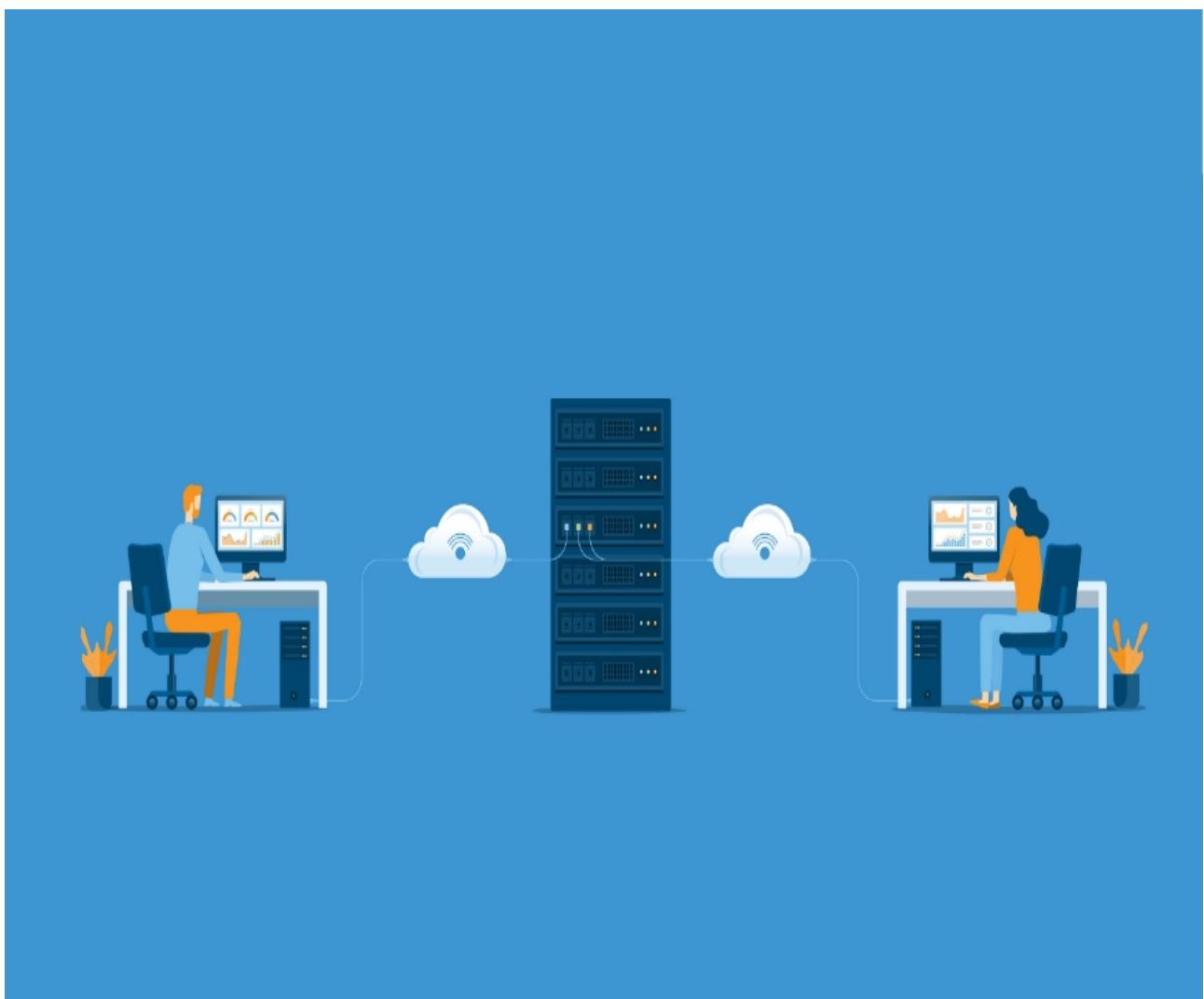


# TAREA UNIDAD 2

## Configuración y administración de servidores de aplicaciones



CFGS DAW. Módulo: Despliegue de Aplicaciones Web (2023/24)  
**Dayro Morales Cruz**

## **Indice**

<b>Ejercicio 1.....</b>	<b>3</b>
<b>Ejercicio 2.....</b>	<b>6</b>
<b>Ejercicio 3.....</b>	<b>8</b>
<b>Ejercicio 4.....</b>	<b>14</b>

# Ejercicio 1

- **Apartado 1.a**

Describe los módulos y enumera los módulos más importantes de Apache y de Tomcat. Y además, describe los principales archivos de configuración de Apache.

- **Módulos de Apache más importantes:**

- **mod\_deflate:** Permite comprimir el contenido antes de enviarlo al navegador usando gzip.
- **mod\_security:** Permite bloquear contenidos sobre la base de datos de forma posterior,.
- **mod\_speling:** Permite realizar la corrección ortográfica en una url.
- **mod\_cache:** Permite un contenido caché introducido a URI.
- **mod\_setenvif:** Permite manejar diferentes variables de entorno.
- **mod\_auth\*:** Permite autenticar casi cualquier módulo.
- **mod\_ssl:** Soporte HTTPS para el servidor web Apache.
- **mod\_proxy:** Permite a Apache convertirse en un Reenviador (Proxy inverso).
- **mod\_benchmark:** Permite almacenar información de referencia y proporciona gráficos.

- **Módulos de Tomcat más importantes:**

- **bin:** Contiene scripts y ejecutables para el arranque y cierre de Tomcat.
- **common:** Contiene clases comunes que pueden utilizar Catalina y las aplicaciones web.
- **conf:** Contiene ficheros XML y los correspondientes DTD para la configuración de Tomcat.
- **logs:** Contiene los logs de Catalina y de las aplicaciones.
- **server:** Contiene clases utilizadas solamente por Catalina.

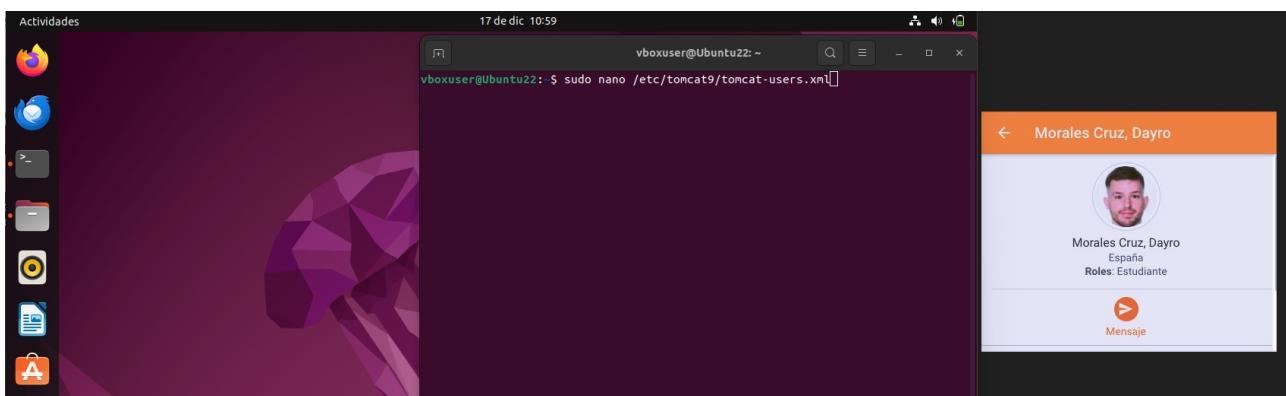
- **Archivos de configuración de Apache más importantes:**
  - **httpd.conf**: Es el archivo principal de configuración de Apache. En algunas versiones de Apache se utiliza el archivo apache2.conf.
  - **.htaccess**: Estos archivos permiten realizar cambios en la configuración en contexto directorio.
  - **apache2.conf**: Este archivo contiene los datos básicos de la configuración de Apache

- **Apartado 1.b**

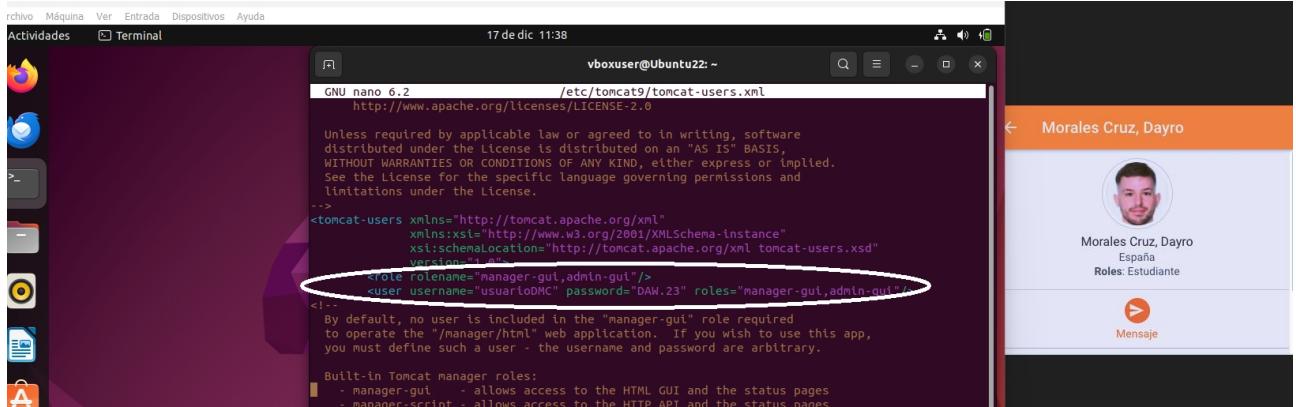
Utiliza la **instalación** del servidor **Apache Tomcat** que se realizó en la tarea online 1 y realiza el siguiente apartado:

- **Configura** Tomcat, para que puedas acceder vía web con un **usuario** al **gestor de aplicaciones** de Tomcat.
- **Crea un usuario** que permita acceder a la interfaz html y permita el acceso a la interfaz de texto sin formato llamado **usuarioXXX** donde las X sean las 3 últimas cifras de tu nº de DNI, y password **DAW.23**.

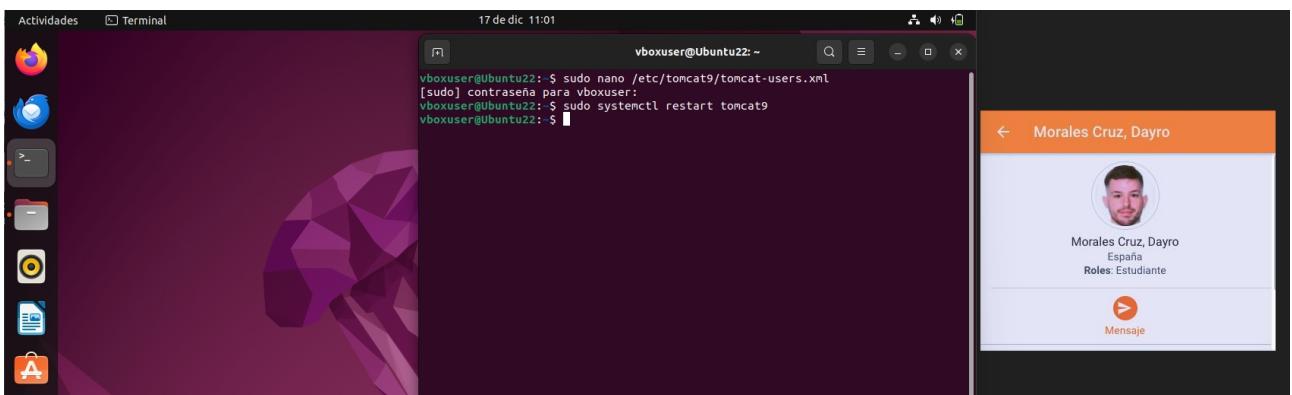
Introducimos el comando con sudo para acceder a la carpeta donde está el archivo *tomcat-users.xml*, en nuestro caso está en el directorio */etc/tomcat9/tomcat-users.xml*.



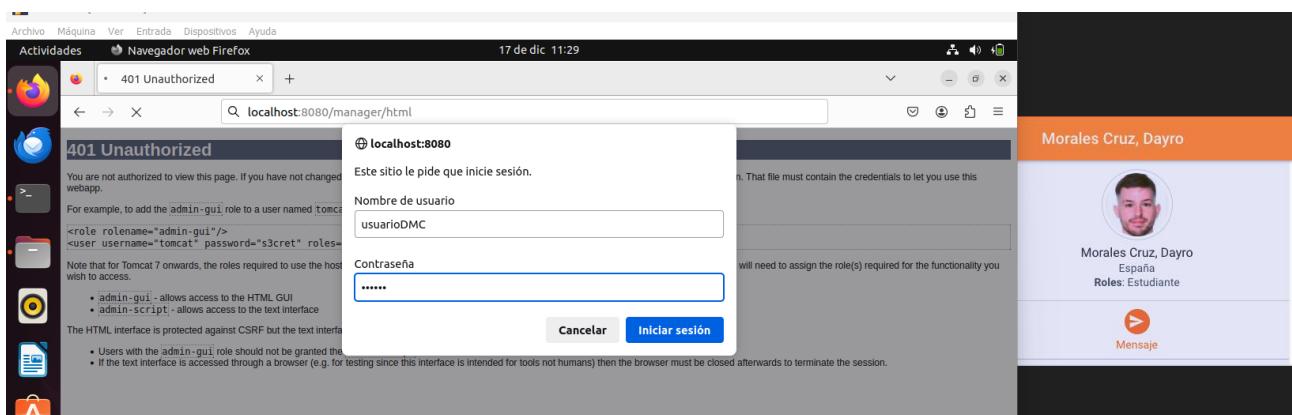
Ingresamos en el documento y le añadimos la siguiente entrada  
**<user username="usuarioDMC" password="DAW.23" roles="manager-gui,admin-gui>** para que no pida usuario y contraseña para poder entrar en tomcat



Y reiniciamos el sistema con el comando **sudo systemctl restart tomcat9**



Ahora al ingresar en el navegador localhost/manager/html nos aparecerá la siguiente ventana donde ingresaremos nuestro usuario y contraseña previamente descritas.



Mostramos la página del gestor de aplicaciones web de tomcat tras ingresar los datos de usuario y contraseña

Ruta	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Ninguno especificado		true	0	<input type="button" value="Arrancar"/> <input type="button" value="Parar"/> <input type="button" value="Recargar"/> <input type="button" value="Replegar"/> <input type="button" value="Expirar sesiones sin trabajar ≥ 30 minutos"/>
/docs	Ninguno especificado	Tomcat Documentation	true	0	<input type="button" value="Arrancar"/> <input type="button" value="Parar"/> <input type="button" value="Recargar"/> <input type="button" value="Replegar"/> <input type="button" value="Expirar sesiones sin trabajar ≥ 30 minutos"/>

## Ejercicio 2

Se realizará a partir de un programa de calculadora dado por el profesor en el foro de la unidad 2. Prueba la aplicación antes de desplegarla en NetBeans y despliégalos en el servidor Tomcat que instalamos en la tarea online 1. Realiza cada uno de los siguientes pasos:

- Genera un archivo **.war** con el contenido de la aplicación web. Este apartado es opcional, ya que el archivo .war también se encuentra ya generado dentro de la carpeta 'dis' del .rar proporcionado.
- Despliega el archivo war en el servidor **Apache Tomcat** de dos formas distintas:
  - Copiando directamente la carpeta en la ruta adecuada.

Mediante el comando **sudo cp /home/vboxuser/Escritorio/Calculadora/dist/Calculadora.war** (ruta hasta el fichero, en nuestro caso el escritorio) **/etc/tomcat9/webapps** (donde se ubica nuestro tomcat)

- Desplegando el archivo war desde el entorno web de Apache (página web del servidor de aplicaciones).

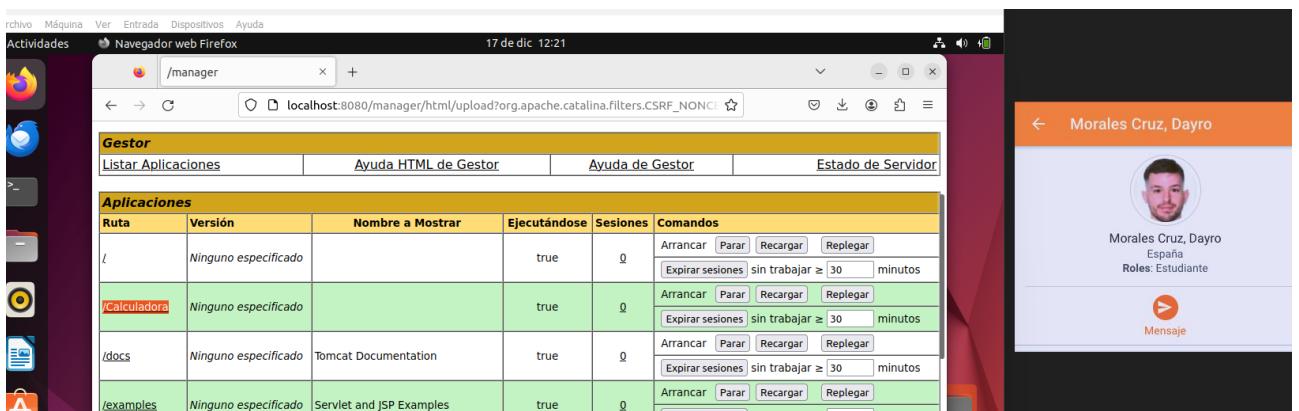
Primero ingresaremos a **localhost:8080/manager/html** e ingresamos nuestro **usuario** y **contraseña** establecidas en el paso anterior. Una vez dentro vamos a listar aplicaciones y vamos a subir nuestro archivo .war directamente desde tomcat



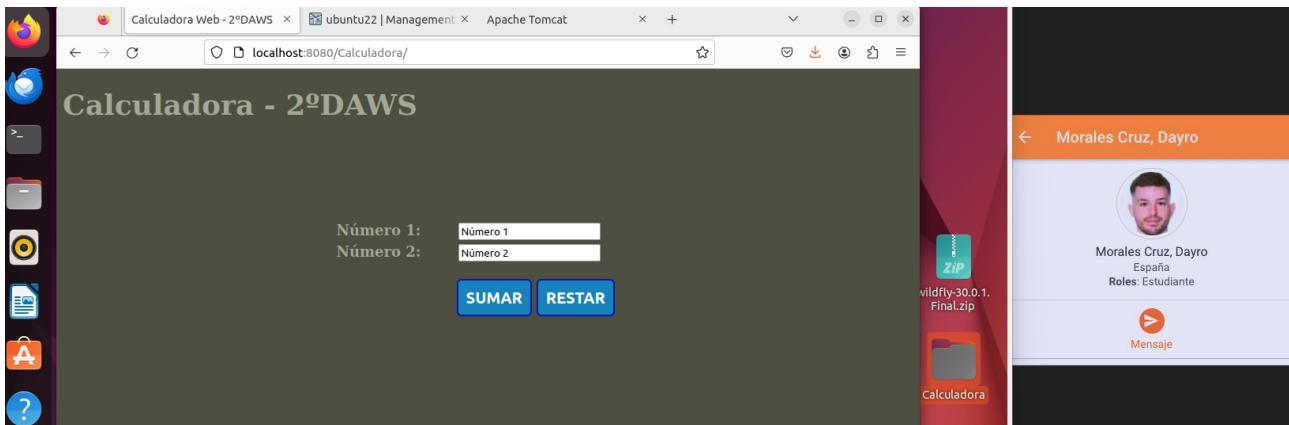
pulsamos en examinar y vamos a la ubicación donde se encuentra nuestro .war, en nuestro caso en el escritorio y desplegamos



Tras desplegar la aplicación comprobamos que está entre nuestras aplicaciones y podemos arrancar, parar, recargar o replegar nuestro programa



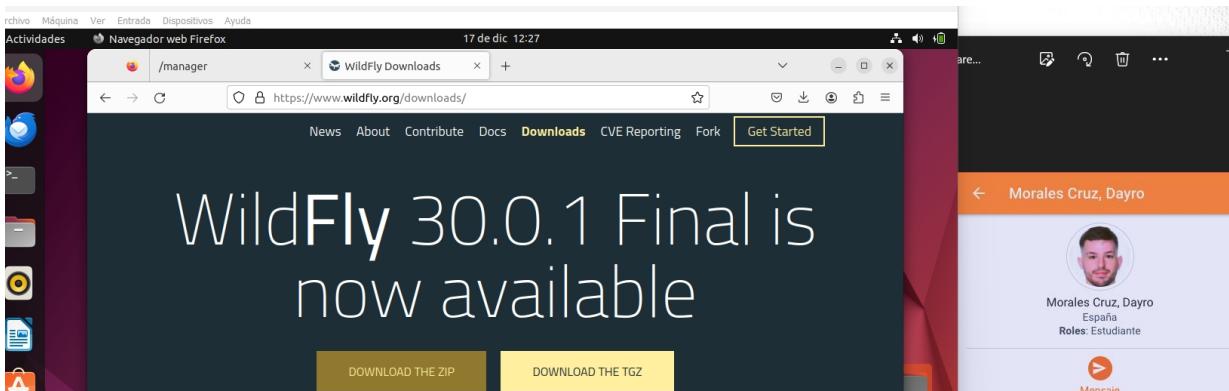
Comprobamos que todo esta correcto



## Ejercicio 3

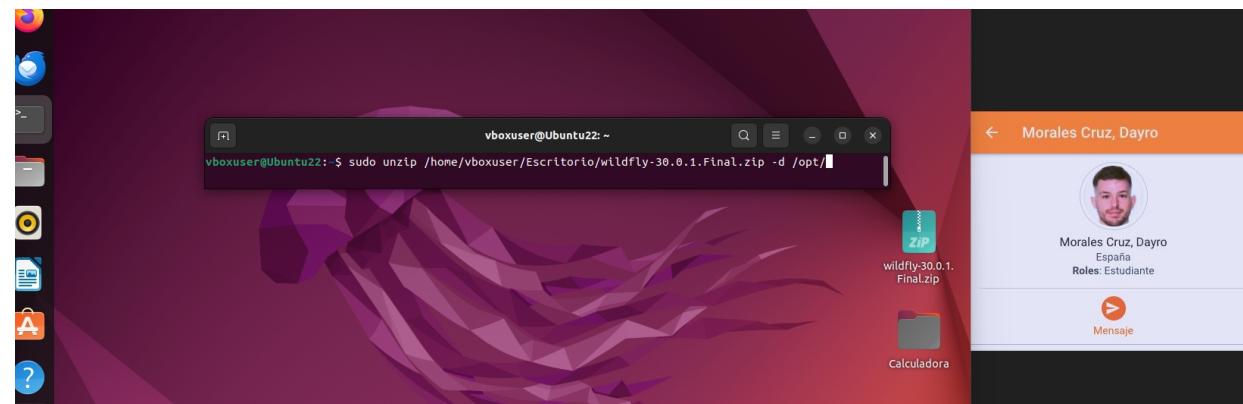
- Descárgate e instala el servidor de aplicaciones **Wildfly** en la máquina virtual de Ubuntu 20/22.

Descargamos nuestra versión final desde la página oficial de descargas de [Wildfly](#) y descargamos la versión .zip

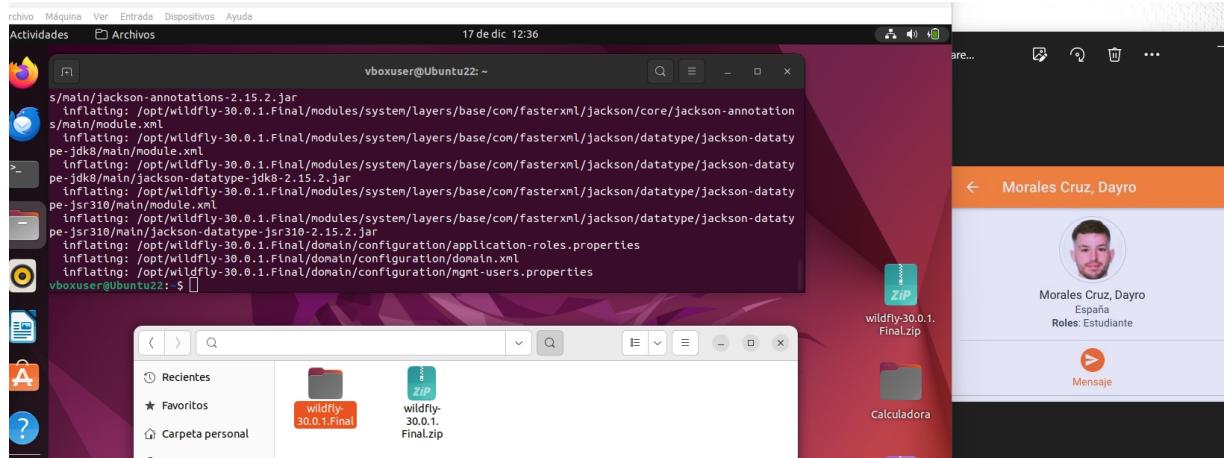


Descomprimimos nuestro archivo en la terminal mediante el comando **sudo unzip /home/vboxuser/Escritorio/wildfly-30.0.1.Final.zip -d /opt/**

Lo moveremos a esta carpeta ya la que en nuestro caso se encuentra vacía.



Nos aseguramos de que el paquete se descomprimió correctamente en nuestra carpeta

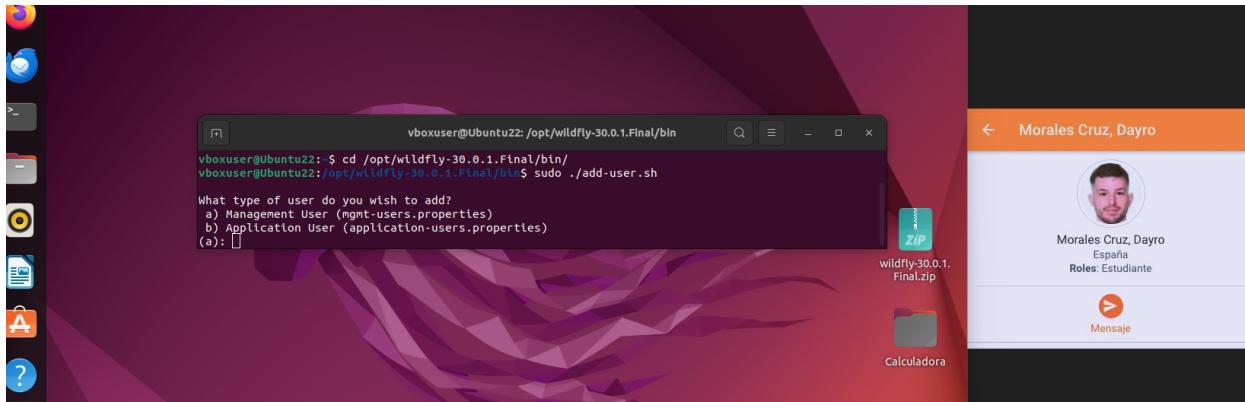


- Crea un usuario para poder acceder vía web al panel de administración de aplicaciones, con el script **add-user.sh**.

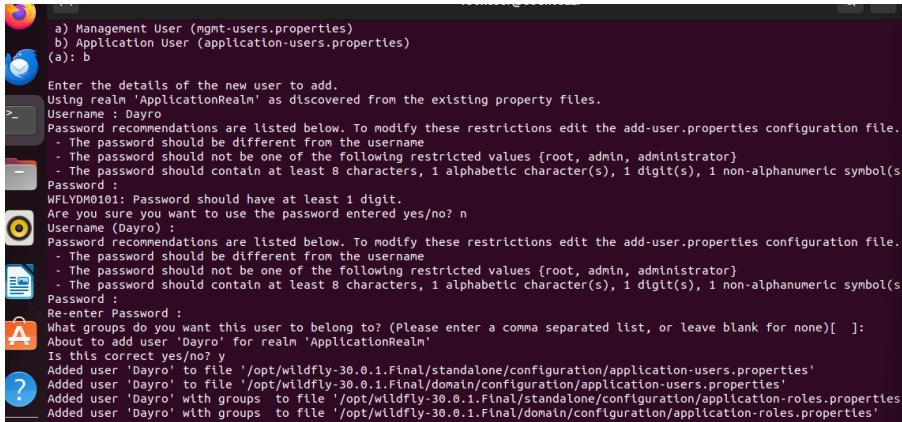
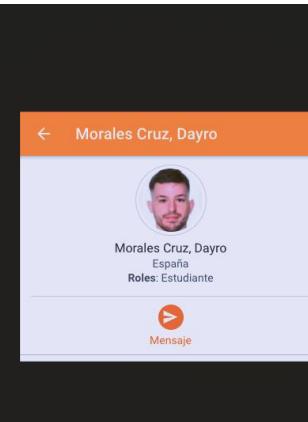
En la terminal, accedemos al archivo **add-user.sh** ubicado en la carpeta `/bin/` de wildfly mediante el comando

**cd /opt/wildfly-30.0.1.Final/bin/**

Y una vez en la carpeta abrimos el archivo y nos guaremos por el asistente



Continuamos hasta que creamos nuestro Management user con nombre **Dayro** y nuestra contraseña

```

a) Management User (mgmt-users.properties)
b) Application User (application-users.properties)
(a): b

Enter the details of the new user to add.
Using realm 'ApplicationRealm' as discovered from the existing property files.

Username : Dayro
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)

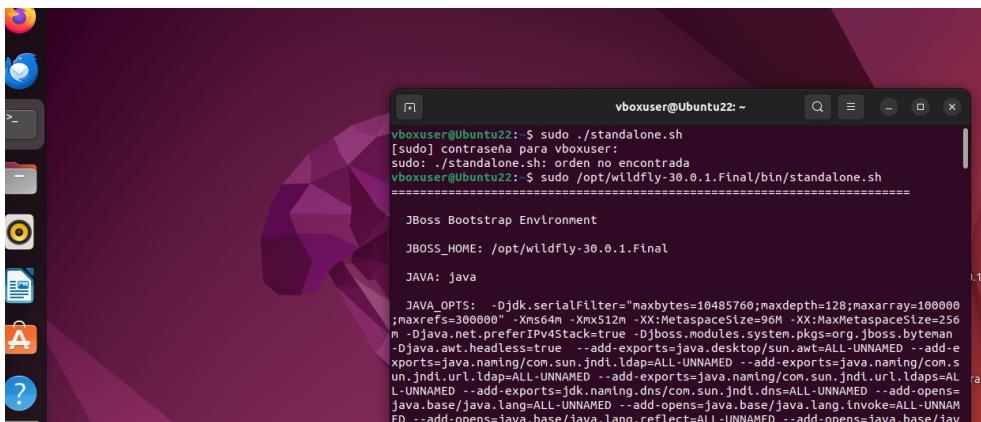
Password :
WFLYDM0101: Password should have at least 1 digit.
Are you sure you want to use the password entered yes/no? n
Username (Dayro) :
Password recommendations are listed below. To modify these restrictions edit the add-user.properties configuration file.
- The password should be different from 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 :
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 'Dayro' for realm 'ApplicationRealm'
Is this correct yes/no? y
Added user 'Dayro' to file '/opt/wildfly-30.0.1.Final/standalone/configuration/application-users.properties'
Added user 'Dayro' to file '/opt/wildfly-30.0.1.Final/domain/configuration/application-users.properties'
Added user 'Dayro' with groups to file '/opt/wildfly-30.0.1.Final/standalone/configuration/application-roles.properties'
Added user 'Dayro' with groups to file '/opt/wildfly-30.0.1.Final/domain/configuration/application-roles.properties'

```

- Accede al **panel de administración** y despliega en el servidor el **war** generado en el ejercicio 2.

Primero iniciaremos el servidor de Wildfly mediante el comando  
**sudo /opt/wildfly-30.0.1.Final/bin/standalone.sh**

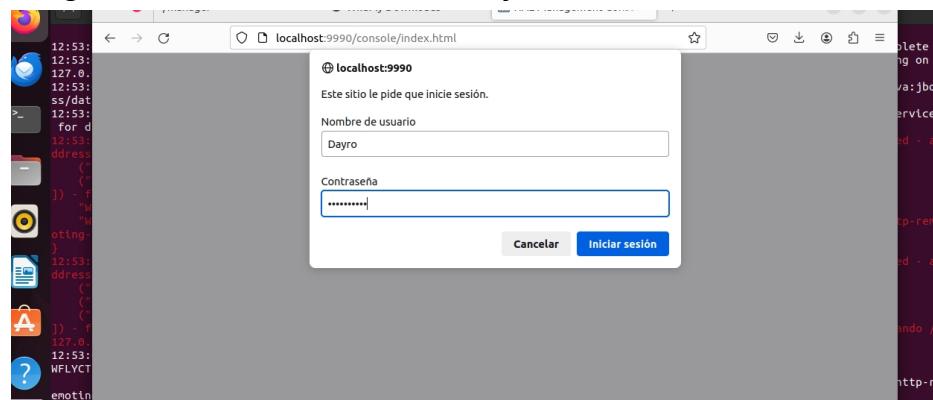
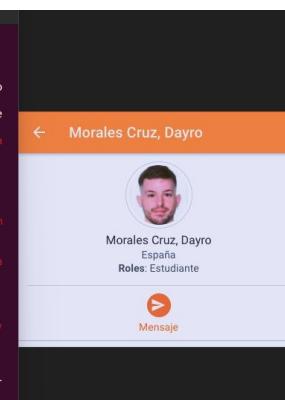



```

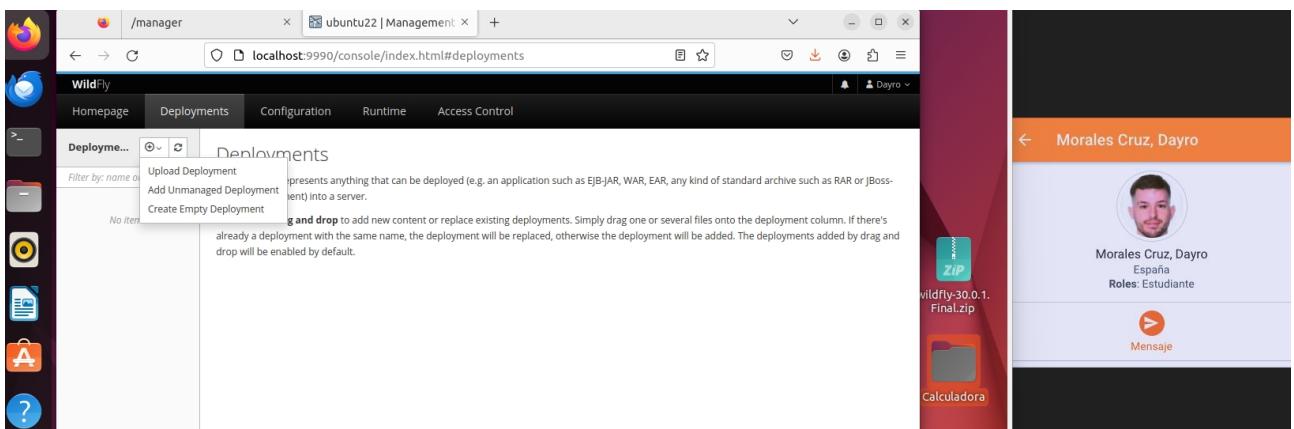
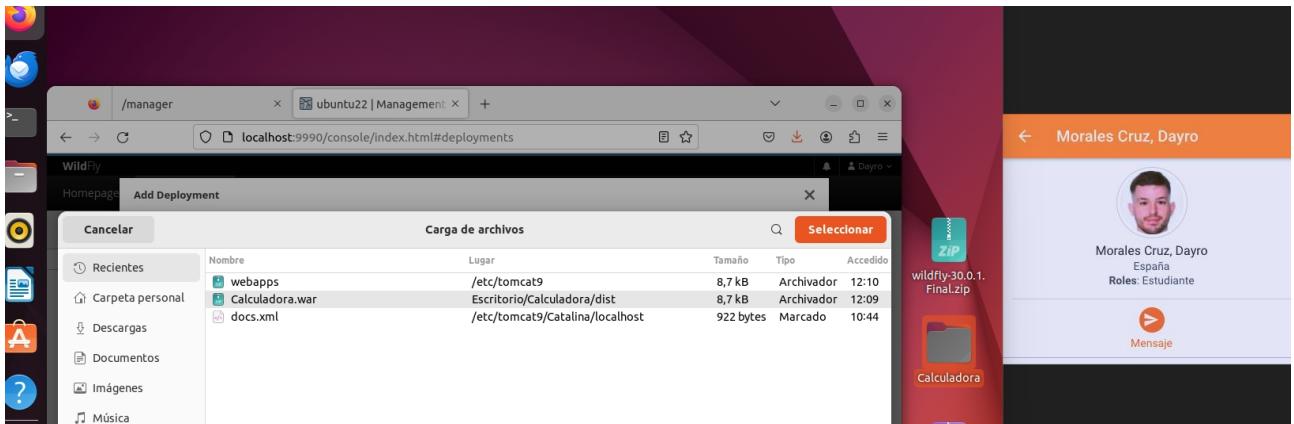
vboxuser@Ubuntu22: $ sudo ./standalone.sh
[sudo] contraseña para vboxuser:
sudo: ./standalone.sh: orden no encontrada
vboxuser@Ubuntu22: $ sudo /opt/wildfly-30.0.1.Final/bin/standalone.sh
=====
JBoss Bootstrap Environment
JBOSS_HOME: /opt/wildfly-30.0.1.Final
JAVA: java
JAVA_OPTS: -Djboss.serialfilter="maxbytes=10485760;maxdepth=128;maxarray=100000
;maxrefs=300000" -Xms64M -Xmx512m -XX:MetaspaceSize=96M -XX:MaxMetaspaceSize=256
m -Djava.net.preferIPv4Stack=true -Djboss.modules.system.pkgs=org.jboss.byteman
-Djava.awt.headless=true --add-exports=java.desktop/sun.awt=ALL-UNNAMED --add-e
xports=java.naming/com.sun.jndt.ldap=ALL-UNNAMED --add-exports=java.naming/com.s
un.jndi.url.ldap=ALL-UNNAMED --add-exports=java.naming/com.sun.jndt.url.ldap=AL
L-UNNAMED --add-exports=jdk.naming.dns/com.sun.jndt.dns=ALL-UNNAMED --add-opens=
java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.lang.invoke=ALL-UNNAM
ED --add-opens=java.base/java.lang.reflect=ALL-UNNAMED --add-opens=java.base/jav

```

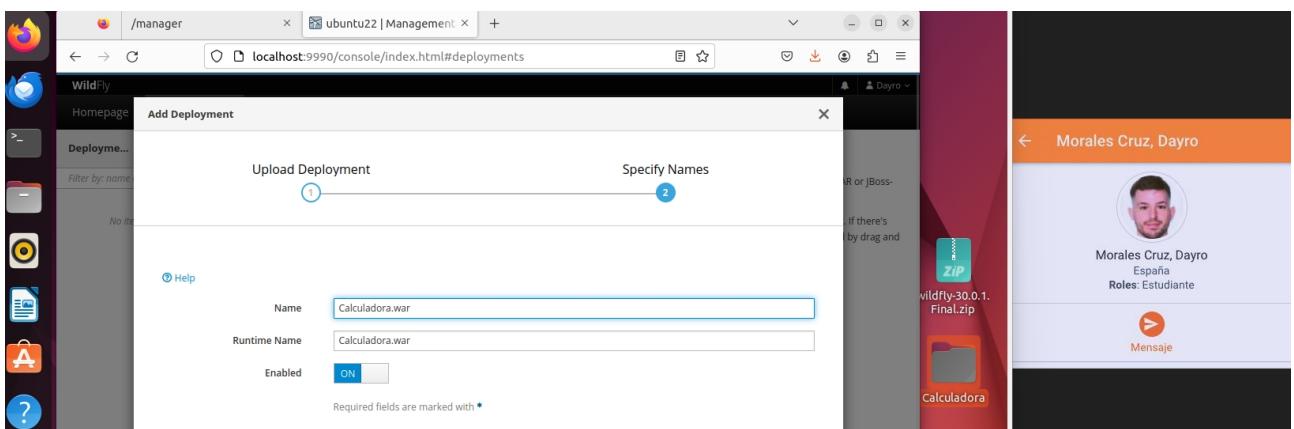
Ingresamos al panel de administración mediante navegador ingresando  
**localhost:9990**  
e ingresaremos nuestro usuario y contraseña

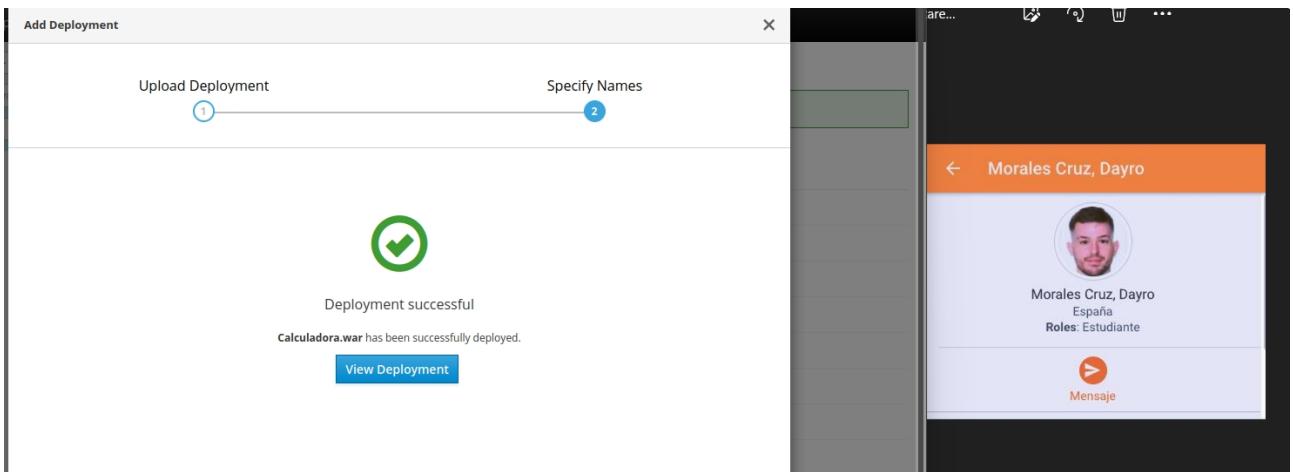



A continuación dentro de la página añadiremos nuestro .war y lo subiremos.



Continuamos subiendo nuestro archivo.



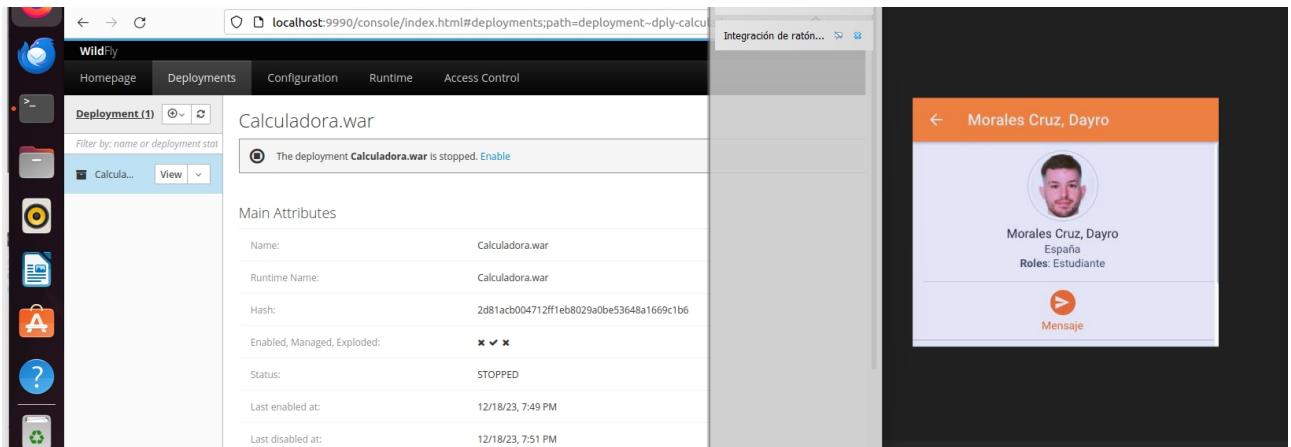


Una vez cargado correctamente comprobamos que funciona correctamente

- Para el servidor.

Para parar el servidor pulsamos sobre disable y lo paramos completamente

Comprobamos que está completamente parado, si nos fijamos en el apartado Status aparece STOPPED



The screenshot shows the WildFly management console interface. On the left, there's a sidebar with various icons. The main area has tabs for 'Homepage', 'Deployments', 'Configuration', 'Runtime', and 'Access Control'. Under 'Deployments', it shows one deployment named 'Calculadora.war'. A message box says 'The deployment Calculadora.war is stopped. Enable'. Below this, under 'Main Attributes', there are several entries:

Name:	Value
Runtime Name:	Calculadora.war
Hash:	2d81acb004712ff1eb8029a0be53648a1669c1b6
Enabled, Managed, Exploded:	x ✓ x
Status:	STOPPED
Last enabled at:	12/18/23, 7:49 PM
Last disabled at:	12/18/23, 7:51 PM

Para este apartado debes realizar:

- Documento de texto con las capturas de pantalla necesarias, que sean legibles, indicativas de que has realizado lo que se pide, que demuestren que todo ha funcionado correctamente, y en las que aparezca tu perfil de la plataforma de enseñanza.

## Ejercicio 4

Automatiza una tarea con la aplicación **Ant**. Para ello:

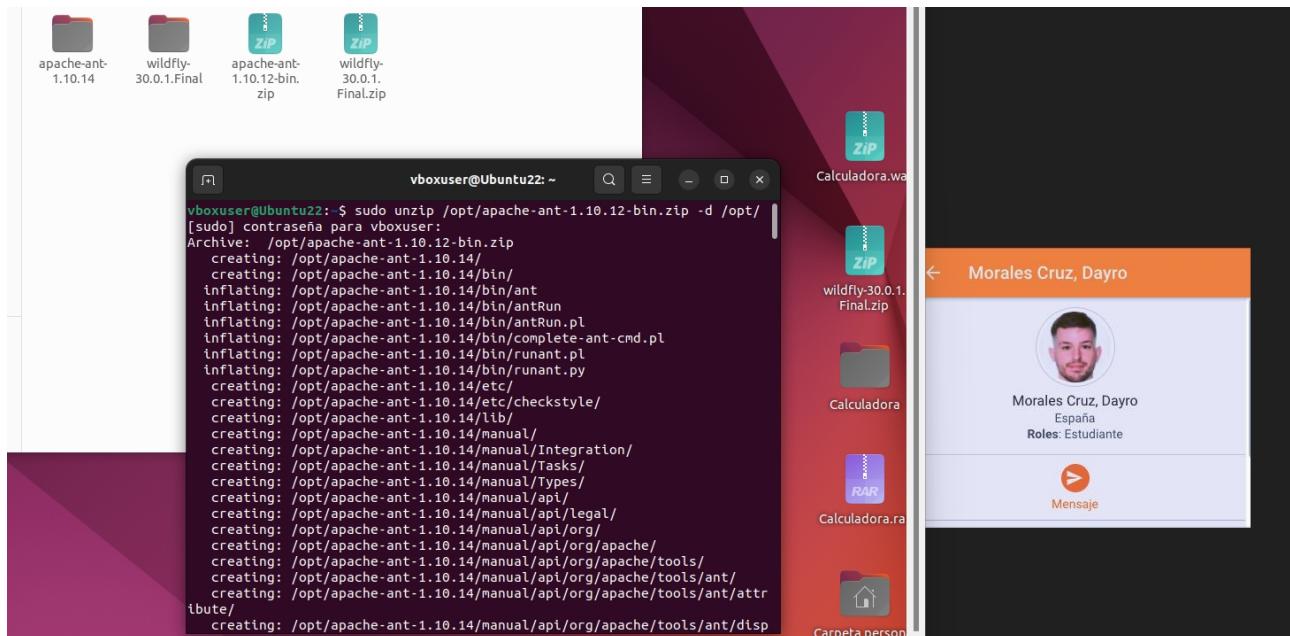
- Instala la aplicación **Ant** en el servidor de forma manual (*sin usar los repositorios de Ubuntu*) desde la [web oficial](#).

Primero moveremos nuestro archivo descargado a nuestra carpeta /opt/  
**sudo mv /home/vboxuser/Escritorio/apache-ant-1.10.14-bin.zip /opt/**

y lo descomprimimos con el comando

**sudo unzip /opt/ apache-ant-1.10.14-bin.zip -d /opt/**

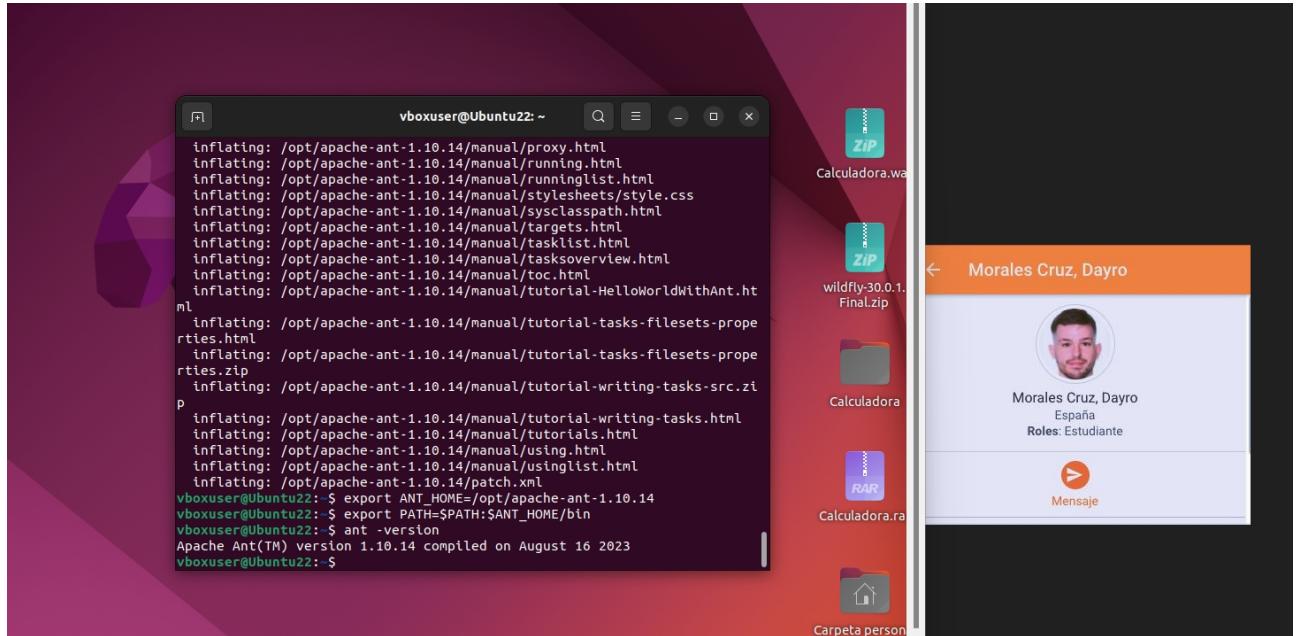
Y comprobamos que se descomprimió correctamente



Mediante el comando **export ANT\_HOME=/opt/apache-ant-1.10.14** establecemos la variable que apunte a la carpeta del programa

Mediante el comando **export PATH=\$PATH:\$ANT\_HOME/bin** nos permitirá ejecutar ant desde cualquier lugar

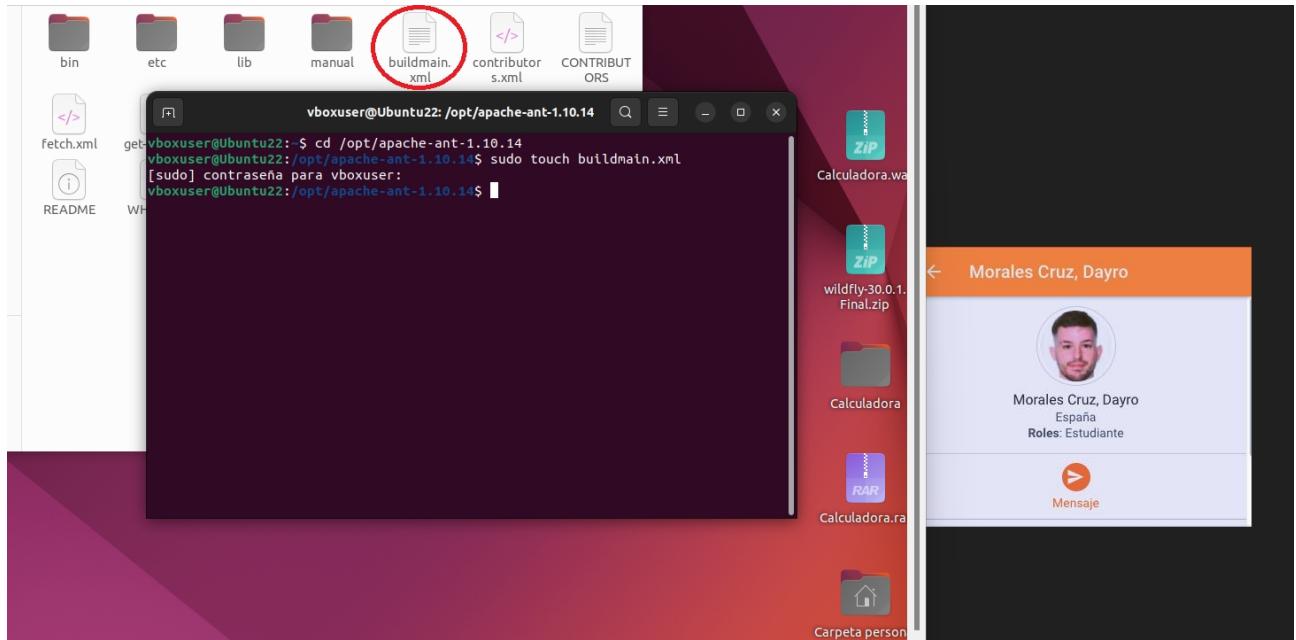
Por ultimo ejecutamos **ant -version** y confirmamos que está en la version correcta



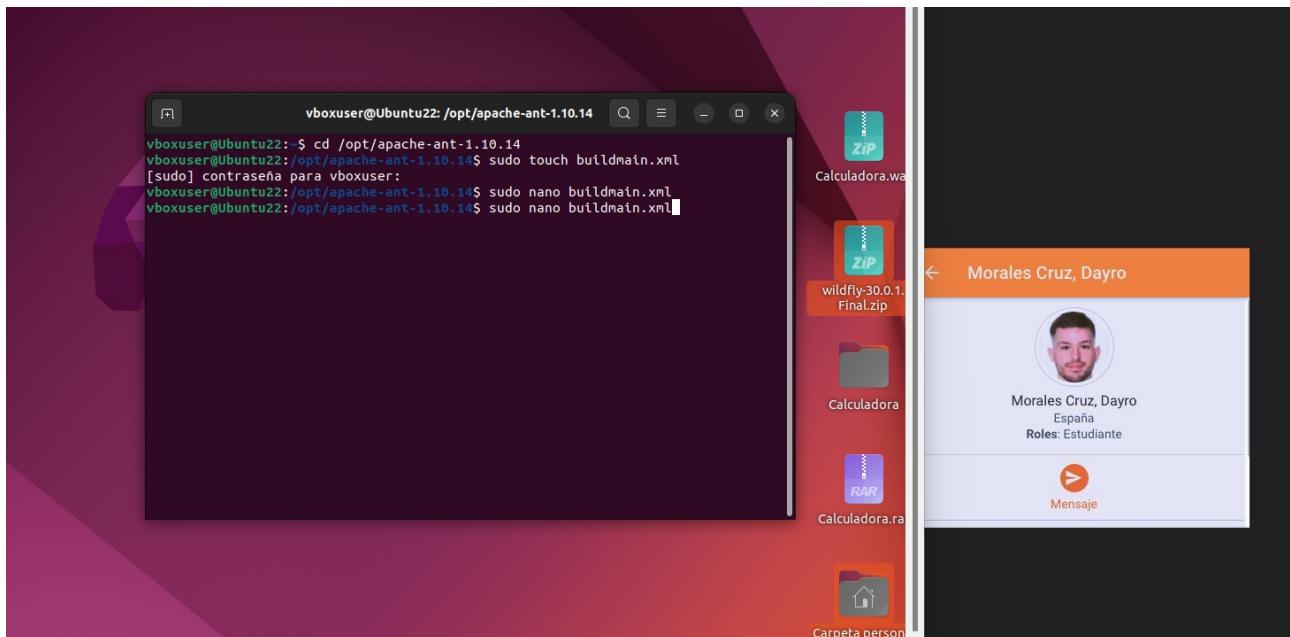
- Elabora un fichero **buildmain.xml** de forma que realice las siguientes acciones:
  - Cree un directorio llamado "compilado"
  - Debe compilar todos los archivos java del directorio "origen" y almacenarlos en el directorio "compilado".
  - Cree un directorio en "distribucion/lib"
  - En este directorio se debe crear un fichero "jar" de forma que contenga todas las clases compiladas del directorio "compilado".
  - El archivo jar se nombrará "daw23-24\_nombre\_apellidos\_yyyyMMdd.jar", donde yyyyMMdd corresponde a la fecha del sistema.
  - Borre posteriormente el directorio "compilado" y todo su contenido.

Primero ingresado en la carpeta de nuestro programa con el comando  
**cd /opt/apache-ant-1.10.14**

Dentro de la carpeta creamos nuestro archivo **buildmain.xml** con el comando  
**sudo touch buildmain.xml**



A continuación mediante el comando **sudo nano buildmain.xml**



Ingresamos el siguiente código en el archivo **buildmain.xml** comentado para identificar cada parte:

```
<project name="Proyecto" default="dist" basedir=".">
    <!-- Define las propiedades del proyecto -->
    <property name="src.dir" value="origen"/>
    <property name="build.dir" value="compilado"/>
    <property name="dist.dir" value="distribucion"/>
    <property name="jar.dir" value="${dist.dir}/lib"/>
    <property name="main-class" value="nombre_apellidos"/>
    <property name="jar.file" value="${jar.dir}/daw23-24_${main-
class}_yyyyMMdd.jar"/>

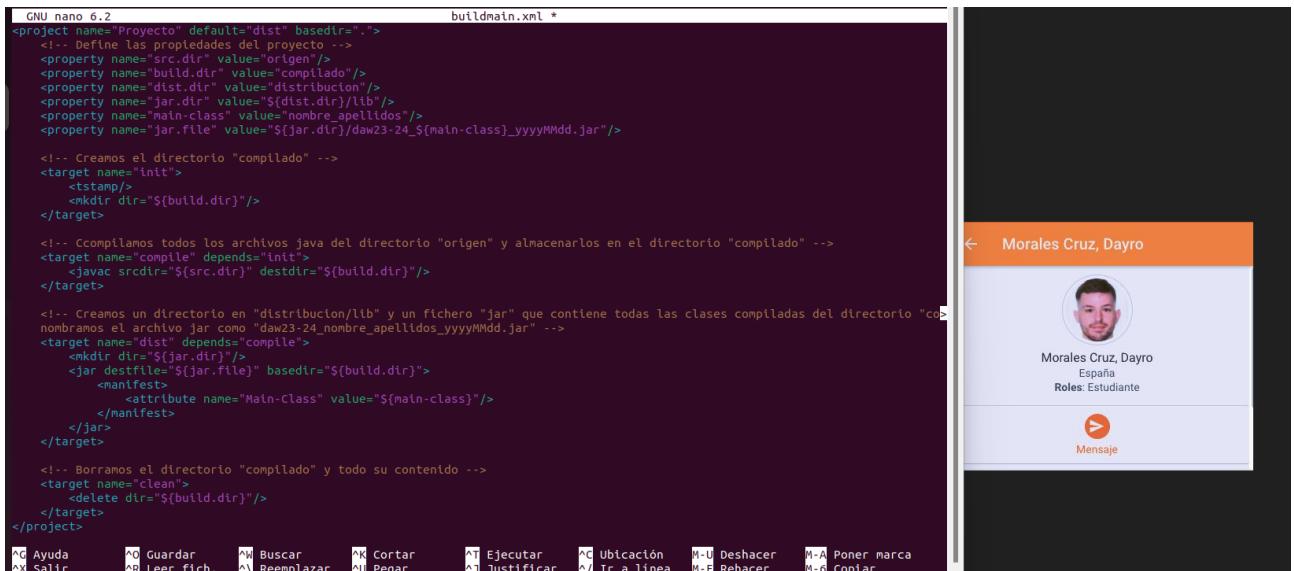
    <!-- Creamos el directorio "compilado" -->
    <target name="init">
        <tstamp/>
        <mkdir dir="${build.dir}" />
    </target>

    <!-- Ccompilamos todos los archivos java del directorio
    "origen" y almacenarlos en el directorio "compilado" -->
    <target name="compile" depends="init">
        <javac srcdir="${src.dir}" destdir="${build.dir}" />
    </target>

    <!-- Creamos un directorio en "distribucion/lib" y un fichero
    "jar" que contiene todas las clases compiladas del directorio
    "compilado"
    nombramos el archivo jar como "daw23-
    24_nombre_apellidos_yyyyMMdd.jar" -->
    <target name="dist" depends="compile">
        <mkdir dir="${jar.dir}" />
        <jar destfile="${jar.file}" basedir="${build.dir}" />
        <manifest>
            <attribute name="Main-Class" value="${main-
class}" />
        </manifest>
        </jar>
    </target>

    <!-- Borramos el directorio "compilado" y todo su contenido -->
    <target name="clean">
        <delete dir="${build.dir}" />
    </target>
</project>
```

una vez terminado guardamos y salimos.



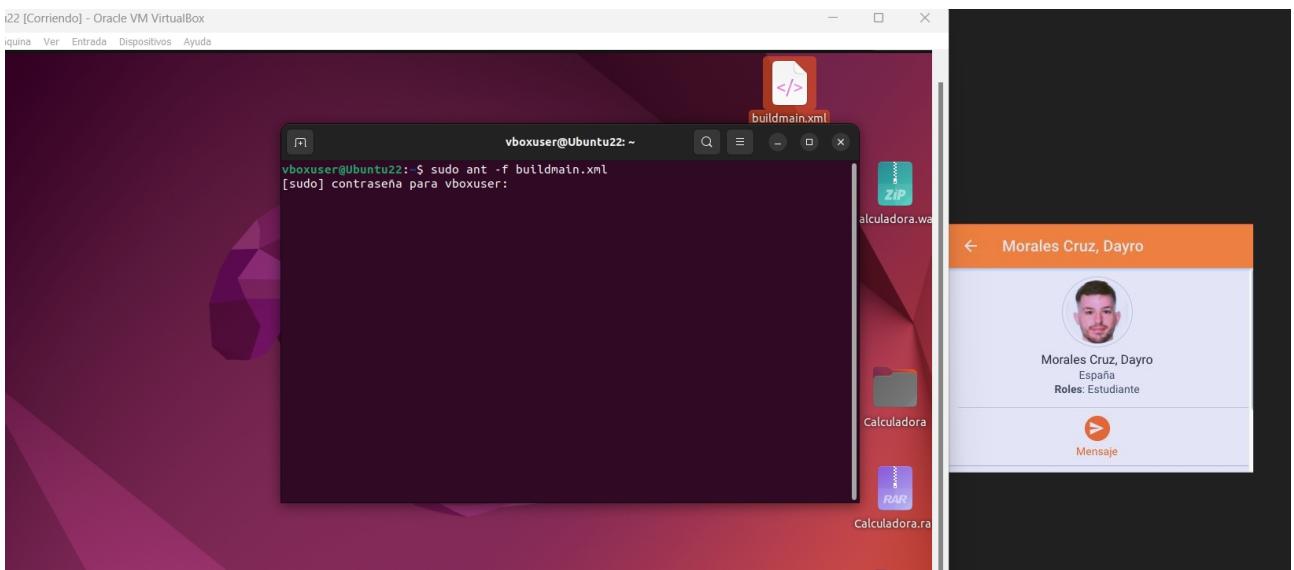
```
GNU nano 6.2          buildmain.xml *
<project name="Proyecto" default="dist" basedir=".">
  <!-- Define las propiedades del proyecto -->
  <property name="src.dir" value="origen"/>
  <property name="build.dir" value="compilado"/>
  <property name="dist.dir" value="distribucion"/>
  <property name="jar.dir" value="${dist.dir}/lib"/>
  <property name="main-class" value="nombre_apellidos"/>
  <property name="jar.file" value="${jar.dir}/daw23-24_${main-class}_yyyyMMdd.jar"/>
  <!-- Creamos el directorio "compilado" -->
  <target name="init">
    <tstamp/>
    <mkdir dir="${build.dir}"/>
  </target>
  <!-- Ccompilamos todos los archivos java del directorio "origen" y almacenarlos en el directorio "compilado" -->
  <target name="compile" depends="init">
    <javac srcdir="${src.dir}" destdir="${build.dir}"/>
  </target>
  <!-- Creamos un directorio en "distribucion/lib" y un fichero "jar" que contiene todas las clases compiladas del directorio "origen". nombramos el archivo jar como "daw23-24_nombre_apellidos_yyyyMMdd.jar" -->
  <target name="dist" depends="compile">
    <mkdir dir="${jar.dir}"/>
    <jar destfile="${jar.file}" basedir="${build.dir}"/>
    <manifest>
      <attribute name="Main-Class" value="${main-class}"/>
    </manifest>
  </jar>
  </target>
  <!-- Borramos el directorio "compilado" y todo su contenido -->
  <target name="clean">
    <delete dir="${build.dir}"/>
  </target>
</project>
```

Ahora nos situamos en la carpeta de ant

**cd /opt/apache-ant-1.10.14**

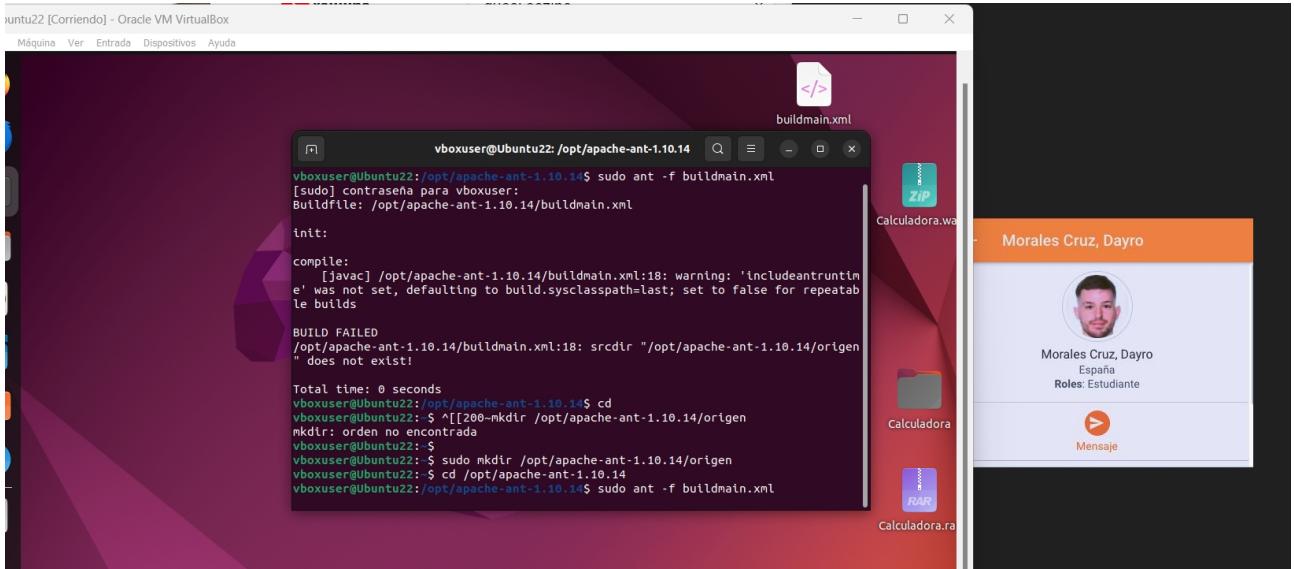
Y ejecutamos el archivo con el comando

**ant -f buildmain.xml**

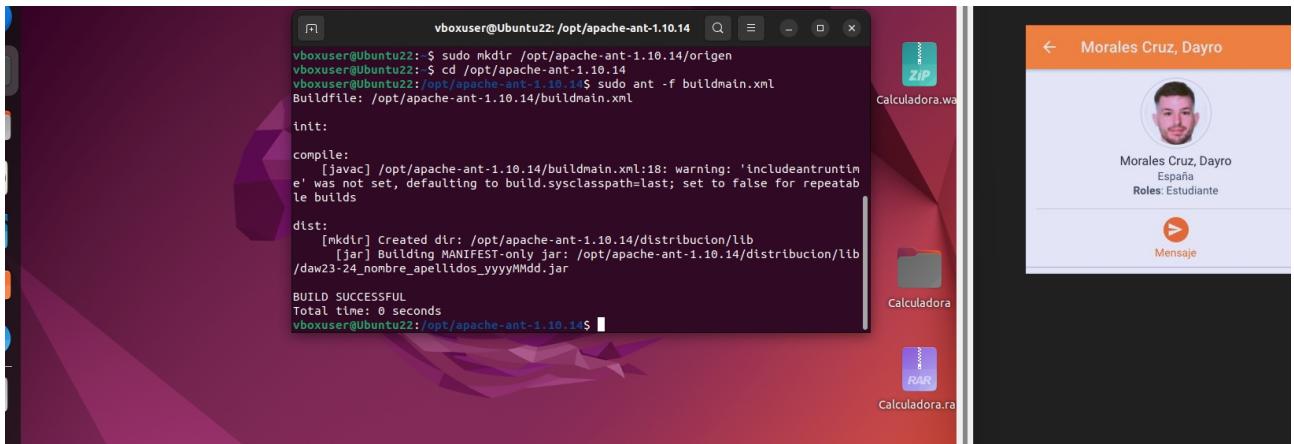


```
i22 [Corriendo] - Oracle VM VirtualBox
quina Ver Entrada Dispositivos Ayuda
vboxuser@Ubuntu22: ~          buildmain.xml
vboxuser@Ubuntu22: $ sudo ant -f buildmain.xml
[sudo] contraseña para vboxuser:
```

Nos salta el siguiente fallo donde nos comenta que no existe la carpeta origen



Lo arreglamos creando la carpeta mediante **sudo mkdir /opt/apache-ant-1.10.14/origen** y volvemos a ejecutar nuestro archivo y vemos que se terminó correctamente



Para este apartado debes realizar:

- Documento de texto con las capturas de pantalla necesarias, que sean legibles, indicativas de que has realizado lo que se pide, que demuestren que todo ha funcionado correctamente, y en las que aparezca tu perfil de la plataforma de enseñanza.
- Archivo buildmain.xml generado.