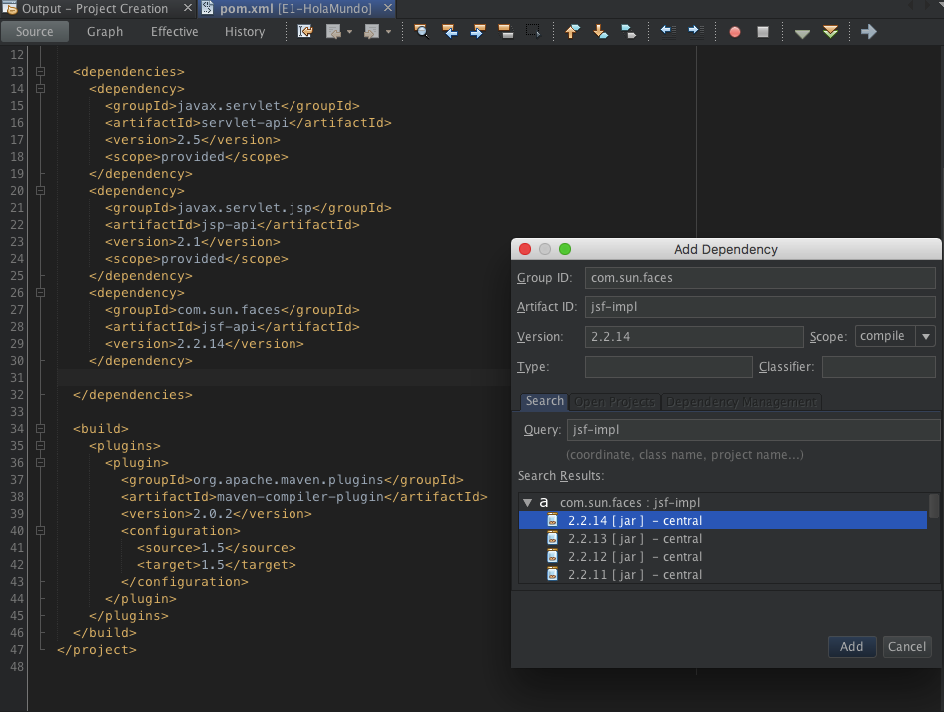


Ilustración 7 - Agregando dependencias desde el asistente

Dar clic en el botón **Clean and Build**  para descargar las bibliotecas necesarias especificadas en el pom.xml y observar los resultados como se muestra a continuación:

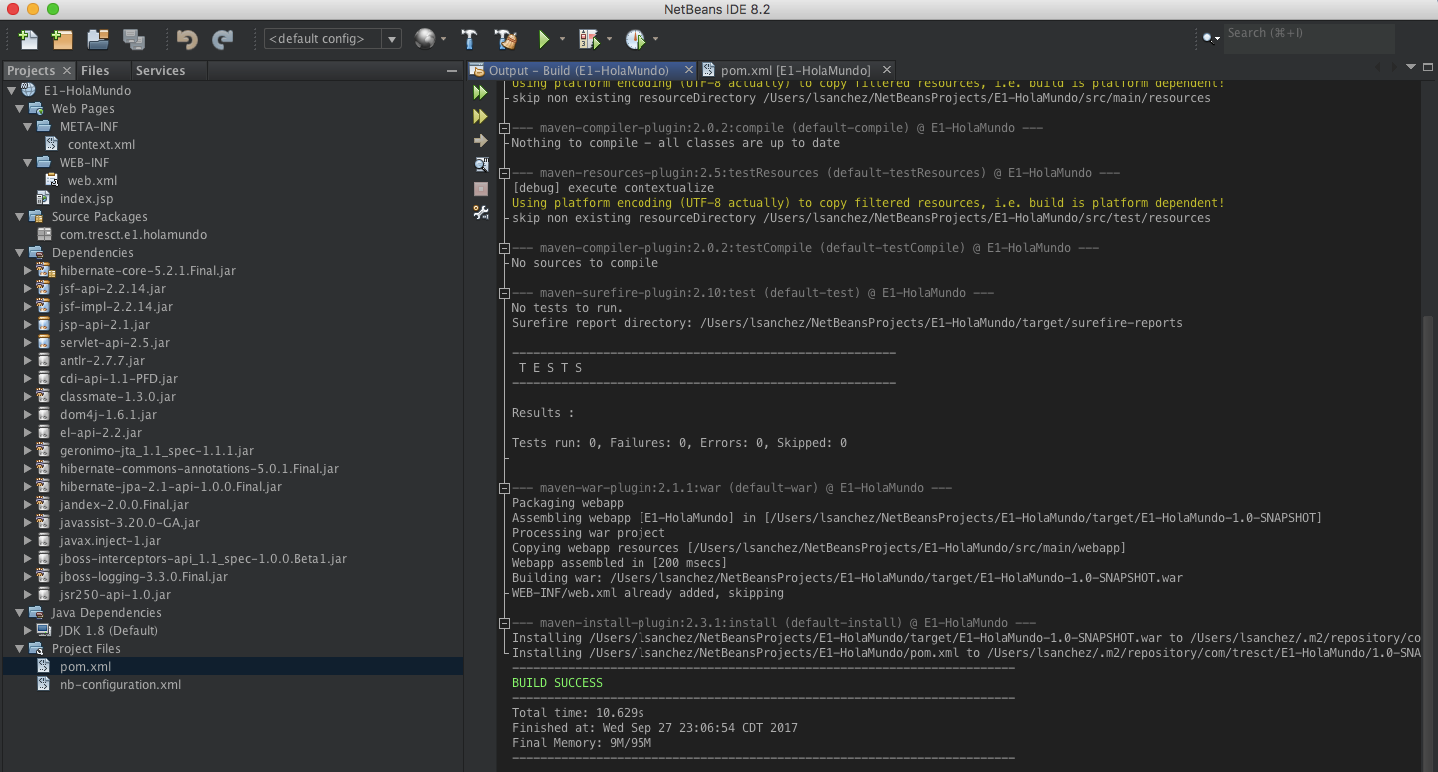


Ilustración 8 - Bibliotecas agregadas al proyecto

1. Remplazar el index.jsp por un index.xhtml

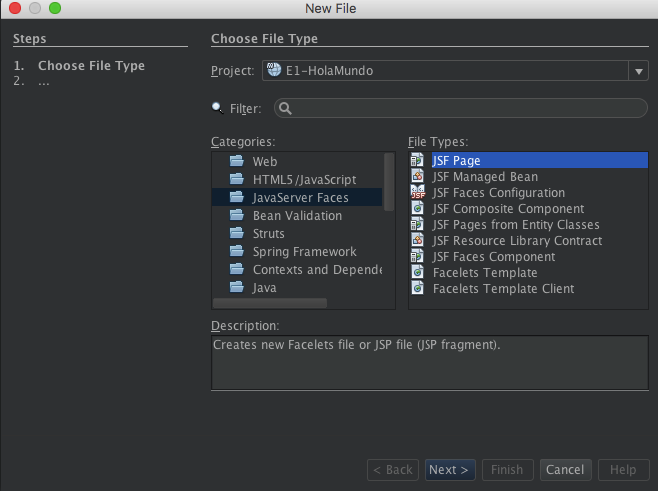


Ilustración 9 - Nuestro primer Facelet

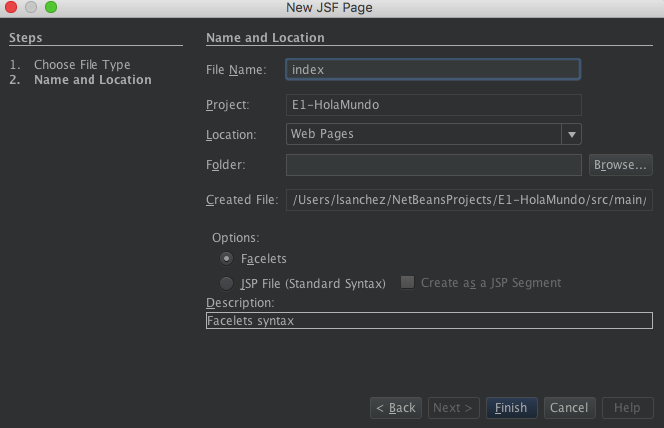
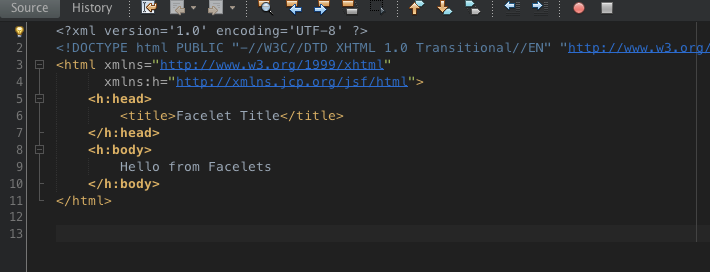


Ilustración 10 - Configuración del Facelet

1. Descripción de nuestro index.xhtml



1. Remplazar el código del index.html por el siguiente:

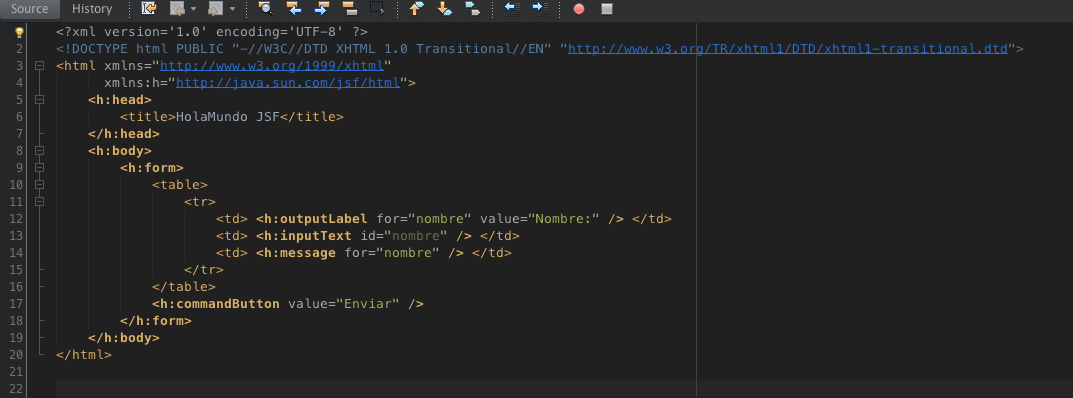


Ilustración 11 - Nuevo index.xhtml con un formulario JSF

|  |  |  |
| --- | --- | --- |
|  |  |  |

1. <? xml version = '1.0'
2. encoding = 'UTF-8' ?> < !DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd" > < html xmlns = "http://www.w3.org/1999/xhtml"
4. 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
5. for = "nombre"
6. value = "Nombre:" / > < /td> < td > < h: inputText id = "nombre" / > < /td> < td > < h: message
7. for = "nombre" / > < /td> < /tr>  < /table> < h: commandButton value = "Enviar" / > < /h:form> < /h:body> < /html>
8. Limpiar y construir el proyecto:

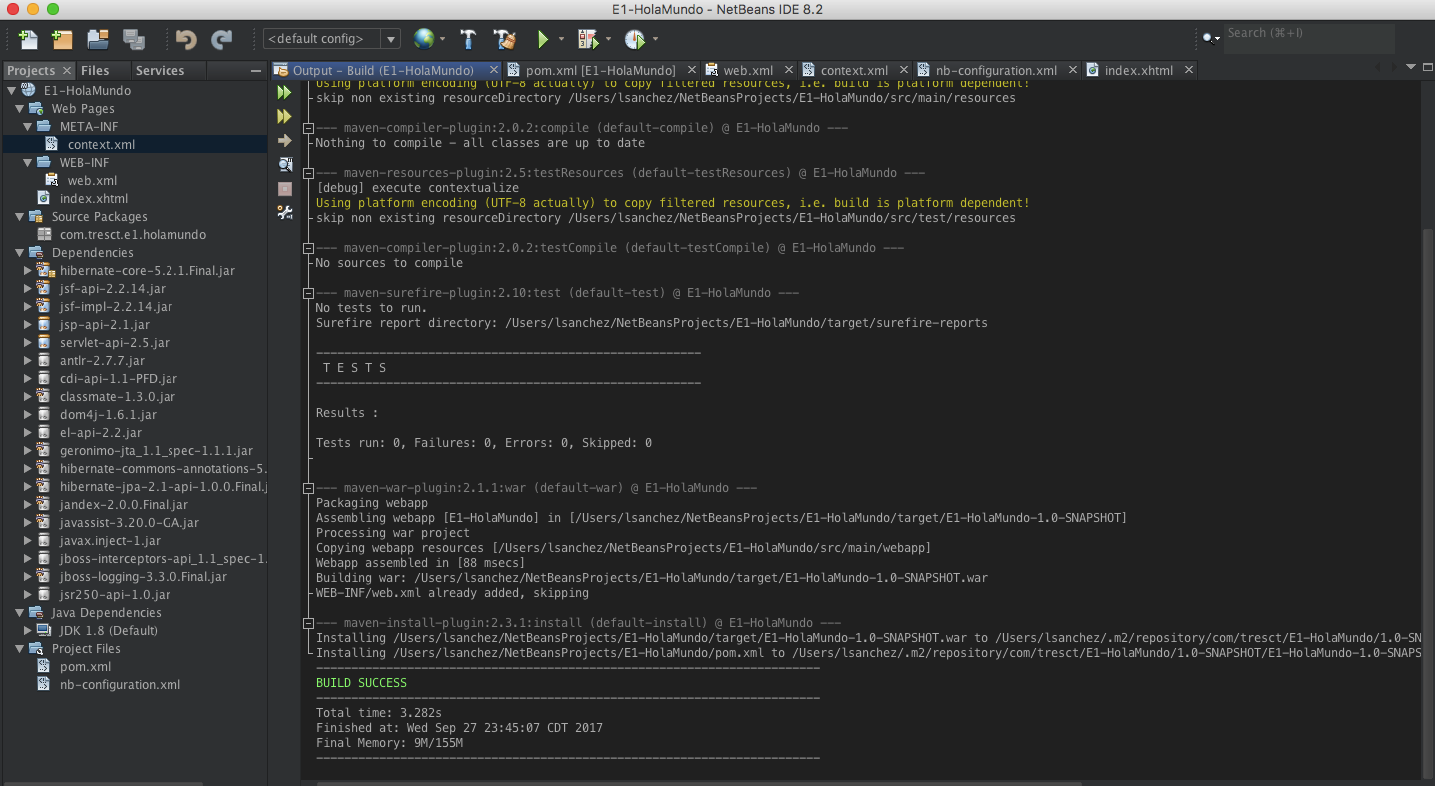


Ilustración 12 -Construcción del proyecto

1. Hacer el despliegue de la aplicación WEB.

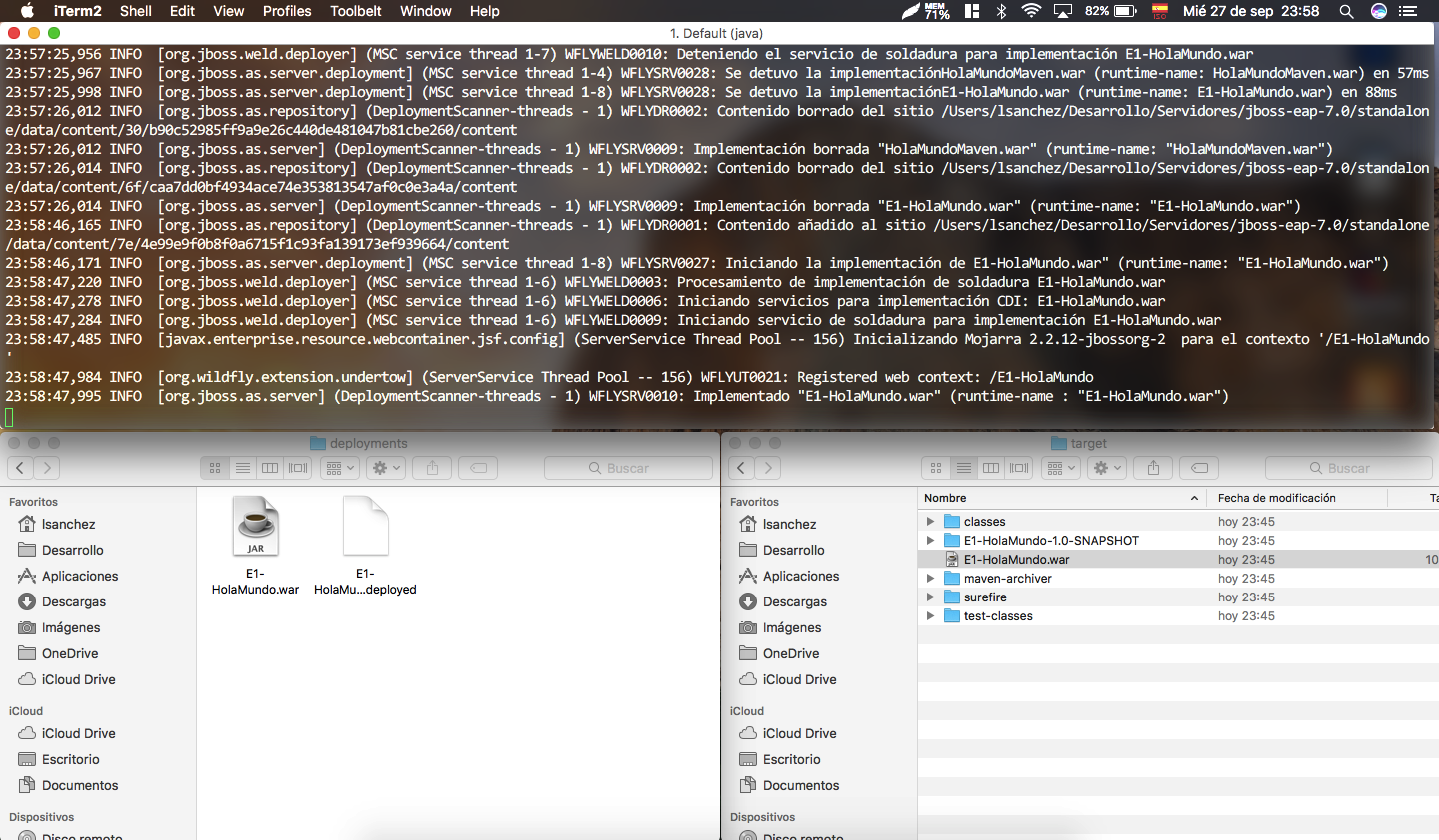


Ilustración 13 - Aplicación desplegada

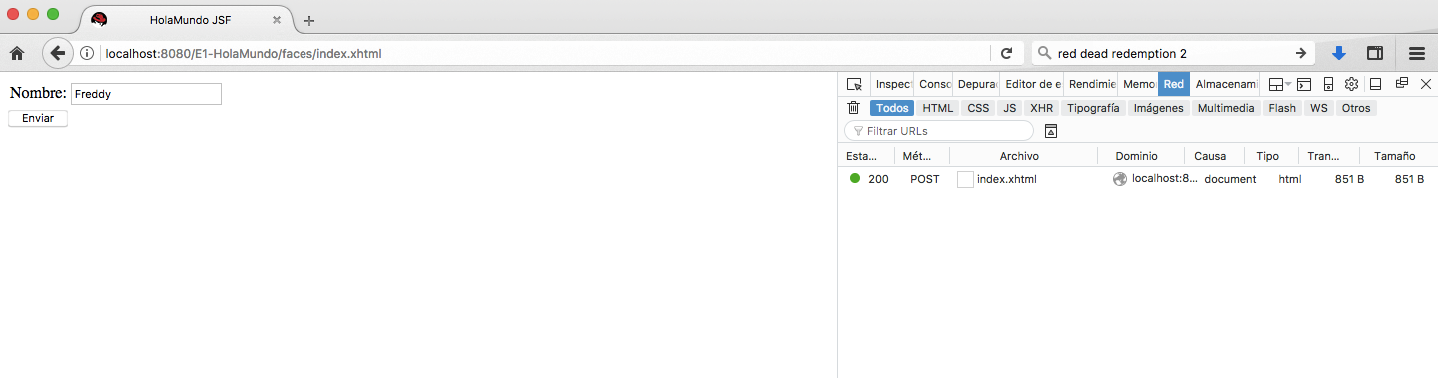


Ilustración 14 - Verificación del funcionamiento de nuestra aplicación en Mozilla Firefox

## Ejercicio #2 Model Managed Bean

* Objetivo: Introducir el concepto de este componente en nuestra aplicación y ubicar en que parte del MVC trabaja.
* Utilizar el concepto de anotaciones basado en las características de JSF 2.0 para declarar el Bean Administrado.
* Comprender la relación del Managed Bean con la Pagina JSF para desplegar valores a las vistas

Pasos a seguir

1. Agregar un Bean administrado al proyecto:

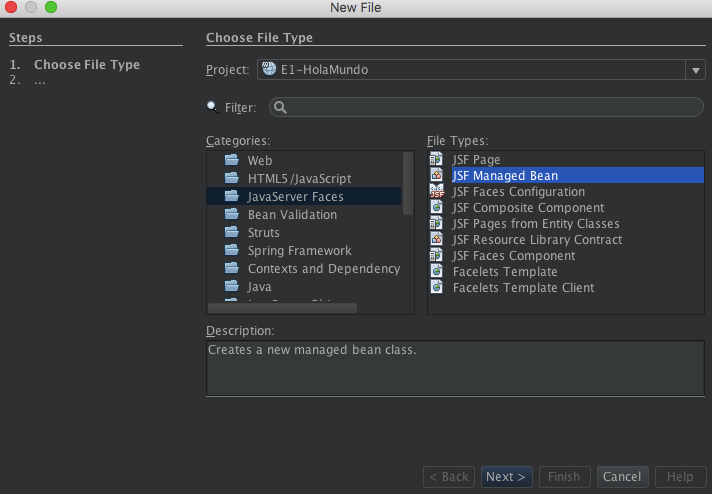


Ilustración 15 - Agregando un Bean Administrado por JSF

1. Configurarlo de la siguiente manera:

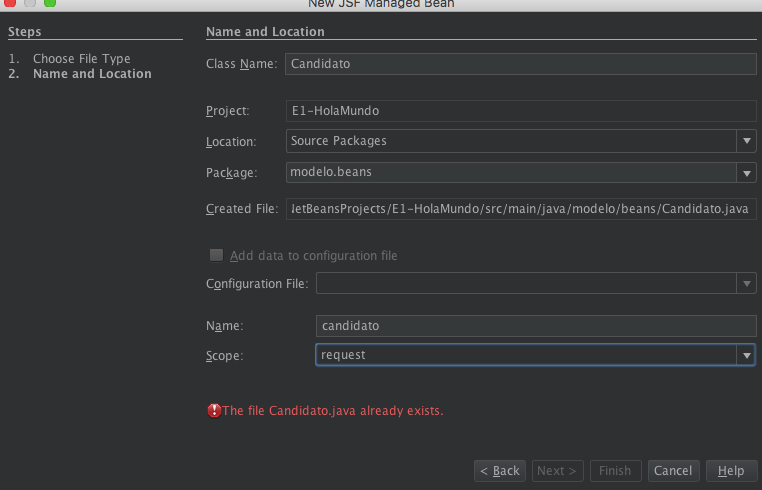
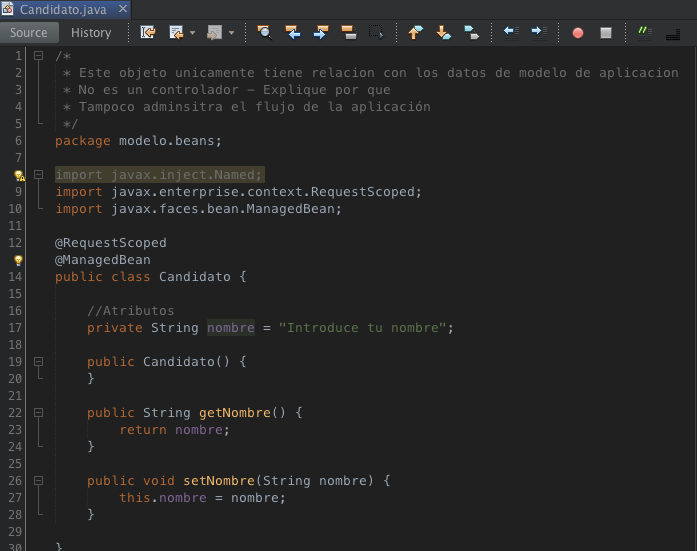


Ilustración 16 - Configuración del Bean Administrado

1. Escribir el siguiente código en el cuerpo del Bean Administrado



1. Enlazamos el Bean Administrado con la pagina JSF.

Ubicar el contenido de la página ***index.xhtml*** :

***<h:inputText id=”nombre” />***

***Sustituir con lo siguiente:***

***<h:inputText id=”nombre” value=”#{candidato.nombre}” />***

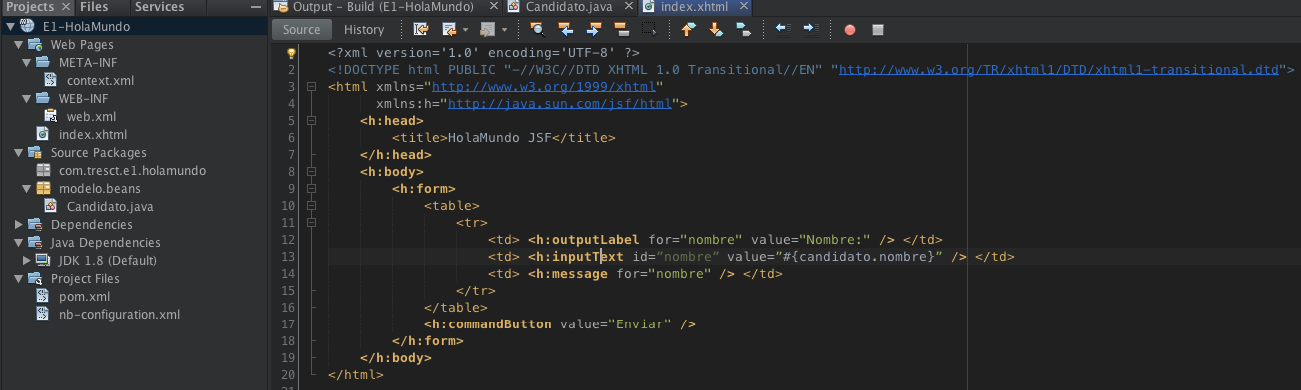
******

Ilustración 17 - Código del **index.xhtml**

1. Ejecutar la aplicación y observar el resultado:

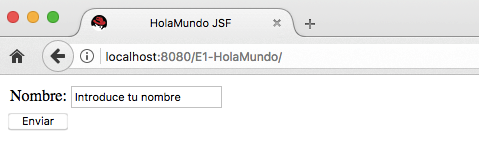


Ilustración 18 - Enlazando valores del Bean a la vista por medio de EL...

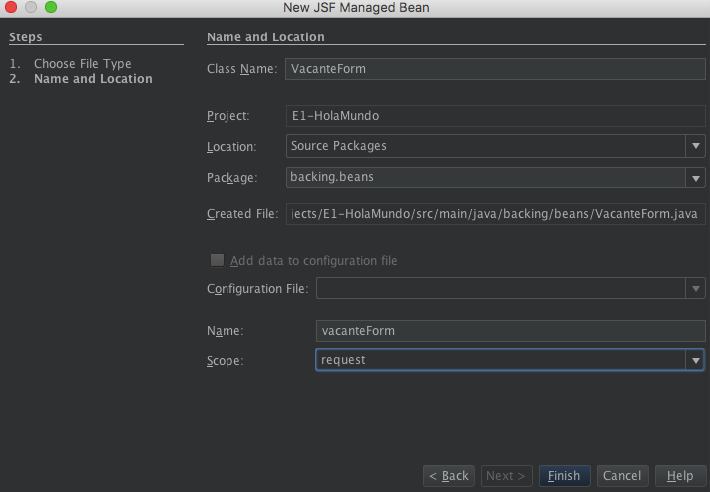
## Ejercicio # 3 Backing Beans

Objetivo:

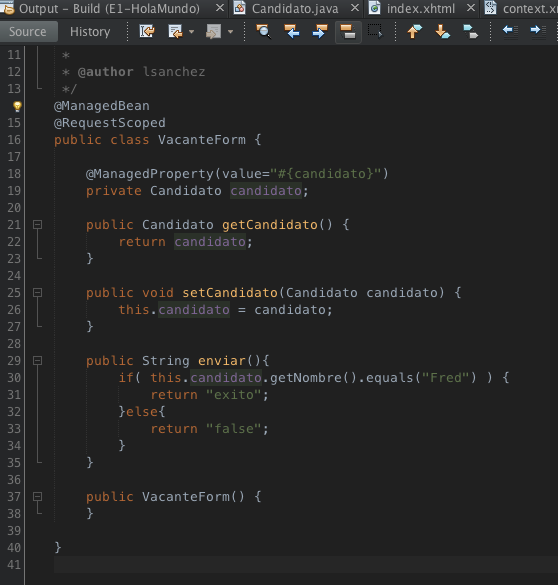
* Introducir el concepto de Backing Beans a nuestra aplicación.
* Utilizar el concepto de anotaciones para inyectar dependencias entre Managed Beans
* Adicionalmente agregar paginas JSF con las respectivas reglas de navegación con notación JSF 2.0
* Asociar funcionalidad al botón Enviar de nuestra página index.xhtml para ejecutar el flujo de la aplicación.

Pasos a seguir:

1. Agregar un JSF Managed Bean:

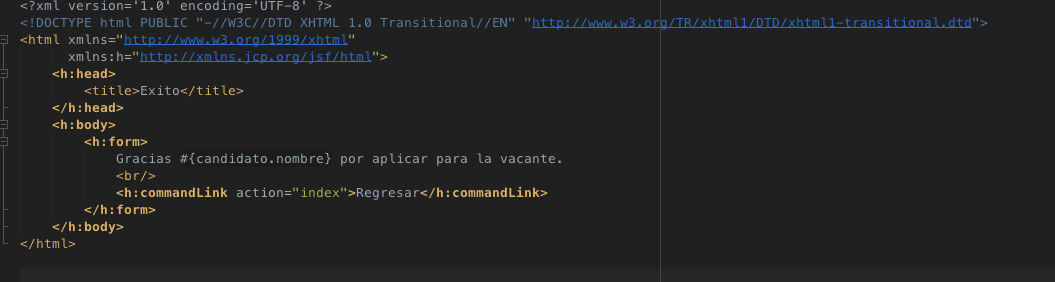


1. Escribir el siguiente código:

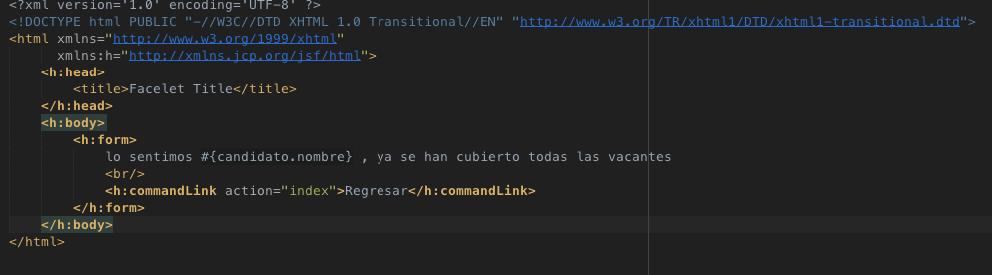


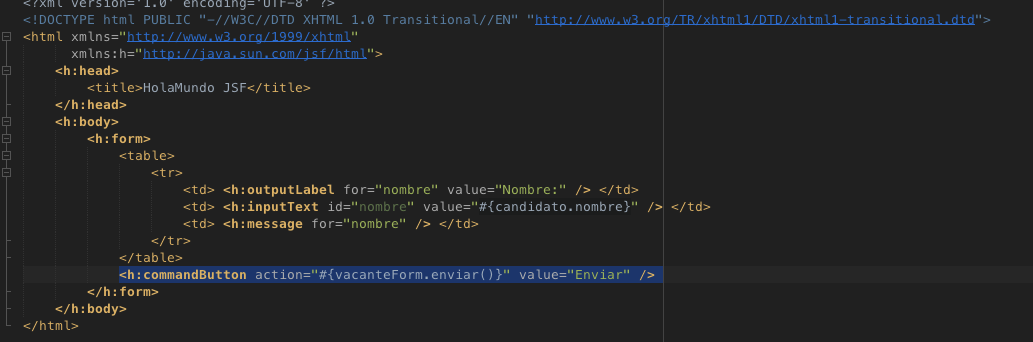
1. Agregar mensajes para el caso de éxito y fallo del flujo de la aplicación:

* Mensaje de éxito: Agregar un nuevo xhtml con el siguiente código:



1. Agregamos el xhtml para el fallo:



1. Finalmente se modifica el index de la siguiente manera:
2. Comente sus resultados:

