

Practica 3

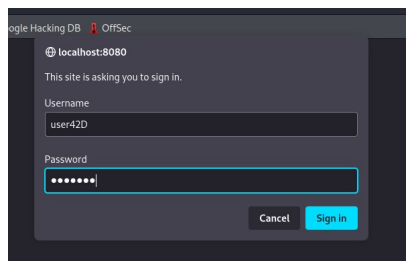
Ejercicio 1

En los siguientes pasos, procederemos a instalar el servidor Apache Tomcat y configurarlo para permitir el acceso de un usuario a través de la interfaz web del gestor de aplicaciones de Tomcat.

Para crear el usuario, agregaremos un fragmento de código al archivo "tomcat-users.xml" ubicado en la ruta /opt/tomcat10/apache-tomcat-10.0.27/conf/tomcat-users.xml. El código quedaria de esta forma:

```
<tomcat-users xmlns="http://tomcat.apache.org/xml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
  version="1.0">
  <role rolename="manager-gui"/>
  <user username="user42D" password="dedaw22" roles="manager-gui"/>
</tomcat-users>
```

Una vez que hayamos guardado los cambios en el archivo, debemos reiniciar Tomcat para que se apliquen los cambios en la máquina. Luego, podemos verificar el acceso abriendo un navegador y navegando a la dirección "localhost:8080", lo que debería mostrarse de la siguiente manera:



localhost:8080

This site is asking you to sign in.

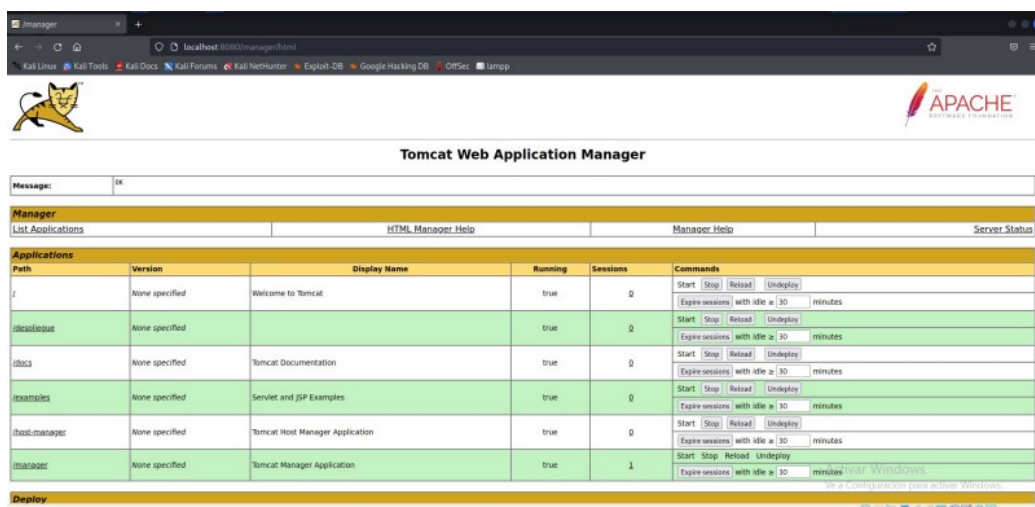
Username

user42D

Password

dedaw22

Cancel Sign in



Tomcat Web Application Manager

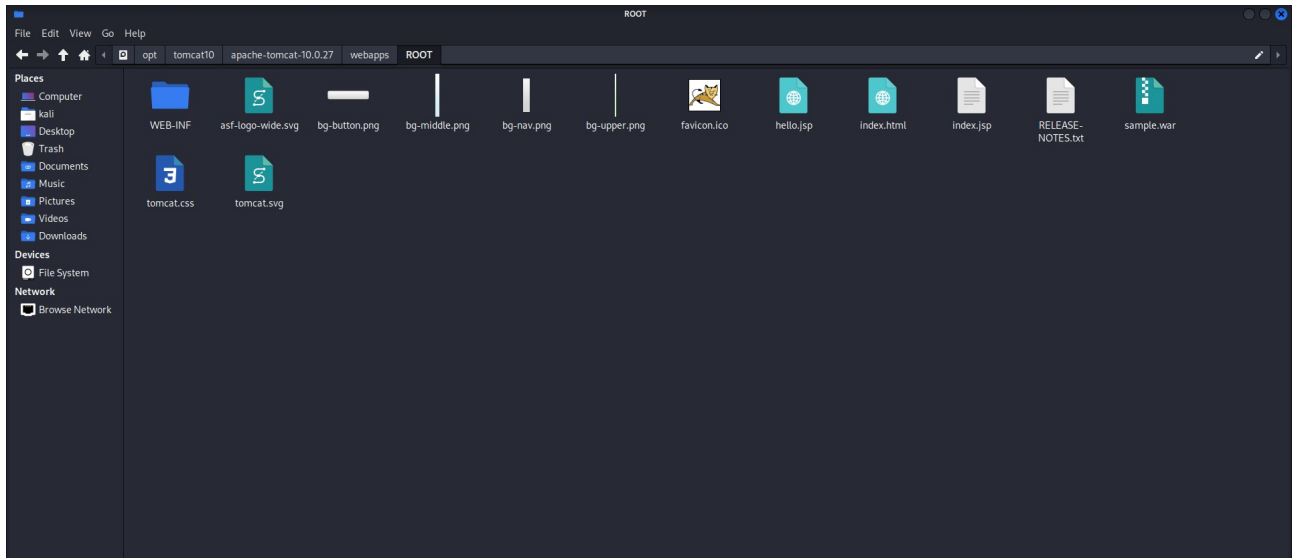
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle a 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle a 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle a 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle a 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle a 30 minutes

Javier Manzano Gamaza

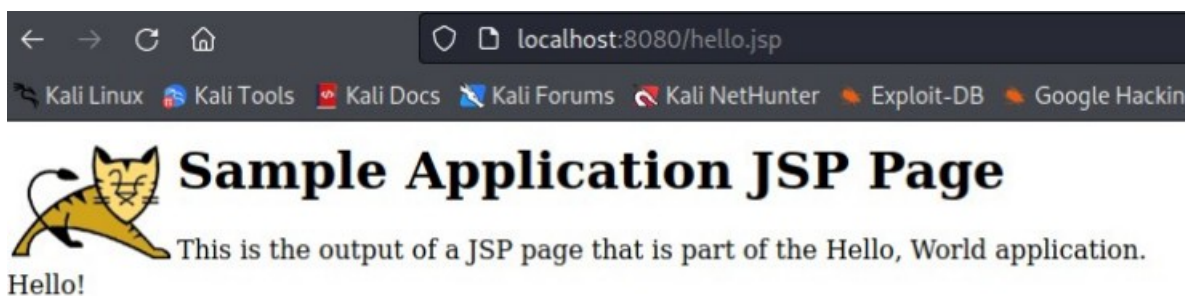
Ejercicio 2

En este ejercicio, desplegaremos un archivo WAR siguiendo los pasos a continuación:

Copiaremos el archivo WAR que deseamos desplegar en la siguiente ruta:



Ponemos la ruta y ya debería salir.

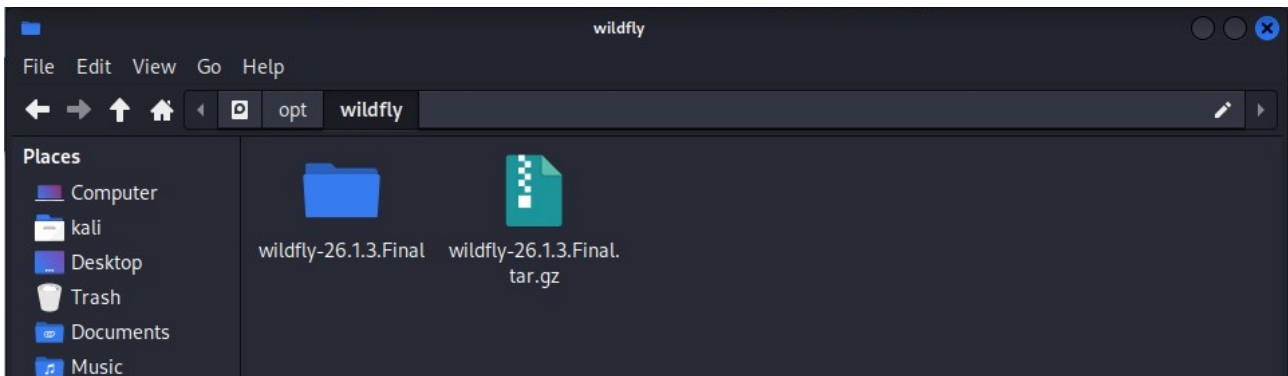


Una vez desplegado el archivo WAR, accederemos a la interfaz web y haremos clic en el enlace "Deploy". Seleccionaremos el archivo WAR que deseamos desplegar y pulsaremos "Deploy" de nuevo. La aplicación se desplegará y estará disponible en la URL "localhost:8080", pero deberemos especificar el nombre de la aplicación para acceder a ella.

Ejercicio 3

Para llevar a cabo esta sección, deberemos descargar la versión 26.x de WildFly desde su sitio web oficial. Una vez que la descarga esté completa, procederemos a descomprimir el archivo.

Para hacerlo, crearemos una carpeta llamada "wildfly" en la ruta "/opt" y descomprimiremos el contenido descargado dentro de esa carpeta.



Ahora vamos a usar el comando siguiente para iniciar el wildfly

```
(root@kali)-[/opt/wildfly/wildfly-26.1.3.Final]
# ./bin/standalone.sh

JBoss Bootstrap Environment

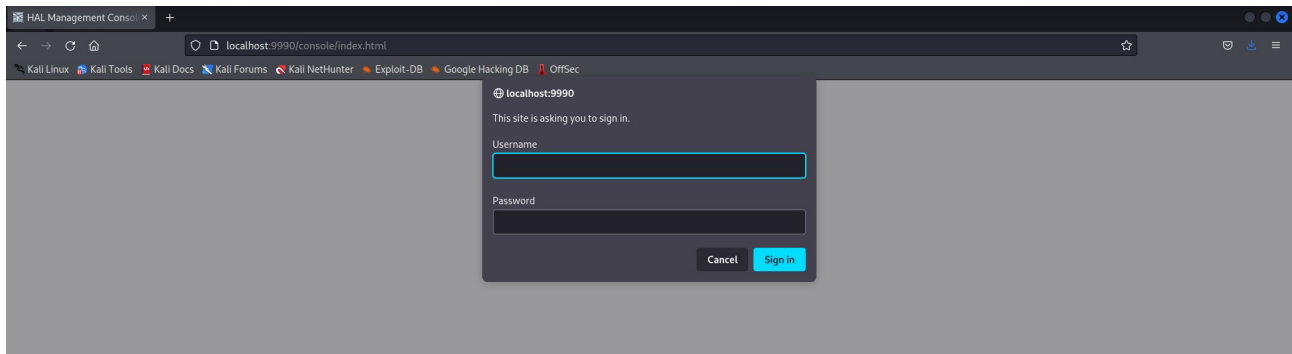
JBOSS_HOME: /opt/wildfly/wildfly-26.1.3.Final

JAVA: java

JAVA_OPTS: -server -Xms64m -Xmx512m -XX:MetaspaceSize=96M -XX:MaxMetaspaceSize=256m -Djava.net.preferIPv4Stack=true -Djboss.modules.system.pkgs=org.jboss.byteman -Djava.awt.headless=true --add-exports=java.desktop/sun.awt=ALL-UNNAMED --add-exports=java.naming/com.sun.jndi.ldap=ALL-UNNAMED --add-exports=java.naming/com.sun.jndi.url.ldap=ALL-UNNAMED --add-exports=java.naming/com.sun.jndi.url.ldaps=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.lang.invoke=ALL-UNNAMED --add-opens=java.base/java.lang.reflect=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.security=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.management/javax.management=ALL-UNNAMED --add-opens=java.naming/javax.naming=ALL-UNNAMED -Djava.security.manager=allow

12:13:58,798 INFO [org.jboss.modules] (main) JBoss Modules version 2.0.2.Final
12:14:00,128 INFO [org.jboss.msc] (main) JBoss MSC version 1.4.13.Final
12:14:00,143 INFO [org.jboss.threads] (main) JBoss Threads version 2.4.0.Final
12:14:00,540 INFO [org.jboss.as] (MSC service thread 1-2) WFLYSRV0049: WildFly Full 26.1.3.Final (WildFly Core 18.1.2.Final) starting
```

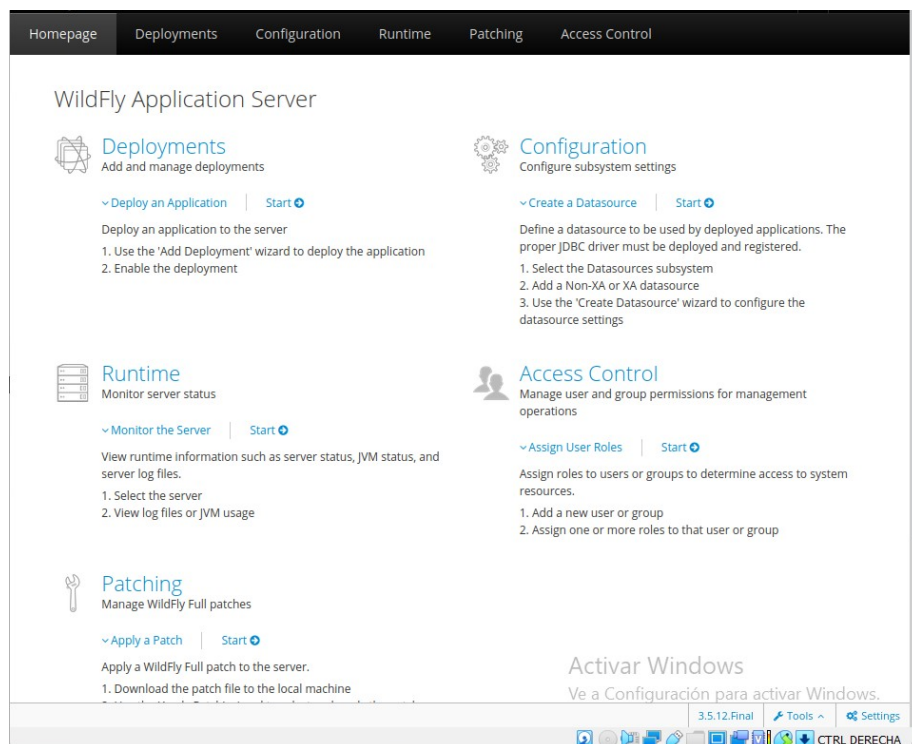
Una vez que se hayan completado todas las operaciones de configuración, debemos verificar que WildFly se haya iniciado correctamente. Para hacerlo, abriremos un navegador web y navegaremos a la dirección "localhost:9990". Si podemos acceder a la página de inicio de WildFly, significará que se ha iniciado correctamente.



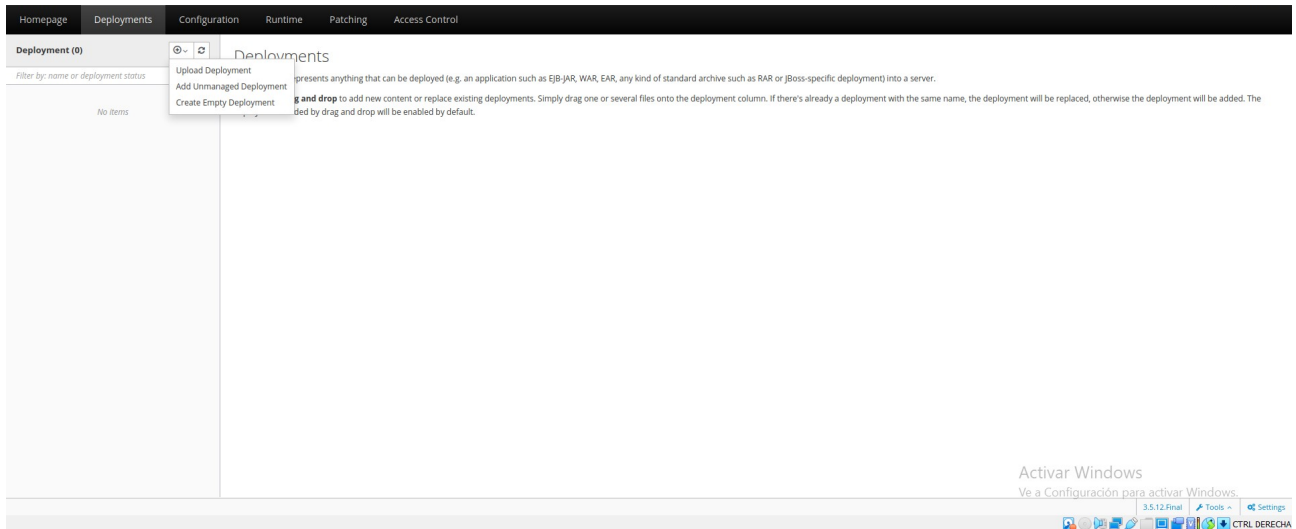
Creamos un nuevo usuario en Wildfly

Después de crear el usuario, procederemos a iniciar WildFly nuevamente y abriremos un navegador web. Navegaremos a la dirección "localhost:9990" y nos autenticaremos con las credenciales del usuario que acabamos de crear.

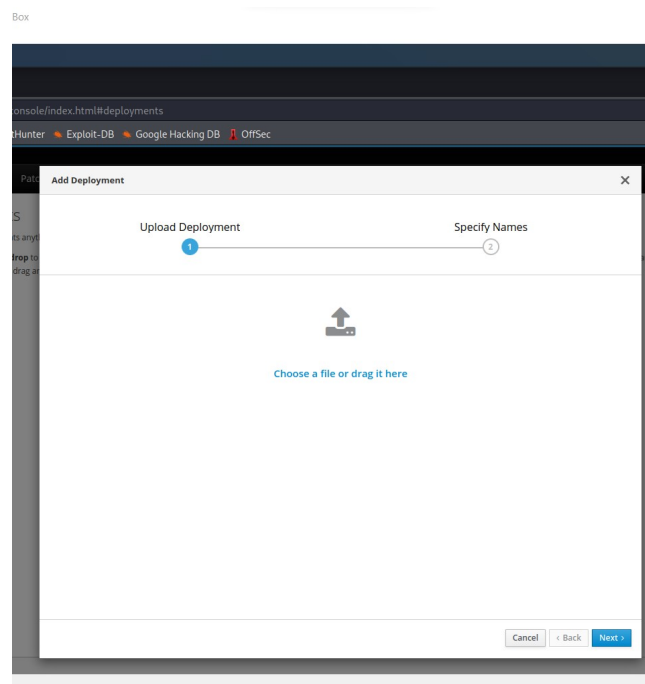
Una vez iniciada la sesión, se mostrará la página principal del panel de administración de WildFly, donde podremos realizar varias configuraciones y despliegues de aplicaciones.



Javier Manzano Gamaza



Elegimos el war que generamos antes



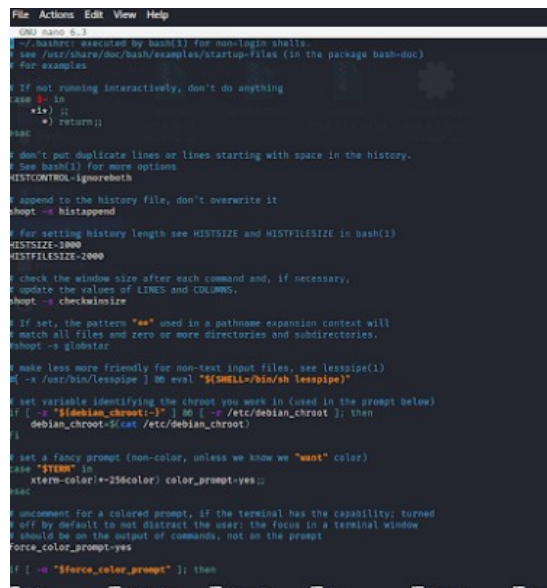
Ponemos el localhost con el puerto correspondiente y ya funcionaria.

Practica 4

Para llevar a cabo esta actividad, seguiremos los siguientes pasos:

Descargaremos el archivo ZIP de Ant desde la siguiente ruta: <http://ant.apache.org/bindownload.cgi>. Descomprimos el archivo en una ubicación adecuada en nuestro sistema.

Configuraremos la variable de entorno ANT_HOME para que apunte a la ubicación donde descomprimos el archivo de Ant. Esto nos permitirá usar Ant desde cualquier ubicación en nuestro sistema.



```
File Actions Edit View Help
GNU nano 2.9.3
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
  *i) ;;
  *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

# If set, the pattern "ee" used in a pathname expansion context will
# match all files and zero or more directories and subdirectories.
shopt -s globstar

# make less more friendly for non-text input files, see lesspipe(1)
[ -x /usr/bin/lesspipe ] && eval "SHELL=\"/usr/bin/lesspipe\""

# set variable identifying the chroot you work in (used in the prompt below)
[ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ] && then
  debian_chroot=$(cat /etc/debian_chroot)
fi

# set a fancy prompt (non-color, unless we know we "want" color)
case "$TERM" in
  xterm-color | *-256color) color_prompt=yes;;
esac

# uncomment for a colored prompt, if the terminal has the capability; turned
# off by default to not distract the user: the focus in a terminal window
# should be on the output of commands, not on the prompt
force_color_prompt=yes

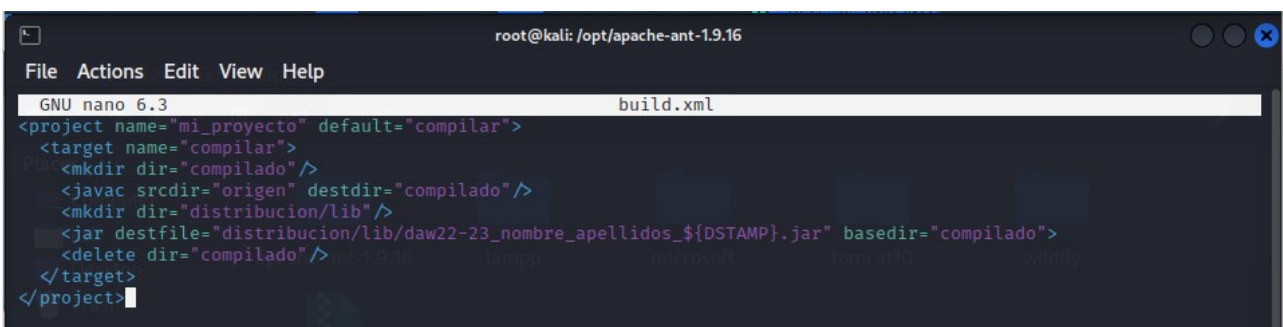
if [ -n "$force_color_prompt" ]; then
  if [ -x "$(command -v tput)" ]; then
    tput setaf 1
  fi
  PS1="tput setaf 1; echo -n 'root@kali: /opt/apache-ant-1.9.16'; tput sgr0; echo -n 'build.xml'"
fi
```

Crearemos el archivo build.xml en la ubicación donde se encuentran nuestros archivos de código fuente.

Agregaremos la configuración necesaria al archivo build.xml para compilar y empaquetar nuestra aplicación. Esta configuración variará dependiendo de la naturaleza de nuestra aplicación y del lenguaje de programación utilizado.

Ejecutaremos el comando "ant" en la terminal desde la ubicación donde se encuentra nuestro archivo build.xml para iniciar el proceso de compilación y empaquetado.

Una vez tenemos el archivo creado, nos pondremos en la ruta donde hemos echo el fichero build.xml y vamos a ejecutar el comando ant



```
root@kali: /opt/apache-ant-1.9.16
File Actions Edit View Help
GNU nano 6.3 build.xml
<project name="mi_proyecto" default="compilar">
  <target name="compilar">
    <mkdir dir="compilado" />
    <javac srcdir="origen" destdir="compilado" />
    <mkdir dir="distribucion/lib" />
    <jar destfile="distribucion/lib/daw22-23_nombre_apellidos_${DSTAMP}.jar" basedir="compilado">
      <delete dir="compilado" />
    </target>
  </project>
```

Cuando lo ejecutas ya saldra el archivo jar.

Practica 5

1. **server.xml:** Este archivo es el archivo de configuración principal de Tomcat y controla la configuración del servidor Tomcat. En este archivo se definen los puertos de red que se utilizarán para la conexión, se establecen los parámetros de seguridad, se definen los hosts virtuales y se configura el acceso a las aplicaciones web.
2. **web.xml:** Este archivo es el archivo de configuración principal de una aplicación web en Tomcat y define la estructura de la aplicación y su configuración. Aquí se configuran los servlets, los filtros, los listeners y otros componentes de la aplicación web.
3. **context.xml:** Este archivo se encuentra en el directorio de contexto de una aplicación web y se utiliza para configurar parámetros específicos del contexto de la aplicación. En este archivo se pueden establecer propiedades como el nombre del contexto, la base de datos, las fuentes de datos, etc.
4. **catalina.properties:** Este archivo se utiliza para configurar los valores predeterminados del servidor Tomcat. Aquí se pueden configurar valores como los directorios base, la configuración de seguridad, la configuración de registro, etc.
5. **tomcat-users.xml:** Este archivo se utiliza para configurar la autenticación y la autorización en Tomcat. Aquí se definen los usuarios y grupos, y se les otorgan permisos de acceso a las aplicaciones web.
6. **logging.properties:** Este archivo se utiliza para configurar el registro de Tomcat. Aquí se pueden configurar los niveles de registro, la ubicación del archivo de registro y otros detalles de registro.