

Introducción al desarrollo de aplicaciones Web con JAVA

Guía rápida

Introducción al desarrollo de aplicaciones Web

Guía rápida

Este documento te guiará por los primeros pasos para desarrollar aplicaciones web usando NetBeans IDE. Te mostrará como crear una aplicación web simple, su despliegue en el servidor y su presentación en un navegador. La aplicación utiliza una pagina JavaServer Pages (JSP) para que ingreses tu nombre. Luego utiliza un componente JavaBeans para que el nombre persista durante toda la sesión HTTP, y devuelve el nombre en la segunda página JSP.

Para completar esta guía, necesitaras el siguiente software y recursos.

Software	Versión
NetBeans IDE	Instalación Web o Java EE versión 6.0 en adelante
Java Developer Kit (JDK)	Versión 5 en adelante
Servidor de Aplicaciones GlassFish ó Contenedor de Servlets Tomcat	V2 Versión 6.x

Nota

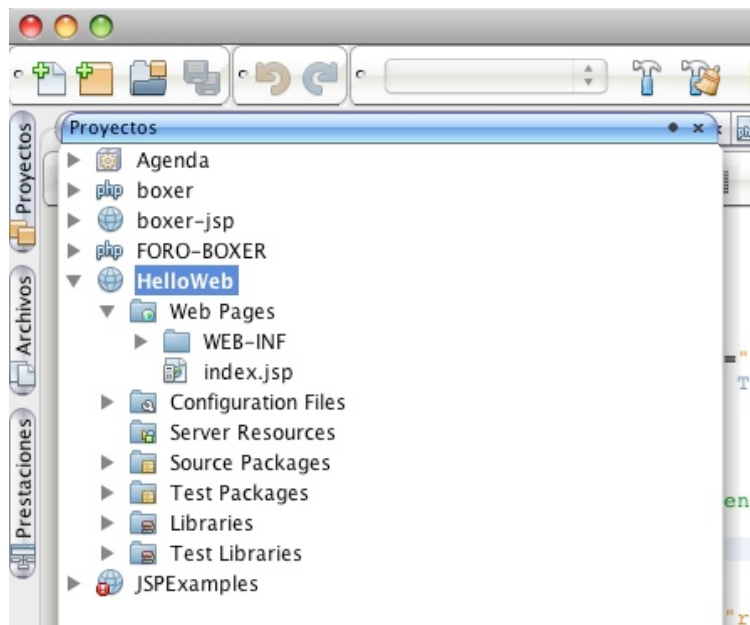
- Las instalaciones Web y Java EE te permitirán que instales, de manera opcional, el servidor de aplicaciones GlassFish V2 y el contenedor de servlets Apache Tomcat 6.0.x. Debes instalar una de las aplicaciones anteriores para realizar el tutorial.

Configurando un Proyecto de Aplicación Web

- 1.- En Netbeans selecciona **File** -> **New Project** (Ctrl-Shift-N) del menú principal. En Categories, selecciona **Web**. Bajo Projects, selecciona **Web Application** entonces da click en Next.

- 2.– Introduce **HelloWeb** en el campo **Project Name**. Observa que el Context Path se convierte en **/HelloWeb**.
- 3.– Especifica en **Project Location**, cualquier directorio en tu PC.
- 4.– Da click en Next y selecciona el servidor en el cual vas a desplegar tu aplicación. Solamente se listaran servidores que estén registrados en el IDE.
- 5.– En el panel de Frameworks, da click en Finish para crear el proyecto.

El IDE crea la carpeta del proyecto **/Applications/MAMP/htdocs/HelloWord** (para este ejemplo). La carpeta del proyecto contiene todas las fuentes y metadata del proyecto, como el script de compilación **Ant**. El proyecto HelloWeb se abre en el IDE. La página de bienvenida, **index.jsp**, se abre en el Source Editor de la ventana principal. Podras ver la estructura de archivos del proyecto en la ventana Files (CTRL-2), y su estructura lógica en la ventana Projects (CTRL-1):



Creando y Editando los Archivos Fuente de la Aplicación Web

La creación y edición de archivos fuente es la función más importante que proporciona NetBeans. NetBeans proporciona una amplia variedad de herramientas que pueden complementar el estilo personal de cualquier desarrollador, aun si prefieres escribir todo a mano o si esperas que el IDE genere el código por ti.

Creando un Paquete Java y un Archivo Fuente Java

- 1. En la ventana **Projects**, expande el nodo **Source Packages**. Nota que el nodo Source Packages solo contiene un nodo por defecto vacío.
- 2. Da click derecho en el nodo **Source Package** y seleccione **New -> Java Class**. Ingresa **NameHandler** en el campo **Class Name** y escribe **org.mypackage.hello** en el campo Package. Da click en Finish. Nota que el nuevo archivo **NameHandler.java** se abre en Source Editor.
- 3. En NameHandler.java, escribimos la siguiente clase:

```
package org.mypackage.hello;

/**@author BOXER*/
public class NameHandler {

    private String name;

    /** Creamos una nueva instancia de NameHandler */
    public NameHandler() {
        name = null;
    }

    /**@return the name*/
    public String getName() {
        return name;
    }

    /**@param name the name to set*/
    public void setName(String name) {
        this.name = name;
    }
}
```

Existe una forma más rápida de crear esta clase utilizando las herramientas de NetBeans

- 3. En Source Editor, declaramos una variable **String** escribiendo la siguiente línea directamente debajo de la declaración de la clase:

```
private String name;
```

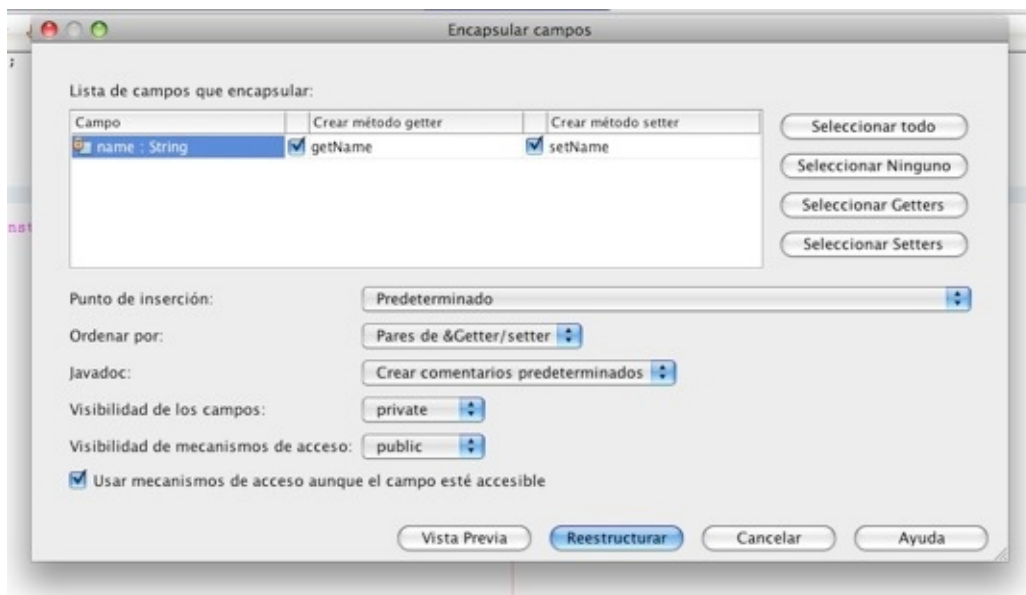
- 4. Añadimos el siguiente constructor a la clase:

```
public NameHandler() {
    name = null;
}
```

```
}
```

Generando Métodos Getter y Setter

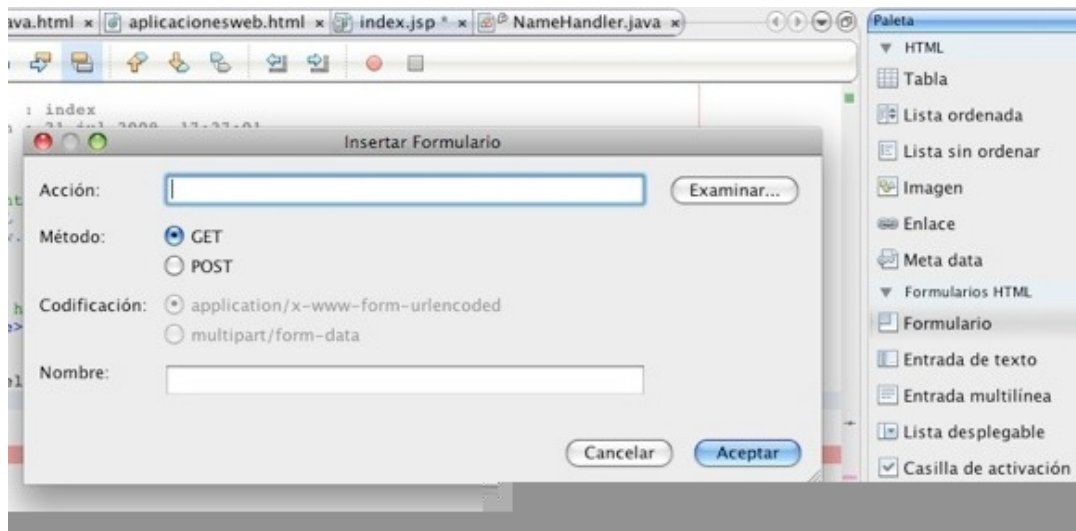
- 1. Da click derecho en el campo name en NameHandler.java y seleccione **Estructurar** -> **Encapsular Campos**. Se abre el dialogo Encapsular Campos. Nota que **Visibilidad de campos** está por defecto establecido como **private**, y **Acceso de visibilidad como public**, indicando que el modificador de acceso para la declaración de variable de clase será especificada como privada, mientras que los métodos getter y setter se generaran con modificadores public y private respectivamente.



- 2. Da click en Reestructurar. Los métodos getter y setter son generados para el campo name. El modificador para la variable de clase esta definido como private mientras que los métodos getter y setter son generados como modificadores públicos. La clase Java ahora se verá similar a la escrita anteriormente.

Editando el Archivo JSP (JavaServer Pages)

- 1. Regresa al archivo **index.jsp** haciendo click en la pestaña disponible sobre Source Editor.
- 2. En **Paleta** (Ctrl-Shift-8) localizada en la parte derecha de Editor Source, expande **HTML Forms** y arrastra el ítem **Form** dentro de Source Editor, a un punto después de las etiquetas `<h1>`. La caja de dialogo de Insert Form se muestra así:



Especifique los siguientes valores:

- **Action:** response.jsp
- **Method:** GET
- **Name:** Name Input Form

Da click en OK. Un formulario HTML es añadido al archivo **index.jsp**.

- 3. Arrastra un ítem **Text Input** (entrada de texto) a un punto antes de la etiqueta `</form>`, entonces especifica los siguientes valores:
- **Name:** name
- **Type:** text

Da click en OK. Una etiqueta HTML `<input>` ha sido añadida dentro de las etiquetas `<form>`.

- 4. Arrastra un ítem **Button** a un punto justo antes de la etiqueta `</form>`.
Especifica los siguientes valores:
- **Label:** OK
- **Type:** submit

Da click en OK. Un boton HTML a sido añadido entre las etiquetas `<form>`.

- 5. Escribe "Ingresa tu nombre": justo antes de la etiqueta `<input>`, entonces cambia el texto "Hello World!" que esta entre las etiquetas `<h1>` a "Formulario de Entrada".
- 6. Da click derecho dentro de Source Editor y seleccione Format (Alt-Shift-F) para limpiar el formato del código. Tu archivo **index.jsp** se parecerá al siguiente:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>

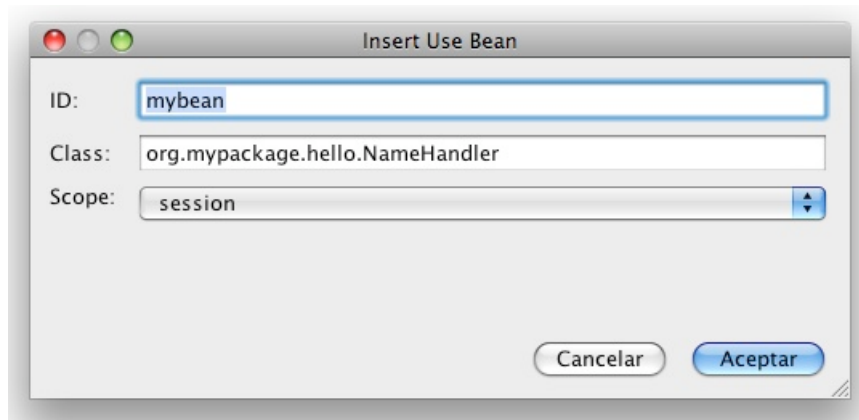
  <body>
    <h1>Formulario de Entrada</h1>

    <form name="Name Input Form" action="response.jsp">
      <p>Ingresa tu nombre:</p>
      <input type="text" name="name" size="30" />
      <input type="submit" value="OK" />
    </form>

  </body>
</html>
```

Creando un Archivo JSP (JavaServer Pages)

- 1. En la ventana **Projects**, da click derecho en el nodo del proyecto HelloWeb y selecciona **New** -> **JSP**. El asistente New JSP File se abrirá. Dale el nombre **response**, y da click en Finish. Observa que el nodo del archivo **response.jsp** se muestra en la ventana Projects debajo de **index.jsp**, y el nuevo archivo se abre en Source Editor.
- 2. En **Paleta** a la derecha de Source Editor, expande JSP y arrastra el ítem **Use Bean** a un punto debajo de la etiqueta **<body>** en Source Editor. La caja de dialogo de Insert Use Bean se abre. Ingresa los siguientes valores:



- **ID:** mybean
- **Class:** org.mypackage.hello.NameHandler
- **Scope:** session

Da click en OK. Observa que las etiquetas `<jsp:useBean>` aparecen debajo de la etiqueta `<body>`.

- 3. Arrastra el ítem **Set Bean Property** de **Paleta** a un punto despues de la etiqueta `<jsp:useBean>` y da click en OK. En la etiqueta `<jsp:setProperty>` , borra el atributo vacío **value** y cambialo por lo siguiente:

```
<jsp:setProperty name="mybean" property="name" />
```

Como se indica en la documentación de `<jsp:setProperty>`, puedes indicar un valor de propiedad de varias maneras. En este caso, el ingreso del usuario que viene de `index.jsp` se convierte en un par **nombre/valor** que pasa al objeto **request object**. Cuando defines una propiedad usando la etiqueta `<jsp:setProperty>`, podras especificar el valor de acuerdo con el nombre de la propiedad contenida en el objeto **request**. Por tanto, al definir **property** a **name**, podras recuperar el valor especificado por el ingreso del usuario.

- 4. Cambia el texto entre las etiquetas `<h1>` para que se vea como los siguiente:

```
<h1>Hello, !</h1>
```

- 5. Arrastra el ítem **Get Bean Property** de **Paleta** y sueltalo despues de la coma entre las etiquetas `<h1>`. Especifica los siguientes valores en la caja de dialogo de **Insert Get Bean Property**:
 - **Bean Name**: mybean
 - **Property Name**: name
- Da click en OK. Observa que las etiquetas `<jsp:getProperty>` están entre las etiquetas `<h1>`.
- 6. Daga click derecho en Source Editor y seleccione **Format** (Alt-Shift-F) para limpiar el formato de tu código. Las etiquetas `<body>` del archivo **response.jsp** deberán ser similares a:

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>JSP Page</title>
</head>

<body>
```

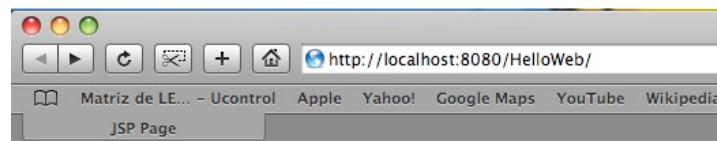


```
<jsp:useBean id="mybean" scope="session"
    class="org.mypackage.hello.NameHandler" />
<jsp:setProperty name="mybean" property="name" />
<h1>Hello,<jsp:getProperty name="mybean" property="name" />!</h1>
</body>
</html>
```

Compilando y Ejecutando un Proyecto de Aplicación Web

El IDE usa un script de compilación **Ant** y ejecuta las aplicaciones Web. El IDE genera el script de compilación basado en las opciones que especifique en el asistente **New Project**, así como los de la caja de dialogo Project Properties del proyecto (En la ventana Projects, seleccione Properties del menú del nodo del proyecto).

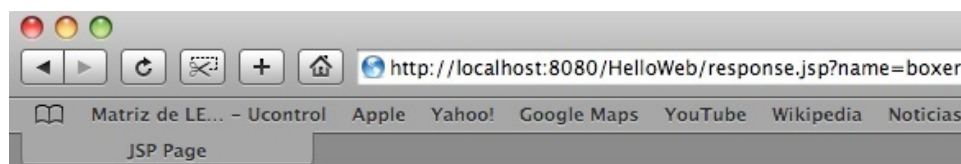
- 1. En la ventana **Projects**, da click derecho sobre el nodo del proyecto **HelloWeb** y seleccione **Run** (F6). El IDE compila la aplicación web y la despliega en el servidor que especificó cuando creaste el proyecto. La página **index.jsp** se abre en tu navegador:



Formulario de Entrada

Ingresa tu nombre:

- 2. Ingresa tu nombre en la caja de texto, y da click en OK. La página **response.jsp** se mostrará, saludándote:



Hello,boxer!

Suerte!