



SEP

SES

TecNM

INSTITUTO TECNOLÓGICO DE TOLUCA

Computo en la Nube

“Tema 1 Práctica 2”

PRESENTA:
González Ramírez Alejandro Emmanuel
No. CONTROL:
19280735

Profesor:
Luis Antonio Estrada Manuel

METEPEC, ESTADO DE MÉXICO, septiembre de 2023

Contenido

Instalación de un servidor de aplicaciones java	2
Tomcat	2
GlassFish.....	8
Administración del servidor con Cockpit	15
Filezilla	17

Instalación de un servidor de aplicaciones java

Tomcat

Primero, se realiza la instalación del paquete con la versión por defecto de java para el SO, en este caso con Debian 12, la versión 17

```
17:20:10.12 - Putty
File /etc/apt/sources.list saved
root@debian:~# apt-get update
Hit:1 https://packages.sury.org/php bookworm InRelease
Hit:2 http://deb.debian.org/debian bookworm InRelease
Get:3 http://security.debian.org/debian-security bookworm-security InRelease [48.0 kB]
Get:4 http://deb.debian.org/debian bookworm/non-free amd64 Packages [98.6 kB]
Get:5 http://security.debian.org/debian-security bookworm-security/main amd64 Packages [62.0 kB]
Get:6 http://deb.debian.org/debian bookworm/non-free Translation-en [67.2 kB]
Get:7 http://deb.debian.org/debian bookworm/non-free amd64 DEP-11 Metadata [4,428 B]
Get:8 http://security.debian.org/debian-security bookworm-security/main Translation-en [37.5 kB]
Fetched 318 kB in 2s (182 kB/s)
Reading package lists... Done
N: Repository 'Debian bookworm' changed its 'non-free component' value from 'non-free' to 'non-free non-free-firmware'
N: More information about this can be found online in the Release notes at: https://www.debian.org/releases/bookworm/amd64/release-notes/ch-information.html#non-free-split
root@debian:~# apt install default-jre default-jdk ←
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
default-jre is already the newest version (2:1.17-74).
default-jdk is already the newest version (2:1.17-74).
0 upgraded, 0 newly installed, 0 to remove and 31 not upgraded.
8 not fully installed or removed.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Se configura la variable de entorno de java y se verifica la versión

```
GNU nano 7.2           /etc/environment
JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64/"

[ Read 1 line ] ^G Help ^O Write Out ^W Where Is ^K Cut
^X Exit ^R Read File ^\ Replace ^U Paste
```

Configurar glassfish en Debian 11

```
# apt install -y unzip wget
Necesitamos Java antes de glassfish
# java -version
//probablemente nos de un error de que no conoce el comando java, significa que no tenemos instalado java

Para instalar jdk 17          NO LO RECOMIENDO
# apt install openjdk-17-jdk

Para instalar jdk 11
# apt install default-jre default-jdk

# java -version
openjdk version "11...."
openjdk version "17...."

Para cambiar de java version
# update-alternatives --config java

Agregamos la variable de ambiente JAVA_HOME
# nano /etc/environment

//agregamos al final del archivo
JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64/"
//guardamos y salimos

Se recarga el archivo
# source /etc/environment

# echo $JAVA_HOME
/usr/lib/jvm/java-11-openjdk-amd64/
# wget https://download.eclipse.org/ee4j/glassfish/glassfish-6.2.5.zip
# mv glassfish-6.2.5.zip /opt/
# cd /opt
```

```
192.168.1.11 ~# java --version
openjdk 17.0.8 2023-07-18
OpenJDK Runtime Environment (build 17.0.8+7-Debian-1deb12u1)
OpenJDK 64-Bit Server VM (build 17.0.8+7-Debian-1deb12u1, mixed mode, sharing)
root@debian:~#
```

Se aplica el archivo en el Shell y se verifica la variable de entorno

```
root@debian:/# source /etc/environment
root@debian:/# echo $JAVA_HOME
/usr/lib/jvm/java-11-openjdk-amd64/
root@debian:/#
```

Una vez realizado, se instala el servidor de aplicaciones java

apt update

apt install -y tomcat10 tomcat10-admin

The terminal window shows the command: `root@debian:~# apt install -y tomcat10 tomcat10-admin`. The file editor window shows a script named "Configurar_debian.txt" with the following content:

```
Archivo Editar Ver
# apt install -y cockpit
# systemctl status cockpit.socket
# ufw allow 9090/tcp

Desde un cliente abrir navegador web
https://IP:9090

Tomcat 9
# apt update
# apt install -y tomcat9 tomcat9-admin
# ufw allow 8080/tcp
# ufw allow 8443/tcp

Creamos un certificado autofirmado y creado por nosotros
# mkdir /etc/keys

# keytool -genkey -alias tomcat9 -keyalg RSA -storetype PKCS12 -keystore /etc/keys/tomcat9.jks
Tomcat
Y nos pide muchos datos, confirmamos con yes

Editamos la llave
# nano /etc/tomcat9/server.xml

Dentro del bloque Service llamado «Catalina» añadiremos una directiva Connector para las conexiones seguras:

<Connector port="8443" maxThreads="150" scheme="https" secure="true"
Ln 297, Col 39
```

Una vez realizado, se arreglan las reglas de puertos que permitirán el acceso al servidor

ufw allow 8080/tcp

ufw allow 8443/tcp

Y se configura el acceso al servicio con **keytool -genkey -alias tomcat10 -keyalg RSA -storetype PKCS12 -keystore /etc/keys/tomcat10.jks** junto a una corta configuración de los datos

```
172.20.10.12 - PuTTY
root@debian:~# ufw allow 8080/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
root@debian:~# ufw allow 8443/tcp
Skipping adding existing rule (v6)
root@debian:~# mkdir /etc/keys
mkdir: cannot create directory '/etc/keys': File exists
root@debian:~# keytool -genkey -alias tomcat10 -keyalg RSA -storetype PKCS12 -keystore /etc/keys/tomcat10.jks
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: Alejandro Gonzalez
What is the name of your organizational unit?
[Unknown]: 1234
What is the name of your organization?
[Unknown]: 1234
What is the name of your City or Locality?
[Unknown]: 1234
What is the name of your State or Province?
[Unknown]: 1234
What is the two-letter country code for this unit?
[Unknown]: 1234
Is CN=Alejandro Gonzalez, OU=1234, O=1234, L=1234, ST=1234, C=1234 correct?
[no]: Y

root@debian:~# ufw allow 8080/tcp
root@debian:~# ufw allow 8443/tcp
root@debian:~# apt update
root@debian:~# apt install -y tomcat9 tomcat9-admin
root@debian:~# ufw allow 8080/tcp
root@debian:~# ufw allow 8443/tcp
root@debian:~# keytool -genkey -alias tomcat10 -keyalg RSA -storetype PKCS12 -keystore /etc/keys/tomcat10.jks
Tomcat
Y nos pide muchos datos, confirmamos con yes
Editamos la llave
root@debian:~# nano /etc/tomcat9/server.xml

Dentro del bloque Service llamado «Catalina» añadiremos una directiva Connector para las conexiones seguras:

<Connector port="8443" maxThreads="150" scheme="https" secure="true"
           SSLEnabled="true" keystoreFile="/etc/keys/tomcat9.jks"
           keystorePass="XXXXXXXX" clientAuth="false" keyAlias="tomcat9"
           sslProtocol="TLS" />

Guardamos y salimos
```

Is CN=Alejandro Gonzalez, OU=1234, O=1234, L=1234,
ST=1234, C=1234 correct?
[no]: Y

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 90 days

```
for: CN=Alejandro Gonzalez, OU=1234, O=1234  
, L=1234, ST=1234, C=1234  
root@debian:~#
```

Una vez configurado se manipula el archivo de servidor para asignar las propiedades de la conexión, debido a que no es posible acceder a la página con el protocolo https, se configura la redirección de puertos con la siguiente configuración.

```

GNU nano 7.2                               /etc/tomcat10/server.xml

<Service name="Catalina">

    <!--The connectors can use a shared executor, you can define one or more named thread pools-->
    <!--
        <Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
            maxThreads="150" minSpareThreads="4"/>
    -->

    <!-- A "Connector" represents an endpoint by which requests are received
        and responses are returned. Documentation at :
            HTTP Connector: /docs/config/http.html
            AJP  Connector: /docs/config/ajp.html
        Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
    -->

    <Connector executor="tomcatThreadPool"
        port="8443" protocol="HTTP/1.1"
        connectionTimeout="20000"
        redirectPort="8443" />

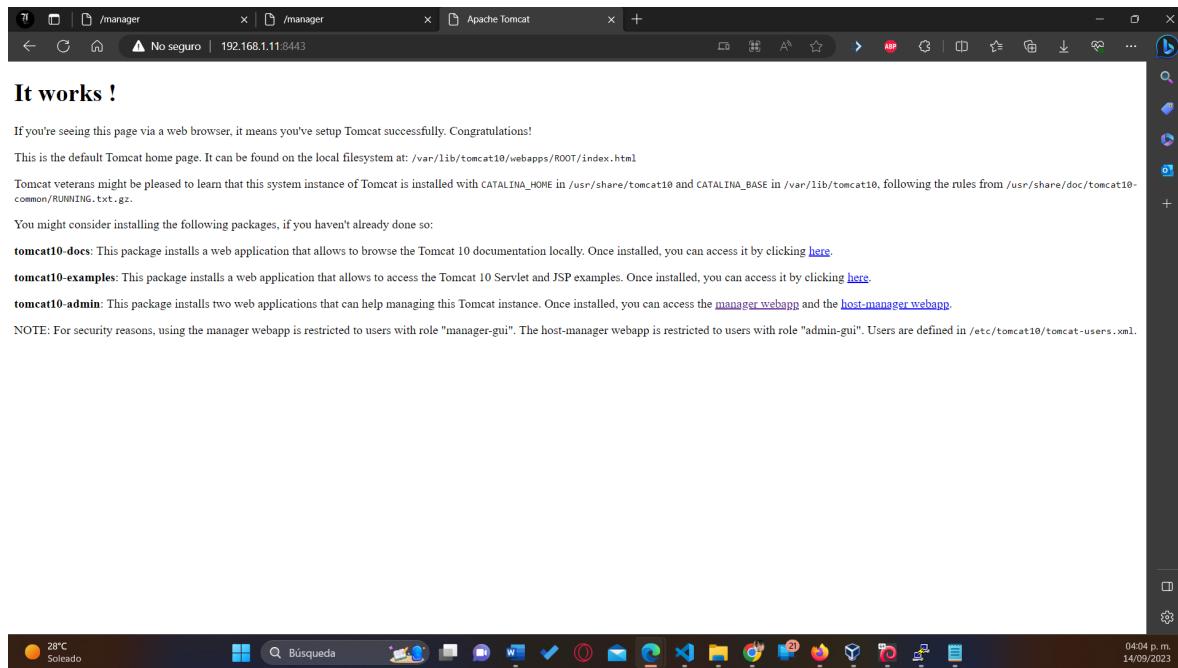
    <Connector port="8443" maxThreads="150" scheme="https" secure="true"
        SSLEnabled="true" keystoreFile="/etc/keys/tomcat10.jks"
        keystorePass="XXXXXXXXX" clientAuth="false" keyAlias="tomcat10"
        sslProtocol="TLS" />
    <!-- A "Connector" using the shared thread pool-->
    <!--
        <Connector executor="tomcatThreadPool"
            port="8080" protocol="HTTP/1.1"

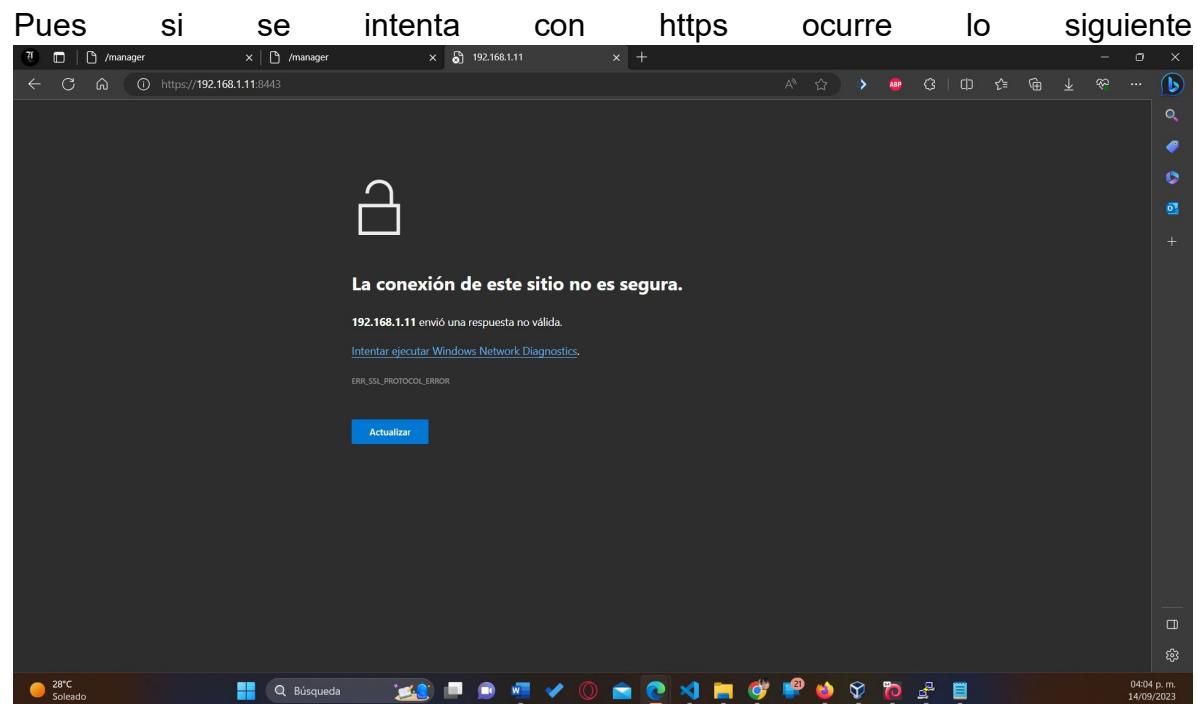
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
 ^X Exit ^R Read File ^A Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo

Para que sea posible acceder a la página del servidor con la url

<http://192.168.1.11:8443/>





Finalmente, se agregan los usuarios para el acceso

```
GNU nano 7.2          /etc/tomcat10/tomcat-users.xml *
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 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="" roles="manager-gui"/>
<user username="robot" password="" roles="manager-scr"/>
-->

<user username="usu" password="PWD" roles="manager-gui,admin-gui"/> ← Yellow arrow points here
<user username="alex" password="GOD_PIECE" roles="manager-gui,admin-gui"/>
<user username="jacquie" password="PRINCESS" roles="manager-gui,admin-gui"

<!--
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 examples web application, do not forget to remove the <!...> that surrounds them. You will also need to set the passwords to something appropriate.
-->
<!--
<role rolename="tomcat"/>
<role rolename="role1"/>
<user username="tomcat" password="" roles="tomcat"/>
<user username="both" password="" roles="tomcat,role1"/>
<user username="role1" password="" roles="role1"/>
-->
</tomcat-users>
```

Configuramos usuarios para tomcat

```
# nano /etc/tomcat10/tomcat-users.xml
```

Dentro del bloque tomcat-users añadiremos una directiva user:

```
<user username="usu" password="PWD" roles="manager-gui,admin-gui"/>
```

Guardamos y salimos

```
https://192.168.122.13:8443/manager
```

Nos pide el usuario y contraseña

Ejemplo

```
pepe/pepe123
juan/juan123
roberto/roberto123
chonto/chonto123
```

GlassFish en Debian 11

```
# apt install -y unzip wget
```

Necesitamos Java antes de glassfish

```
# java -version
//probablemente nos de un error de que no conoce el comando java, significa que no tenemos instalado java
```

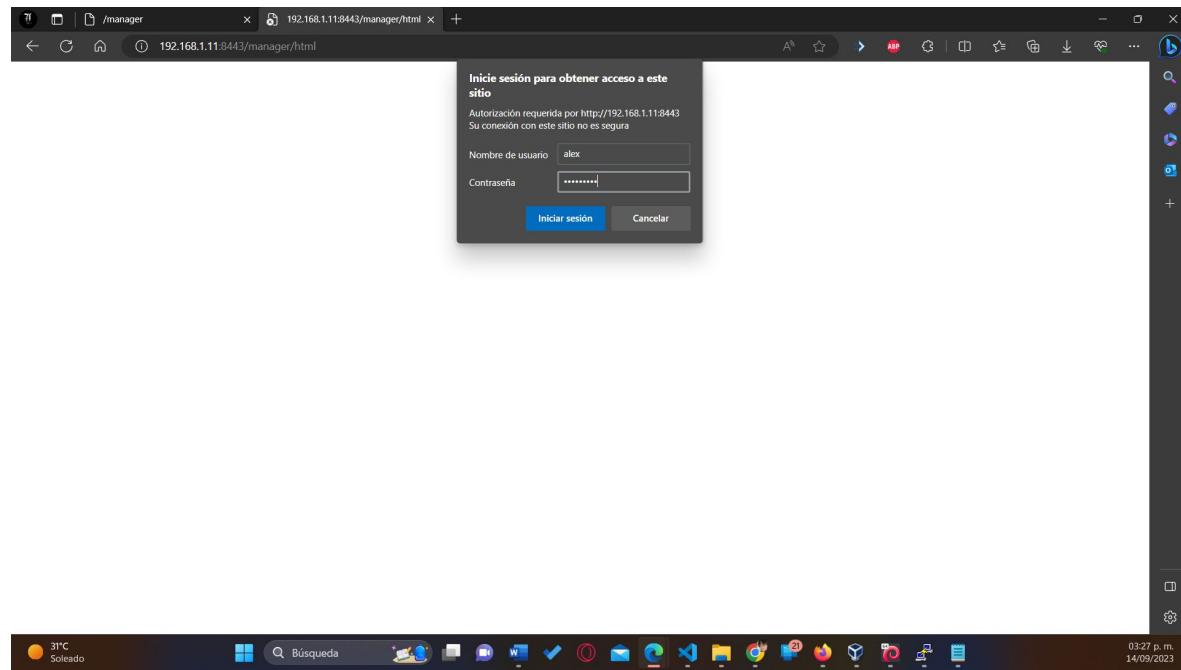
Para instalar jdk 17 NO LO RECOMIENDO

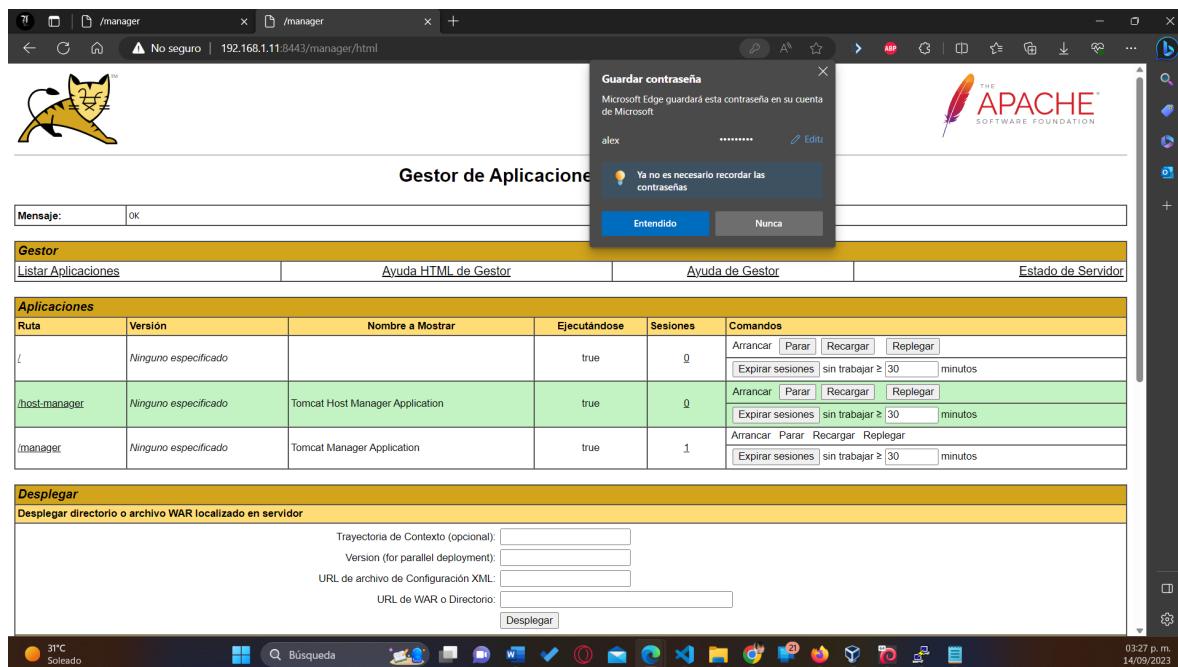
```
# apt install openjdk-17-jdk
```

Para instalar jdk 11

```
# apt install default-jre default-jdk
```

Y con la url <http://192.168.1.11:8443/manager> se accede





GlassFish

Se configura el paquete y se verifica la versión de java junto a la variable de entorno

```
root@debian:/# apt install -y unzip wget
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
unzip is already the newest version (6.0-28).
wget is already the newest version (1.21.3-1+b2).
0 upgraded, 0 newly installed, 0 to remove and 27 not upgraded.
root@debian:/# java -version
openjdk version "17.0.8" 2023-07-18
OpenJDK Runtime Environment (build 17.0.8+7-Debian-1deb12u1)
OpenJDK 64-Bit Server VM (build 17.0.8+7-Debian-1deb12u1, mixed mode, sharing)
root@debian:/# nano /etc/environment
root@debian:/# source /etc/environment
root@debian:/# echo $JAVA_HOME
/usr/lib/jvm/java-11-openjdk-amd64/ ← Orange arrow points here
root@debian:/#
```

```
Configurar_debian.txt
Archivo Editar Ver
# java -version
//probablemente nos de un error de que no conoce el comando java, significa que no tenemos instalado java
Para instalar jdk 17 NO LO RECOMIENDO
# apt install openjdk-17-jdk
Para instalar jdk 11
# apt install default-jre default-jdk
# java -version
openjdk version "11...."
openjdk version "17...."
Para cambiar de java version
# update-alternatives --config java
Agregamos la variable de ambiente JAVA_HOME
# nano /etc/environment
//agregamos al final del archivo
JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64/"
//guardamos y salimos
Se recarga el archivo
# source /etc/environment
# echo $JAVA_HOME
/usr/lib/jvm/java-11-openjdk-amd64/
# wget https://download.eclipse.org/ee4j/glassfish/glassfish-6.2.5.zip
# mv glassfish-6.2.5.zip /opt/
# cd /opt
# unzip glassfish-6.2.5.zip
Creamos un usuario para glassfish
# useradd -s /sbin/nologin glassfish
```

Se descargan los paquetes

```

root@debian:/# wget https://download.eclipse.org/ee4j/glassfish/glassfish-6.2.5.zip
--2023-09-14 19:00:25-- https://download.eclipse.org/ee4j/glassfish/glassfish-6.2.5.zip
Resolving download.eclipse.org (download.eclipse.org)...
198.41.30.199
Connecting to download.eclipse.org (download.eclipse.org)|198.41.30.199|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 118284616 (113M) [application/zip]
Saving to: 'glassfish-6.2.5.zip'

glassfish-6.2.5. 11% 12.80M 5.67MB/s

```

```

# java -version
//probablemente nos de un error de que no conoce el comando java, significa que no tenemos instalado java

Para instalar jdk 17 NO LO RECOMIENDO
# apt install openjdk-17-jdk

Para instalar jdk 11
# apt install default-jre default-jdk

# java -version
openjdk version "11...."
openjdk version "17...."

Para cambiar de java version
# update-alternatives --config java

Agregamos la variable de ambiente JAVA_HOME
# nano /etc/environment

//agregamos al final del archivo
JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64/"
//guardamos y salimos

Se recarga el archivo
# source /etc/environment

# echo $JAVA_HOME
/usr/lib/jvm/java-11-openjdk-amd64

# wget https://download.eclipse.org/ee4j/glassfish/glassfish-6.2.5.zip

# mv glassfish-6.2.5.zip /opt/
# cd /opt/
# unzip glassfish-6.2.5.zip

Creamos un usuario para glassfish
# useradd -s /sbin/nologin glassfish

```

Y se extraen los paquetes descargados

mv glassfish-6.2.5.zip /opt/

cd /opt

unzip glassfish-6.2.5.zip

```

inflating: glassfish6/mq/lib/images/admin/folder.gif
inflating: glassfish6/mq/lib/images/admin/splash_openmq.gif
inflating: glassfish6/mq/lib/img.jar
inflating: glassfish6/mq/lib/imgadmin.jar
inflating: glassfish6/mq/lib/imgbridgemgr.jar
inflating: glassfish6/mq/lib/imgbroker.jar
inflating: glassfish6/mq/lib/imghttp.war
inflating: glassfish6/mq/lib/imghttps.war
inflating: glassfish6/mq/lib/imgjmsbridge.jar
inflating: glassfish6/mq/lib/imgjmsra.rar
inflating: glassfish6/mq/lib/imgjmx.jar
inflating: glassfish6/mq/lib/imgservlet.jar
inflating: glassfish6/mq/lib/imgstomp.jar
inflating: glassfish6/mq/lib/imgums.war
inflating: glassfish6/mq/lib/imgutil.jar
inflating: glassfish6/mq/lib/imgxmx.jar
inflating: glassfish6/mq/lib/jakarta.json.jar
inflating: glassfish6/mq/lib/jakarta.servlet-api.jar
inflating: glassfish6/mq/lib/jakarta.transaction-api.jar
inflating: glassfish6/mq/lib/jaxm-api.jar
inflating: glassfish6/mq/lib/jhall.jar
inflating: glassfish6/mq/lib/jms.jar
inflating: glassfish6/mq/lib/props/broker/default.properties
inflating: glassfish6/mq/lib/props/broker/install.properties
inflating: glassfish6/mq/lib/tyrus-standalone-client.jar
root@debian:/opt#

```

```

# java -version
openjdk version "11...."
openjdk version "17...."

Para cambiar de java version
# update-alternatives --config java

Agregamos la variable de ambiente JAVA_HOME
# nano /etc/environment

//agregamos al final del archivo
JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64/"
//guardamos y salimos

Se recarga el archivo
# source /etc/environment

# echo $JAVA_HOME
/usr/lib/jvm/java-11-openjdk-amd64

# wget https://download.eclipse.org/ee4j/glassfish/glassfish-6.2.5.zip

# mv glassfish-6.2.5.zip /opt/
# cd /opt/
# unzip glassfish-6.2.5.zip

Creamos un usuario para glassfish
# useradd -s /sbin/nologin glassfish

Cambiamos propietario
# chown -R glassfish:glassfish /opt/glassfish6

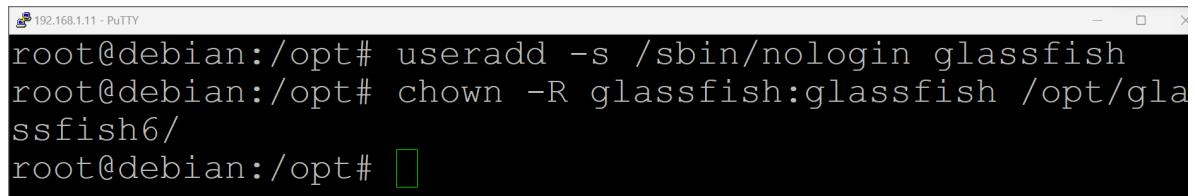
# nano /usr/lib/systemd/system/glassfish.service

[Unit]

```

Creamos un usuario para glassfish

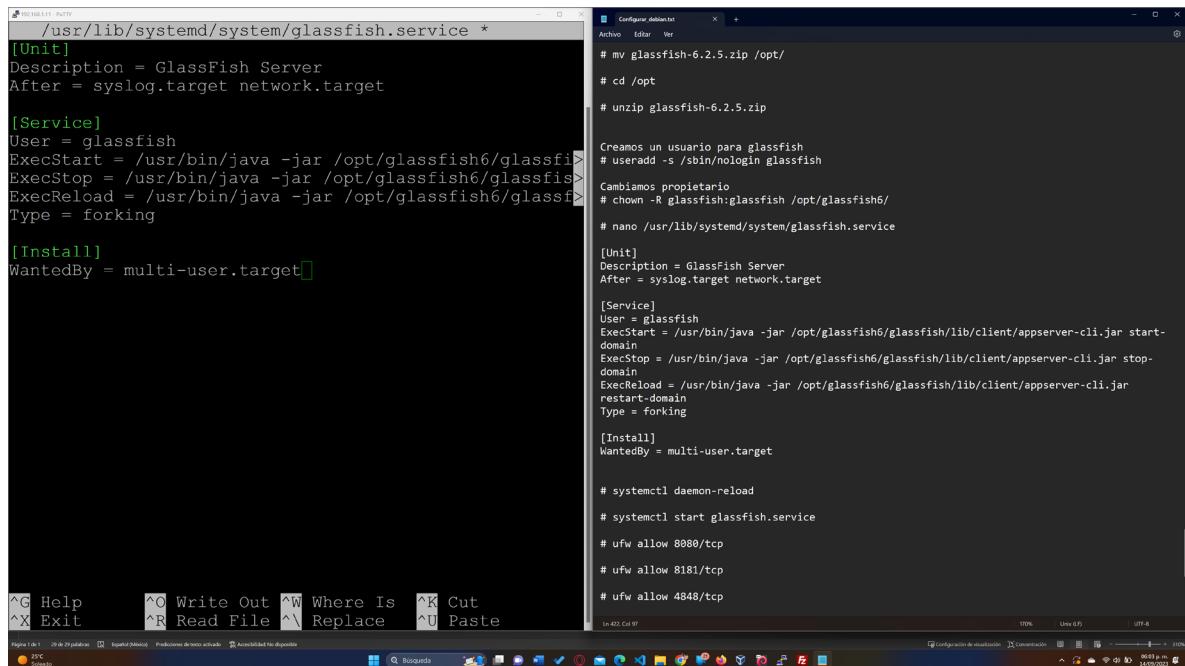
useradd -s /sbin/nologin glassfish y se agregan los permisos con **chown -R glassfish:glassfish /opt/glassfish6/**



```
root@debian:/opt# useradd -s /sbin/nologin glassfish
root@debian:/opt# chown -R glassfish:glassfish /opt/glassfish6/
root@debian:/opt#
```

Editamos el archivo de configuración:

nano /usr/lib/systemd/system/glassfish.service



```
/usr/lib/systemd/system/glassfish.service *
[Unit]
Description = GlassFish Server
After = syslog.target network.target

[Service]
User = glassfish
ExecStart = /usr/bin/java -jar /opt/glassfish6/glassfish>
ExecStop = /usr/bin/java -jar /opt/glassfish6/glassfish>
ExecReload = /usr/bin/java -jar /opt/glassfish6/glassfi>
Type = forking

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

```
Archivo Editar Ver
# mv glassfish-6.2.5.zip /opt/
# cd /opt
# unzip glassfish-6.2.5.zip

Creamos un usuario para glassfish
# useradd -s /sbin/nologin glassfish
Cambiamos propietario
# chown -R glassfish:glassfish /opt/glassfish6/
# nano /usr/lib/systemd/system/glassfish.service

[Unit]
Description = GlassFish Server
After = syslog.target network.target

[Service]
User = glassfish
ExecStart = /usr/bin/java -jar /opt/glassfish6/glassfish/lib/client/appserver-cl>
ExecStop = /usr/bin/java -jar /opt/glassfish6/glassfish/lib/client/appserver-cl.jar stop->
ExecReload = /usr/bin/java -jar /opt/glassfish6/glassfish/lib/client/appserver-cl.jar
restart-domain
Type = forking

[Install]
WantedBy = multi-user.target

# systemctl daemon-reload
# systemctl start glassfish.service
# ufw allow 8080/tcp
# ufw allow 8181/tcp
# ufw allow 4848/tcp
```

Habilitamos el servicio y lo iniciamos con

systemctl daemon-reload

systemctl start glassfish.service

ufw allow 8080/tcp

ufw allow 8181/tcp

ufw allow 4848/tcp

systemctl enable glassfish.service

```

root@debian:/opt# useradd -s /sbin/nologin glassfish
root@debian:/opt# chown -R glassfish:glassfish /opt/glassfish6/
root@debian:/opt# nano /usr/lib/systemd/system/glassfish6.service
root@debian:/opt# systemctl daemon-reload
root@debian:/opt# systemctl start glassfish.service
root@debian:/opt# ufw allow 8080/tcp
Rule added
Rule added (v6)
root@debian:/opt# ufw allow 8181/tcp
Rule added
Rule added (v6)
root@debian:/opt# ufw allow 4848/tcp
Rule added
Rule added (v6)
root@debian:/opt# systemctl enable glassfish.service
Created symlink /etc/systemd/system/multi-user.target.wants/glassfish.service → /lib/systemd/system/glassfish.service.
root@debian:/opt# ./opt/glassfish6/bin/asadmin --port 4848 change-admin-password
Enter admin user name [default: admin]>[REDACTED]

```

```

[Service]
User = glassfish
ExecStart = /usr/bin/java -jar /opt/glassfish6/glassfish/lib/client/appserver-cli.jar start-domain
ExecStop = /usr/bin/java -jar /opt/glassfish6/glassfish/lib/client/appserver-cli.jar stop-domain
ExecReload = /usr/bin/java -jar /opt/glassfish6/glassfish/lib/client/appserver-cli.jar restart-domain
Type = forking

[Install]
WantedBy = multi-user.target

# systemctl daemon-reload
# systemctl start glassfish.service
# ufw allow 8080/tcp
# ufw allow 8181/tcp
# ufw allow 4848/tcp
# systemctl enable glassfish.service
# ./opt/glassfish6/bin/asadmin --port 4848 change-admin-password
admin
PWD
PWD

```

Se cambia la contraseña con `/opt/glassfish6/bin/asadmin --port 4848 change-admin-password`

```

root@debian:/opt# ls
glassfish6  glassfish-6.2.5.zip
root@debian:/opt# ./opt/glassfish6/bin/asadmin --port 4848 change-admin-password
Enter admin user name [default: admin]>
Enter the admin password>
Enter the new admin password>
Enter the new admin password again>
Command change-admin-password executed successfully.
root@debian:/opt# [REDACTED]

```

```

[Service]
User = glassfish
ExecStart = /usr/bin/java -jar /opt/glassfish6/glassfish/lib/client/appserver-cli.jar start-domain
ExecStop = /usr/bin/java -jar /opt/glassfish6/glassfish/lib/client/appserver-cli.jar stop-domain
ExecReload = /usr/bin/java -jar /opt/glassfish6/glassfish/lib/client/appserver-cli.jar restart-domain
Type = forking

[Install]
WantedBy = multi-user.target

# systemctl daemon-reload
# systemctl start glassfish.service
# ufw allow 8080/tcp
# ufw allow 8181/tcp
# ufw allow 4848/tcp
# systemctl enable glassfish.service
# ./opt/glassfish6/bin/asadmin --port 4848 change-admin-password
admin
PWD

```

```
root@debian:/opt# /opt/glassfish6/bin/asadmin enable-secure-admin
Enter admin user name> admin
Enter admin password for user "admin"
You must restart all running servers for the change in secure admin to take effect.
Command enable-secure-admin executed successfully.
root@debian:/opt# /opt/glassfish6/bin/asadmin --port 4848 enable-secure-admin
Enter admin user name> admin
Enter admin password for user "admin"
You must restart all running servers for the change in secure admin to take effect.
Command enable-secure-admin executed successfully.
root@debian:/opt# systemctl restart glassfish.service
root@debian:/opt# systemctl STATUS glassfish.service
Unknown command verb STATUS.
root@debian:/opt# systemctl status glassfish.service
● glassfish.service - GlassFish Server
   Loaded: loaded (/lib/systemd/system/glassfish.service)
   Active: active (running) since Thu 2023-09-14 19:33:58 UTC
     Process: 3361 ExecStart=/usr/bin/java -jar /opt/glassfish6/glassfish/lib/glassfish.jar
    Main PID: 3385 (java)
      Tasks: 93 (limit: 9474)
     Memory: 260.5M
        CPU: 10.879s
       CGroup: /system.slice/glassfish.service
               └─3385 /usr/lib/jvm/java-17-openjdk-amd64/bin/java -jar /opt/glassfish6/glassfish/lib/glassfish.jar

Sep 14 19:14:08 debian systemd[1]: Starting glassfish>
Sep 14 19:14:10 debian java[3361]: Waiting for domain>
Sep 14 19:14:10 debian java[3361]: Successfully starte>
Sep 14 19:14:10 debian java[3361]: domain Location:>

[Configure database] x +
Archivo Edición Ver
restar-domain
Tipo = forking

[Install]
WantedBy = multi-user.target

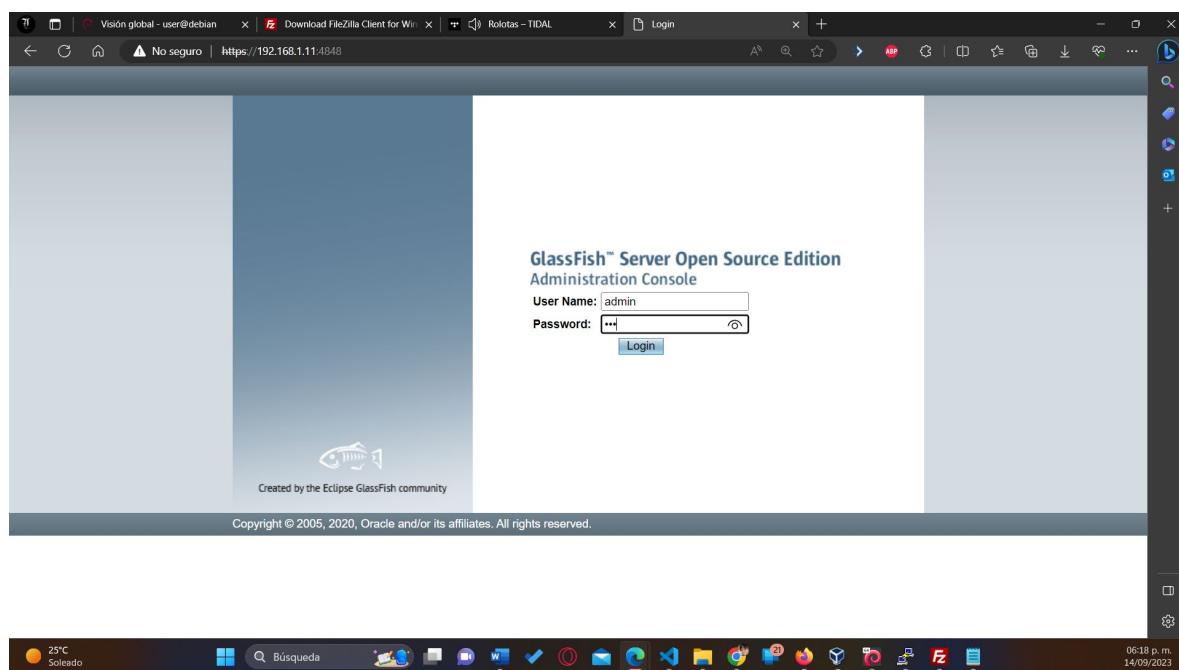
# systemctl daemon-reload
# systemctl start glassfish.service
# ufw allow 8080/tcp
# ufw allow 8181/tcp
# ufw allow 4848/tcp
# systemctl enable glassfish.service
# /opt/glassfish6/bin/asadmin --port 4848 change-admin-password
admin
PWD
PWD

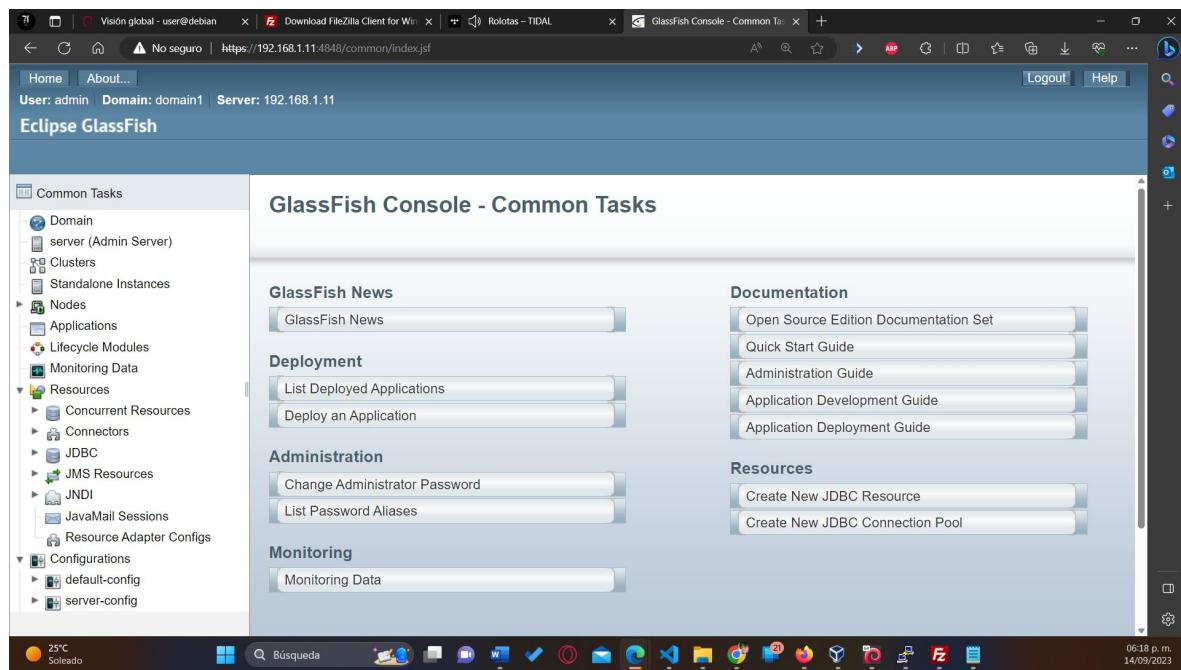
# /opt/glassfish6/bin/asadmin enable-secure-admin
# /opt/glassfish6/bin/asadmin --port 4848 enable-secure-admin
# systemctl restart glassfish.service
```

Posteriormente, se coloca el comando **/opt/glassfish6/bin/asadmin enable-secure-admin** para habilitar la seguridad del administrador y **/opt/glassfish6/bin/asadmin --port 4848 enable-secure-admin** para los puertos, además de reiniciar el servicio

```
192.168.1.11 - PuTTY
root@debian:/opt# /opt/glassfish6/bin/asadmin enable-secure-admin
Enter admin user name> admin
Enter admin password for user "admin">
You must restart all running servers for the change in
secure admin to take effect.
Command enable-secure-admin executed successfully.
root@debian:/opt#
```

```
root@debian:/opt/glassfish6/bin/asadmin --port 4848 enable-secure-admin
Enter admin user name> admin
Enter admin password for user "admin">
You must restart all running servers for the change in
secure admin to take effect.
Command enable-secure-admin executed successfully.
root@debian:/opt# systemctl restart glassfish.service
root@debian:/opt# systemctl STATUS glassfish.service
Unknown command verb STATUS.
root@debian:/opt# systemctl status glassfish.service
● glassfish.service - GlassFish Server
   Loaded: loaded (/lib/systemd/system/glassfish.service)
   Active: active (running) since Thu 2023-09-14 19:22:21 -05
     Process: 3361 ExecStart=/usr/bin/java -jar /opt/glassfish6/glassfish
      Main PID: 3385 (java)
        Tasks: 93 (limit: 9474)
       Memory: 260.5M
          CPU: 10.879s
        CGroup: /system.slice/glassfish.service
                  └─3385 /usr/lib/jvm/java-17-openjdk-amd64/jre/bin/java
```





Administración del servidor con Cockpit

Para realizar la administración del servidor con un entorno gráfico en otro dispositivo se implementa el paquete Cockpit, primero instalándolo con el comando

apt update

apt install -y cockpit

Posterior a ello se corrobora el estatus del servicio con **systemctl status cockpit.socket** y se agrega el acceso al puerto del paquete con **ufw allow 9090/tcp**

The screenshot shows a terminal window with several command-line sessions. The first session shows the installation of Cockpit:

```
root@debian:~# apt install -y cockpit
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
cockpit is already the newest version (287-1).
0 upgraded, 0 newly installed, 0 to remove and 27 not upgraded
.
root@debian:~# systemctl status cockpit.socket
● cockpit.socket - Cockpit Web Service Socket
  Loaded: loaded (/lib/systemd/system/cockpit.socket; enabled; preset: enabled)
  Active: active (listening) since Thu 2023-09-14 16:17:38 UTC
    Docs: man:cockpit-ws(8)
   Triggers: ● cockpit.service
   Listen: [::]:9090 (Stream)
  Process: 602 ExecStartPost=/usr/share/cockpit/motd/update
  Process: 609 ExecStartPost=/bin/ln -snf active.motd /run/cockpit/motd
  Tasks: 0 (limit: 9474)
  Memory: 168.0K
    CPU: 10ms
   CGroup: /system.slice/cockpit.socket
```

The second session shows the status of the cockpit.socket service:

```
Sep 14 16:17:38 debian systemd[1]: Starting cockpit.socket
Sep 14 16:17:38 debian systemd[1]: Listening on cockpit.socket
root@debian:~# ufw allow 9090/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
root@debian:~#
```

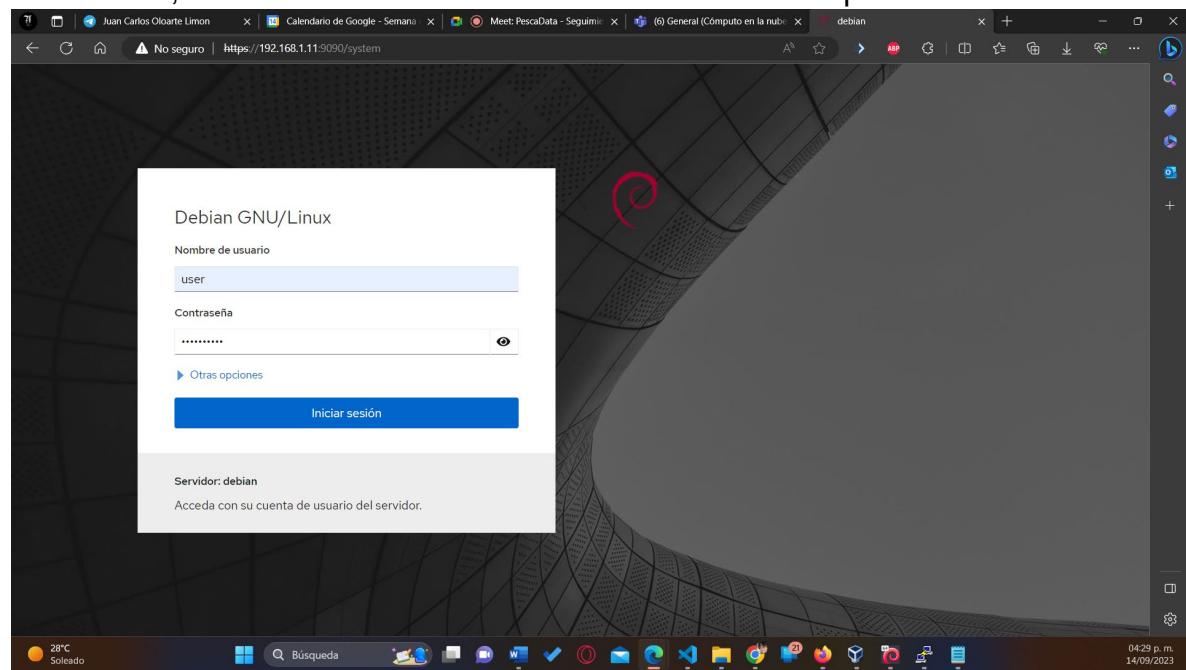
The third session shows the configuration of the firewall to allow port 9090:

```
# apt update
# apt install -y cockpit
# systemctl status cockpit.socket
# ufw allow 9090/tcp
Desde un cliente abrir navegador web
https://IP:9090

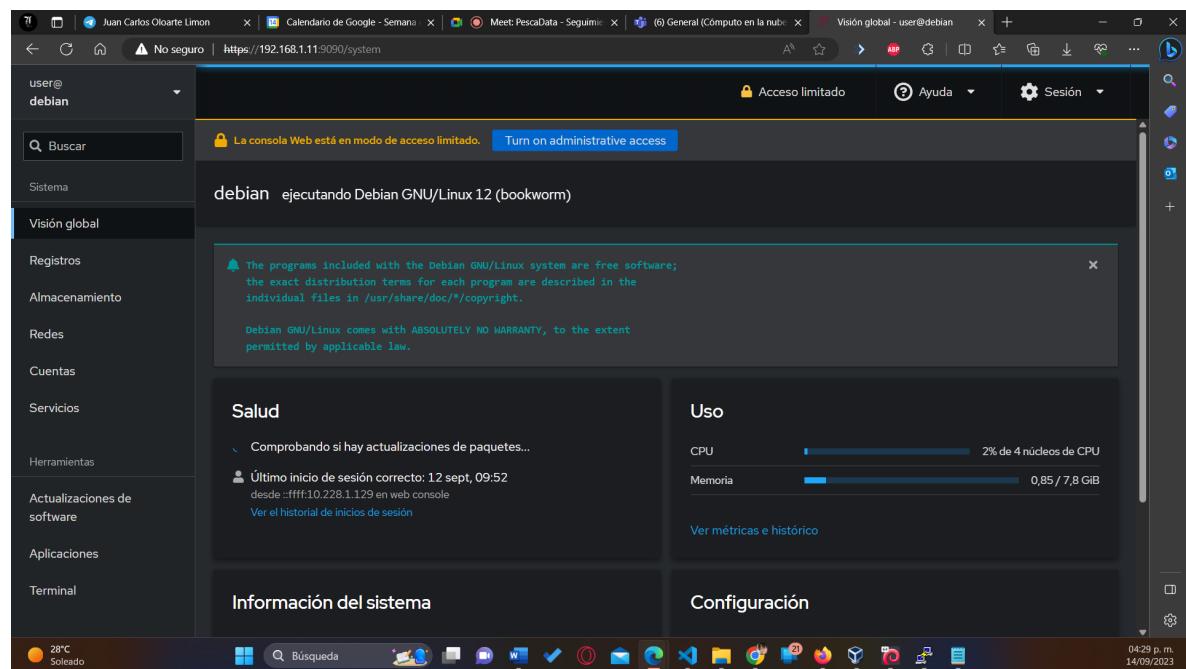
Tomcat 9
# apt update
# apt install -y tomcat10 tomcat10-admin
# ufw allow 8080/tcp
# ufw allow 8443/tcp
Creamos un certificado autofirmado y creado por nosotros
# mkdir /etc/keys
# keytool -genkey -alias tomcat10 -keyalg RSA -storetype PKCS12 -keystore /etc/keys/tomcat10.jks
Tomcat
Y nos pide muchos datos, confirmamos con yes
Editamos la llave
# nano /etc/tomcat10/server.xml
```

The terminal window also displays a file titled "Configurar debinit.txt" which contains notes about FileZilla and Tomcat 9 configuration.

Finalmente, se accede con un usuario que no sea root



Y ahora se puede monitorear el servidor acorde a los privilegios del usuario

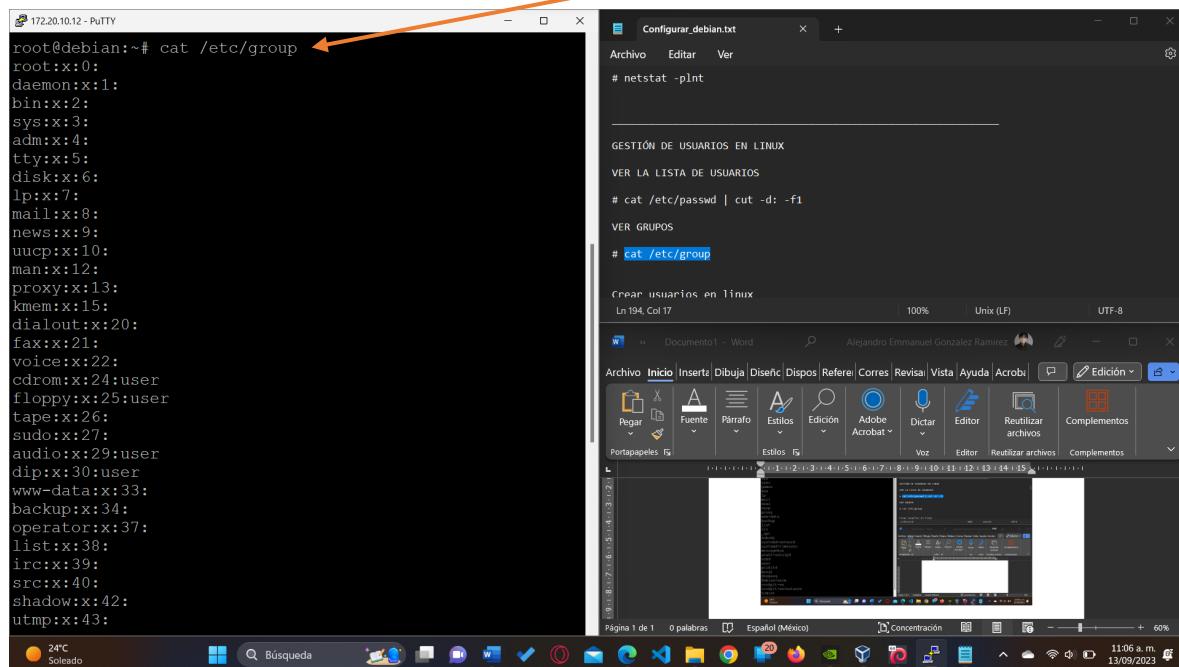
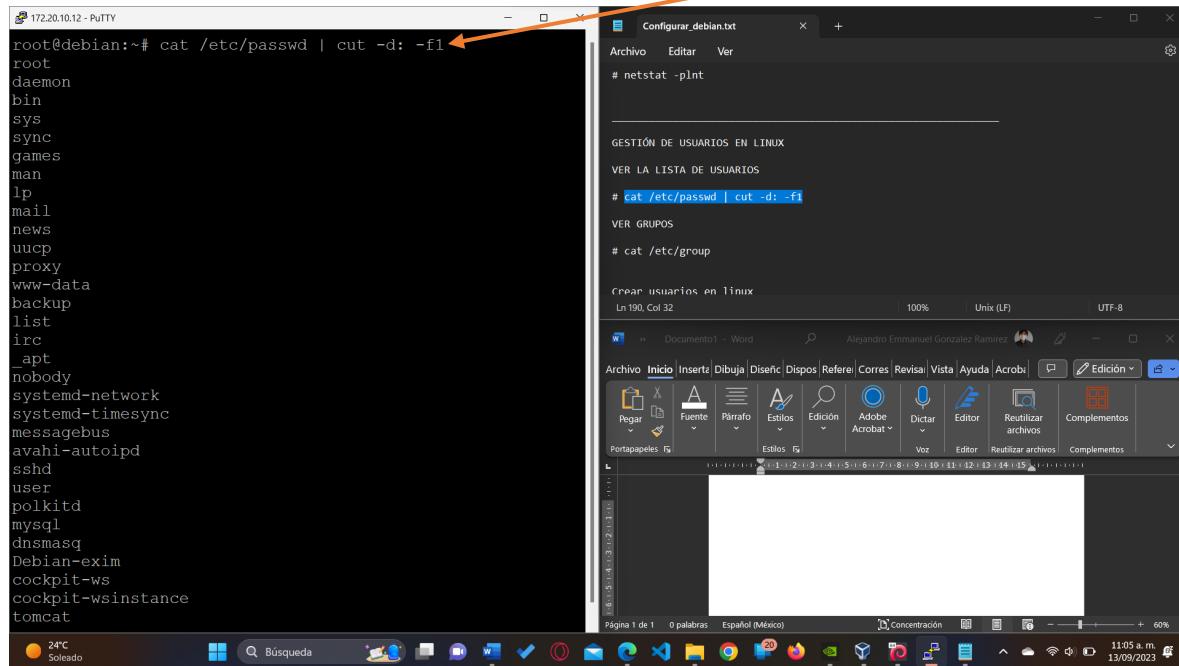


Filezilla

Para una modificación de archivos de forma remota, se emplea Filezilla

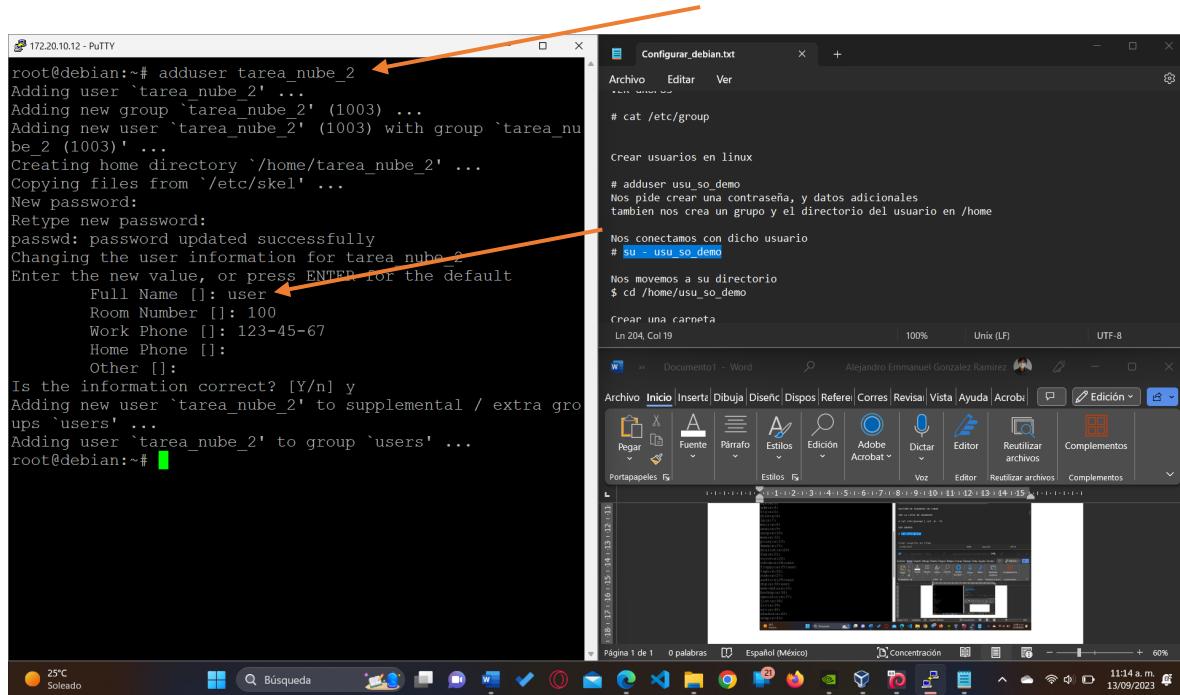
Para esto, se verifican los usuarios del servidor con:

cat /etc/passwd | cut -d: -f1 y sus grupos # cat /etc/group



Ahora, para probar la transferencia, se crea un nuevo usuario con:

adduser tarea_nube_2 y posteriormente se configuran los datos del usuario



Después, se crea una carpeta nueva accediendo al usuario con **su – tarea_nube_2** en la carpeta de usuario con: **Cd /home/tarea_nube_2/** creamos una carpeta para hacer las operaciones **mkdir ejemplo_carpeta** además, ver los permisos y los detalles con **ls -la**. Posterior a ello se asignan los permisos de la carpeta para que tenga todos los permisos de lectura, escritura y ejecución con **chmod 775 -R ejemplo_carpeta/**. Finalmente para la configuración del usuario se cambia de propietario una carpeta (solo se puede hacer con usuario root)

Su -

chown tarea_nube_2:tarea_nube2 -R demo1/

The screenshot shows a terminal window titled 'tarea_nube_2@debian: ~'. The user is root. The terminal history is as follows:

```
tarea_nube_2@debian:~# su - tarea_nube_2
tarea_nube_2@debian:~$ cd ~/home/tarea_nube_2/
tarea_nube_2@debian:~$ ls
tarea_nube_2@debian:~$ mkdir ejemplo_carpet
tarea_nube_2@debian:~$ ls -la
total 28
drwx----- 3 tarea_nube_2 tarea_nube_2 4096 Sep 13 19:31 .
drwxr-xr-x  6 root      root      4096 Sep 13 12:13 ..
-rw-----  1 tarea_nube_2 tarea_nube_2   20 Sep 13 19:31 .bash_hi
story
-rw-r--r--  1 tarea_nube_2 tarea_nube_2  220 Sep 13 12:13 .bash_lo
gout
-rw-r--r--  1 tarea_nube_2 tarea_nube_2 3526 Sep 13 12:13 .bashrc
drwxr-xr-x  2 tarea_nube_2 tarea_nube_2 4096 Sep 13 19:31 ejemplo_
carpet
-rw-r--r--  1 tarea_nube_2 tarea_nube_2   807 Sep 13 12:13 .profile
tarea_nube_2@debian:~$ chmod 744 -R ejemplo_carpet/
tarea_nube_2@debian:~$ chmod 744 -R ejemplo_carpet/nom_archivo.t
xt
chmod: cannot access 'ejemplo_carpet/nom_archivo.txt': No such f
ile or directory
tarea_nube_2@debian:~$ chmod 775 -R ejemplo_carpet/
tarea_nube_2@debian:~$ su -
Password:
root@debian:~# chown tarea_nube_2:tarea_nube_2 -R demo1/
chown: cannot access 'demo1/': No such file or directory
root@debian:~# chown tarea_nube_2:tarea_nube_2 -R ejemplo_carpet/
chown: cannot access 'ejemplo_carpet/': No such file or director
y
root@debian:~# chown tarea_nube_2:tarea_nube_2 -R /home/tarea_nub
e_2/ejemplo_carpet/
root@debian:~#
```

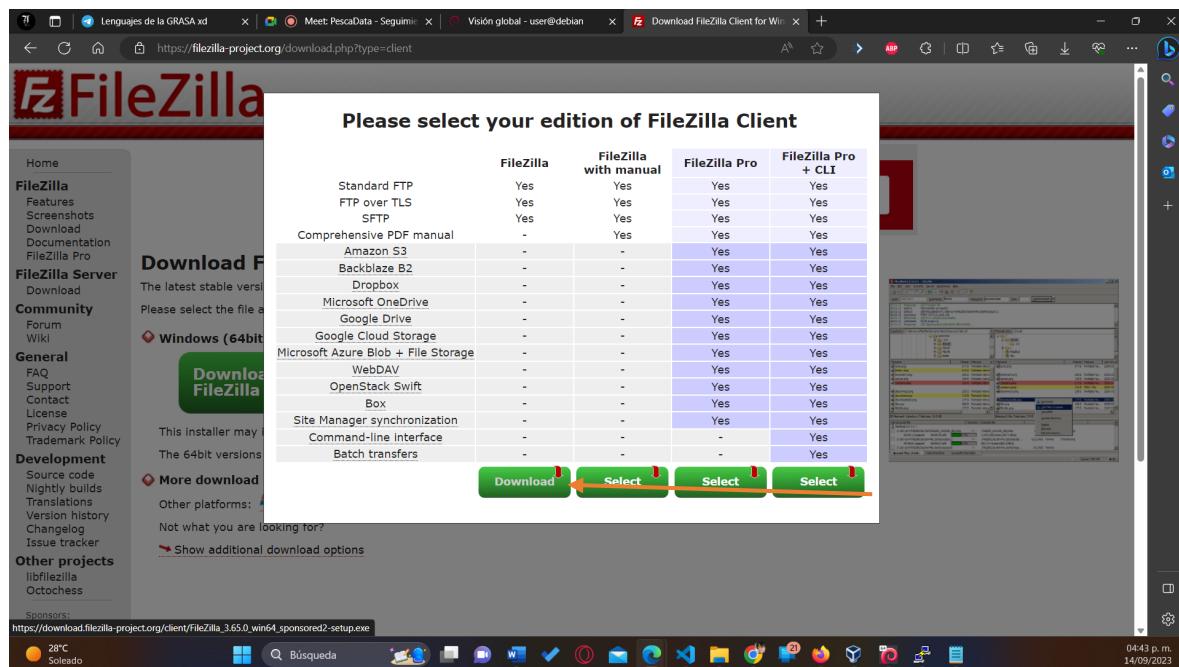
Para la instalación de Filezilla se ejecuta:

apt update

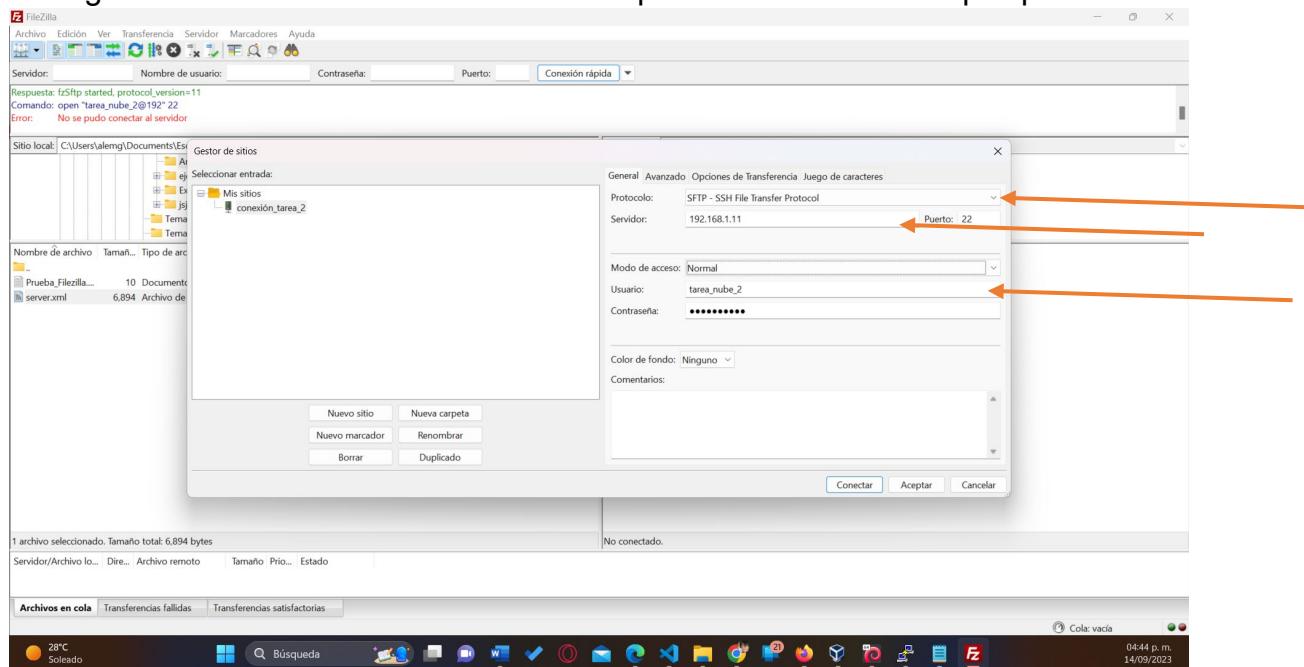
apt install filezilla

Y en el host de Windows se descarga el .exe desde el sitio oficial

<https://filezilla-project.org/>



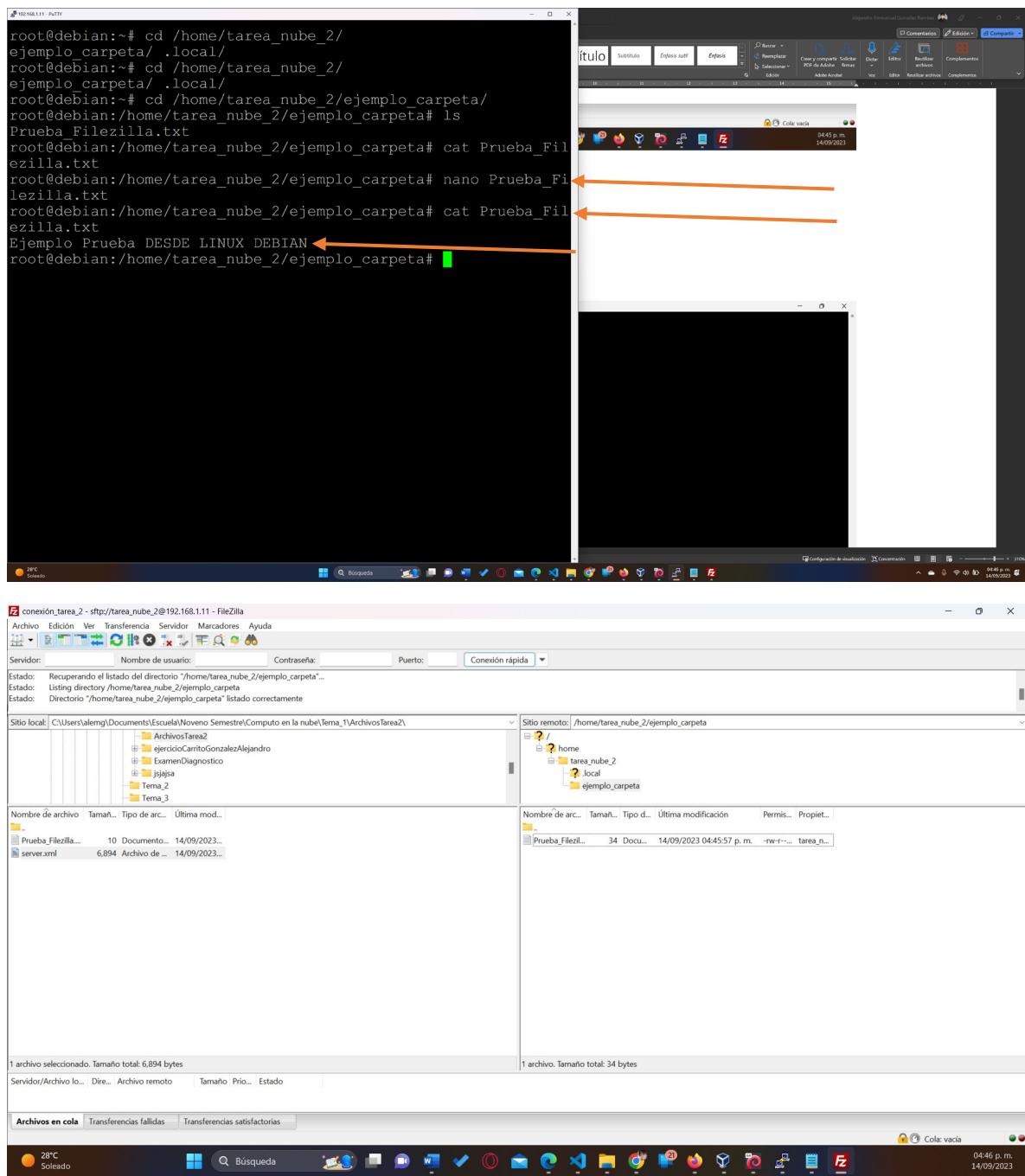
Configuramos el sitio con SFTP que es ssh + ftp puerto 22



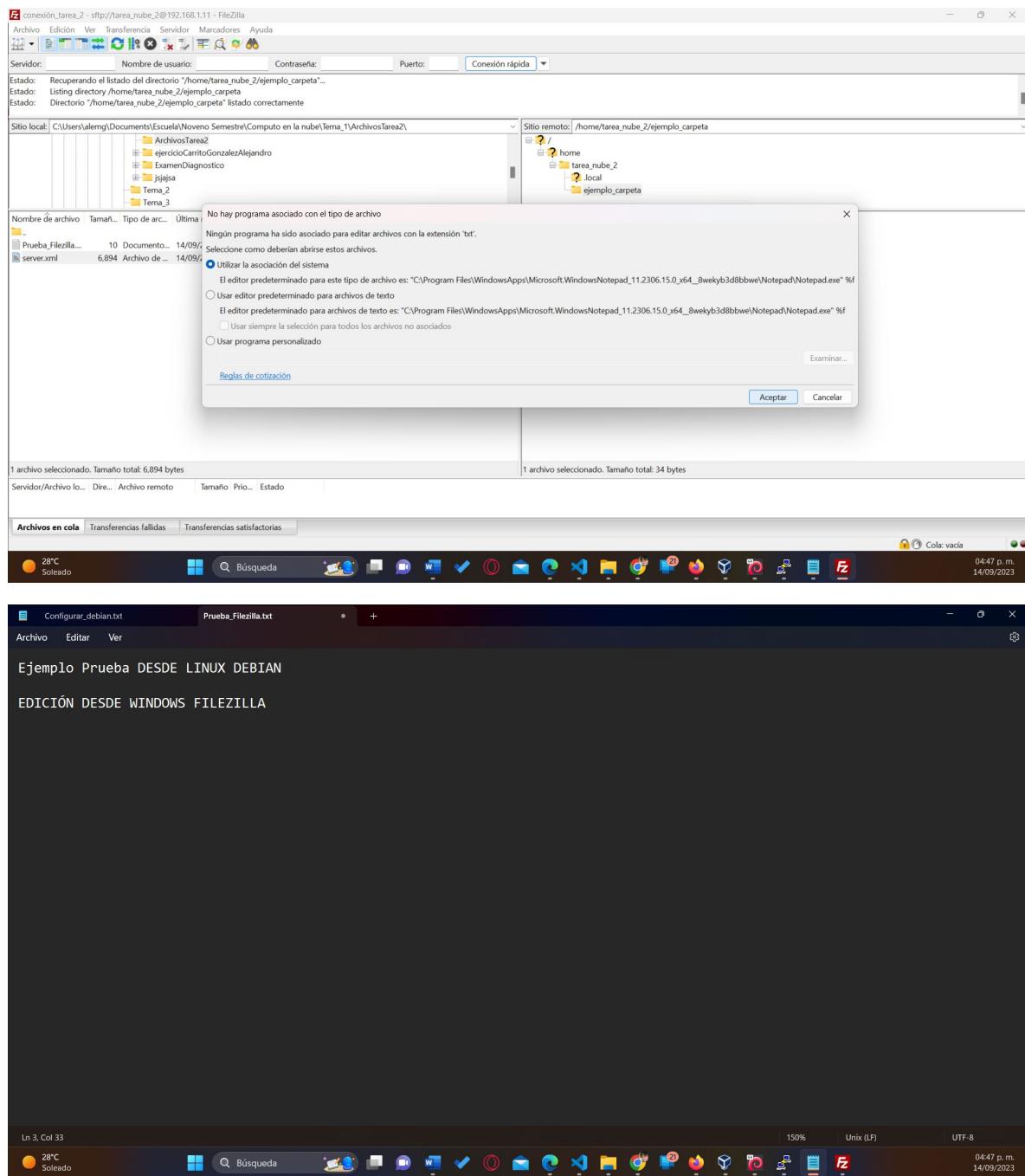
Antes de probar la conexión, se crea un archivo de prueba para ver si aparece con:

Nano Prueba_Filezilla.txt

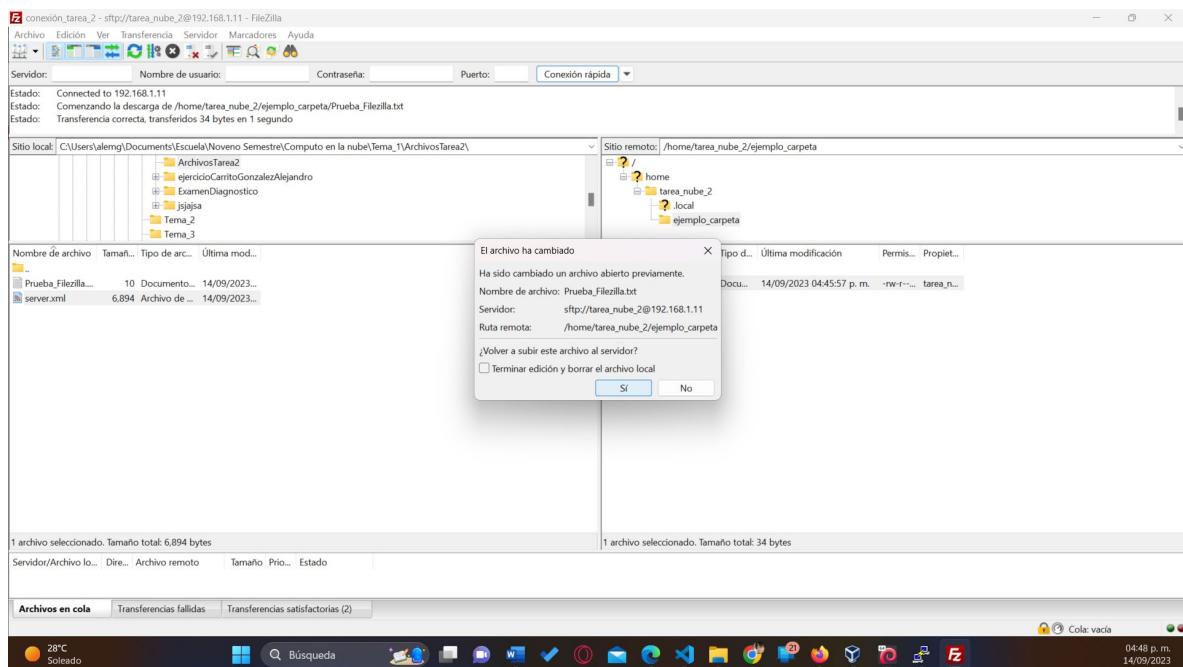
Y un texto de ejemplo



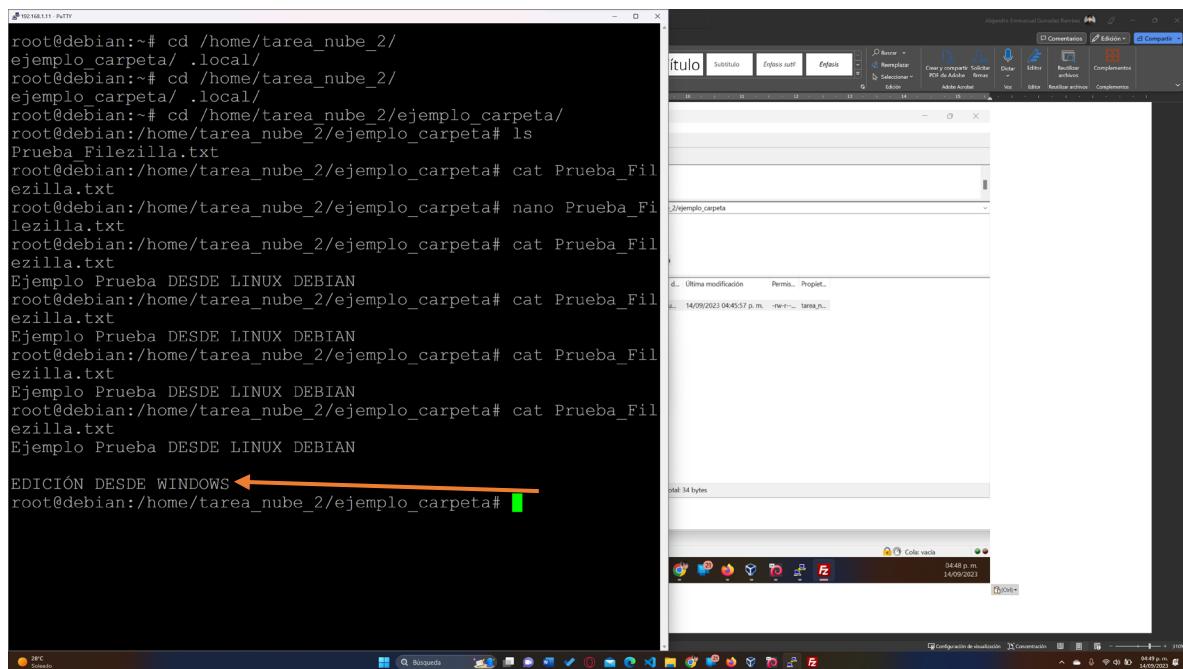
Ahora, una vez teniendo acceso al servidor es posible realizar acciones como edición:



Se modifica desde windows

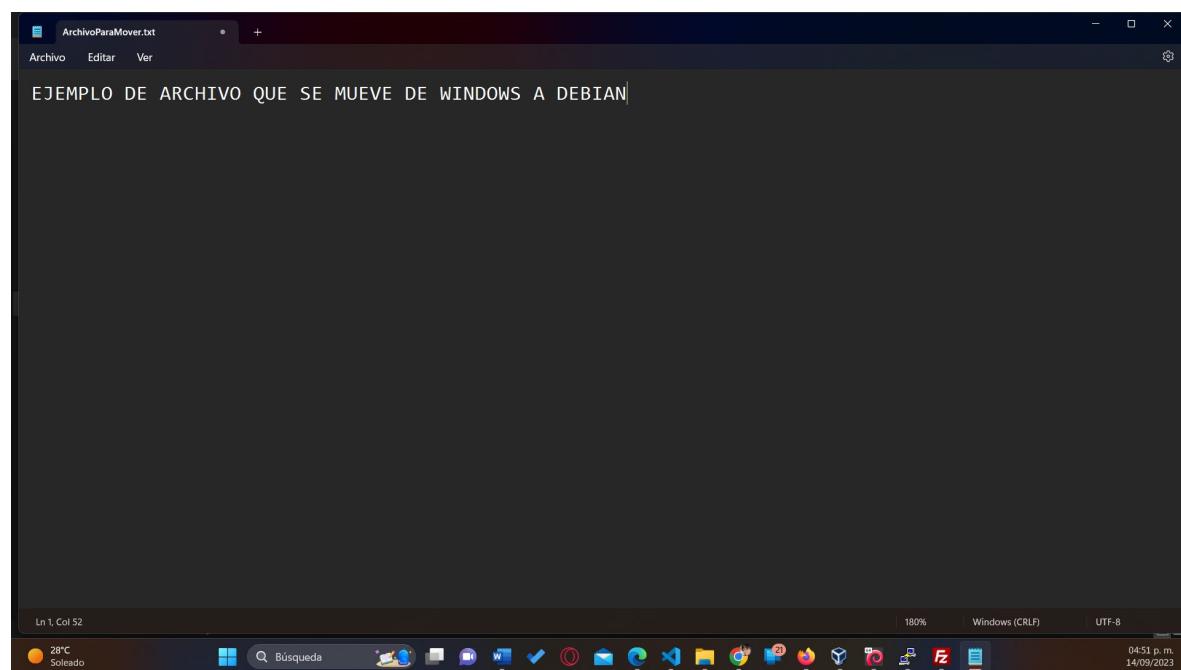
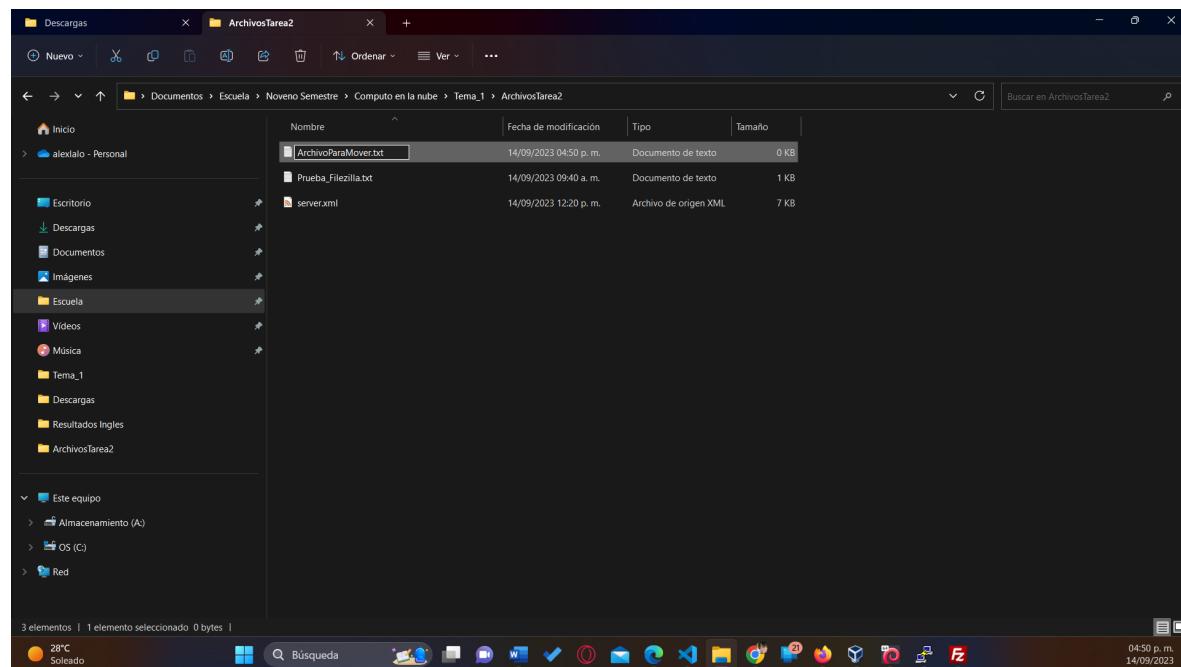


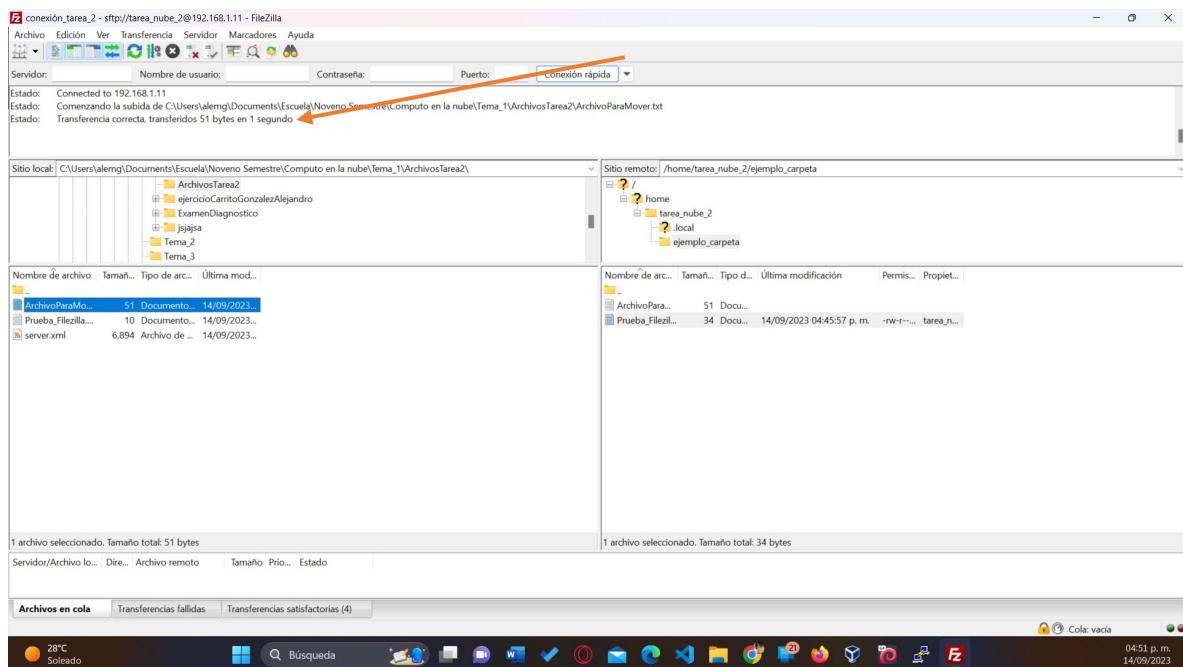
Se confirman y verifican los cambios



Y se pueden ver los cambios aplicados

Para mover archivos, se crea un nuevo archivo en windows



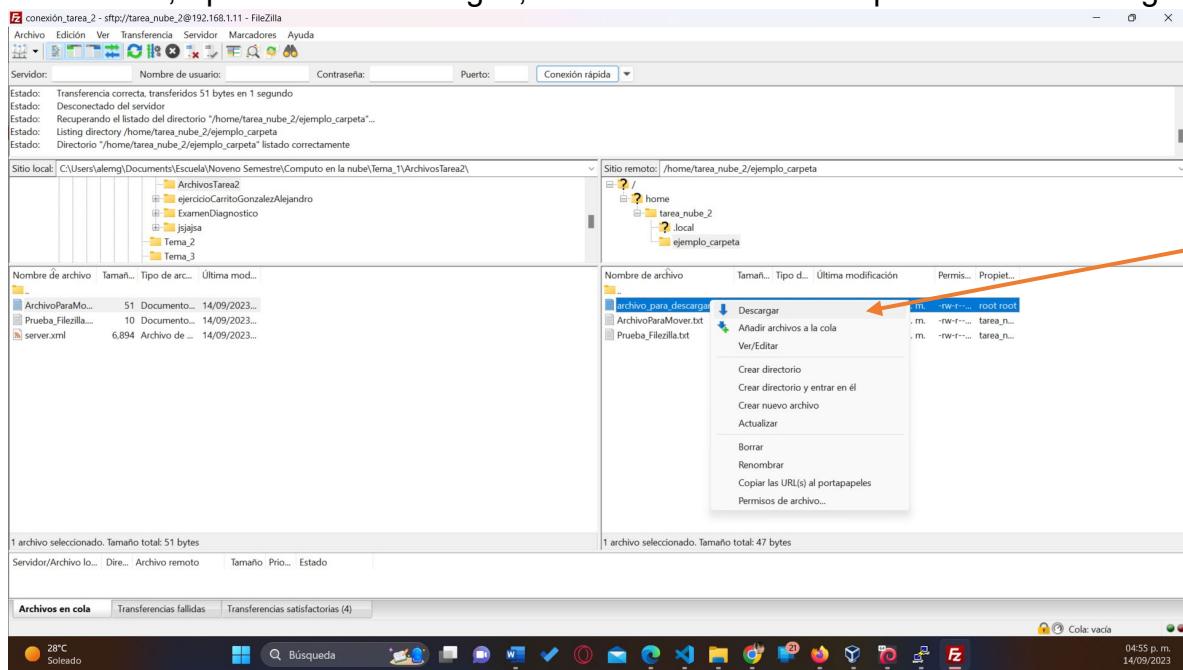


Se verifica que la transferencia fue exitosa y para probar la descarga, se crea un archivo desde debian

```
root@debian:~# cd /home/tarea_nube_2/
ejemplo_carpetas/.local/
root@debian:~# cd /home/tarea_nube_2/
ejemplo_carpetas/.local/
root@debian:~# cd /home/tarea_nube_2/ejemplo_carpetas/
root@debian:/home/tarea_nube_2/ejemplo_carpetas# ls
Prueba_Filezilla.txt
root@debian:/home/tarea_nube_2/ejemplo_carpetas# cat Prueba_Fil
ezilla.txt
root@debian:/home/tarea_nube_2/ejemplo_carpetas# nano Prueba_Fi
lezilla.txt
root@debian:/home/tarea_nube_2/ejemplo_carpetas# cat Prueba_Fil
ezilla.txt
Ejemplo Prueba DESDE LINUX DEBIAN
root@debian:/home/tarea_nube_2/ejemplo_carpetas# cat Prueba_Fil
ezilla.txt
Ejemplo Prueba DESDE LINUX DEBIAN
root@debian:/home/tarea_nube_2/ejemplo_carpetas# cat Prueba_Fil
ezilla.txt
Ejemplo Prueba DESDE LINUX DEBIAN
root@debian:/home/tarea_nube_2/ejemplo_carpetas# cat Prueba_Fil
ezilla.txt
Ejemplo Prueba DESDE LINUX DEBIAN
root@debian:/home/tarea_nube_2/ejemplo_carpetas# cat Prueba_Fil
ezilla.txt
Ejemplo Prueba DESDE LINUX DEBIAN

EDICIÓN DESDE WINDOWS
root@debian:/home/tarea_nube_2/ejemplo_carpetas# nano archivo_p
ara_descargar.txt
root@debian:/home/tarea_nube_2/ejemplo_carpetas# cat archivo_pa
ra_descargar.txt
Archivo para descargar en windows desde debian
root@debian:/home/tarea_nube_2/ejemplo_carpetas# cat archivo_para_descargar.txt
Archivo para descargar en windows desde debian
root@debian:/home/tarea_nube_2/ejemplo_carpetas#
```

Finalmente, para las descargas, se selecciona la opción de descargar



Y se puede apreciar que fue exitoso

