



# Servidores Web

- Sesión 2: Desarrollo y configuración de aplicaciones web



# Puntos a tratar

- Desarrollo de aplicaciones web
  - Construcción de la aplicación
  - Despliegue de la aplicación
- Configuración de Tomcat
- Configuración de aplicaciones web
  - El contexto
  - Los *Valves*
  - El descriptor de despliegue



# Directorio de desarrollo

- No debemos desarrollar la aplicación en el mismo directorio de Tomcat
  - Crearemos la aplicación en un directorio de desarrollo
  - Para probar la aplicación la copiaremos al directorio de Tomcat (despliegue)
- Directorio de desarrollo:
  - `src`: código fuente (servlets y otras clases)
  - `web` o `WebContent`: elementos que no necesitan ser compilados (recursos estáticos, JSPs, `WEB-INF/web.xml`, librerías)
  - `build`: Aplicación completa tal como se instalará en el servidor
  - `dist`: Aplicación empaquetada (fichero WAR)
  - ... veremos otros directorios relevantes más adelante*



# Preparación

- Crear el directorio `build`
- Copiar el contenido de `web` (o `WebContent`) a `build`

```
<target name="prepare">
  <mkdir dir="${build.home}" />
  <mkdir dir="${build.home}/WEB-INF" />
  <mkdir dir="${build.home}/WEB-INF/classes" />
  <mkdir dir="${build.home}/WEB-INF/lib" />
  <copy todir="${build.home}">
    <fileset dir="${web.home}" />
  </copy>
</target>
```



# Compilación

- Compilar los fuentes de `src`
- Generar la salida en `build/WEB-INF/classes`

```
<target name="compile" depends="prepare"
        description="Compila los fuentes Java">
    <javac srcdir="${src.home}"
          destdir="${build.home}/WEB-INF/classes"
          debug="${compile.debug}"
          deprecation="${compile.deprecation}"
          optimize="${compile.optimize}">
        <classpath refid="compile.classpath"/>
    </javac>
</target>
```



# Classpath

- En el *classpath* debemos tener las librerías de J2EE para compilar
- Cogemos la implementación de J2EE incluida en el servidor de aplicaciones que vamos a usar
  - En Tomcat se incluye parte de la especificación de J2EE

```
<path id="compile.classpath">
  <pathelement location="{catalina.home}/common/classes"/>
  <fileset dir="{catalina.home}/common/endorsed">
    <include name="*.jar"/>
  </fileset>
  <fileset dir="{catalina.home}/common/lib">
    <include name="*.jar"/>
  </fileset>
  <pathelement location="{catalina.home}/shared/classes"/>
  <fileset dir="{catalina.home}/shared/lib">
    <include name="*.jar"/>
  </fileset>
</path>
```



# Empaquetar

- Opcional según el servidor
- Empaquetar el contenido de `build` en un fichero WAR, y guardarlo en `dist`
- El fichero WAR se crea con la herramienta JAR

```
<target name="dist" depends="compile"
        description="Crea el fichero WAR de la
                    aplicación">
    <mkdir dir="${dist.home}" />
    <jar jarfile="${dist.home}/${war.name}"
        basedir="${build.home}" />
</target>
```



# Directorio de aplicaciones

- El servidor web Tomcat tiene un directorio donde se encuentran las aplicaciones web instaladas

```
${tomcat.home}/webapps
```

- Cada aplicación web (o contexto) se guarda como un subdirectorio de `webapps`
- Al contexto se le da por defecto como ruta el nombre de este directorio
  - Si está en `webapps/aplic`, la ruta del contexto será

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

- El contexto `ROOT` creado por defecto tiene la ruta

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

- Podremos cambiar estas rutas en la configuración



# Despliegue

- Entendemos por *despliegue* (*deployment*) el proceso de instalar la aplicación web en el servidor de aplicaciones para que empiece a funcionar
- Alternativas para el despliegue:
  - Copiar la aplicación al directorio `webapps` de Tomcat. Copiamos el directorio de la aplicación o el fichero WAR
  - Utilizar la interfaz HTML del *manager* de Tomcat para subir el fichero WAR
  - Utilizar las tareas de *ant* para el despliegue:
    - Añadir tareas de Tomcat a *ant*
    - Utilizar la tarea de despliegue para subir el fichero WAR



# Copiar al directorio de aplicaciones

- Para desplegar aplicaciones en Tomcat podemos copiarlas directamente al directorio webapps
  - Podemos copiar el directorio completo o el fichero WAR con la aplicación (Tomcat creará el directorio)
  - Deberemos reiniciar el servidor

```
<target name="deploy" depends="compile"
    description="Despliega la aplicacion">
    <mkdir dir="${catalina.home}/${app.name}" />
    <copy todir="${catalina.home}/${app.name}">
        <fileset dir="${build.home}" />
    </copy>
</target>
```



# Interfaz HTML

- Tomcat tiene un manager desde el que podremos desplegar y gestionar las aplicaciones web

The screenshot shows the Tomcat Manager interface with two main sections:

- Install directory or WAR file located on server:** This section contains three input fields: 'Context Path (optional):', 'XML Configuration file URL:', and 'WAR or Directory URL:'. Below these fields is an 'Install' button.
- Upload a WAR file to install:** This section contains a 'Select WAR file to upload' label, an input field, and an 'Examinar...' (Browse...) button. Below the input field is an 'Install' button.

<http://localhost:8080/manager/html>



## Tareas de Ant

- Añadir librería `catalina-ant.jar` al CLASSPATH o a la configuración de Eclipse (*Window – Preferences – Ant – Runtime*)

```
 ${tomcat.home}/server/lib/catalina-ant.jar
```

- Declarar las tareas de Tomcat

```
<taskdef name="deploy"      classname="org.apache.catalina.ant.DeployTask"/>
<taskdef name="install"    classname="org.apache.catalina.ant.InstallTask"/>
<taskdef name="list"       classname="org.apache.catalina.ant.ListTask"/>
<taskdef name="reload"     classname="org.apache.catalina.ant.ReloadTask"/>
<taskdef name="remove"     classname="org.apache.catalina.ant.RemoveTask"/>
<taskdef name="resources"  classname="org.apache.catalina.ant.ResourcesTask"/>
<taskdef name="roles"      classname="org.apache.catalina.ant.RolesTask"/>
<taskdef name="start"      classname="org.apache.catalina.ant.StartTask"/>
<taskdef name="stop"       classname="org.apache.catalina.ant.StopTask"/>
<taskdef name="undeploy"   classname="org.apache.catalina.ant.UndeployTask"/>
```



# Despliegue con Ant

- Se puede automatizar el despliegue con la tarea `deploy`

```
<target name="deploy" depends="dist"
        description="Despliega la aplicacion">
  <deploy url="{manager.url}"
          username="{manager.nombre}"
          password="{manager.passwd}"
          path="{app.path}"
          war="file:{dist.home}/{war.name}" />
</target>
```



# Configuración de Tomcat

- La configuración general de Tomcat se encuentra en 4 ficheros en la carpeta `conf`
  - `server.xml`: fichero principal de configuración
  - `web.xml`: fichero descriptor con configuración global a todas las aplicaciones
  - `tomcat-users.xml`: lista de usuarios y contraseñas para autenticación
  - `catalina.policy`: fichero con las políticas de seguridad de Tomcat



# Configuración del host

- La etiqueta *Host* de *server.xml* define la configuración general para un host o host virtual (es decir, un subgrupo de aplicaciones dentro del servidor web)

```
<Host name="localhost" debug="0" appbase="webapps"
      unpackWars="true" autoDeploy="true">
```

Atributo	Significado	Valor por defecto
name	nombre del host o host virtual	ninguno
debug	nivel de mensajes de depuración	0
appBase	directorio donde se instalarán las aplicaciones de este host (si es relativa, la ruta se supone con respecto al directorio de Tomcat)	ninguno
unpackWARs	si es true, las aplicaciones empaquetadas en WAR se desempaquetan antes de ejecutarse	true
autoDeploy	si es true, se utiliza el despliegue automático de aplicaciones	true
liveDeploy	si es true, se utiliza despliegue automático sin necesidad de rearrancar Tomcat	true



# Contexto de una aplicación web

- Elemento Context

Atributo	Significado	Valor por defecto
<code>crossContext</code>	indica si la aplicación web puede comunicarse con otras	<code>false</code> (por razones de seguridad)
<code>debug</code>	nivel de mensajes de depuración	0
<code>docBase</code>	directorio donde está la aplicación, bien especificado de modo absoluto o con respecto al <code>appBase</code> del Host	ninguno
<code>path</code>	<i>path</i> de esta aplicación, que se hace coincidir con la URL de la petición para ver qué aplicación debe servirla.	ninguno
<code>reloadable</code>	permite monitorizar los cambios de clases en <code>/WEB-INF/classes</code> y <code>/WEB-INF/lib</code> para que no sea necesario reiniciar Tomcat	<code>false</code> (se suele poner a <code>true</code> durante el desarrollo)



# Dónde configurar la aplicación

- En un elemento `Context` dentro de `server.xml`

```
<Context docBase="conversor" path="/conversor"  
.../>
```

- Desaconsejado
- Despliegue automático
  - En un elemento `Context` dentro de un xml dejado en `conf/nombre_engine/nombre_host/fichero.xml`
  - Sin nada: se toma el contexto "por defecto" (`DefaultContext`)



# Valves

- Componentes que se insertan en el ciclo de procesamiento de la petición para controlar varios aspectos:
  - Registro de accesos
  - Filtro de hosts o de IPs
  - Volcado de la petición
  - ...
- Clases Java que el usuario podría implementar
- Se pueden poner a distintos niveles: **engine**, **host** o **context**



# Ejemplos de *Valves*

- Log de accesos

```
<Valve  
className="org.apache.catalina.valves.AccessLogValve"  
directory="logs" prefix="localhost_access_log."  
suffix=".txt" pattern="common" resolveHosts="false"/>
```

- Filtro de IPs

```
<Valve  
  className="org.apache.catalina.valves.RemoteAddrValve"  
  allow="127.0.0.1" />
```



# Configuración de la aplicación: web.xml

- Cada aplicación web / contexto debe tener un fichero descriptor de despliegue
  - Fichero `web.xml`
  - Ubicado en la ruta `/WEB-INF/web.xml` del contexto
- Describe y configura la aplicación web
- Fichero XML definido de forma estándar para la configuración de las aplicaciones J2EE



# Descriptor básico

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE web-app PUBLIC
  "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
  "http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>

  <display-name>Mi Aplicacion Web</display-name>
  <description>
    Esta es una aplicacion web sencilla a modo de ejemplo
  </description>

  <!-- Resto de elementos -->

</web-app>
```



# Elementos del descriptor

- En el descriptor de despliegue se configura:
  - Variable globales del contexto
  - Filtros
  - Servlets y mapeo a URLs
  - Listeners (de cambios en el contexto)
  - Sesiones
  - Página de inicio
  - Restricciones de seguridad
  - Librerías de *tags*
  - Etc ...