# **Solución de la tarea online DAW03**

Una aplicación web puede ser desplegada en diferentes servidores web manteniendo su funcionalidad y sin ningún tipo de modificación en su código debido a la especificación servlet 2.2, ¿cúal es la estructura de directorios que debe tener?

Directorio principal (raíz): Contendrá los ficheros estáticos (HTML, imágenes, etc...) y JSPs.

* + - Carpeta WEB-INF: contiene el fichero "web.xml" (descriptor de la aplicación), encargado de configurar la aplicación.
      * Subcarpeta classes: contiene los ficheros compilados (servlets, beans).
      * Subcarpeta lib: librerías adicionales.
    - Resto de carpetas para ficheros estáticos.

Ant se basa en ficheros XML, normalmente configuramos el trabajo a hacer con nuestra aplicación en un fichero llamado build.xml. Indica alguna de las etiquetas con las que podemos formar el contenido de este archivo, así como la función de cada una de ellas.

**project**: este es el elemento raíz del fichero XML, y como tal, solamente puede haber uno en todo el fichero, el que se corresponde a nuestra aplicación Java.

**target**: un target u objetivo es un conjunto de tareas que queremos aplicar a nuestra aplicación en algún momento. Se puede hacer que unos objetivos dependan de otros, de forma que eso lo trate Ant automáticamente.

**task**: un task o tarea es un código ejecutable que aplicaremos a nuestra aplicación, y que puede contener distintas propiedades (como por ejemplo el classpath). Ant incluye ya muchas básicas, como compilación y eliminación de ficheros temporales, pero podemos extender este mecanismo si nos hace falta. Luego veremos algunas de las disponibles.

**property**: una propiedad o property es simplemente algún parámetro (en forma de par nombre-valor) que necesitamos para procesar nuestra aplicación, como el nombre del compilador, etc. Ant incluye ya las más básicas, como son BaseDir para el directorio base de nuestro proyecto, ant.file para el path absoluto del fichero build.xml, y ant.java.version para la versión de la JVM.

Dispones de una máquina que cuenta con el sistema operativo **Ubuntu 18.04 LTS** o posterior recientemente actualizado, en la que está el entorno de red configurado y además dispones de conexión a Internet, estás trabajando con la cuenta del usuario root. Indica cada uno de los pasos y comandos implicados en ellos para conseguir hacer lo siguiente:Instalar el JDK 11. Para instalar la versión JDK (OpenJDK), ejecutamos el comando:

# apt update

# apt install default-jdk

Crear usuario para WildFly.

sudo groupadd -r wildfly

sudo useradd -r -g wildfly -d /opt/wildfly -s /sbin/nologin wildfly

Descargar e instalar Wildfly.

Para proceder a su instalación primeramente lo descargamos a la carpeta **/tmp**:

WILDFLY\_VERSION=19.1.0.Final

sudo wget https://download.jboss.org/wildfly/$WILDFLY\_VERSION/wildfly-$WILDFLY\_VERSION.tar.gz -P /tmp

A continuación descomprimimos el archivo y lo movemos al driectorio **/opt**:

sudo tar xf /tmp/wildfly-$WILDFLY\_VERSION.tar.gz -C /opt/

Creamos el enlace simbólico para Wildfly:

sudo ln -s /opt/wildfly-$WILDFLY\_VERSION /opt/wildfly

Le pasamos la propiedad del directorio al usuario **wildfly**:

sudo chown -RH wildfly: /opt/wildfly

Configurar **system.d** y archivo **widfly.conf**.

Creamos el directorio para guardar el archivo de configuración:

sudo mkdir -p /etc/wildfly

Copiamos el archivo de configuración al directorio:

sudo cp /opt/wildfly/docs/contrib/scripts/systemd/wildfly.conf /etc/wildfly/

Copiamos el script **launch.sh** script al directorio **/opt/wildfly/bin/**:

sudo cp /opt/wildfly/docs/contrib/scripts/systemd/launch.sh /opt/wildfly/bin/

Le ponemos el atributo ejecutable a los ficheros del directorio **bin**:

sudo sh -c 'chmod +x /opt/wildfly/bin/\*.sh'

Finalmente, copiamos el archivo del servicio al directorio**/etc/systemd/system/**:

sudo cp /opt/wildfly/docs/contrib/scripts/systemd/wildfly.service /etc/systemd/system/

Recargamos **system.d**, arrancamos el servicio Wildfly y comprobamos que esté funcionado:

sudo systemctl daemon-reload

sudo systemctl start wildfly

sudo systemctl status wildfly

Configurar la autenticación de Wildfly.

Agregamos un usuario ejecutando el script **add.user.sh**:

sudo /opt/wildfly/bin/add-user.sh

Nos aparecerá una pregunta, y seleccionamos la opción **a**.

What type of user do you wish to add?

a) Management User (mgmt-users.properties)

b) Application User (application-users.properties)

(a):

A continuación añadimos un usuario, en mi caso **usuario/usuario**:  
Añadimos posteriormente ejecutaríamos **#source /etc/profile** para que el sistema recoja el contenido de las variables creadas sin necesidad de reiniciar el equipo.

Enter the details of the new user to add. Using realm 'ManagementRealm' as discovered from the existing property files. Username : usuario Password recommendations are listed below. To modify these restrictions edit the add-user.properties configuration file. - The password should be different from the username - The password should not be one of the following restricted values {root, admin, administrator} - The password should contain at least 8 characters, 1 alphabetic character(s), 1 digit(s), 1 non-alphanumeric symbol(s) Password : WFLYDM0098: The password should be different from the username Are you sure you want to use the password entered yes/no? y Re-enter Password : What groups do you want this user to belong to? (Please enter a comma separated list, or leave blank for none)[ ]: About to add user 'usuario' for realm 'ManagementRealm' Is this correct yes/no? y Added user 'usuario' to file '/opt/wildfly-19.1.0.Final/standalone/configuration/mgmt-users.properties' Added user 'usuario' to file '/opt/wildfly-19.1.0.Final/domain/configuration/mgmt-users.properties' Added user 'usuario' with groups to file '/opt/wildfly-19.1.0.Final/standalone/configuration/mgmt-groups.properties' Added user 'usuario' with groups to file '/opt/wildfly-19.1.0.Final/domain/configuration/mgmt-groups.properties' Is this new user going to be used for one AS process to connect to another AS process? e.g. for a slave host controller connecting to the master or for a Remoting connection for server to server EJB calls. yes/no? y To represent the user add the following to the server-identities definition <secret value="dXN1YXJpbw==" />

Por último, ya podemos verificar la instalación de Wildfly, introduciendo en el navegador:

http://<dirección\_IP>:8080