

Tema 1 : Servidor de aplicaciones



Ciclo Superior DAW

Asignatura: Despliegue de aplicaciones web

Curso 20/21



Introducción

- En este capítulo veremos aspectos como:
 - Principios de un servidor de aplicaciones.
 - Arquitectura.
 - Instalación.
 - Documentación de la instalación del servidor.
 - Estructura de una aplicación web J2EE



Servidores de aplicaciones

- Un **servidor de aplicaciones** es un **paquete software** que **proporciona servicios a las aplicaciones**, como seguridad, soporte para transacciones, balanceo de carga y gestión de sistemas distribuidos.
- Suele ser un dispositivo software que proporciona servicios de aplicación a los ordenadores cliente.

Recursos necesarios en el servidor de aplicaciones



- Para poder **instalar nuestro propio servidor web** necesitaremos, como mínimo, los siguientes recursos:
 - Máquina con una potencia capaz de atender las peticiones que tenga que procesar. Es aconsejable que se trate de un servidor dedicado o, como mucho, compartido, pero con funciones relacionadas, por ejemplo, un servidor de transferencia de archivos (FTP).
 - Sistema operativo estable

Recursos necesarios en el servidor de aplicaciones



- Dirección IP estática.
- Dirección simbólica configurada en un servidor DNS para que dirija las peticiones a la dirección IP correspondiente.
- Conexión a Internet 24 h.
- Software del servidor.



Entorno que configuraremos

- Para comprobar el funcionamiento de los servidores, montaremos un entorno de virtualización con VirtualBox que conste de tres máquinas virtuales:
 - **Xubuntu 18.04.** La elección de esta distribución se debe a que ofrece un entorno gráfico más simple que permite su ejecución en máquinas de recursos más limitados.
 - **Windows 7**
 - **Máquina cliente**, dejando el sistema operativo a elección del alumno.



Entorno que configuraremos

- Salvo que se indique otra cosa, para las prácticas emplearemos el modo de red “Red interna” con las siguientes IP:

Máquina	IP
Xubuntu 18.04	192.168.0.1
Windows 7	192.168.0.2
Máquina cliente	192.168.0.3



Actividad 1

Siguiendo los pasos anteriormente indicados, configuramos nuestro entorno

Tomcat



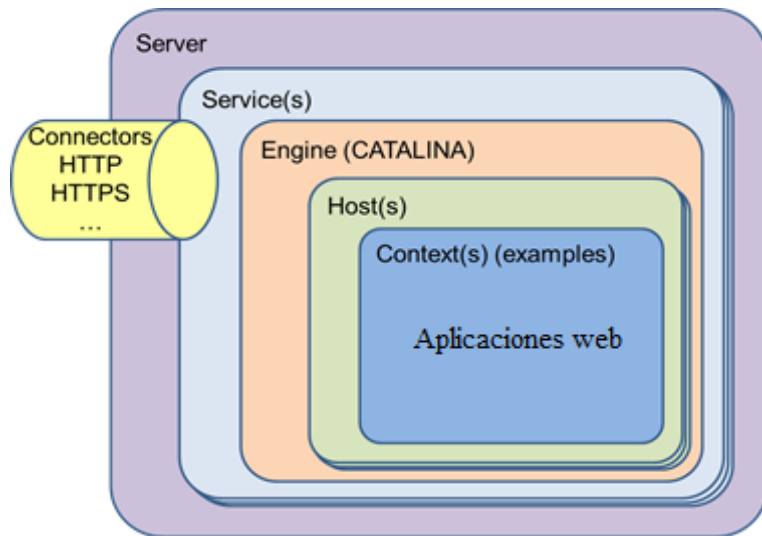
- **Apache Tomcat** (también llamado Jakarta Tomcat o simplemente Tomcat) funciona como un colector de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation.
- Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP).

Tomcat



- Se encarga de:
 - Gestionar las solicitudes y respuestas HTTP
 - Es servidor de aplicaciones o colector de Servlets y JSP.
 - Incluye el compilador Jasper, que compila JSP convirtiéndolas en servlets.

Tomcat. Arquitectura





Tomcat. Arquitectura

- Tomcat dispone de los siguientes componentes:
 - El **Server** es el primer elemento contenedor.
 - Un **Conector (Connector)** es una asociación con puerto IP para manejar las peticiones y las respuestas con los clientes.
 - Un **Servicio (Service)** asocia uno o más Connectors con un único Engine.
 - **Engine** es un colector de uno o más Hosts. Es posible configurar Virtual Hosts. Recibe las peticiones de los Conectores y las traslada al host correspondiente.



Tomcat. Arquitectura

- Tomcat dispone de los siguientes componentes:
 - Un **Host** define un servidor virtual (Virtual Host). Puede contener una o más aplicaciones web. Cada una de ellas se representa por un Context.
 - Cada **Context** representa una aplicación web (HTML, CSS, Servlets, JSP...) ejecutándose dentro de un Host.



HTTP Apache y Apache Tomcat

- **HTTP Apache y Apache Tomcat**, es software que se ejecuta en un servidor con la finalidad de ayudar a desplegar (deploy) un proyecto web.
- La diferencia radica en el tipo de proyecto que se planea implementar



HTTP Apache y Apache Tomcat

- **HTTP Apache** se utiliza principalmente para **ALMACENAR PÁGINAS WEB ESTÁTICAS Y DINÁMICAS**, usando lenguajes como PHP, Perl, Python y Ruby.
- **Apache Tomcat** también soporta páginas web estáticas, pero **SU ESPECIALIDAD SON LAS PÁGINAS WEB DINÁMICAS** desarrolladas con la tecnología Java, como Java Servlet, JavaServer Page (JSP).



Actividad 2

¿Qué diferencias podemos apreciar entre HTTP Apache y Apache Tomcat?

¿Cuál utilizaríais para crear una página de un supermercado online?

¿Y para crear una página de un bar?



Tomcat. Instalación en Linux

- Para instalar Apache Tomcat 9 en Linux tendremos que instalar el Java Development Kit y el propio servidor Tomcat.
- Ambos se encuentran en los repositorios, así que los instalaremos desde consola:

```
sudo apt-get install openjdk-8-jdk
```

```
sudo apt-get install tomcat9
```



Tomcat. Instalación en Linux

- Para poder visualizar también los ejemplos, instalaremos, además:

```
sudo apt-get install tomcat9-examples
```



Tomcat. Instalación en Linux

- Para comprobar que el servidor está lanzado empleamos el comando **ps**
- Este comando nos muestra información sobre los procesos activos del sistema, con las opciones -e (para mostrar todos los procesos) y -f (para que se muestre el listado completo de la información para cada proceso):

```
ps -ef | grep apache
```



Tomcat. Instalación en Linux

- Y para comprobar que está escuchando en el puerto 8080, el comando **netstat**, que muestra, entre otros, las conexiones de red, con las opciones -l (escucha), -t (TCP) y -n (numérico, para IP y puertos):

```
netstat -ltn
```



Tomcat. Instalación en Linux

- Para hacer una prueba de conexión al servidor, en el propio servidor, debe abrirse un navegador web y teclear:

```
http://localhost:8080
```

- Y aparecerá la página que se encuentre en:

```
/var/lib/tomcat9/webapps/ROOT/index.html
```



Tomcat. Instalación en Linux

- Si no se habéis hecho ningún cambio, contendrá, entre otros, el mensaje:

“It works!”.

- Para comprobar que se pueden establecer conexiones desde otras máquinas, desde la máquina cliente tecleamos la IP del servidor Xubuntu:

<http://192.168.0.1:8080>



Actividad 3

*Siguiendo los pasos anteriormente indicados, instalamos Tomcat en nuestro
servidor Linux*



Tomcat. Instalación en Windows

- Para instalar Apache Tomcat 9 en Windows tendremos que descargar los binarios del HTTP Apache de la siguiente dirección:

<https://www.apachehaus.com/downloads/httpd-2.4.46-lre323-x86-vs16.zip>

- Una vez descargado, lo descomprimos y copiamos la carpeta en C:\ de la máquina servidor



Tomcat. Instalación en Windows

- Para poder ejecutarlo, tenemos que haber instalado el Java Development Kit, que deberemos descargar desde la dirección:

<https://www.oracle.com/webapps/redirect/signon?nexturl=https://download.oracle.com/otn/java/jdk/8u271-b09/61ae65e088624f5aaa0b1d2d801acb16/jdk-8u271-windows-i586.exe>

- Una vez instalado deberemos establecer dos variables de entorno:

JAVA_HOME	C:\Program Files\Java\jdk1.8.0_121
JAVA_JRE	C:\Program Files\Java\jre1.8.0_121



Tomcat. Instalación en Windows

- Reiniciamos la máquina virtual del servidor para que los cambios tengan efecto.
- A continuación, descargamos los archivos precisos para la instalación de Apache Tomcat desde la máquina real de la dirección:

<http://apache.rediris.es/tomcat/tomcat-9/v9.0.41/bin/apache-tomcat-9.0.41-windows-x86.zip>



Tomcat. Instalación en Windows

- Una vez reiniciada, descomprimos el fichero:

```
apache-tomcat-9.0.0.M17.zip
```

- Y extraemos la carpeta, colocándola en la ruta C:\

```
apache-tomcat-9.0.0.M17
```



Tomcat. Instalación en Windows

- **Instalamos el servicio tomcat9**, mediante el siguiente comando (debemos encontrarnos en el directorio donde está el archivo tomcat9.exe, es decir: C:\apache-tomcat-9.0.0.M17\bin):

```
tomcat9.exe -k install
```



Tomcat. Instalación en Windows

- A continuación, ejecutamos el siguiente comando:

```
C:\apache-tomcat-9.0.0.M17\bin\startup.bat
```



Tomcat. Instalación en Windows

- Tras esto, comprobaremos que el servidor está escuchando en el puerto 8080 con el comando **netstat**, que **permite monitorizar las conexiones existentes entre nuestro ordenador y el mundo exterior**, con las opciones -a (para que las muestre todas), -p TCP (para que filtre las del protocolo TCP) y -n (para que muestre direcciones y puertos de forma numérica):

```
netstat -a -p TCP -n
```



Tomcat. Instalación en Windows

- Comprobaremos que funciona en local y desde la máquina cliente de la misma manera que hicimos para Linux, teniendo en cuenta únicamente que se debe configurar el firewall de Windows para que permita las conexiones entrantes en el puerto 8080.



Actividad 4

*Siguiendo los pasos anteriormente indicados, instalamos Tomcat en nuestro
servidor Windows*



Estructura de una aplicación web J2EE

- Una aplicación web está organizada en una estructura jerárquica de directorios divididos en dos partes:
 - Un **directorio privado WEB-INF**
 - Un **directorio público**



Estructura de una aplicación web J2EE

- El **Directorio privado WEB-INF** contiene:
 - El descriptor de despliegue de aplicaciones web (web.xml), que describe la estructura de la aplicación web.
 - Archivos de descriptor de bibliotecas de archivos.
 - **class/**: directorio que contiene las clases java compiladas que componen la aplicación (incluidos los servlets).
 - **lib/**: directorio que contiene archivos JAR de bibliotecas (bibliotecas de códigos y bibliotecas de programas de utilidad llamadas por clases del servidor).



Estructura de una aplicación web J2EE

- El **Directorio público** contiene:
 - Los recursos públicos de la aplicación web



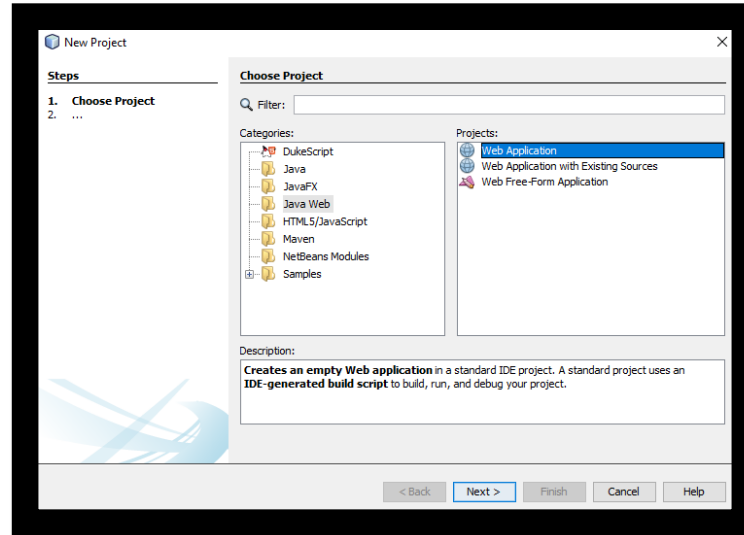
Pasos para crear una aplicación web J2EE

- A continuación, se explicará el proceso seguido para crear una aplicación web J2EE.
- La explicación está hecha con el IDE Netbeans. Si alguno quiere utilizar otro IDE, es libre de hacerlo sin ningún inconveniente.
- Los pasos a seguir son:



Pasos para crear una aplicación web J2EE

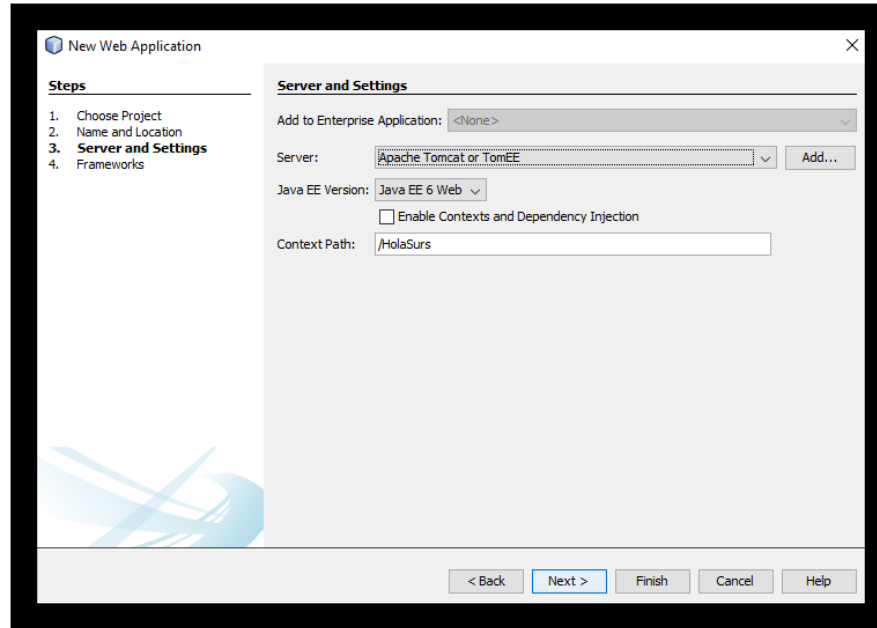
1. Creamos un nuevo proyecto en NetBeans: File > New Project y selecciona Java Web > Web Application





Pasos para crear una aplicación web J2EE

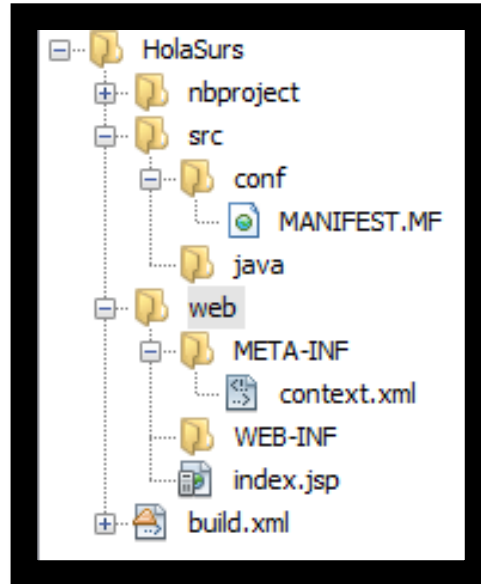
2. Seleccionamos el servidor de aplicaciones en el que se va a desplegar la aplicación





Pasos para crear una aplicación web J2EE

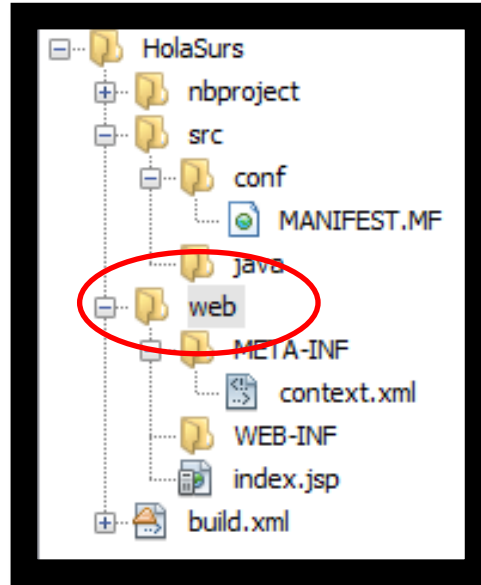
3. Comprobamos la estructura del proyecto accediendo a la pestaña Files.
Debería ser similar a la imagen, si no aparece, clicamos en Windows > Files





Pasos para crear una aplicación web J2EE

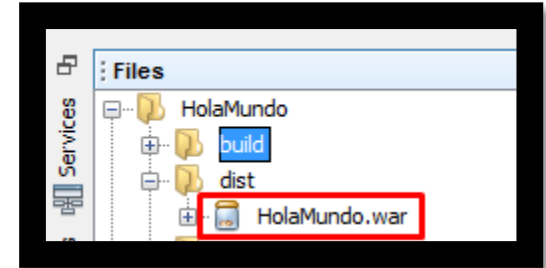
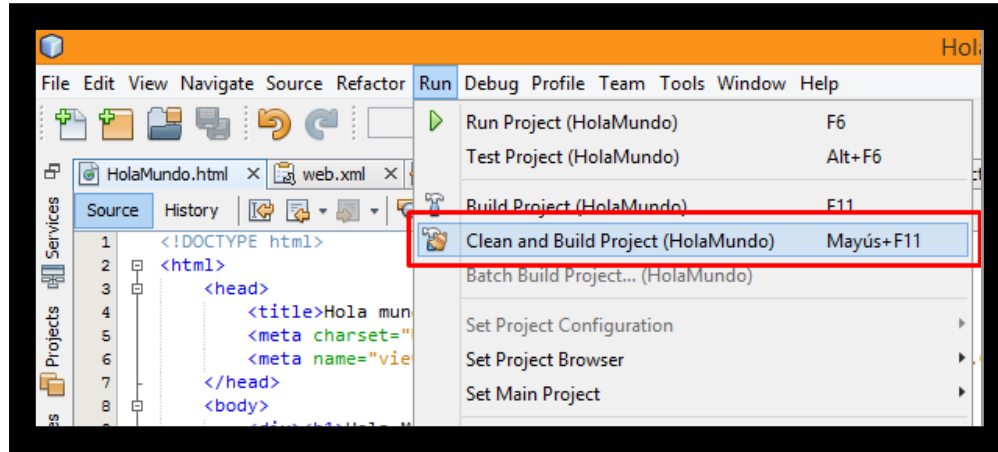
4. En la carpeta “web”, crearemos los ficheros HTML que queremos que se muestren.





Pasos para crear una aplicación web J2EE

5. Seleccionamos la opción “Run > Clean and Build”, y observamos que se han creado las carpetas build y dist; Además, dentro de la carpeta dist se ha creado un archivo llamado HolaMundo.war





Pasos para crear una aplicación web J2EE

6. Para desplegar el proyecto deberemos copiar este archivo en la carpeta webapps de nuestro servidor.
7. Una vez hecho esto, si accedemos al siguiente recurso, comprobaremos que la web se ha desplegado correctamente:

```
http://<IP\_SERVIDOR\_TOMCAT>:8080/<NombreAplicacion>/<NombrePagina>
```



Actividad 5

Siguiendo los pasos indicados, crearemos una aplicación web J2EE y la configuraremos para su correcto despliegue



Descriptor de despliegue

- Un Descriptor de Despliegue es un **documento XML que describe las características de despliegue de una aplicación, un módulo o un componente**
- Cualquier aplicación web tiene que tener un descriptor de despliegue situado en WEB-INF/web.xml



Descriptor de despliegue

- El descriptor de despliegue de TOMCAT es:

<TOMCAT_HOME>/conf/web.xml

- El **objetivo** del descriptor de despliegue es **especificar la configuración general de la aplicación web tal como requiere el estándar J2EE.**



Descriptor de despliegue

- El descriptor de despliegue especifica:
 - Los **valores de parámetros de inicialización** que se proporcionan **para la aplicación web**.
 - Las **clases de servlet utilizadas por la aplicación web** se pueden declarar y darles nombres.
 - **Cada clase de servlet se relaciona con uno o más patrones de URL**: cuando el colector de servlet recibe una solicitud cuya URL coincide con un patrón definido en el archivo web.xml, se utilizará el servlet correspondiente para procesar la solicitud.



Descriptor de despliegue

- El descriptor de despliegue especifica:
 - Si es necesario, **los valores de parámetros de inicialización que se proporcionan para cada servlet:**
 - Información de sesión (como el tiempo de espera).
 - La localización de bibliotecas de códigos personalizadas utilizadas por las páginas JSP.



Descriptor de despliegue

- El descriptor de despliegue tiene los siguientes elementos:
 - **display-name:** Nombre de la aplicación (para tareas de identificación, por ejemplo, en Tomcat Manager aparece en "Nombre a mostrar").
 - **welcome-file-list:** Lista de ficheros que se servirán, por orden, en el caso de que no se indique ninguno en la URL. Cada uno de ellos se encuentra en un elemento welcome-file.
 - **session-config**
 - **session-timeout:** Tiempo, en minutos, que tarda en expirar la sesión.



Descriptor de despliegue

- **Servlet**
 - **servlet-name**: Nombre que le damos al servlet.
 - **servlet-class**: Clase Java en la que se implementa el servlet.
- **servlet-mapping**
 - **servlet-name**: Nombre que le damos al servlet.
 - **url-pattern**: Patrón de URL que deberá usar para acceder.



Descriptor de despliegue

- **security-constraint**
 - web-resource-collection
 - web-resource-name: Nombre del recurso (para identificación).
 - url-pattern: Patrón de archivos a proteger dentro de la aplicación.
 - auth-constraint
 - role-name: Nombre del rol definido en Tomcat para la autenticación.
 - login-config
 - auth-method: Método de autenticación.
 - realm-name: Mensaje que aparecerá en la pantalla de identificación.



Descriptor de despliegue. Ejemplo

- Un ejemplo de descriptor de despliegue puede ser el siguiente archivo web.xml:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app PUBLIC
"-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
"http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">
<web-app>
    <!-- Tus definiciones van aquí -->
</web-app>
```



Descriptor de despliegue. Ejemplo

- Entre las etiquetas `<web-app>` y `</web-app>` estarían los descriptors de despliegue de servlets:

```
<servlet>
  <servlet-name>nombre</servlet-name>
  <servlet-class>package.nombre.MiClass</servlet-class>
</servlet>
```



Desplegando una aplicación web

- Para realizar el despliegue de una aplicación web en Tomcat basta con crear un directorio con el nombre de la aplicación dentro del directorio **webapps**.
- Tomcat reconocerá la ruta y, si el directorio creado es una aplicación web, la desplegará con éxito.



Servlets

- Un **servlet** es un **programa Java que se ejecuta en un servidor Web y construye o sirve páginas web.**
- De esta forma se pueden construir páginas dinámicas, basadas en diferentes fuentes variables: datos proporcionados por el usuario, fuentes de información variable (páginas de noticias, por ejemplo), o programas que extraigan información de bases de datos.



Servlets

- Como veis en la definición, el servlet requiere de programación en código Java. No es lo que vamos a hacer nosotros aquí.
- En esta asignatura nos centraremos en aprender a configurar correctamente un servlet.
- Es decir, a comprobar que se despliega correctamente, a controlar cómo acceder al contenido, etc.



Pasos para crear un servlet

- Para crear nuestro servlet, utilizaremos una aplicación web J2EE ya creada.
- A continuación veremos los pasos que debemos seguir:



Pasos para crear un servlet

1. Sobre una aplicación web existente se crea un nuevo Servlet. Para eso, accedemos a New > Servlet. Es importante añadir la información de que existe un nuevo servlet al fichero web.xml:

Configure Servlet Deployment

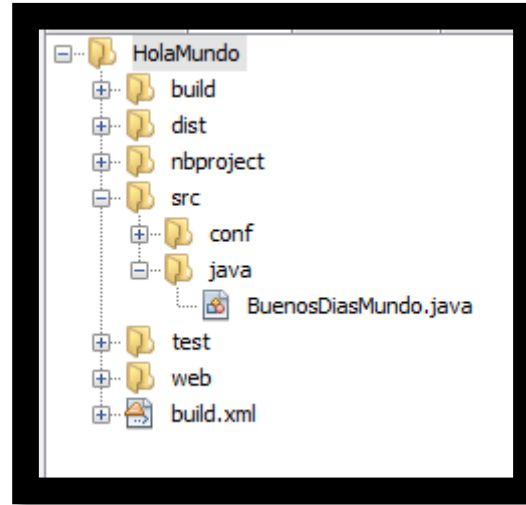
Register the Servlet with the application by giving the Servlet an internal name (Servlet Name). Then specify patterns that identify the URLs that invoke the Servlet. Separate multiple patterns with commas.

☒ Add information to deployment descriptor (web.xml)



Pasos para crear un servlet

2. Comprobamos que en el directorio **src/java** se ha creado el nuevo servlet.





Pasos para crear un servlet

3. Comprobamos que se ha añadido al web.xml

```
<servlet>
  <servlet-name>BuenosDiasMundo</servlet-name>
  <servlet-class>BuenosDiasMundo</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>BuenosDiasMundo</servlet-name>
  <url-pattern>/BuenosDiasMundo</url-pattern>
</servlet-mapping>
```



Pasos para crear un servlet

4. Generamos el fichero WAR aplicando la configuración “Run > Clean and Build” y desplegamos el proyecto:

```
http://<IP_SERVIDOR_TOMCAT>:8080/<NombreServlet>/<URL>
```



Actividad 6

Siguiendo los pasos indicados, crearemos un servlet y lo configuraremos para su correcto despliegue

Tema 1 : Servidor de aplicaciones



Ciclo Superior DAW

Asignatura: Despliegue de aplicaciones web

Curso 20/21