

13-10-2023

Implantación de arquitecturas web. LAMP

Tarea 01

ROBERTO RODRÍGUEZ JIMÉNEZ
roberto.rodjim.1@educa.jcyl.es

Contenido

Tarea online DAW01	2
Enunciado	2
Caso práctico.....	2
¿Qué te pedimos que hagas?	2
Recursos necesarios y recomendaciones	3
Recursos necesarios	3
Recomendaciones	3
Evaluación de la tarea	3
Criterios de evaluación implicados.....	3
¿Cómo valoramos y puntuamos tu tarea?	3
Respuestas.....	4
1. La arquitectura Web es un modelo compuesto de tres capas, ¿cuáles son y cuál es la función de cada una de ellas?	4
2. Una plataforma web es el entorno de desarrollo de software empleado para diseñar y ejecutar un sitio web; destacan dos plataformas web, LAMP y WISA. Explica en qué consiste cada una de ellas.	5
3. Dispones de una máquina que cuenta con el sistema operativo Ubuntu 20.04 LTS recientemente actualizado, esta máquina tiene el entorno de red configurado y, además, dispones de conexión a Internet. Además, estás trabajando con la cuenta del usuario root. Indica cada uno de los pasos, y comandos implicados en ellos, para conseguir hacer lo siguiente:	6
3.1 Instalar el servidor web Apache desde terminal.	7
3.2 Comprobar que está funcionando el servidor Apache desde terminal.....	8
3.3 Comprobar que está funcionando el servidor Apache desde navegador.....	9
3.4 Cambiar el puerto por el cual está escuchando Apache pasándolo al puerto 82.	10
3.5 Instalar el servidor de aplicaciones Tomcat.	11

Tarea online DAW01

Título de la tarea: Instalación de servidores Apache y Tomcat

Unidad: 1

Ciclo formativo y módulo: Desarrollo de Aplicaciones Web - Despliegue de Aplicaciones Web

Curso académico: 2021/2022

Enunciado

Caso práctico

En la empresa BK programación, le han encargado a Juan que monte una máquina de pruebas para un proyecto web que va a empezar. Lo primero que hace Juan es repasar la Arquitectura Web y las distintas Plataformas que hay disponibles para ello.

Como el lenguaje principal de desarrollo va a ser Java, ha visto que necesitará Apache y Tomcat. Juan también se decanta por Ubuntu ya que es una distribución que conoce bastante bien y es la que usan en la empresa para los servidores.

Como parte de su proceso de formación en la empresa, Ada le ha pedido a Juan que documente todo el proceso ya que puede serle útil a su compañera María, que también ha empezado hace poco a trabajar en la misma empresa.

¿Qué te pedimos que hagas?

1. La arquitectura Web es un modelo compuesto de tres capas, ¿cuáles son y cuál es la función de cada una de ellas?
2. Una plataforma web es el entorno de desarrollo de software empleado para diseñar y ejecutar un sitio web; destacan dos plataformas web, LAMP y WISA. Explica en qué consiste cada una de ellas.
3. Dispones de una máquina que cuenta con el sistema operativo Ubuntu 20.04 LTS recientemente actualizado, esta máquina tiene el entorno de red configurado y, además, dispones de conexión a Internet. Además, estás trabajando con la cuenta del usuario root. Indica cada uno de los pasos, y comandos implicados en ellos, para conseguir hacer lo siguiente:
 1. Instalar el servidor web Apache desde terminal.
 2. Comprobar que está funcionando el servidor Apache desde terminal.
 3. Comprobar que está funcionando el servidor Apache desde navegador.
 4. Cambiar el puerto por el cual está escuchando Apache pasándolo al puerto 82.
 5. Instalar el servidor de aplicaciones Tomcat.

NOTA IMPORTANTE

Para el apartado 3 es necesario entregar las capturas de pantalla de los principales pasos realizados, explicando el proceso seguido en cada uno de ellos. Las capturas de pantalla realizadas deben tener como fondo de pantalla la plataforma con tu usuario mostrando claramente la foto de tu perfil. Aquellos apartados/subapartados que no cumplan esta condición no serán corregidos.

Recursos necesarios y recomendaciones

Recursos necesarios

- ✓ Computador con conexión a Internet.
- ✓ Gestor de máquinas virtuales tipo Virtual Box.
- ✓ Máquina virtual con última distribución de Ubuntu 20.04 LTS.

Recomendaciones

Incluye la pregunta que se formula antes de cada una de las respuestas.

Detalla las respuestas de una forma estructurada, e intenta razonar y explicar de forma clara las decisiones adoptadas.

Ten en cuenta que para los ejercicios prácticos existen varias respuestas posibles, siendo todas ellas válidas.

Evaluación de la tarea

Criterios de evaluación implicados

- a) Se han analizado aspectos generales de arquitecturas Web, sus características, ventajas e inconvenientes.
- b) Se han descrito los fundamentos y protocolos en los que se basa el funcionamiento de un servidor Web.
- c) Se ha realizado la instalación y configuración básica de servidores Web.
- d) Se han clasificado y descrito los principales servidores de aplicaciones.
- e) Se ha realizado la instalación y configuración básica de servidores de aplicaciones.
- f) Se han realizado pruebas de funcionamiento de los servidores web y de aplicaciones.
- g) Se han documentado los procesos de instalación y configuración realizados sobre los servidores Web y sobre las aplicaciones.

¿Cómo valoramos y puntuamos tu tarea?

Rúbrica de la tarea

Apartado 1: Arquitectura Web. Respuesta correcta.	2 puntos
Apartado 2: Plataformas Web. Respuesta correcta.	2 puntos
Apartado 3.1: Instalar el servidor web Apache desde terminal.	1 punto
Apartado 3.2: Comprobar que está funcionando el servidor Apache desde terminal.	1 punto
Apartado 3.3: Comprobar que está funcionando el servidor Apache desde navegador.	1 punto
Apartado 3.4: Cambiar el puerto por el cual está escuchando Apache pasándolo al puerto 82.	1 punto
Apartado 3.5: Instalar el servidor de aplicaciones Tomcat.	2 puntos

NOTA IMPORTANTE

Aquellos apartados/subapartados en los que las capturas de pantalla no sean claras o no tengan como fondo de pantalla la plataforma con tu usuario mostrando claramente la foto de tu perfil, no serán corregidos.

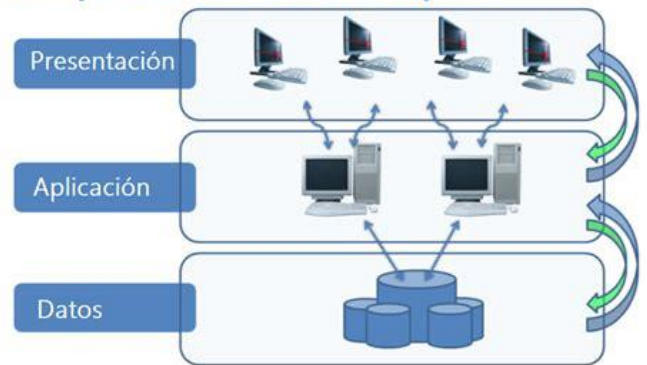
Respuestas

1. La arquitectura Web es un modelo compuesto de tres capas, ¿cuáles son y cuál es la función de cada una de ellas?

Capas:

1. **Datos (Capa de acceso datos):** generalmente son bases de datos en las cuales se almacena la información que se debe mostrar mediante el servicio web.
Esta información es gestionada por Sistemas de Gestión de Bases de Datos (SGBD), que podrían ser del tipo *MySQL*, *PostgreSQL*, etc.
2. **Servidor (Capa de negocio):** el servidor recibe las peticiones realizadas desde el cliente, las valida, obtiene los datos de la *capa de acceso a datos*, y finalmente devuelve al cliente la información pedida.

Arquitectura de 3 capas



Arquitectura de 3 capas.

Fuente: <https://www.estrategiamagazine.com>

Los servidores más conocidos son :

- a. *Apache Web Server*: que sirve aplicaciones web, tanto dinámicas como estáticas, portales (sitios webs que ofrecen diferentes servicios, como por ejemplo la web de una administración pública), CMS (o gestores de contenido), que son sitios configurados para que el propietario del sitio solamente se ocupe del contenido de la web, o las tiendas virtuales que, generalmente, también son CMS, pero ofrecen conexiones con pasarelas de pago o catálogos.
 - b. *Apache Tomcat*: es un servidor de aplicaciones, generalmente desarrolladas en Java, aunque también puede ofrecer contenido estático, aunque su rendimiento baja notablemente.
3. **Cliente (Capa de presentación):** Los clientes acceden al contenido servido a través de navegadores web. Navegadores web existen muchos: Firefox, Chrome, Edge, Opera, etc.

2. Una plataforma web es el entorno de desarrollo de software empleado para diseñar y ejecutar un sitio web; destacan dos plataformas web, LAMP y WISA. Explica en qué consiste cada una de ellas.

LAMP (Linux Apache MySQL PHP)

LAMP combina las cuatro tecnologías de código abierto:

- *Linux* es el sistema operativo en el que se ejecuta LAMP.
No requiere grandes recursos.
- *Apache HTTP Server* es el servidor web.
- *MySQL* gestiona las bases de datos. Es un *SGBD* relacional que utiliza *SQL*.
Actualmente se tiende a usar *MariaDB* ya que *MySQL* no es completamente Open Source al haber sido comprado por Oracle.
- *PHP* es el lenguaje de programación, desarrollado para crear sitios web dinámicos.

Existen alternativas multiplataforma (*XAMP*) o para Windows (*WAMP*).

Otras plataformas modernas pueden usar motores de bases de datos o lenguajes diferentes, como puede ser *MEAN* (MongoDB, Express, Angular, Node).

WISA (Windows, IIS, SQL, ASP)

WISA es el *web stack* de *Microsoft*, por lo que el software es propietario y su uso requiere licencias de uso.

Una de las diferencias con LAMP es que WISA sí desarrolló los componentes para trabajar juntos, por lo que se garantiza un funcionamiento óptimo.

Sus componentes son:

- WISA corre bajo el sistema operativo Windows de Microsoft, el cual necesita una licencia de uso.
- El servidor web es Internet Information Services (ISS).
- SGBD: SQL Server.
- El framework ASP.NET proporciona un conjunto de lenguajes para poder mostrar las webs dinámicas, como Visual Basic.

Con el uso del framework ASP.NET, el acrónimo WISA también se conoce como WISN.

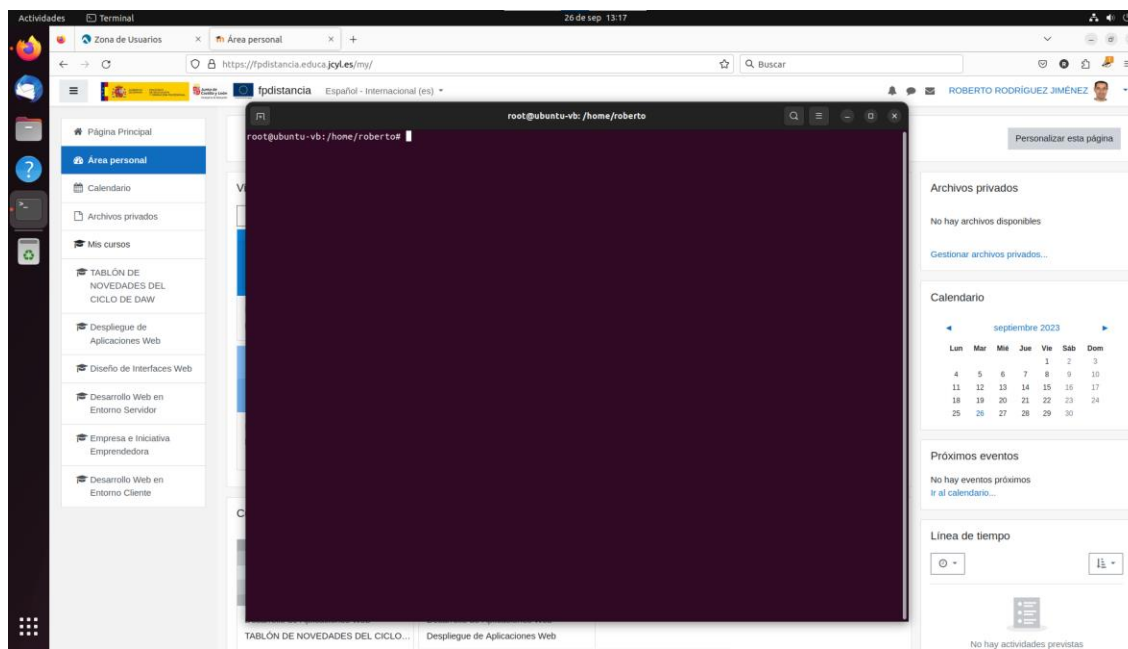
3. Dispones de una máquina que cuenta con el sistema operativo Ubuntu 20.04 LTS recientemente actualizado, esta máquina tiene el entorno de red configurado y, además, dispones de conexión a Internet. Además, estás trabajando con la cuenta del usuario root. Indica cada uno de los pasos, y comandos implicados en ellos, para conseguir hacer lo siguiente:

Sistema operativo: Ubuntu 22.04.3 (64-bit) (septiembre 2023)

Máquina virtual: VirtualBox 7.0

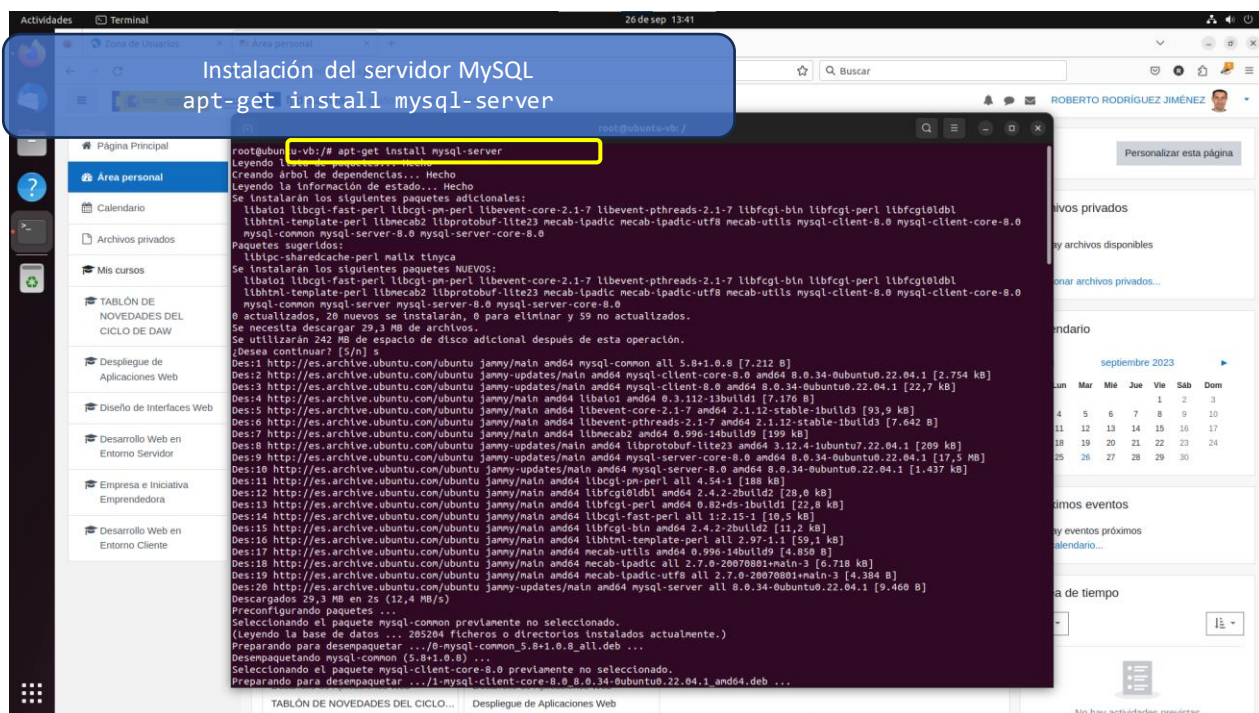
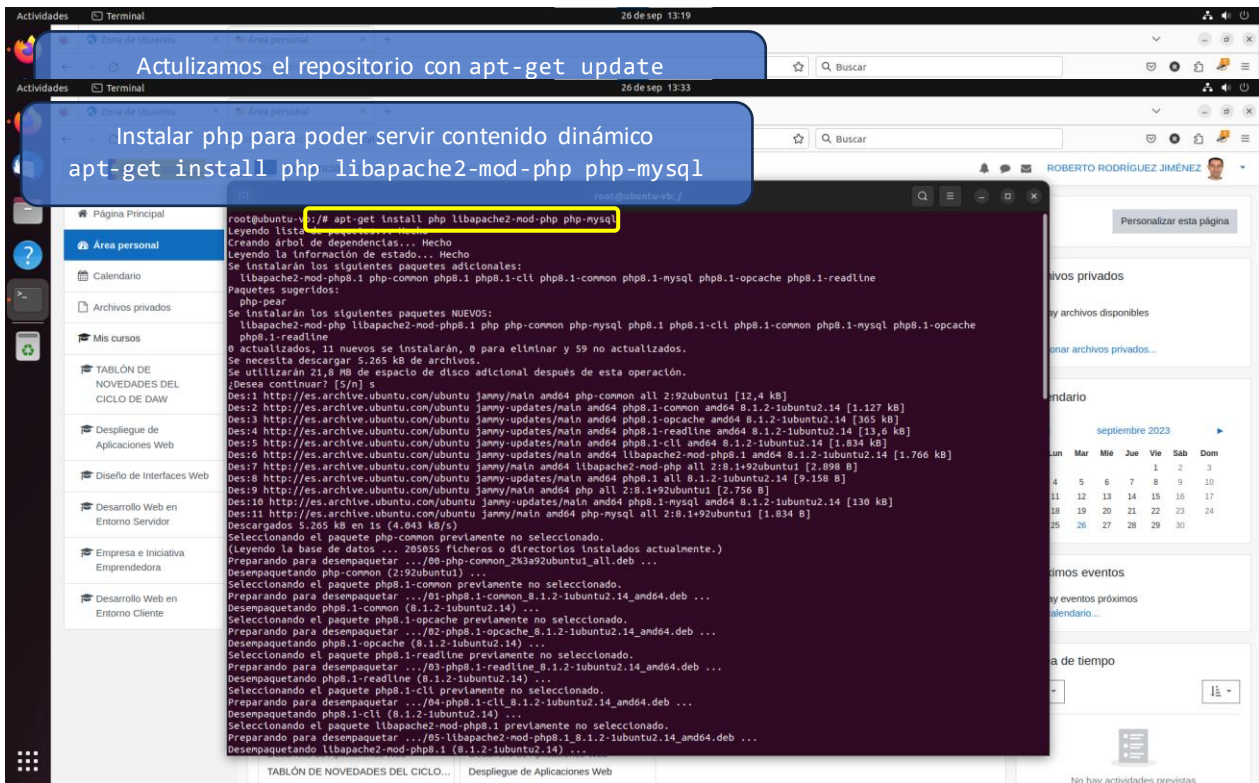
Configuración:

- RAM: 4GB
- Procesadores: 2
- Espacio en disco: 40GB
- Memoria de vídeo: 16MB



Ubuntu 22.04.3 Consola con usuario root

3.1 Instalar el servidor web Apache desde terminal.



Con systemctl status apache2 comprobamos que apache ya se ejecuta desde el inicio.

```
root@ubuntu-vb:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-09-26 13:31:10 CEST; 14min ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 12692 (apache2)
      Tasks: 6 (limit: 4096)
     Memory: 10.0M
        CPU: 136ms
    CGroup: /system.slice/apache2.service
            └─12692 /usr/sbin/apache2 -k start
              12694 /usr/sbin/apache2 -k start
              12695 /usr/sbin/apache2 -k start
              12696 /usr/sbin/apache2 -k start
              12697 /usr/sbin/apache2 -k start
              12698 /usr/sbin/apache2 -k start

sep 26 13:31:10 ubuntu-vb systemd[1]: Starting The Apache HTTP Server...
sep 26 13:31:10 ubuntu-vb apache2[12690]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using
sep 26 13:31:10 ubuntu-vb systemd[1]: Started The Apache HTTP Server.
lines 1-19/19 (END)
root@ubuntu-vb:~#
```

3.2 Comprobar que está funcionando el servidor Apache desde terminal.

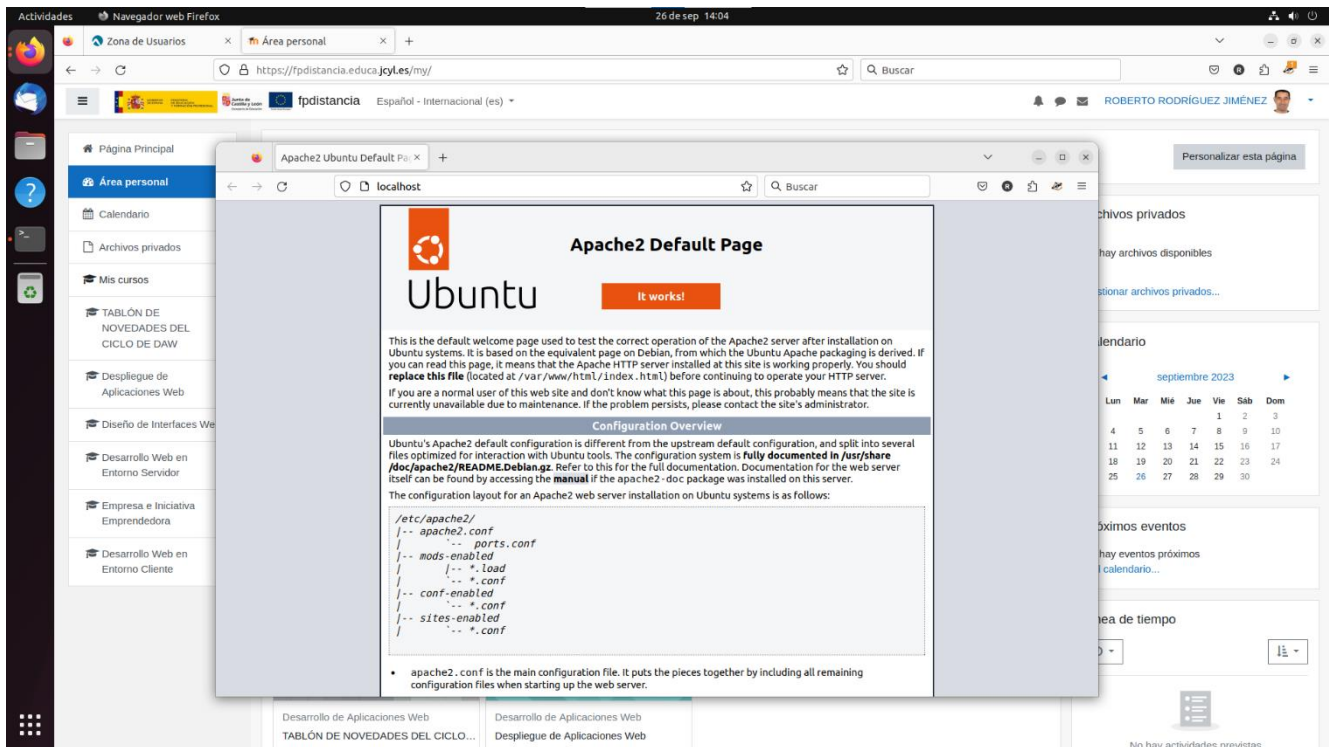
Apachectl configtest muestra que está correcto

```
root@ubuntu-vb:~# apachectl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
root@ubuntu-vb:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-09-26 13:31:10 CEST; 14min ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 12692 (apache2)
      Tasks: 6 (limit: 4096)
     Memory: 10.0M
        CPU: 191ms
    CGroup: /system.slice/apache2.service
            └─12692 /usr/sbin/apache2 -k start
              12694 /usr/sbin/apache2 -k start
              12695 /usr/sbin/apache2 -k start
              12696 /usr/sbin/apache2 -k start
              12697 /usr/sbin/apache2 -k start
              12698 /usr/sbin/apache2 -k start

sep 26 13:31:10 ubuntu-vb systemd[1]: Starting The Apache HTTP Server...
sep 26 13:31:10 ubuntu-vb apache2[12690]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using
sep 26 13:31:10 ubuntu-vb systemd[1]: Started The Apache HTTP Server.
lines 1-19/19 (END)
root@ubuntu-vb:~#
```

Vemos el estado del servidor con systemctl status apache2

3.3 Comprobar que está funcionando el servidor Apache desde navegador.



3.4 Cambiar el puerto por el cual está escuchando Apache pasándolo al puerto 82.

Accedemos al fichero `ports.conf` en el cual se especifica el puerto por defecto por el cual escucha Apache:

```
# vim /etc/apache2/ports.conf
```

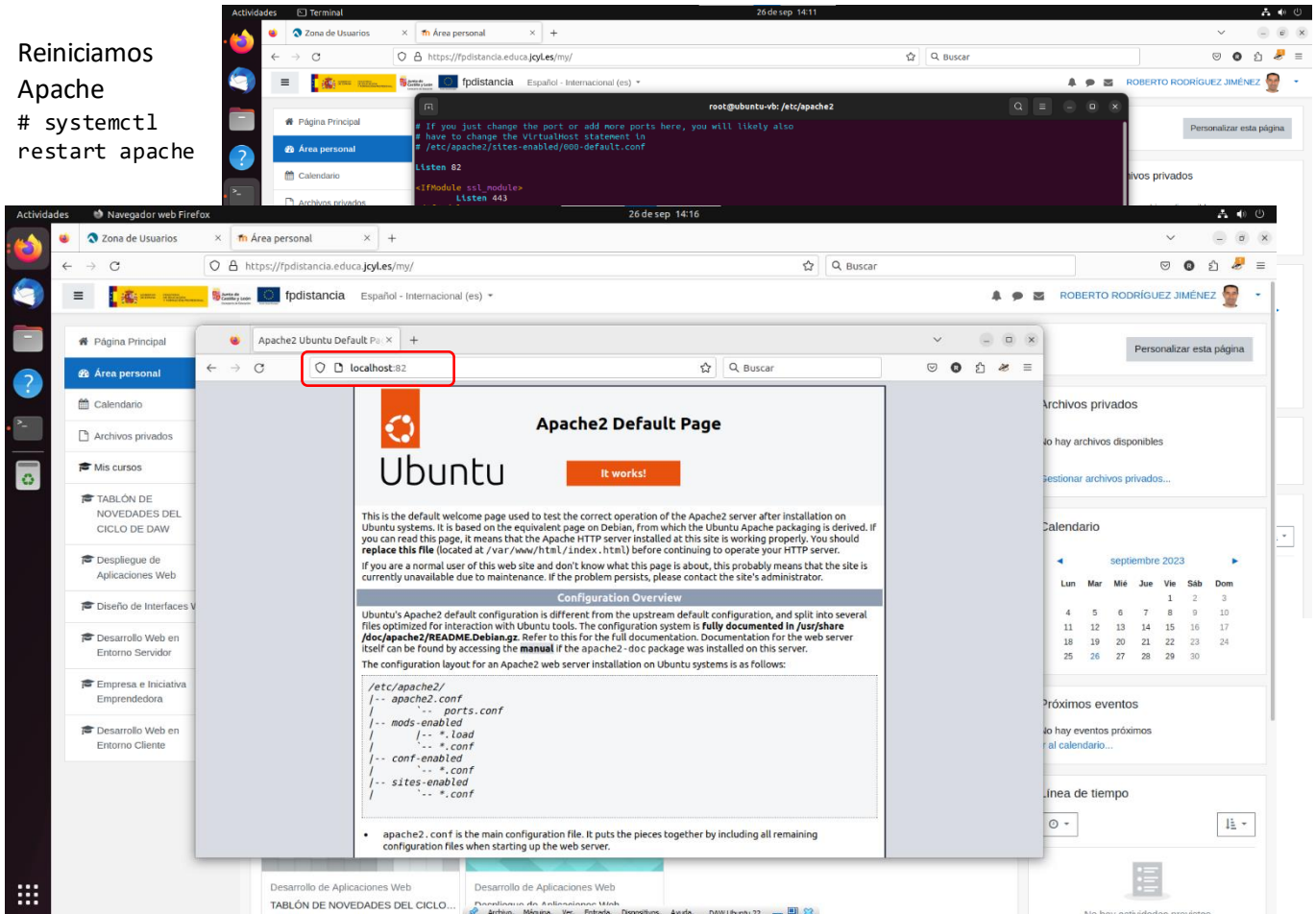
Modificamos el puerto que se indica en `Listen` estableciendo el puerto 82.

Guardamos el archivo y salimos.

Reiniciamos

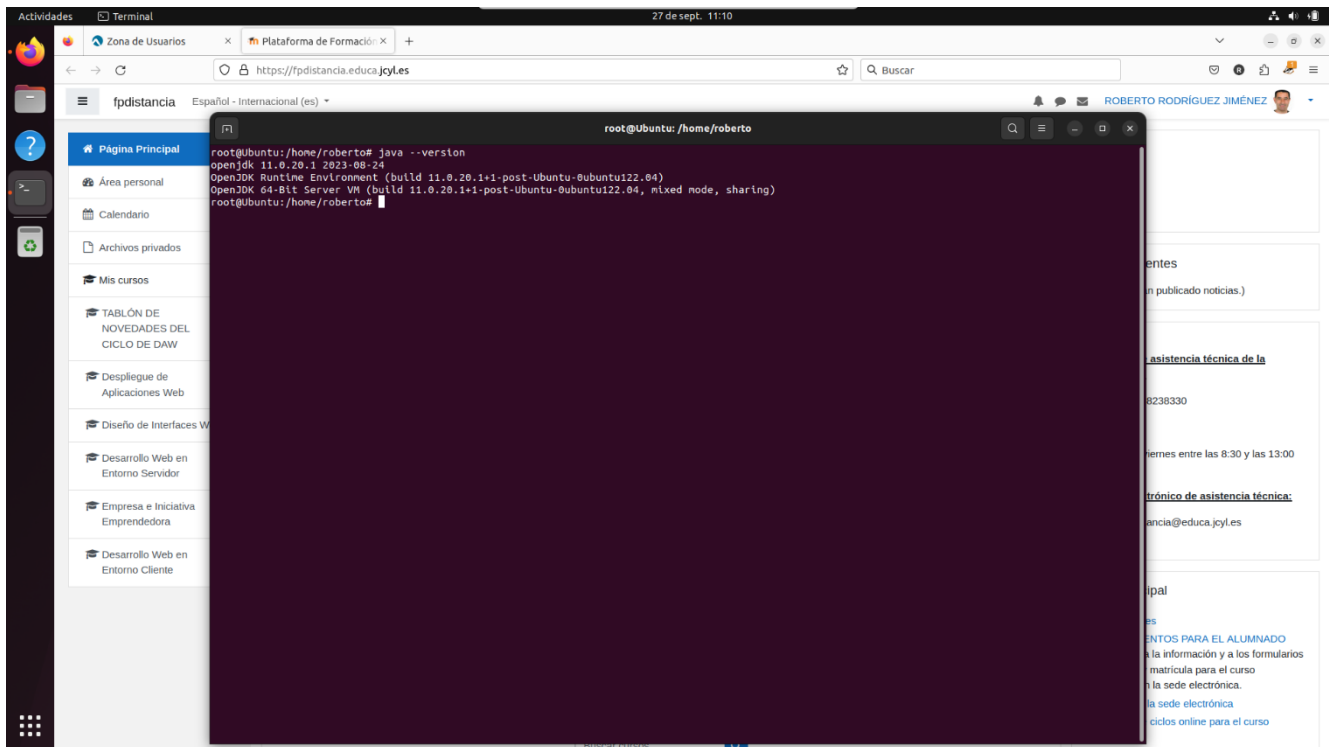
Apache

```
# systemctl  
restart apache
```

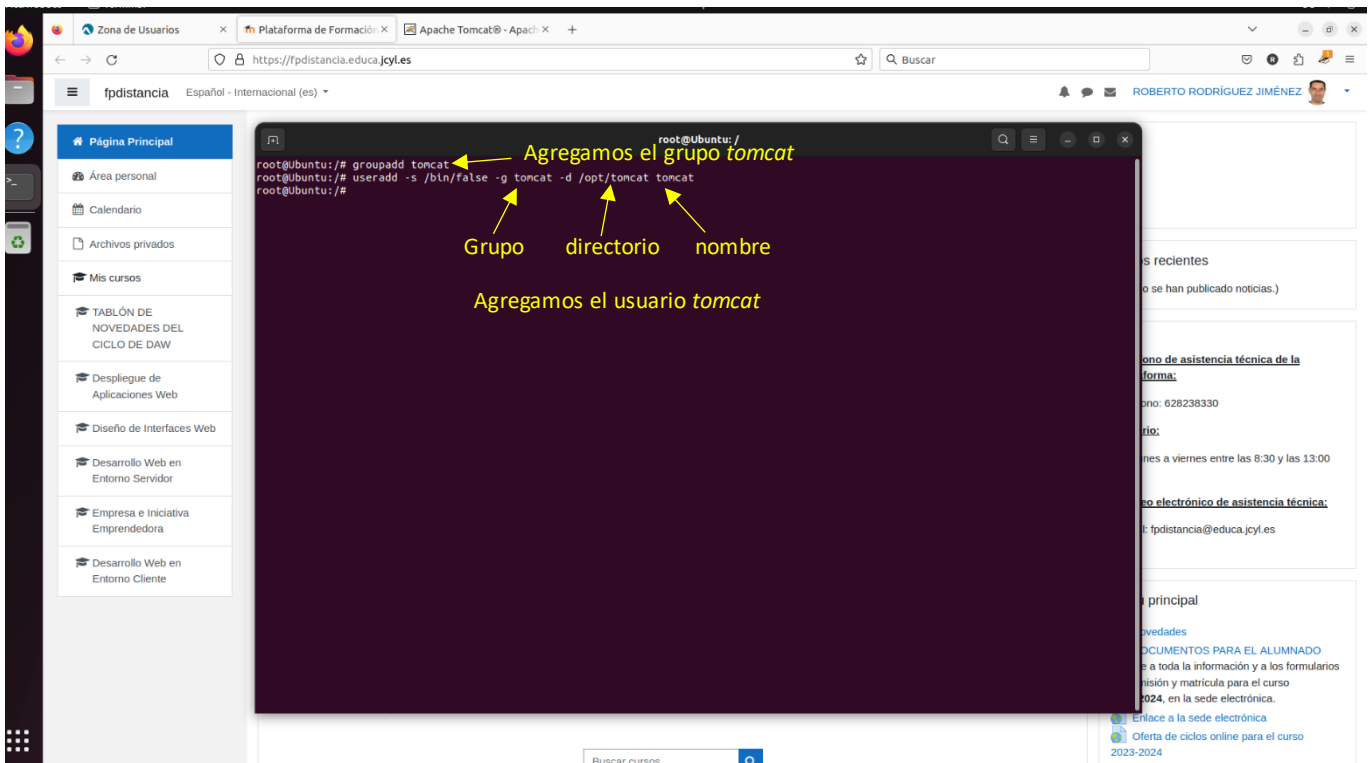


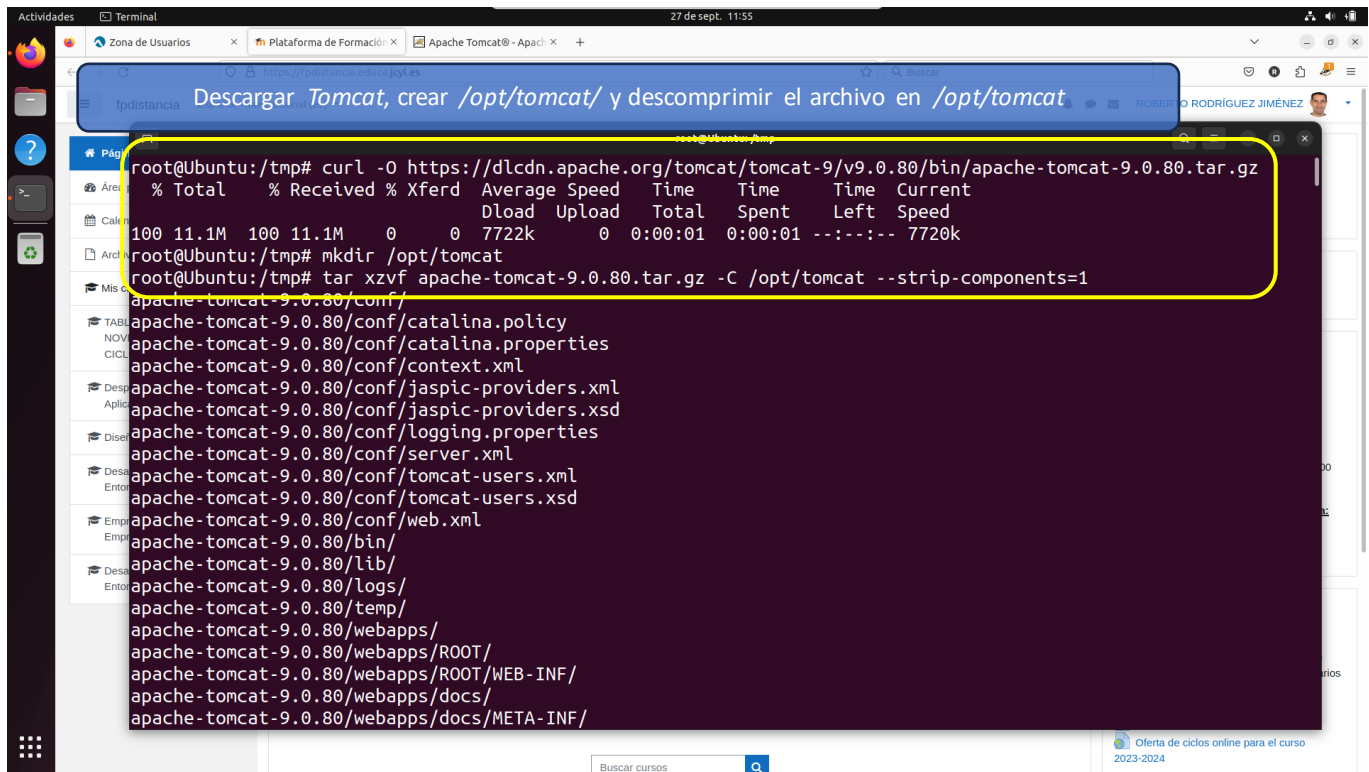
3.5 Instalar el servidor de aplicaciones Tomcat.

Partimos de que ya tenemos instalado el JDK, sólo vamos a instalar el servidor Tomcat.



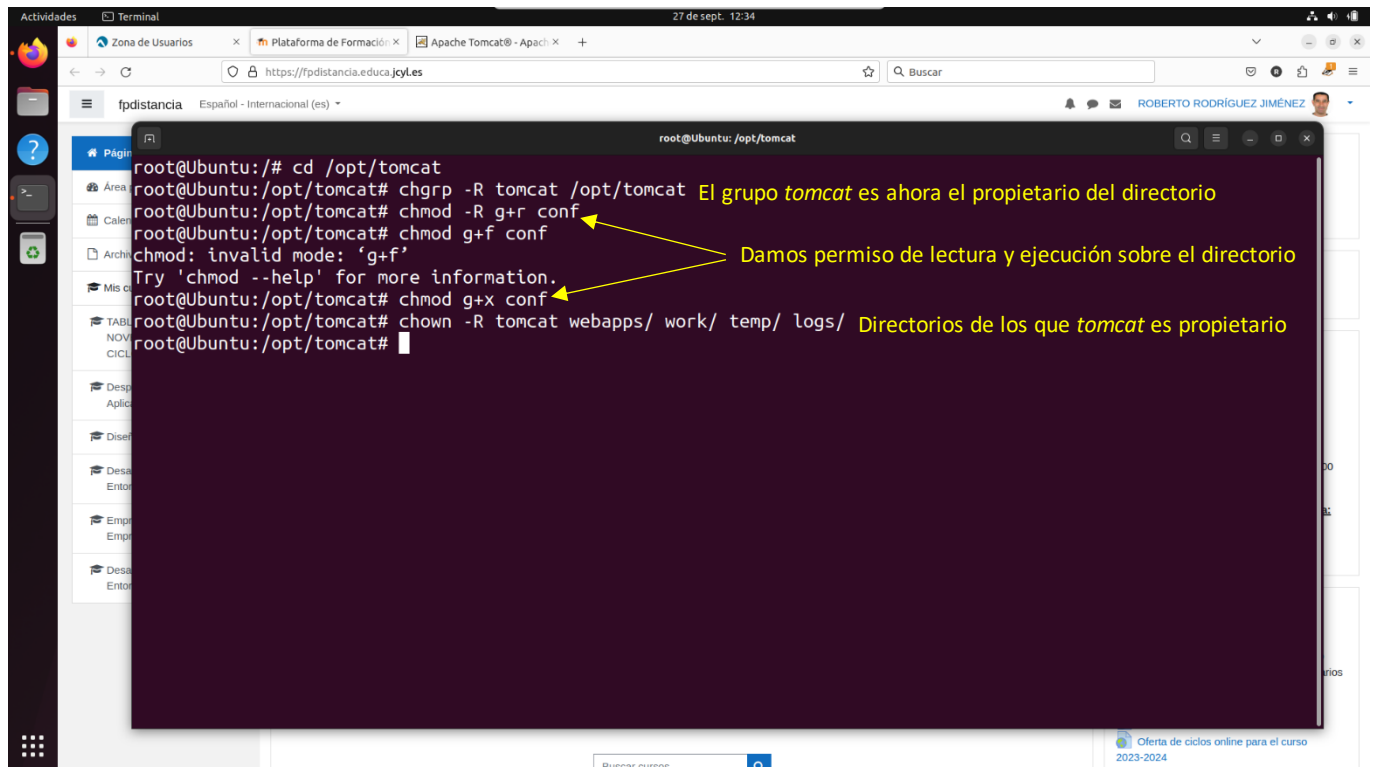
Agregar el grupo y el usuario



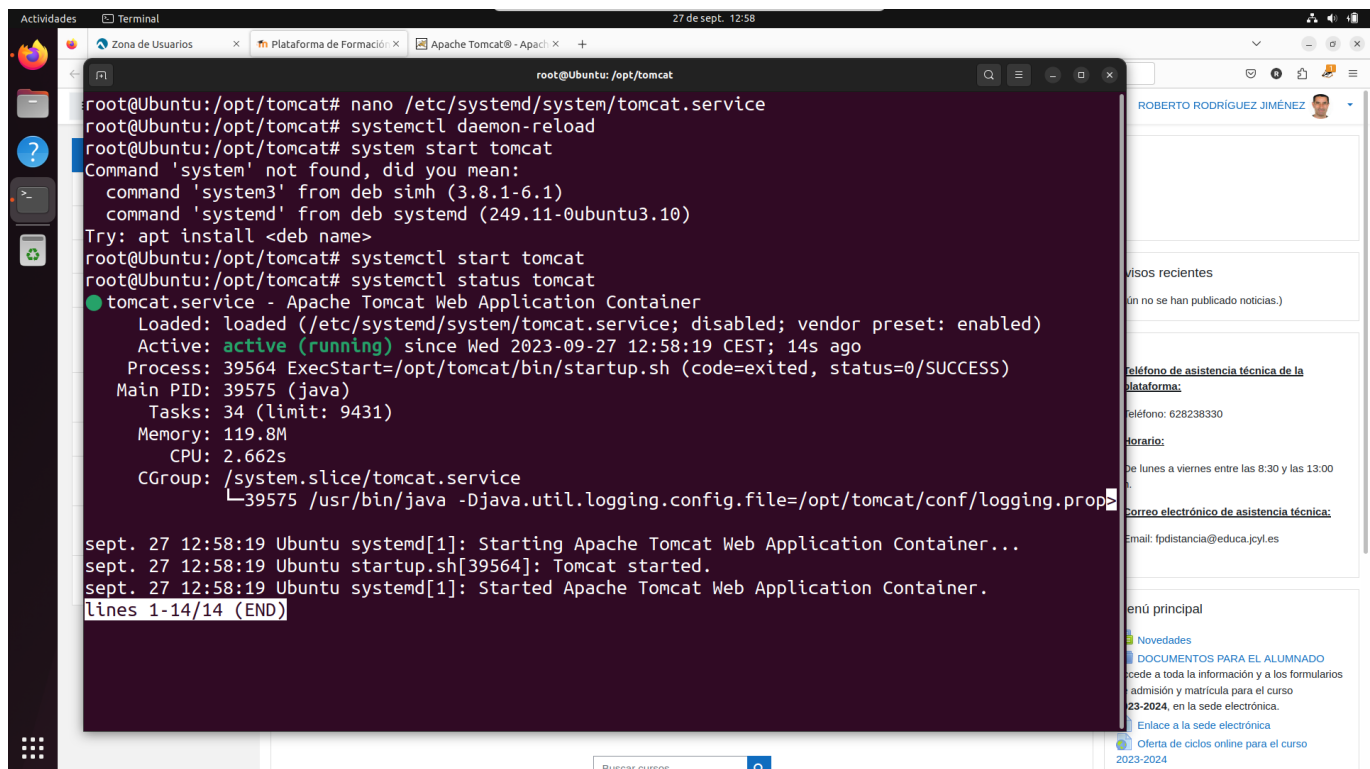
Descargar e instalar *Tomcat*

```
root@Ubuntu:/tmp# curl -O https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.80/bin/apache-tomcat-9.0.80.tar.gz
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 11.1M 100 11.1M 0 0 7722k 0 0:00:01 0:00:01 --:--:-- 7720k
root@Ubuntu:/tmp# mkdir /opt/tomcat
root@Ubuntu:/tmp# tar xzvf apache-tomcat-9.0.80.tar.gz -C /opt/tomcat --strip-components=1
apache-tomcat-9.0.80/conf/
apache-tomcat-9.0.80/conf/catalina.policy
apache-tomcat-9.0.80/conf/catalina.properties
apache-tomcat-9.0.80/conf/context.xml
apache-tomcat-9.0.80/conf/jaspic-providers.xml
apache-tomcat-9.0.80/conf/jaspic-providers.xsd
apache-tomcat-9.0.80/conf/logging.properties
apache-tomcat-9.0.80/conf/server.xml
apache-tomcat-9.0.80/conf/tomcat-users.xml
apache-tomcat-9.0.80/conf/tomcat-users.xsd
apache-tomcat-9.0.80/conf/web.xml
apache-tomcat-9.0.80/bin/
apache-tomcat-9.0.80/lib/
apache-tomcat-9.0.80/logs/
apache-tomcat-9.0.80/temp/
apache-tomcat-9.0.80/webapps/
apache-tomcat-9.0.80/webapps/ROOT/
apache-tomcat-9.0.80/webapps/ROOT/WEB-INF/
apache-tomcat-9.0.80/webapps/docs/
apache-tomcat-9.0.80/webapps/docs/META-INF/
```

Permisos

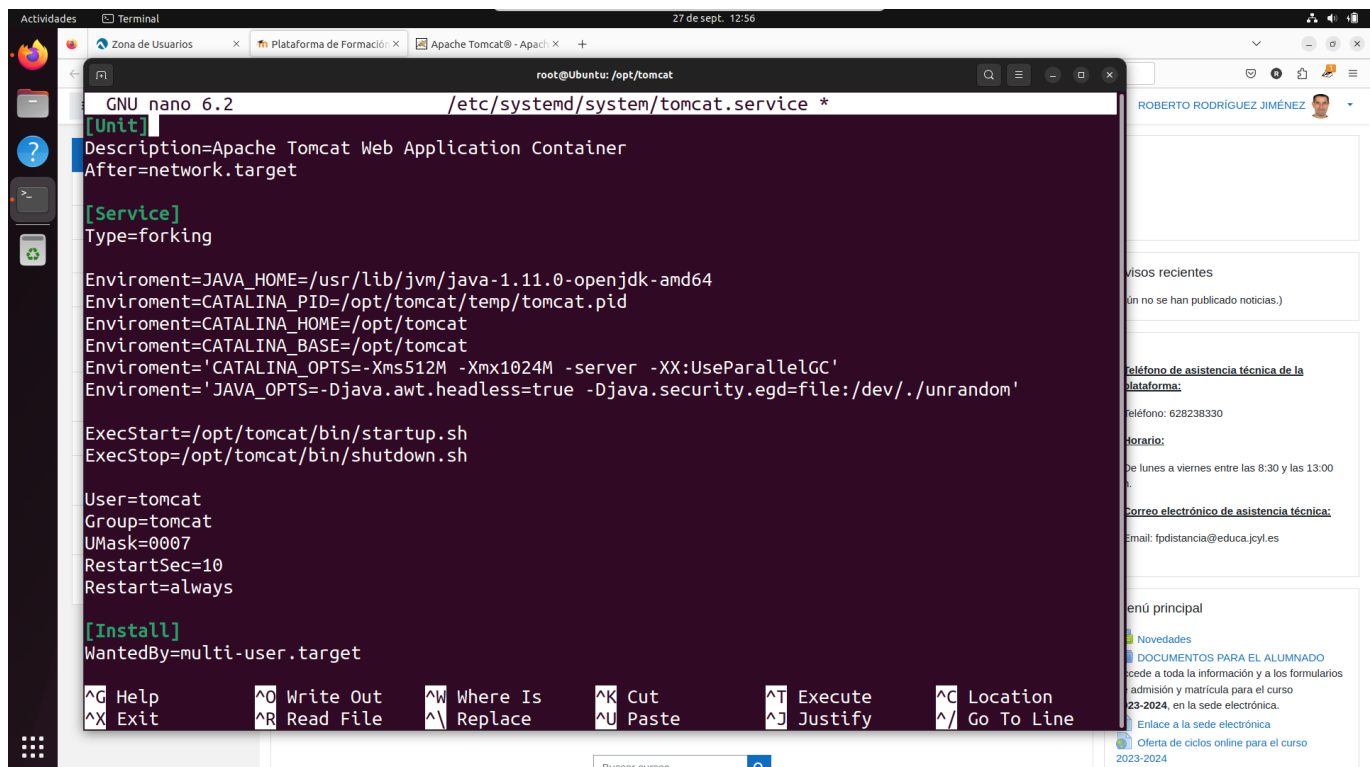


```
root@Ubuntu:/# cd /opt/tomcat
root@Ubuntu:/opt/tomcat# chgrp -R tomcat /opt/tomcat El grupo tomcat es ahora el propietario del directorio
root@Ubuntu:/opt/tomcat# chmod -R g+r conf
root@Ubuntu:/opt/tomcat# chmod g+f conf
chmod: invalid mode: 'g+f'
Try 'chmod --help' for more information.
root@Ubuntu:/opt/tomcat# chmod g+x conf
root@Ubuntu:/opt/tomcat# chown -R tomcat webapps/ work/ temp/ logs/ Directorios de los que tomcat es propietario
root@Ubuntu:/opt/tomcat#
```

Crear el servicio *tomcat.service*

```
root@Ubuntu:/opt/tomcat# nano /etc/systemd/system/tomcat.service
root@Ubuntu:/opt/tomcat# systemctl daemon-reload
root@Ubuntu:/opt/tomcat# system start tomcat
Command 'system' not found, did you mean:
  command 'system3' from deb simh (3.8.1-6.1)
  command 'systemd' from deb systemd (249.11-0ubuntu3.10)
Try: apt install <deb name>
root@Ubuntu:/opt/tomcat# systemctl start tomcat
root@Ubuntu:/opt/tomcat# systemctl status tomcat
● tomcat.service - Apache Tomcat Web Application Container
   Loaded: loaded (/etc/systemd/system/tomcat.service; disabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-09-27 12:58:19 CEST; 14s ago
     Process: 39564 ExecStart=/opt/tomcat/bin/startup.sh (code=exited, status=0/SUCCESS)
    Main PID: 39575 (java)
       Tasks: 34 (limit: 9431)
      Memory: 119.8M
         CPU: 2.662s
        CGroup: /system.slice/tomcat.service
                └─39575 /usr/bin/java -Djava.util.logging.config.file=/opt/tomcat/conf/logging.properties

sept. 27 12:58:19 Ubuntu systemd[1]: Starting Apache Tomcat Web Application Container...
sept. 27 12:58:19 Ubuntu startup.sh[39564]: Tomcat started.
sept. 27 12:58:19 Ubuntu systemd[1]: Started Apache Tomcat Web Application Container.
lines 1-14/14 (END)
```

Archivo *tomcat.service*

```
GNU nano 6.2 /etc/systemd/system/tomcat.service *
[Unit]
Description=Apache Tomcat Web Application Container
After=network.target

[Service]
Type=forking

Environment=JAVA_HOME=/usr/lib/jvm/java-1.11.0-openjdk-amd64
Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat
Environment=CATALINA_BASE=/opt/tomcat
Environment='CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:UseParallelGC'
Environment='JAVA_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./unrandom'

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh

User=tomcat
Group=tomcat
UMask=0007
RestartSec=10
Restart=always

[Install]
WantedBy=multi-user.target

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute   ^C Location
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify   ^/_ Go To Line
```


Carpeta de trabajo de *Tomcat*

```

via a brute-force attack -->
<Realm className="org.apache.catalina.realm.LockOutRealm">
  <!-- This Realm uses the UserDatabase configured in the global JNDI
  resources under the key "UserDatabase". Any edits
  that are performed against this UserDatabase are immediately
  available for use by the Realm. -->
  <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
  resourceName="UserDatabase"/>
</Realm>

<Host name="localhost" appBase="webapps"
      unpackWARs="true" autoDeploy="true">

  <!-- SingleSignOn valve, share authentication between web applications
  Documentation at: /docs/config/valve.html -->
  <!--
  <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
  -->

  <!-- Access log processes all example.
  Documentation at: /docs/config/valve.html
  Note: The pattern used is equivalent to using pattern="common" -->
  <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
        prefix="localhost_access_log" suffix=".txt"
        pattern="%h %l %u %t &quot;%r&quot; %s %b" />

```

Añadir roles al usuario admin.

Ni el nombre de usuario de usuario ni la clave son seguras, pero es sólo para el ejercicio.

```

to operate the "/manager/html" web application. If you wish to use this app,
you must define such a user - the username and password are arbitrary.

Built-in Tomcat manager roles:
- manager-gui - allows access to the HTML GUI and the status pages
- manager-script - allows access to the HTTP API and the status pages
- manager-jmx - allows access to the JMX proxy and the status pages
- manager-status - allows access to the status pages only

The users below are wrapped in a comment and are therefore ignored. If you
wish to configure one or more of these users for use with the manager web
application, do not forget to remove the <!-- ... --> that surrounds them. You
will also need to set the passwords to something appropriate.
-->
<user username="admin" password="0000" roles="manager-gui,admin-gui"/>
<!--
<user username="robot" password="<must-be-changed>" roles="manager-script"/>
-->
<!--
The sample user and role entries below are intended for use with the
examples web application. They are wrapped in a comment and thus are ignored
when reading this file. If you wish to configure these users for use with the
examples web application, do not forget to remove the <!-- ... --> that surrounds
them. You will also need to set the passwords to something appropriate.

```


Tomcat ejecutándose en el navegador

