1- 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?

La estructura estándar definida por la especificación Servlet 2.2 es la siguiente:

```
NombreApp/
├── WEB-INF/
│   ├── web.xml       # Archivo de configuración principal
│   ├── classes/      # Clases Java compiladas (servlets, etc.)
│   └── lib/          # Bibliotecas externas (archivos JAR)
├── META-INF/         # Metadatos adicionales (opcional)
├── index.jsp         # Archivos JSP/HTML/JS/CSS accesibles públicamente
└── recursos/         # Imágenes, CSS, JS, etc.
```

2- 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. Busca otra aplicación para automatizar tareas que sea más actual que Ant e indica brevemente sus características principales.

```
<project name="App" default="deploy">  <!-- Define el proyecto -->
  <property name="src" location="src"/> <!-- Define variables -->
  <target name="compile">             <!-- Define una tarea -->
    <javac srcdir="${src}" destdir="build"/> <!-- Compila código -->
  </target>
  <target name="jar" depends="compile">
    <jar destfile="app.war" basedir="build"/> <!-- Crea archivo WAR -->
  </target>
</project>
```

Herramienta más moderna: Gradle

- Gestión de dependencias integrada (Maven Central, JCenter).
- Soporte para compilación incremental y multi-proyecto.

Ejemplo de build.gradle:

```
plugins {
    id 'war' // Plugin para aplicaciones web
}
repositories {
    mavenCentral()
}
dependencies {
    implementation 'javax.servlet:javax.servlet-api:4.0.1'
}
```

3- Utiliza la instalación de Tomcat que has hecho en la primera tarea y realiza alguna prueba sencilla con un servlet y con una jsp. Comprueba desde el navegador que se ejecutan ambos correctamente.

* Ver práctica 1: Instalación y configuración básica de Tomcat en Ubuntu

Actualizar máquina e instalar Tomcat. Instalo tomcat 10 porque estoy utilizando Ubuntu 24.

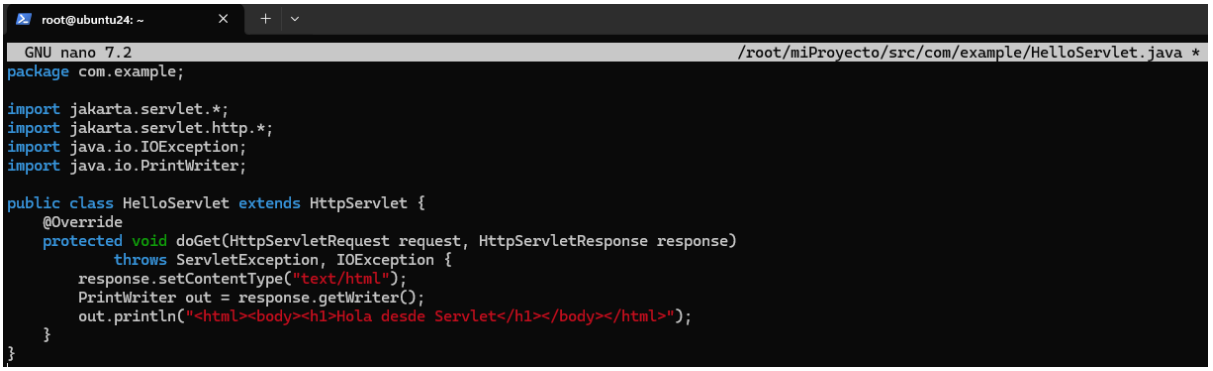
sudo apt update && sudo apt install tomcat10 libtomcat10-java openjdk-21-jdk-headless -y

Crear una carpeta para la aplicación, por ejemplo miapp, dentro de la carpeta de despliegue de Tomcat (usualmente /var/lib/tomcat10/webapps/):

sudo mkdir /var/lib/tomcat10/webapps/miapp
sudo mkdir -p /var/lib/tomcat10/webapps/miapp/WEB-INF/classes

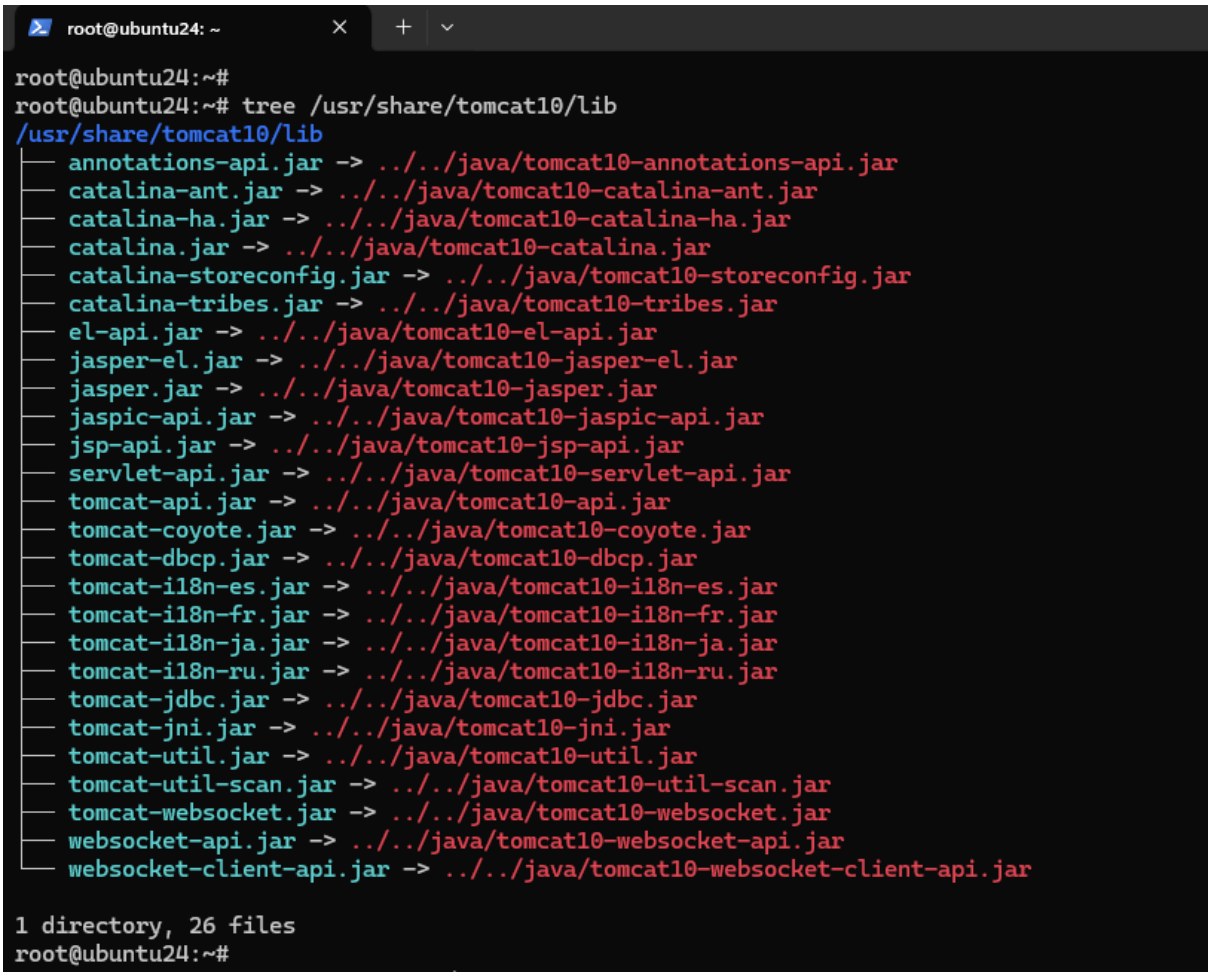
Crear un servlet. Crear un archivo llamado HelloServlet.java con el siguiente contenido en ~/miProyecto/src/com/example/HelloServlet.java

mkdir -p ~/miProyecto/src/com/example
nano ~/miProyecto/src/com/example/HelloServlet.java




```
package com.example;

import jakarta.servlet.*;
import jakarta.servlet.http.*;
import java.io.IOException;
import java.io.PrintWriter;

public class HelloServlet extends HttpServlet {
   @Override
   protected void doGet(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {
      response.setContentType("text/html");
      PrintWriter out = response.getWriter();
      out.println("<html><body><h1>Hola desde Servlet</h1></body></html>");
   }
}
```

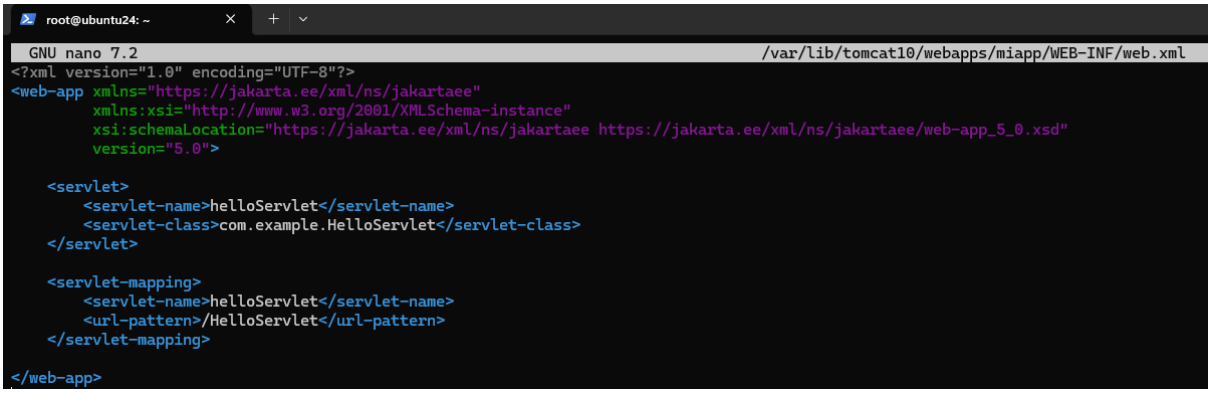
Compilar el Servlet utilizando el API de Servlets de Tomcat ubicados en /usr/share/tomcat10/lib/.



javac -cp /usr/share/tomcat10/lib/servlet-api.jar -d /var/lib/tomcat10/webapps/miapp/WEB-INF/classes ~/miProyecto/src/com/example/HelloServlet.java

Configurar el servlet en web.xml

nano /var/lib/tomcat10/webapps/miapp/WEB-INF/web.xml



<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="https://jakarta.ee/xml/ns/jakartaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee
https://jakarta.ee/xml/ns/jakartaee/web-app_5_0.xsd"
version="5.0">

<servlet>
<servlet-name>helloServlet</servlet-name>
<servlet-class>com.example.HelloServlet</servlet-class>
</servlet>

<servlet-mapping>
<servlet-name>helloServlet</servlet-name>
<url-pattern>/HelloServlet</url-pattern>
</servlet-mapping>

</web-app>

Creación de una página JSP

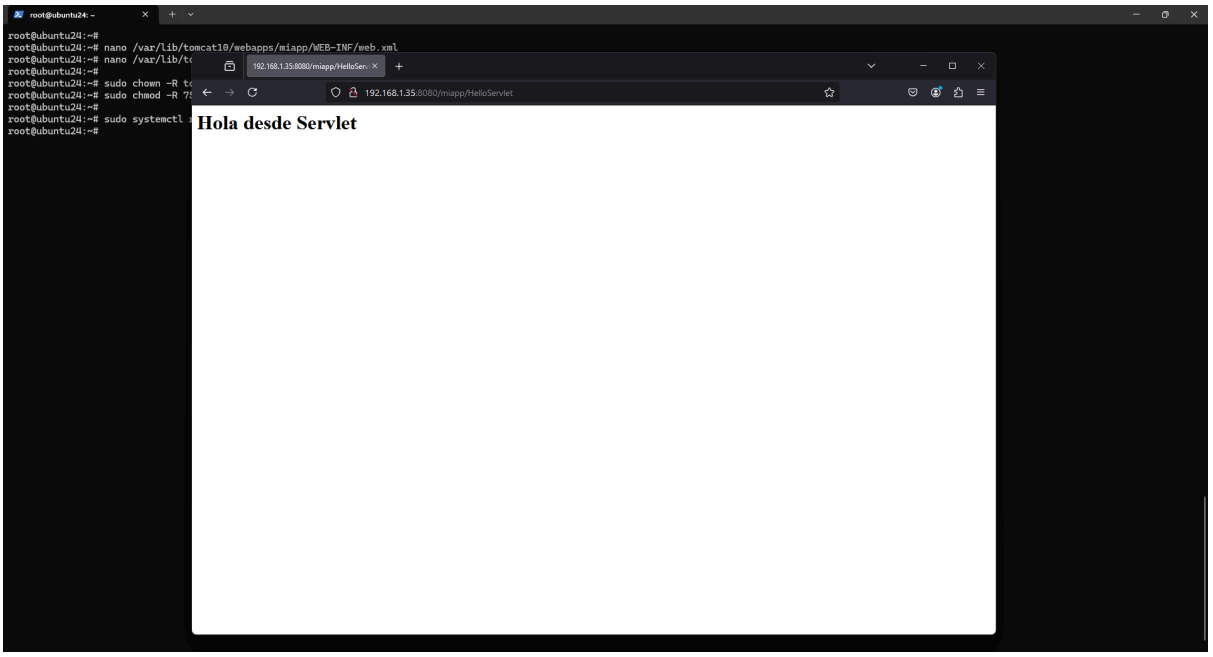
sudo nano /var/lib/tomcat10/webapps/miapp/test.jsp

<html>
<head>
<title>Prueba JSP</title>
</head>
<body>
<h1>Hola desde JSP</h1>
</body>
</html>

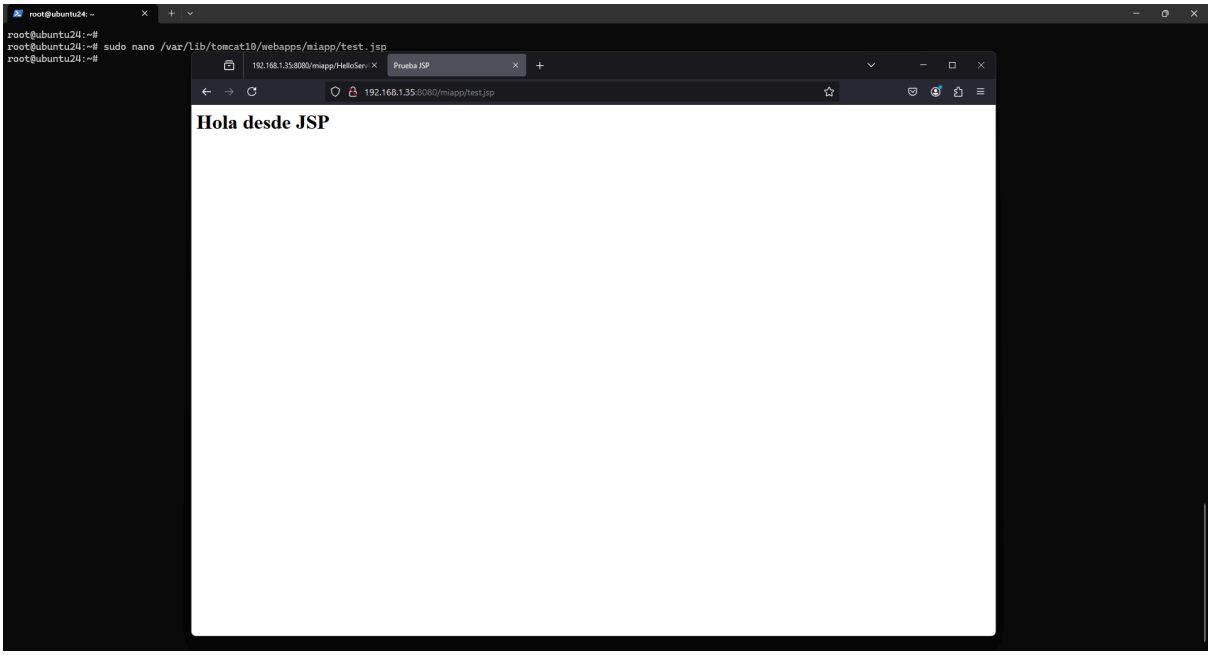
Modificar permisos y reiniciar Tomcat

sudo chown -R tomcat:tomcat /var/lib/tomcat10/webapps/miapp
sudo chmod -R 755 /var/lib/tomcat10/webapps/miapp
sudo systemctl restart tomcat10

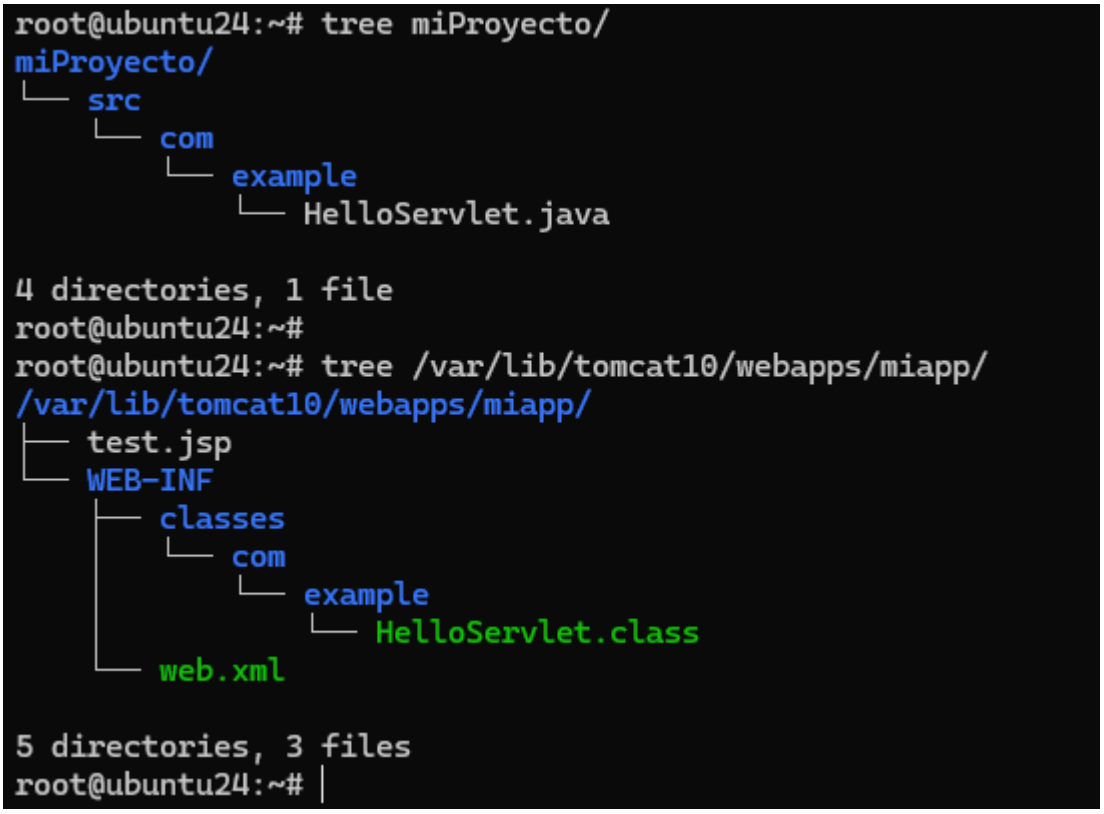
Acceder a http://192.168.1.35:8080/miapp/HelloServlet para comprobar que el servlet funciona.



Acceder a http://192.168.1.35:8080/miapp/test.jsp para comprobar que la JSP funciona.



La estructura final del proyecto es la siguiente:



4- Dispones de una máquina que cuenta con el sistema operativo Ubuntu 22.04 LTS o posterior, recientemente actualizado, en la que está el entorno de red configurado y, además, dispones de conexión a Internet y 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 JDK.**

Ya lo hice en el punto 3.

```
root@ubuntu24:~# java -version
openjdk version "21.0.6" 2025-01-21
OpenJDK Runtime Environment (build 21.0.6+7-Ubuntu-124.04.1)
OpenJDK 64-Bit Server VM (build 21.0.6+7-Ubuntu-124.04.1, mixed mode, sharing)
root@ubuntu24:~#
```

- **Crear usuario para WildFly.**

useradd -r -d /opt/wildfly -s /bin/false wildfly

-r: Crea un usuario del sistema sin capacidad de login.
-d /opt/wildfly: Define el directorio de WildFly.
-s /bin/false: Evita que el usuario inicie sesión.

- **Descargar e instalar WidlFly.**

cd /opt
wget https://github.com/wildfly/wildfly/releases/download/30.0.0.Final/wildfly-30.0.0.Final.tar.gz

```
root@ubuntu24:~# cd /opt/
root@ubuntu24:/opt#
root@ubuntu24:/opt# wget https://github.com/wildfly/wildfly/releases/download/30.0.0.Final/wildfly-30.0.0.Final.tar.gz
--2025-02-22 12:05:42--  https://github.com/wildfly/wildfly/releases/download/30.0.0.Final/wildfly-30.0.0.Final.tar.gz
Resolving github.com (github.com)... 140.82.121.3
Connecting to github.com (github.com)|140.82.121.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/764708/580cbe35-367f-4677-9469-dc87d3370e04?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250222%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250222T120545Z&X-Amz-Expires=300&X-Amz-Signature=c4407c998684ae36572dd1d49104e3542be1a45a349da3cf325fd05d13bf67dc&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dwildfly-30.0.0.Final.tar.gz&response-content-type=application%2Foctet-stream [following]
--2025-02-22 12:05:43--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/764708/580cbe35-367f-4677-9469-dc87d3370e04?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250222%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250222T120545Z&X-Amz-Expires=300&X-Amz-Signature=c4407c998684ae36572dd1d49104e3542be1a45a349da3cf325fd05d13bf67dc&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dwildfly-30.0.0.Final.tar.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 235787802 (225M) [application/octet-stream]
Saving to: 'wildfly-30.0.0.Final.tar.gz'

wildfly-30.0.0.Final.tar.gz                   100%[==========================================================================================================>] 224,86M  7,34MB/s    in 32s

2025-02-22 12:06:15 (7,04 MB/s) - 'wildfly-30.0.0.Final.tar.gz' saved [235787802/235787802]
```

tar -xvzf wildfly-30.0.0.Final.tar.gz
mv wildfly-30.0.0.Final wildfly
chown -R wildfly:wildfly /opt/wildfly

```
root@ubuntu24:/opt# ll
total 230276
drwxr-xr-x  3 root root       4096 feb 22 12:06 ./
drwxr-xr-x 23 root root       4096 nov 23 20:09 ../
drwxr-xr-x 12  501 staff      4096 oct 18  2023 wildfly-30.0.0.Final/
-rw-r--r--  1 root root  235787802 oct 18  2023 wildfly-30.0.0.Final.tar.gz
root@ubuntu24:/opt#
root@ubuntu24:/opt# mv wildfly-30.0.0.Final wildfly
root@ubuntu24:/opt#
root@ubuntu24:/opt# chown -R wildfly:wildfly /opt/wildfly
root@ubuntu24:/opt#
root@ubuntu24:/opt# ll
total 230276
drwxr-xr-x  3 root    root         4096 feb 22 12:07 ./
drwxr-xr-x 23 root    root         4096 nov 23 20:09 ../
drwxr-xr-x 12 wildfly wildfly      4096 oct 18  2023 wildfly/
-rw-r--r--  1 root    root    235787802 oct 18  2023 wildfly-30.0.0.Final.tar.gz
root@ubuntu24:/opt#
```

- **Configurar systemd y el archivo wildfly.conf.**

mkdir -p /etc/wildfly
nano /etc/wildfly/wildfly.conf

Añadir lo siguiente:

WILDFLY_HOME="/opt/wildfly"
WILDFLY_USER=wildfly
WILDFLY_MODE=standalone
WILDFLY_CONFIG=standalone.xml
WILDFLY_BIND=0.0.0.0

```
root@ubuntu24: /opt
 GNU nano 7.2                                  /etc/wildfly/wildfly.conf *
WILDFLY_HOME="/opt/wildfly"
WILDFLY_USER=wildfly
WILDFLY_MODE=standalone
WILDFLY_CONFIG=standalone.xml
WILDFLY_BIND=0.0.0.0
```

Crear el servicio en systemd

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

Cambiar la directiva ExecStart de este servicio por la siguiente (versión standalone)

ExecStart=/opt/wildfly/bin/launch.sh $WILDFLY_MODE $WILDFLY_CONFIG $WILDFLY_BIND

```
root@ubuntu24: /opt
 GNU nano 7.2                                                    /etc/systemd/system/wildfly.service
[Unit]
Description=The WildFly Application Server
After=syslog.target network.target
Before=httpd.service

[Service]
Environment=LAUNCH_JBOSS_IN_BACKGROUND=1
EnvironmentFile=-/etc/wildfly/wildfly.conf
User=wildfly
LimitNOFILE=102642
PIDFile=/run/wildfly/wildfly.pid
#ExecStart=/opt/wildfly/bin/launch.sh $WILDFLY_MODE $WILDFLY_CONFIG $WILDFLY_BIND
ExecStart=/opt/wildfly/bin/standalone.sh -c standalone.xml -b 0.0.0.0
StandardOutput=null

[Install]
WantedBy=multi-user.target
```

Reiniciar servicios

systemctl daemon-reload
systemctl enable --now wildfly

Comprobar que el servicio se ha iniciado correctamente.

systemctl status wildfly

```
root@ubuntu24:/opt# systemctl status wildfly
● wildfly.service - The WildFly Application Server
     Loaded: loaded (/etc/systemd/system/wildfly.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-02-22 12:14:14 UTC; 4min 18s ago
   Main PID: 2920 (standalone.sh)
      Tasks: 60 (limit: 9442)
     Memory: 283.4M (peak: 293.1M)
        CPU: 12.727s
     CGroup: /system.slice/wildfly.service
             ├─2920 /bin/sh /opt/wildfly/bin/standalone.sh -c standalone.xml -b 0.0.0.0
             └─3043 java "-D[Standalone]" -Xms64m -Xmx512m -XX:MetaspaceSize=96M -XX:MaxMetaspaceSize=256m -Djava.net.preferIPv4Stack=true -Djboss.modules.system.pkgs=org.jboss.byteman -Djava.awt.headless=tru

feb 22 12:14:14 ubuntu24 systemd[1]: Started wildfly.service - The WildFly Application Server.
lines 1-12/12 (END)
```

- **Configurar la autenticación de Wildfly.**

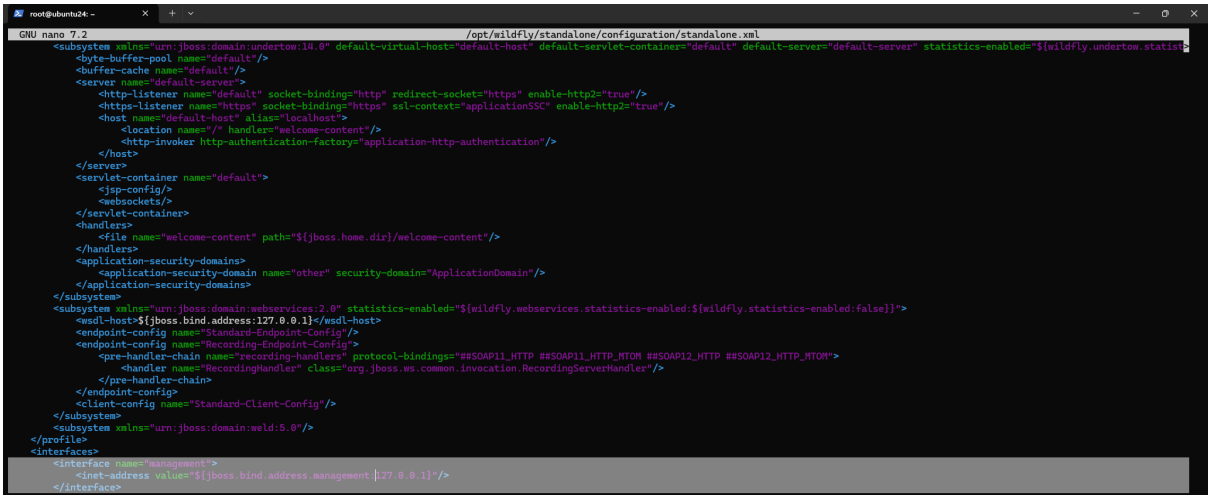
Crear un usuario administrador añadiéndolo al grupo Administrators.

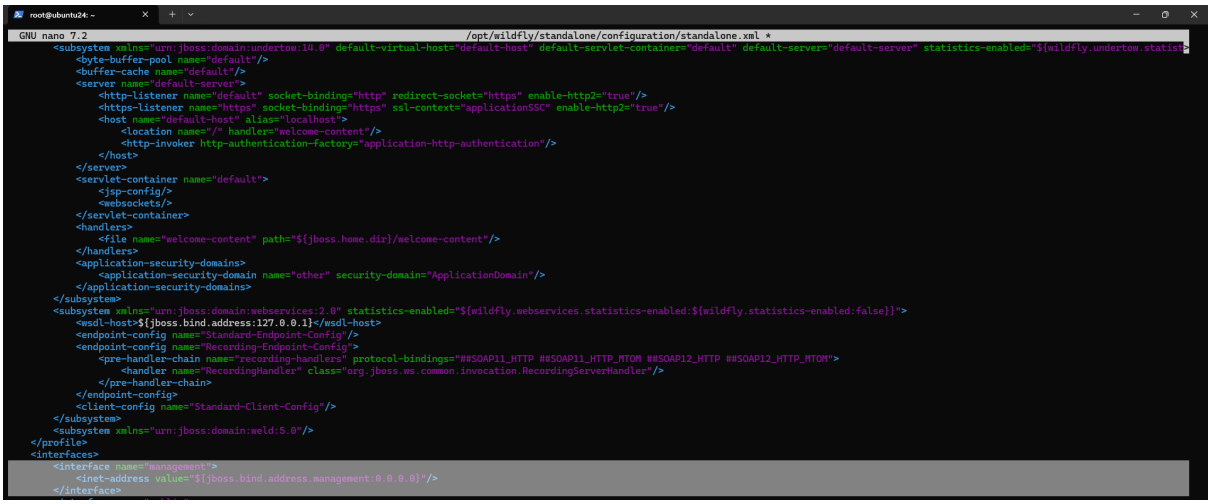
/opt/wildfly/bin/add-user.sh

```
root@ubuntu24:~# /opt/wildfly/bin/add-user.sh

What type of user do you wish to add?
 a) Management User (mgmt-users.properties)
 b) Application User (application-users.properties)
(a): a

Enter the details of the new user to add.
Using realm 'ManagementRealm' as discovered from the existing property files.
Username : diego
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 :
WFLYDM0099: Password should have at least 8 characters!
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)[  ]: Administrators
About to add user 'diego' for realm 'ManagementRealm'
Is this correct yes/no? y
Added user 'diego' to file '/opt/wildfly/standalone/configuration/mgmt-users.properties'
Added user 'diego' to file '/opt/wildfly/domain/configuration/mgmt-users.properties'
Added user 'diego' with groups Administrators to file '/opt/wildfly/standalone/configuration/mgmt-groups.properties'
Added user 'diego' with groups Administrators to file '/opt/wildfly/domain/configuration/mgmt-groups.properties'
root@ubuntu24:~#
```

Por defecto Wildfly sólo escucha peticiones de localhost. Como estoy accediendo desde el navegador de otra máquina es necesario modificar este comportamiento editando el archivo /opt/wildfly/standalone/configuration/standalone.xml.

nano /opt/wildfly/standalone/configuration/standalone.xml





Reiniciar Wildfly (systemctl restart wildfly) y acceder a la consola de administración en http://192.168.1.35:9990 para comprobar que funciona. Autenticarse con el usuario administrador creado previamente.