MEMORIA FINAL DE PROYECTO

IMPLANTAR SERVIDOR VPN CON 'WIREGUARD' EN UN ENTORNO EMPRESARIAL.

CICLO FORMATIVO DE GRADO SUPERIOR:

ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS EN RED

AUTOR:

MIGUEL ÁNGEL RIVERA ROLDÁN

TUTORA:

TERESA GONZÁLEZ GONZÁLEZ

COORDINADORA:

MARÍA JOSÉ VILLAR RUIBAL

CURSO:

2019 – 2020

I.E.S CLARA DEL REY

ÍNDICE

1. INTRODUCCIÓN

Descripción:

Instalación y configuración de servidores en Linux para una empresa, con la implementación tanto en el servidor como en el cliente del protocolo VPN "WireGuard" en máquinas virtuales.

- La idea de este proyecto surge por las noticias de ciberseguridad que se han ido conociendo desde que comenzó la pandemia del Covid-19. Todo este tiempo y debido al estado en el que se encuentra el mundo, millones de personas se han visto obligadas a realizar telestudios o a teletrabajar con sus consecuentes riesgos, cuando antes no tenían necesidad de hacerlo .

Fuente: ViktorHanacek.cz

- Aunque mayoritariamente los ciberataques están automatizados y no siempre van dirigidos a una persona en concreto, existiendo miles de IPs maliciosas que buscan algún resquicio en la seguridad de nuestros sistemas y con ello obtener algún tipo de beneficio. Probablemente las empresas es donde mayor daño pueden hacer y mayor beneficio económico pueden sacar estos atacantes.

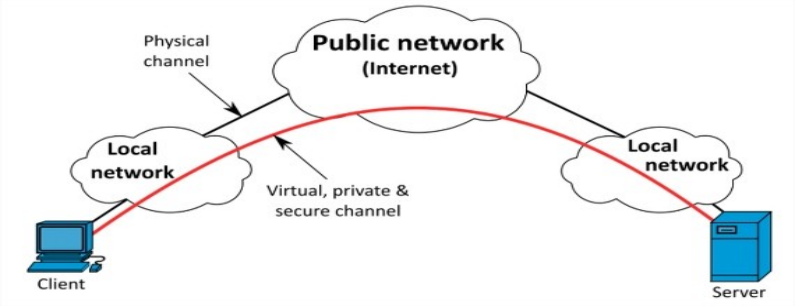
- Cada día que pasa es más importante la ciberseguridad en un entorno empresarial, evitando en la mayor medida la perdida o el robo de información relevante de la empresa y de los propios clientes.

2. ALCANCE DEL PROYECTO

Fuente : INCIBE

- En el proyecto se simulará el entorno de una empresa (PYME) con los diferentes servicios básicos necesarios con servidores implantados en Linux, a los cuales se añadirá un servidor con WireGuard y un cliente se conectará desde casa.

- Una de las formas más efectivas de evitar la perdida o el robo de información aunque no la única es utilizando una **VPN[1] (Virtual Private Network)**, que es una conexión encriptada a través de Internet desde un dispositivo a una red. La conexión cifrada ayuda a garantizar que los datos confidenciales se transmitan de forma segura. Evita que personas no autorizadas interpreten nuestra información y permite al usuario realizar el trabajo de forma remota. La tecnología VPN es ampliamente utilizada en entornos corporativos.



Fuente : INCIBE

1 https://www.cisco.com/c/en/us/products/security/vpn-endpoint-security-clients/what-is-vpn.html

2.1 OBJETIVO

- Con este proyecto se pretende demostrar la funcionalidad, versatilidad y posibilidades de WireGuard en un entorno empresarial. Así como afianzar algunos de los conocimientos obtenidos durante el ciclo de ASIR.

3. ANÁLISIS DE SOLUCIÓN ESCOGIDA

Fuente : WireGuard

3.1 ¿QUÉ ES WIREGUARD?

- De todos los protocolos VPN que existen se utilizará **“ WireGuard[2] “** por ser de código abierto, por su utilización en todos los sistemas operativos basados en Unix (Android, iOS, macOS, Linux) de forma nativa, ademas de otras características que hacen a este protocolo muy interesante. Se trata de una aplicación de software libre y un protocolo de comunicación que implementa técnicas de red privada virtual (VPN) para crear conexiones seguras punto a punto en configuraciones enrutadas o puenteadas, con técnicas de encriptación modernas como son : **Noise_protocol_framework, Curve25519, ChaCha20, Poly1305, BLAKE2, SipHas24, HKDF**

- *Jason A. Donenfeld* el fundador de **Edge Security**, creó este código. Se dio cuenta de que los mismos métodos utilizados para infiltrarse en una red eran los que la protegían. WireGuard contiene muchas características ocultas ya integradas y proporciona un túnel VPN fiable que supera las tecnologías obsoletas que usamos. Admite la capa 3 para IPv4 en IPv6 y puede encapsular IPv4 en IPv6 y viceversa. **WireGuard parece ser el futuro de los protocolos VPN.**

2 https://www.wireguard.com/

- Este protocolo que lleva desde el año 2016 tratándose de forma experimental por los desarrolladores, ha llegado para quedarse. Prueba de ello es que el responsable de la pila de red de Linux, David Miller aceptó la inclusión de este protocolo VPN en el nuevo kernel 5.6 en Diciembre de 2019 y en Enero de 2020 Linux Torvalds lo añadió al árbol de dicho kernel.

- Se dice que es el protocolo más fácil de usar, seguro y efectivo disponible en la actualidad. Este protocolo ofrece un salto de rendimiento superior al que ofrece otros protocolos como OpenVPN, debido a que se ejecuta como un módulo dentro del kernel y no como ocurre con otros protocolos que lo hace como una aplicación. Esto hace que los servicios criptográficos y sus procesos de cifrado y descifrado se ejecuten en WireGuard mucho más rápido.

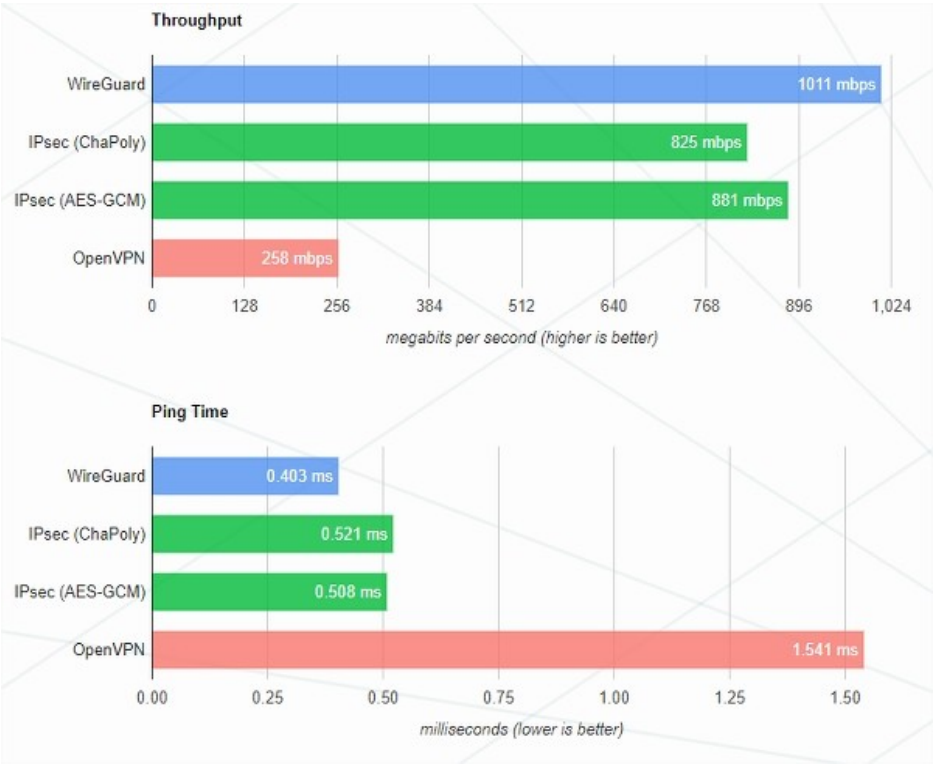


Figura 1: Comparativa protocolos
Fuente : https://www.wireguard.com/performance/

3.2 PLANIFICACIÓN DE TAREAS

Proyecto de Implementación de servidor VPN con Wireguard			
Lista de tareas	**Fecha Inicio**	**Duración**	**Fecha Final**
Analisis e Investigación	27/04/2020	12	09/05/2020
Topología de Red	07/05/2020	12	19/05/2020
Creación de Máquinas Virtuales	20/05/2020	8	28/05/2020
Configuración de Máquinas Virtuales	24/05/2020	15	08/06/2020
Pruebas de conexión	31/05/2020	9	09/06/2020
Realización de Memoria	04/05/2020	37	10/06/2020

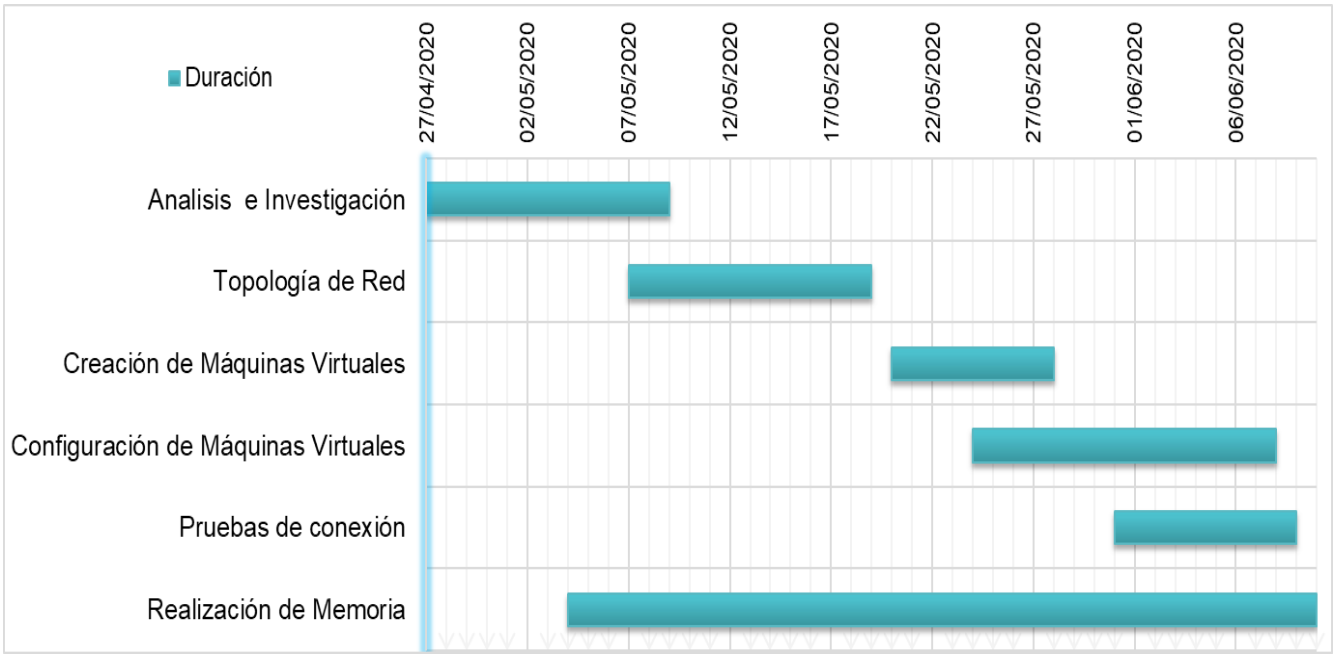


Figura 2: Diagrama de Gannt

Fuente : Propia

4. TOPOLOGÍA DE RED UTILIZADA

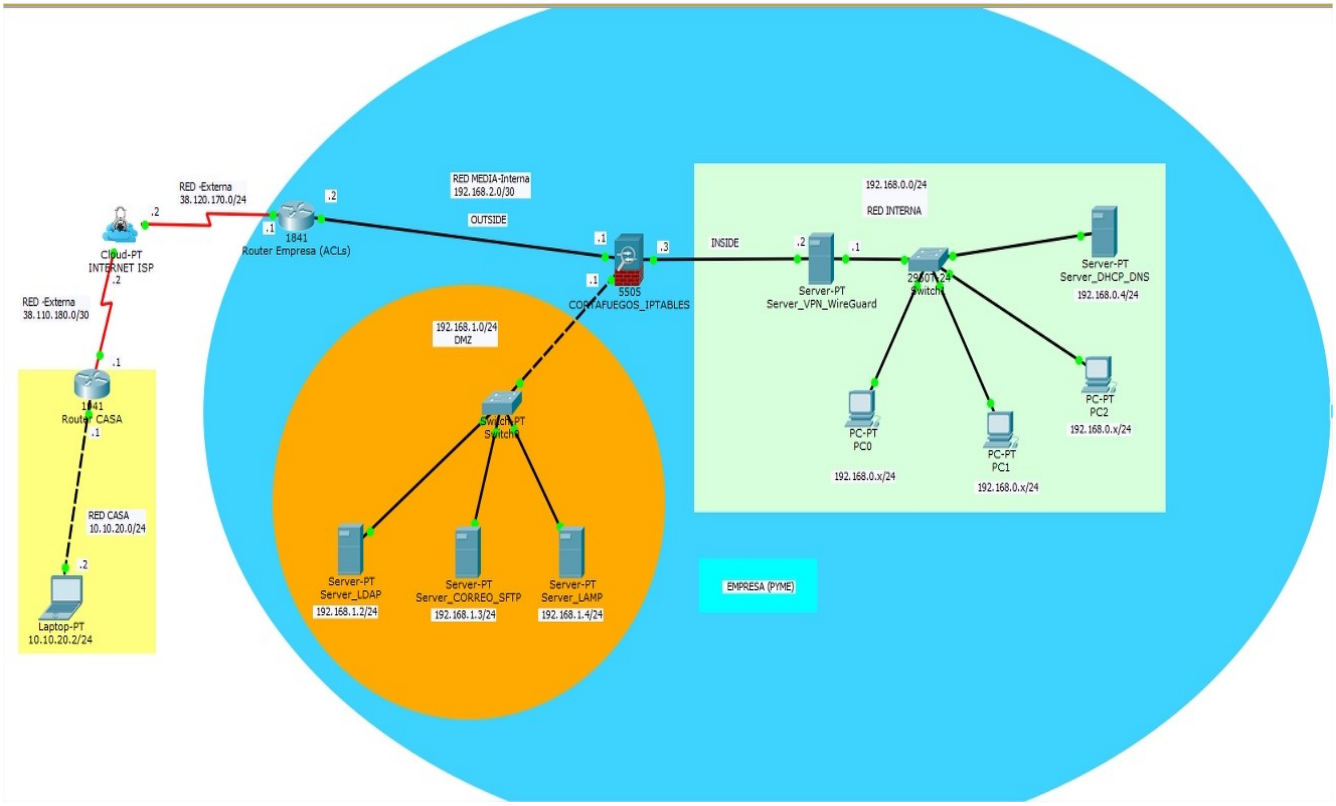


Figura 3: Topología de red (PYME)
Fuente : Propia

- Para realizar la topología de red se ha utilizado el software de Cisco “ Packet Tracer “.
- El diseño cuenta con cuatro (4) zonas :
  (a) “ Red Interna “ donde se encontraran los empleados, un servidor DHCP-DNS y el servidor VPN con WireGuard.
  (b) “ Zona DMZ “ donde se encontraran los diferentes servidores utilizados por la empresa (OpenLDAP, Correo, SFTP y LAMP).
  (c) “ Red Media interna “ donde el router de la empresa se conectará con el cortafuegos.
  (d) “ Red Casa “ desde donde se conectará el empleado.

5. IMPLEMENTACIÓN DEL DISEÑO

- Para realizar la creación y configuración del diseño se usarán recursos de diversas fuentes y los conocimientos adquiridos en los diferentes módulos del ciclo de ASIR:
  - Implantación y Administración de sistemas operativos.
  - Administración y Servicios de redes.
  - Seguridad y alta disponibilidad.
  - Administración de sistemas gestores de bases de datos.
  - Implantación de aplicaciones web.
- Se creará un laboratorio virtual para la implantación y configuración de los servidores en máquinas virtuales con prueba del servicio VPN WireGuard con el software de Oracle “ VM VirtualBox “.

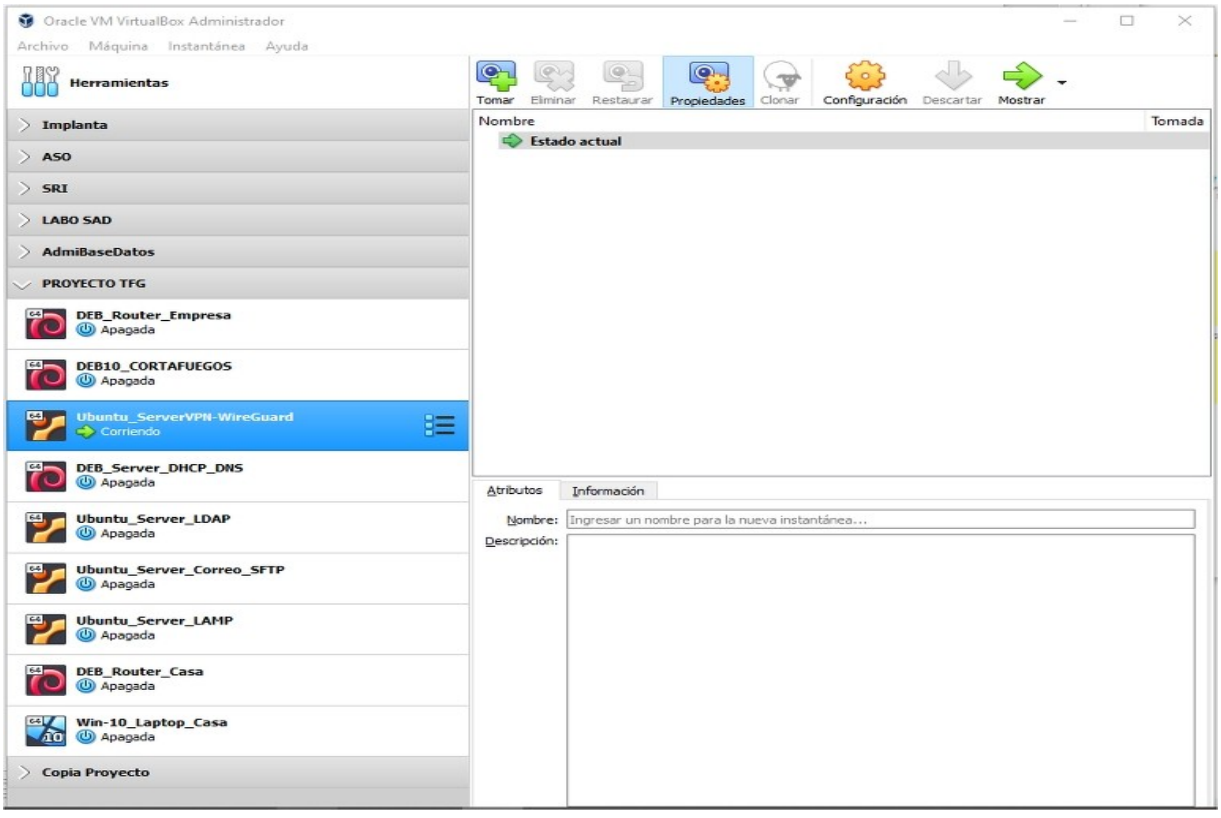


Figura 4: Máquinas creadas en VirtualBox

Fuente : Propia

5.1 CREACIÓN DE MÁQUINAS VIRTUALES

1. Router Empresa con 2 interfaces de red
   a) Adaptador puente (Red Externa)
   b) Red Media-Interna (192.168.2.2/30)
2. Cortafuegos_iptables con 3 interfaces de red
   a) Red Media-Interna (192.168.2.1/30)
   b) Red DMZ (192.168.1.1/24)
   c) Red Interna (192.168.0.3/24)
3. Server_DHCP_DNS con 1 interfaz de red
   a) Red Interna (192.168.0.4/24)
4. Server_LDAP_DNS maestro con 1 interfaz de red
   a) Red DMZ (192.168.1.2/24)
5. Server_Correo_SFTP con 1 interfaz de red
   a) Red DMZ (192.168.1.3/24)
6. Server_LAMP con 1 interfaz de red
   a) Red DMZ (192.168.1.4/24)
7. Router Casa con 2 interfaces de red
   a) Adaptador puente (Red Externa)
   b) Red Casa (10.10.20.1/24)
8. Server_VPN_WireGuard con 2 interfaces de red
   a) Red Interna (192.168.0.1/24)
   b) Red Interna (192.168.0.2/24)
9. Laptop_Casa con 1 interfaz de red
   a) Red Casa (10.10.20.2/24)

5.1.1 CONFIGURACIÓN DE MÁQUINAS VIRTUALES

Todos los comandos se ejecutan en un terminal como " root "

1- Router Empresa

- Las configuraciones de red se realizan con un editor de texto en el fichero: */etc/network/interfaces*
- Este fichero quedará configurado con dos interfaces de red de la siguiente forma:

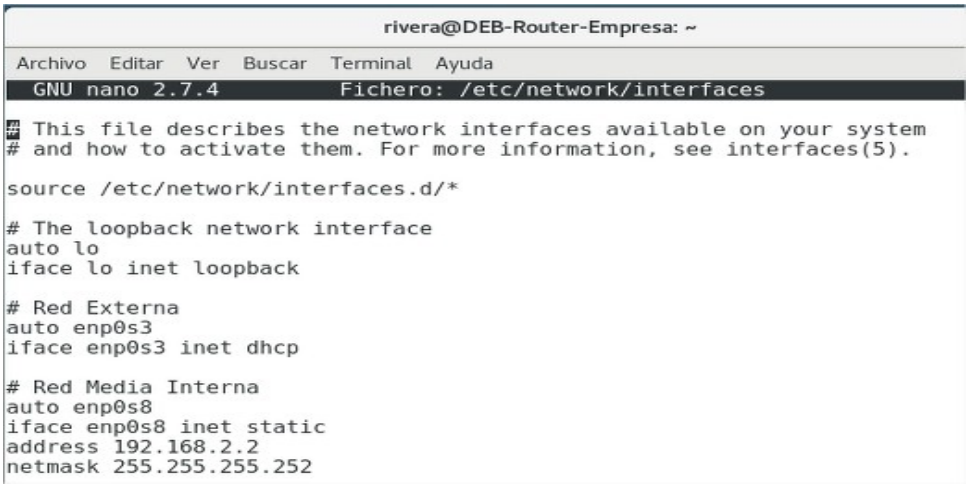


Figura 5: Fichero /etc/network/interfaces

Fuente : Propia

- *S*e activará de forma permanente el reenvío de datagramas en el fichero */etc/sysctl.conf* eliminando (#) de la línea *net.ipv4.ip_forward = 1* y se ejecuta el comando *# sysctl -p* para hacer efectivos los cambios.
- Se utilizará el siguiente comando para hacer NAT *:*

  # iptables –t nat –A POSTROUTING –o enp0s3 –j MASQUERADE
- Para hacerlo permanente primero se crea el fichero *nat* con el comando :

  # iptables-save > /etc/nat
- *Después se añadirá al final del fichero /etc/network/interfaces la orden :*

  pre-up iptables-restore < /etc/nat

- Los *routers* ofrecen un medio de filtrado del tráfico de la red mediante las ACLs. Para realizar este filtrado en un router Cisco o en el Packet tracer se pueden aplicar una serie de ACEs y de esta forma conseguir que el router haga NAT-P, también se les podrá asignar IPs publicas a los host de la red interna.

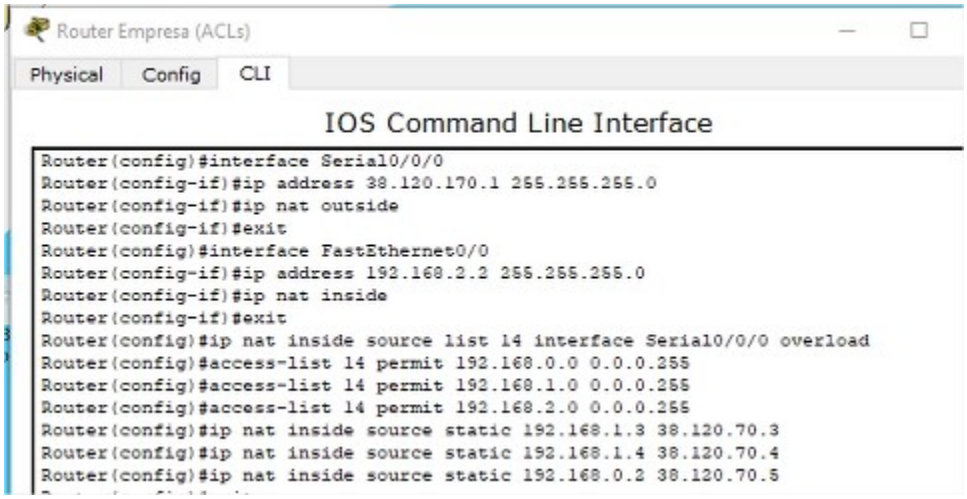


Figura 6: ACLs Packet tracer

Fuente : Propia

- Ahora se pasará a configurar las DNS en el fichero /etc/resolv.conf donde se pondrá el dominio, las dns de Google, las de Opendns[3] y nuestro servidor DNS maestro.

Figura 7: Fichero resolv.conf

Fuente : Propia

3 https://www.opendns.com/

- Por último se reiniciará el servicio con el comando :

  # /etc/init.d/networking restart

```
root@DEB-Router-Empresa:/home/rivera# /etc/init.d/networking restart
[ ok ] Restarting networking (via systemctl): networking.service.
root@DEB-Router-Empresa:/home/rivera#
```

Figura 8: Reinicio de servicio

Fuente : Propia

2- Cortafuegos_iptables

- En esta máquina se configuran 3 interfaces de red de forma estática y activamos el reenvío de datagramas como en la máquina anterior.

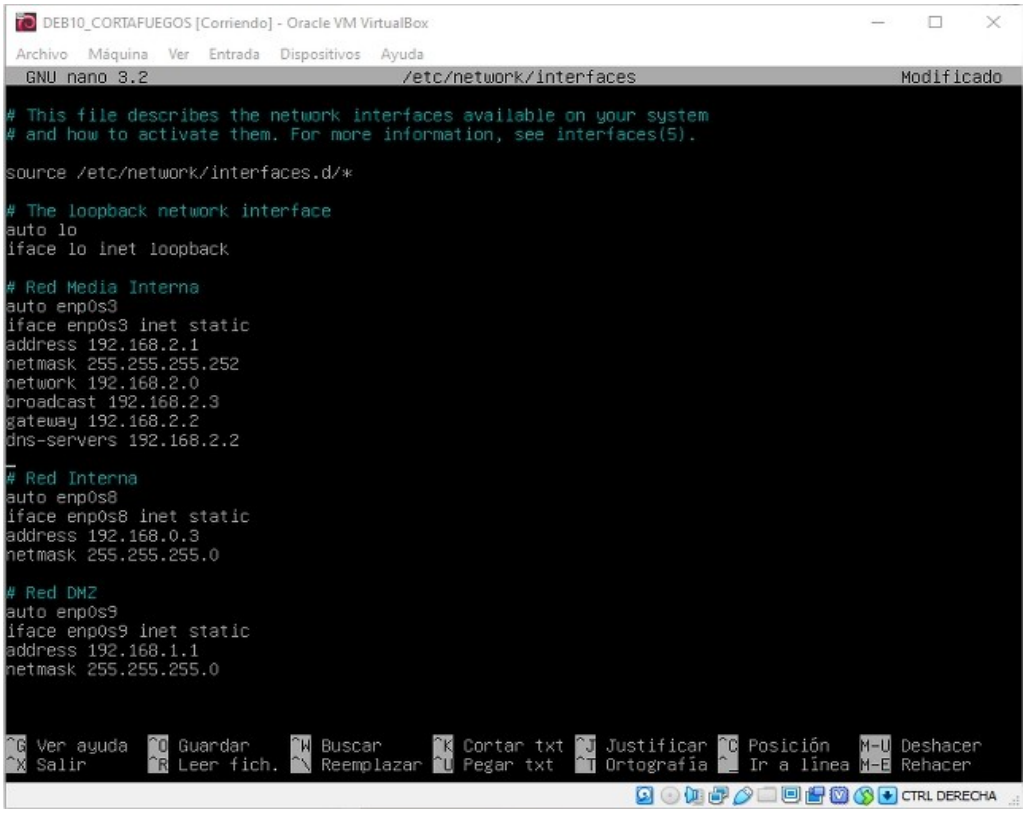


Figura 9: Fichero /etc/network/interfaces

Fuente : Propia

- Para realizar el cortafuegos con iptables, se podrá implementar Systemd[4] como servicio del sistema por su facilidad de uso. Se realizaran unos scripts para su correcta configuración y su posterior funcionamiento.
- Se crea el script " cortafuegos-on "

```
root@DEB10CORTAFUEGOS:~# nano /usr/lib/systemd/scripts/cortafuegos-on_
```

```
#! /bin/sh

# Borrar Reglas
iptables -F
iptables -X
iptables -Z
iptables -t nat -F
# Políticas por defecto
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT
# Enmascaramos la Red Interna y la DMZ
iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE
# Aceptamos al localhost
/sbin/iptables -A INPUT -i lo -j ACCEPT
# Aceptamos a la Red Interna acceder al Cortafuegos por ssh
iptables -A INPUT -s 192.168.0.0/24 -p tcp --dport 22 -j ACCEPT
iptables -A OUTPUT -d 192.168.0.0/24 -p tcp --sport 22 -j ACCEPT
## Permitimos las conexiones Establecidas y Relacionadas hacia la Red Interna desde la Red Externa
## y desde la DMZ
iptables -A FORWARD -d 192.168.0.0/24 -p tcp -m state --state ESTABLISHED,RELATED -j ACCEPT
## Aceptamos el tráfico desde la Red Interna hacia la Red Externa y la DMZ
iptables -A FORWARD -s 192.168.0.0/24 -p tcp -j ACCEPT
iptables -A FORWARD -s 192.168.0.0/24 -p udp -j ACCEPT
iptables -A FORWARD -s 192.168.0.0/24 -p icmp -j ACCEPT
# Permitimos las respuestas PING hacia la Red Interna
iptables -A FORWARD -d 192.168.0.0/24 -p icmp --icmp-type echo-reply -j ACCEPT
## Aceptamos el tráfico hacia el Servidor WEB
iptables -A FORWARD -d 192.168.1.4/24 -p tcp --dport 80 -j ACCEPT
iptables -A FORWARD -d 192.168.1.4/24 -p tcp --dport 443 -j ACCEPT
## Permitimos las conexiones Establecidas desde el servidor WEB
iptables -A FORWARD -d 192.168.1.4/24 -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
iptables -A FORWARD -d 192.168.1.4/24 -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT
## Aceptamos el tráfico hacia el Servidor (Correo,ftp-sftp)
iptables -A FORWARD -d 192.168.1.3/24 -p tcp --dport 20,21,22,25 -j ACCEPT
## Permitimos las conexiones Establecidas y Relacionadas para el Servidor (correo,ftp-sftp)
iptables -A FORWARD -d 192.168.1.3/24 -p tcp --sport 20,21,22,25 -m state --state ESTABLISHED,RELATED -j ACCEPT
## Aceptamos las conexiones Establecidas desde la Red Externa hacia la Red Interna por udp
iptables -A FORWARD -d 192.168.0.0/24 -i enp0s3 -p udp -m state --state ESTABLISHED -j ACCEPT
## Ahora hacemos DNAT para las peticiones recibidas  en los Servidores
iptables -t nat -A PREROUTING -i enp0s9 -p tcp --dport 80 -j DNAT --to-destination 192.168.1.4/24:80
iptables -t nat -A PREROUTING -i enp0s9 -p tcp --dport 443 -j DNAT --to-destination 192.168.1.4/24:443
iptables -t nat -A PREROUTING -i enp0s9 -p tcp --dport 22,25 -j DNAT --to-destination 192.168.1.3/24:22,25
## Ver RESULTADO iptables
iptables -L -n
iptables -t nat -L -n -v

exit 0
```

Figura 10: Fichero script " cortafuegos-on "
Fuente : Propia

4 https://wiki.archlinux.org/index.php/Systemd_(Espa%C3%B1ol)

- Se crea el script " cortafuegos-off "

```
root@DEB10CORTAFUEGOS:~# nano /usr/lib/systemd/scripts/cortafuegos-off_
```

```
  GNU nano 3.2                                  cortafuegos-off

#! /bin/sh

iptables -F
iptables -X
iptables -t nat -F

# Ver Resultado
iptables -L -n -v
iptables -t nat -L -n -v

exit 0
```

Figura 11: Fichero script " cortafuegos-off "
Fuente : Propia

- Ahora se les dará permisos de ejecución a los dos scripts.

```
root@DEB10CORTAFUEGOS:~# chmod +x /usr/lib/systemd/scripts/cortafuegos-off
root@DEB10CORTAFUEGOS:~# chmod +x /usr/lib/systemd/scripts/cortafuegos-on
```

- El siguiente paso es crear el fichero del servicio " cortafuegos-on.service " con el comando : # *usr*/lib/systemd/system/cortafuegos-on.service

```
  GNU nano 3.2              /usr/lib/systemd/system/cortafuegos-on.service

[Unit]
Description=Packet Filtering Framework

[Service]
Type=oneshot
ExecStart=/usr/lib/systemd/scripts/cortafuegos-on
ExecReload=/usr/lib/systemd/scripts/cortafuegos-on
ExecStop=/usr/lib/systemd/scripts/cortafuegos-off
RemainAfterExit=yes

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

Figura 12: Fichero del servicio
Fuente : Propia

- Después estando en el directorio */usr*/lib/systemd/system/, solo faltaría habilitar el servicio e iniciar el cortafuegos, para ello se utilizaran los siguientes comandos :

  # systemctl enable cortafuegos-on.service (Habilitar)

  # systemctl “start/stop” cortafuegos-on.service (Iniciar/Parar)

  # systemctl status cortafuegos-on.service (Ver estado)

```
root@DEB10CORTAFUEGOS:/usr/lib/systemd/system# systemctl enable cortafuegos-on.service
Created symlink /etc/systemd/system/multi-user.target.wants/cortafuegos-on.service → /lib/systemd/sy
stem/cortafuegos-on.service.
root@DEB10CORTAFUEGOS:/usr/lib/systemd/system# systemctl start cortafuegos-on.service
root@DEB10CORTAFUEGOS:/usr/lib/systemd/system# systemctl status cortafuegos-on.service
● cortafuegos-on.service - Packet Filtering Framework
   Loaded: loaded (/lib/systemd/system/cortafuegos-on.service; enabled; vendor preset: enabled)
   Active: active (exited) since Wed 2020-06-03 02:32:53 CEST; 14s ago
  Process: 613 ExecStart=/usr/lib/systemd/scripts/cortafuegos-on (code=exited, status=0/SUCCESS)
 Main PID: 613 (code=exited, status=0/SUCCESS)
```

Figura 13: Habilitar servicio

Fuente : Propia

3 - Server_DHCP_DNS

- En esta máquina virtual situada en la red interna se implantará un servidor DHCP y DNS caché sencillo con DNSMASQ[5]. La única interfaz de red se configura con una IP estática desde el fichero /*etc/network/*interfaces.

```
# Red Interna
auto enp0s3
iface enp0s3 inet static
address 192.168.0.4
netmask 255.255.255.0
gateway 192.168.0.1
dns-nameservers 192.168.0.4 192.168.1.2
dns-search Rivera.org
```

Figura 14: Fichero /etc/network/interfaces

Fuente : Propia

5 https://www.gamificafp.com/pluginfile.php/8111/mod_resource/content/0/2-Servidor-DHCP-y-DNS.pdf

- De esta forma se tendrá un servidor caché en la red interna que resolverá las IPs, los nombres y adicionalmente asignará las IPs de manera dinámica dentro del rango establecido.

- Se ejecutaran en un terminal para su instalación y arranque los comandos :

  # apt-get install dnsmasq

  # /etc/init.d/dnsmasq restart

- Si pretendemos el arranque automático del servicio al iniciar la máquina, podemos utilizar la herramienta gráfica “ boot-up manager[6] “ .

- Para su instalación se utilizará el comando que aparece en la siguiente imagen :

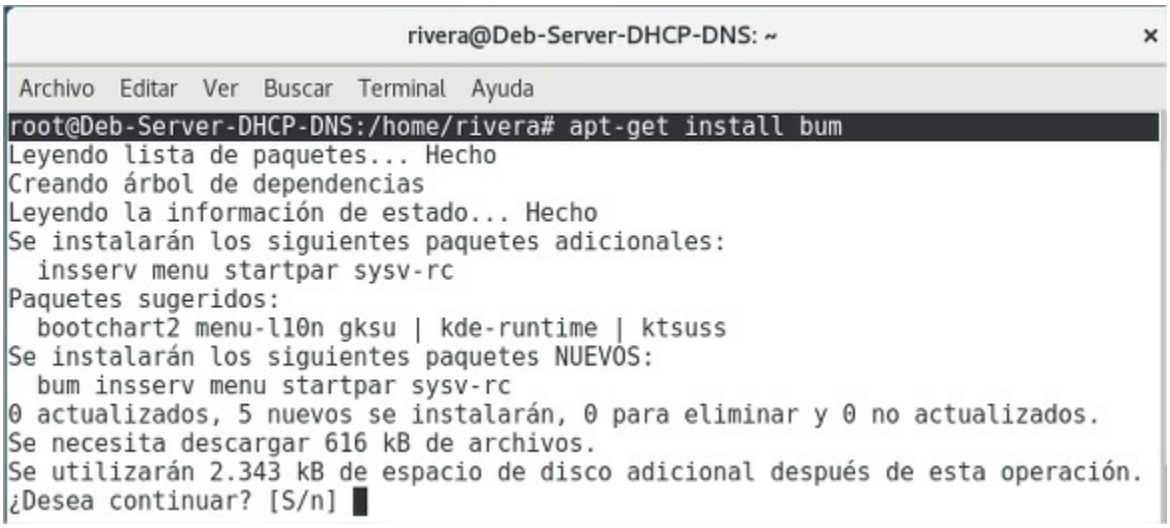


Figura 15: Comando instalar bum

Fuente : Propia

6 http://somebooks.es/administrar-servicios-demonios-de-ubuntu-con-boot-up-manager/

- Con esta herramienta la gestión de los servicios se vería de la siguiente forma :

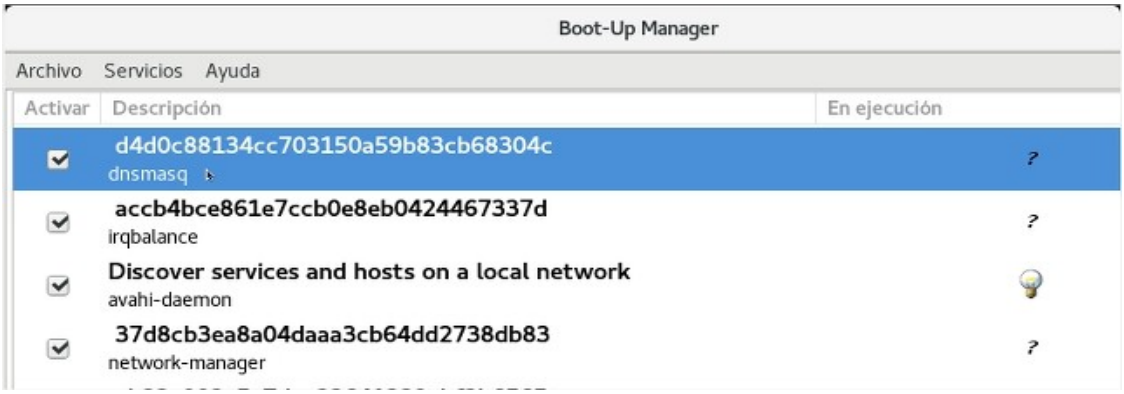


Figura 16: Herramienta gráfica " boot-up manager

Fuente : Propia

- Ahora en el fichero de configuración /*etc*/dnsmasq.conf se añadirá el rango y el tiempo de cesión de IPs :

Figura 17: Fichero /etc/dnsmasq.conf

Fuente : Propia

- Como la puerta de enlace de los clientes será otra máquina distinta a este servidor se le indicará de la siguiente forma, terminando así la configuración de este servidor :

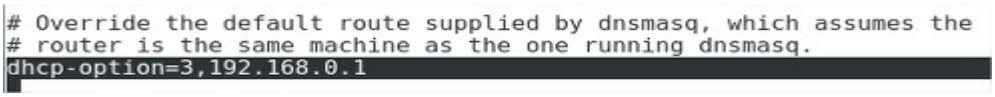


Figura 18: Fichero /etc/dnsmasq.conf

Fuente : Propia

4 – Server_LDAP_DNS maestro

- LDAP, es un protocolo que ofrece el acceso a un servicio de directorio implementado sobre un entorno de red, con el objeto de acceder a una determinada información.

- En este servidor también instalaremos el servidor DNS Bind como servidor maestro, que será capaz de resolver peticiones de nombres internos para nuestra red de empresa.

A) DNS

- El paquete bind9 será instalado para este fin, para ello se hace uso de los siguientes comandos: # sudo apt-get update y # sudo apt-get install bind9

- Después de la instalación, se modificará el fichero /etc/bind/named.conf.local y en este fichero se declaran las zonas asociadas al servidor de dominio en cuestión. Para ello, se utiliza el comando : # nano /etc/bind/named.conf.local

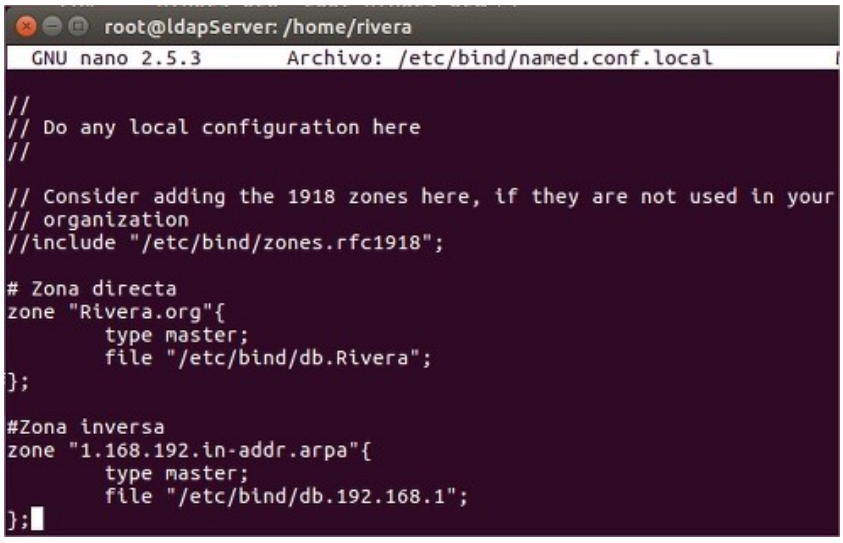


Figura 19: Fichero named.conf.local

Fuente : Propia

- A continuación se hará una copia de los ficheros creados con la instalación /etc/bind/db.local nombrándolo como db.Rivera y /etc/bind/db.127 como db.192.168.1. En estos ficheros se cambiará " localhost " por el nombre de dominio y 127.0.0.1 por la dirección IP del servidor, ademas se añadirán los servidores de correo, web y SFTP. La configuración de estos ficheros quedará como en las siguientes imágenes:

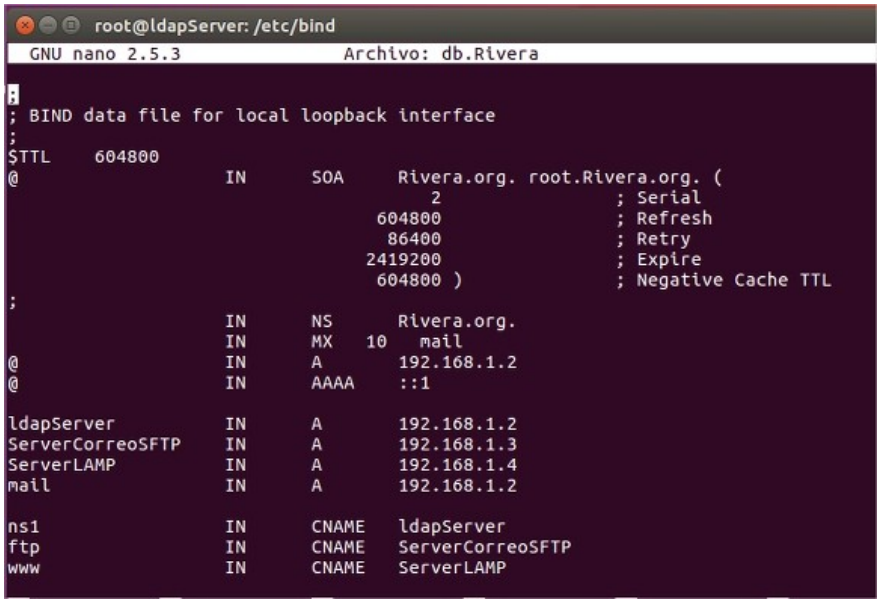


Figura 20: Fichero zona directa

Fuente : Propia

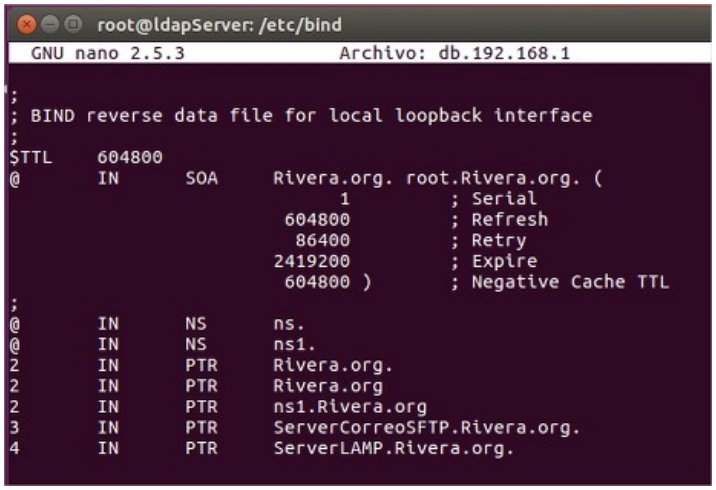


Figura 21: Fichero zona inversa

Fuente : Propia

- Se puede comprobar la correcta sintaxis de las zonas con el comando :

  # named-checkzone Rivera.org /etc/bind/db.Rivera

  # named-checkzone Rivera.org /etc/bind/db.192.168.1

```
root@ldapServer:/etc/bind# named-checkzone Rivera.org /etc/bind/db.Rivera
zone Rivera.org/IN: loaded serial 2
OK
root@ldapServer:/etc/bind# named-checkzone Rivera.org /etc/bind/db.192.168.1
zone Rivera.org/IN: loaded serial 1
OK
root@ldapServer:/etc/bind#
```

Figura 22: Comprobar zonas

Fuente : Propia

- El fichero de configuración también se puede comprobar con :

  # named-checkconf -z /etc/bind/named.conf

```
root@ldapServer:/etc/bind# named-checkconf -z /etc/bind/named.conf
zone Rivera.org/IN: loaded serial 2
zone 1.168.192.in-addr.arpa/IN: loaded serial 1
zone localhost/IN: loaded serial 2
zone 127.in-addr.arpa/IN: loaded serial 1
zone 0.in-addr.arpa/IN: loaded serial 1
zone 255.in-addr.arpa/IN: loaded serial 1
root@ldapServer:/etc/bind#
```

Figura 23: Comprobar named.conf

Fuente : Propia

- **Importante :**

  - A todas las maquinas de la empresa se les indicará el servidor DNS y el dominio.

  dns-nameservers 192.168.1.2

  name-search Rivera.org

B) LDAP

- La finalidad de nuestro servidor LDAP es que sirva de almacén de usuarios y grupos para autentificar sistemas linux y servicios como ftp y web y se deberá crear una estructura que parta de la base de nuestro directorio, para almacenar dicha información.

- Se instalará el paquete SLAPD y un paquete adicional que contiene las utilidades de administración de LDAP, ldap-utils. Se pedirá la contraseña para la administración de LADP. *# sudo apt-get install slapd ldap-utils*

- A continuación, se instalará la librería NSS para LDAP. Esta librería ofrece una interfaz para acceder y configurar distintas bases de datos utilizadas para almacenar cuentas de usuario. # sudo apt install libnss-ldap

- Nos aparecerá el menú de configuración de ldap-auth-config, primero nos solicita la dirección URI del servidor LDAP. Es importante dejarla como en la siguiente imagen y poner la IP de nuestro servidor:

Figura 24: URI servidor

Fuente : Propia

- Lo siguiente será indicarle el dominio y la cuenta de administrador.

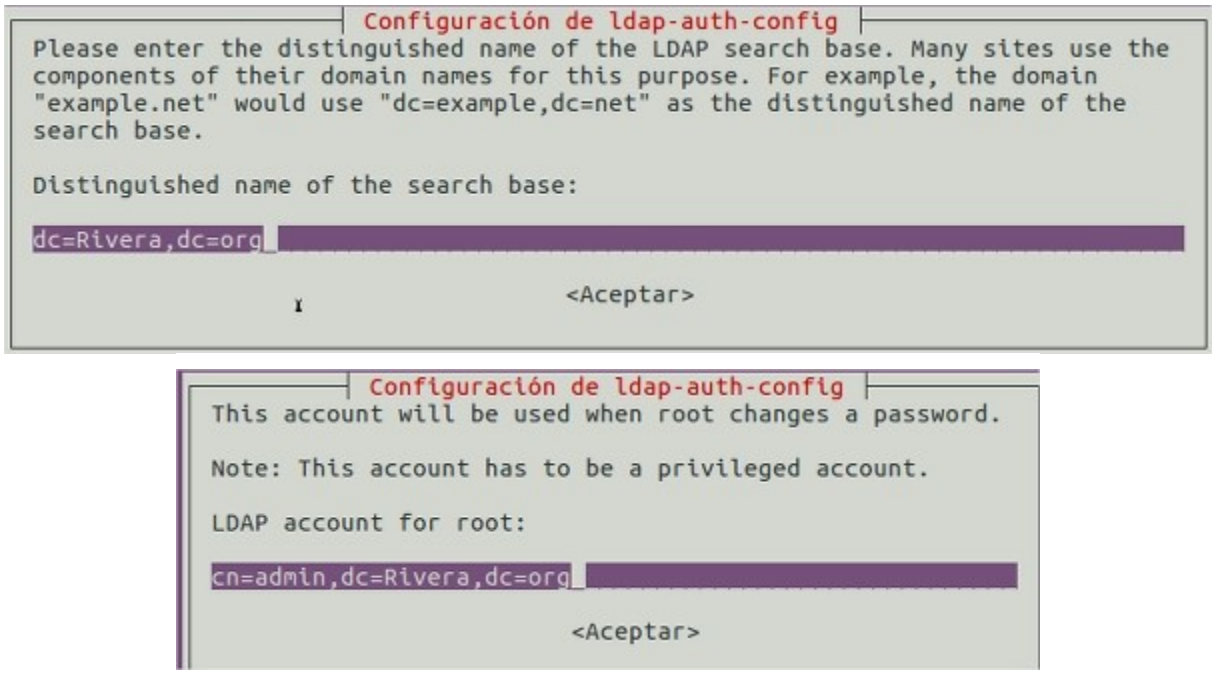


Figura 25: Dominio y cuenta admin
Fuente : Propia

- Indicaremos la versión de protocolo " 3 ", utilidades PAM " si " y después " no " para la necesidad de identificarse al realizar consultas.

- Se usará un script que nos ayuda a modificar los archivos de configuración de PAM y NSS. Para usar este script, ejecutamos el siguiente comando:

```
# sudo auth-client-config -t nss -p lac_ldap
```

- Seguidamente, se deberá actualizar la configuración de las políticas de autenticación predeterminadas de PAM, esto permitirá autenticar usuarios. Para ello, se usará el siguiente comando:

```
# sudo pam-auth-update
```

- En el asistente de configuración se selccionarán los módulos para autenticar usuarios y una vez haya terminado se confirmará en el fichero /*etc/*ldap.conf que los siguientes datos son los correctos.

```
Host 192.168.1.2
base dc=Rivera,dc=org
uri ldap://192.168.1.2/
rootbinddn cn=admin,dc=Rivera,dc=org
ldap_version 3
bind_policy soft
```

- Con esto se habrá terminado la configuración del servidor LDAP. Ahora está listo para autenticar usuarios.

5 - Server_Correo_SFTP

A) SFTP

- Al igual que con los otros servidores, se tiene que configurar una IP estática desde el fichero /*etc/network/*interfaces.

```
# Red DMZ
auto enp0s3
iface enp0s3 inet static
address 192.168.1.3
netmask 255.255.255.0
gateway 192.168.1.1
```

Figura 26: Fichero /etc/network/interfaces

Fuente : Propia

- Para instalar el servicio SFTP[7] se utilizará el comando :

  # apt-get install openssh-server

- Una vez instalado se creará una carpeta en el servidor y se comprobará el estado de la conexión SSH como se puede apreciar en la siguiente imagen.

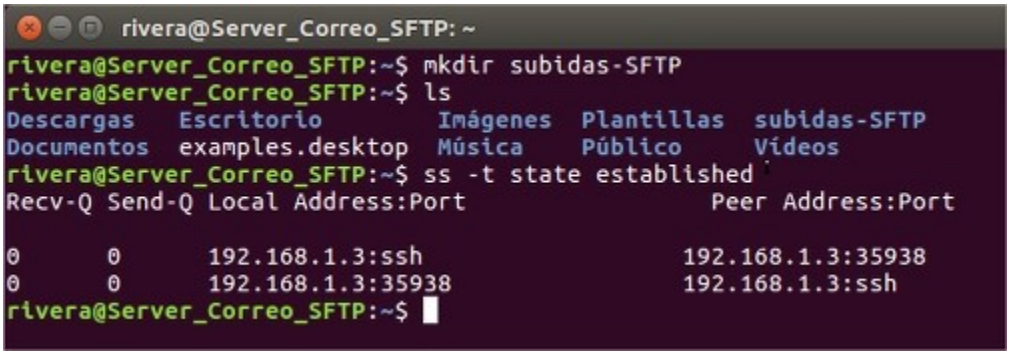


Figura 27: Creación de carpeta y estado de la conexión

Fuente : Propia

- Comprobamos en el archivo de la configuración del servidor (/etc/ssh/shhd_config) que la siguiente cláusula está descomentada:

  AuthorizedKeysFile .ssh/authorized_keys

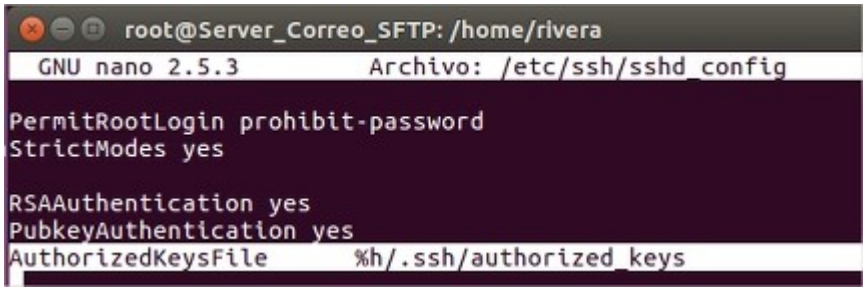


Figura 28: Autorizar claves

Fuente : Propia

7 https://www.ssh.com/ssh/sftp/

- Reiniciamos el servicio y se coprueba su estado con los siguientes comandos :

  # systemctl restart ssh.service

  # systemctl status ssh.service

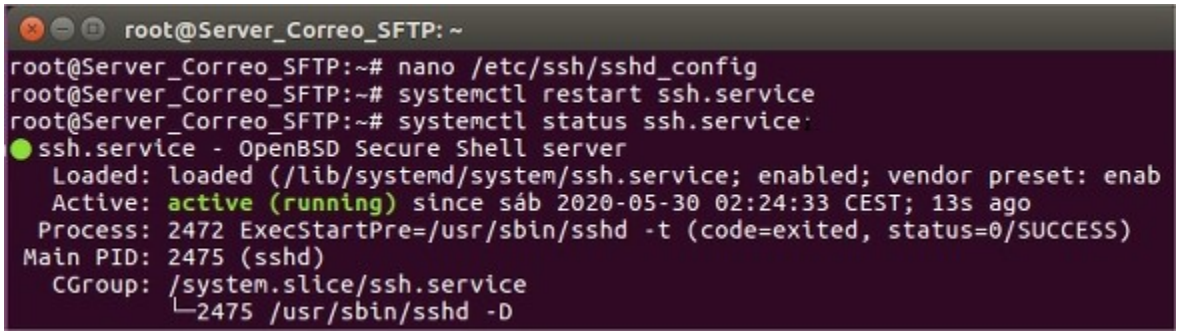


Figura 29: Estado del servicio ssh

Fuente : Propia

- Como se puede comprobar en la imagen anterior el servicio está activo y es el momento de generar las claves RSA[8] y copiar la clave pública en el directorio del usuario.

  - Comando para generar las claves : # ssh-keygen -t rsa

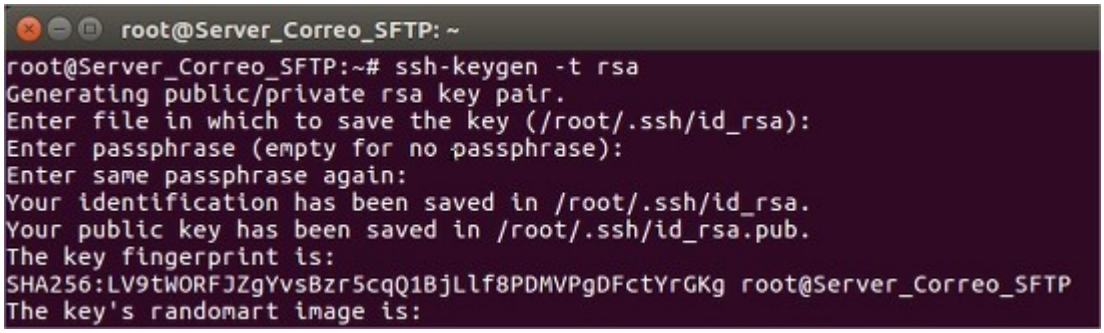


Figura 30: Generar claves ssh

Fuente : Propia

8 https://www.ecured.cu/RSA

- Comando para copiar clave publica en el directorio del usuario :

ssh-copy-id “usuario” @ ” IP servidor “

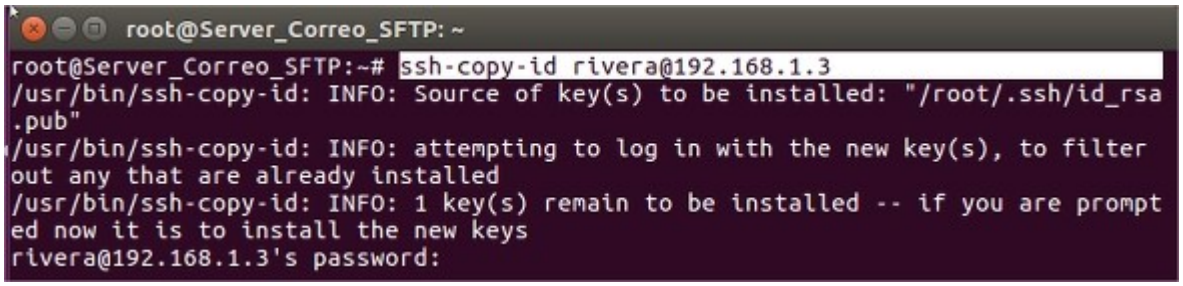


Figura 31: Copiar clave publica

Fuente : Propia

- Para terminar nos conectamos con el usuario al servidor SFTP

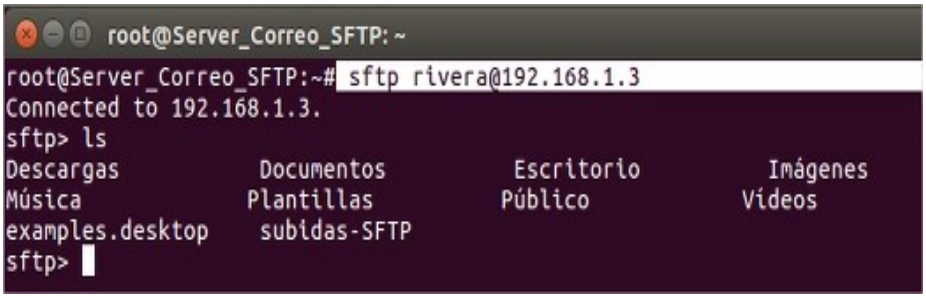


Figura 32: Conexión al servidor SFTP

Fuente : Propia

B) MailServer

- En esta misma máquina se implementará el servidor de correo, para ello se instalaran un agente de transferencia de correos (MTA) que en este caso será “Postfix[9]”, un agente de entrega de correo (MDA) como “Dovecot[10]” y un cliente de correo web (MUA) como es el caso de “SquirrelMail[11]”.

- Se empezará instalando y configurando Postfix, para ello se usará el comando :

  # apt-get install postfix

- Durante la instalación se solicitara la configuración básica, se debe elegir la opción “ Sitio de Internet ” como lo indica la imagen siguiente :

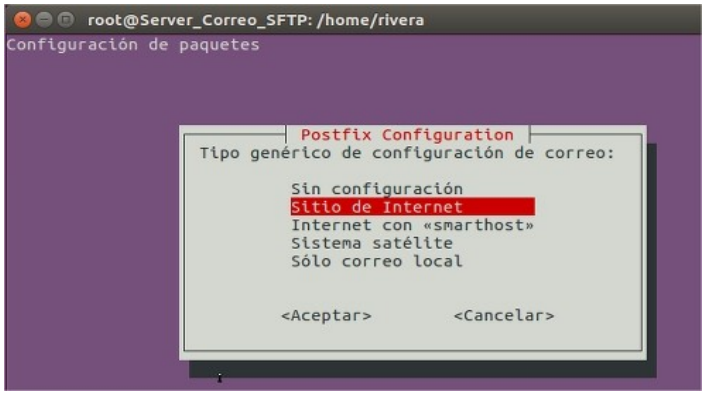


Figura 33: Conexión al servidor SFTP

Fuente : Propia

9 https://wiki.archlinux.org/index.php/Postfix_(Espa%C3%B1ol)

10 https://wiki.archlinux.org/index.php/Dovecot

11 https://wiki.archlinux.jp/index.php/Squirrelmail

- Ahora se le asigna el nombre del sistema de correos o dominio.

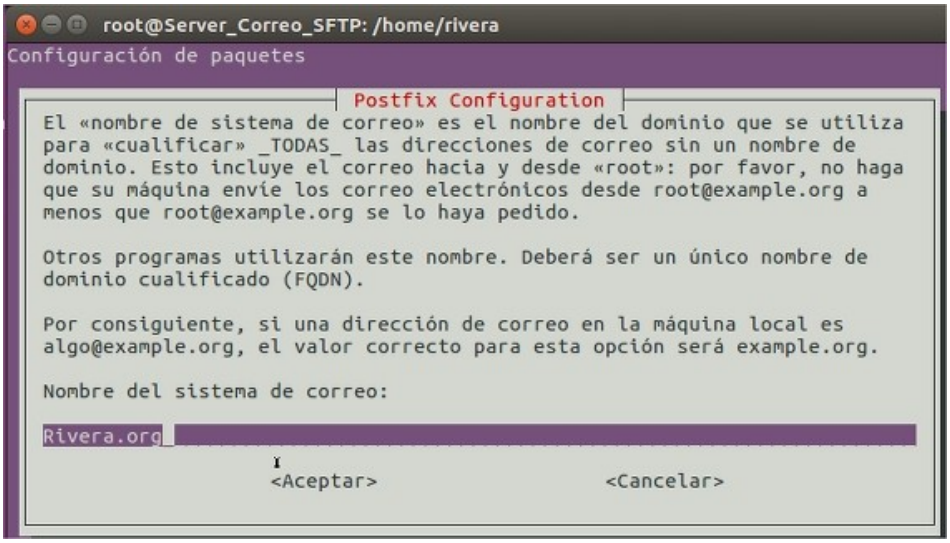


Figura 34: Conexión al servidor SFTP

Fuente : Propia

- El siguiente paso para poner el servidor de correo en marcha, es instalar Dovecot, que será el servidor de IMAP/POP3. Se ejecuta en terminal el comando y se deja la configuración por defecto :

  # apt-get install dovecot-core dovecot-imapd dovecot-pop3d

- A continuación se instalará el cliente de correo web SquirrelMail y para su correcto funcionamiento se deberá instalar también Apache2[12] para que funcione en el servidor . Los comandos serán :

  # apt-get install apache2

  # apt-get install squirrelmail

12 https://httpd.apache.org/

- La configuración de SquirrelMail es bastante sencilla y se abre ejecutando en un terminal lo siguiente:
  # squirrelmail-configure

Figura 35: Configuración SquirrelMail

Fuente : Propia

- Se ingresa D para establecer una configuración predeterminada para el servidor IMAP donde se elegirá dovecot, a continuación, se ingresa 2 para acceder a la configuración del servidor, luego 1 para configurar el dominio y S para guardar los cambios. La configuración se muestra en las siguientes imágenes :

Figura 36: Configuración Imap server

Fuente : Propia

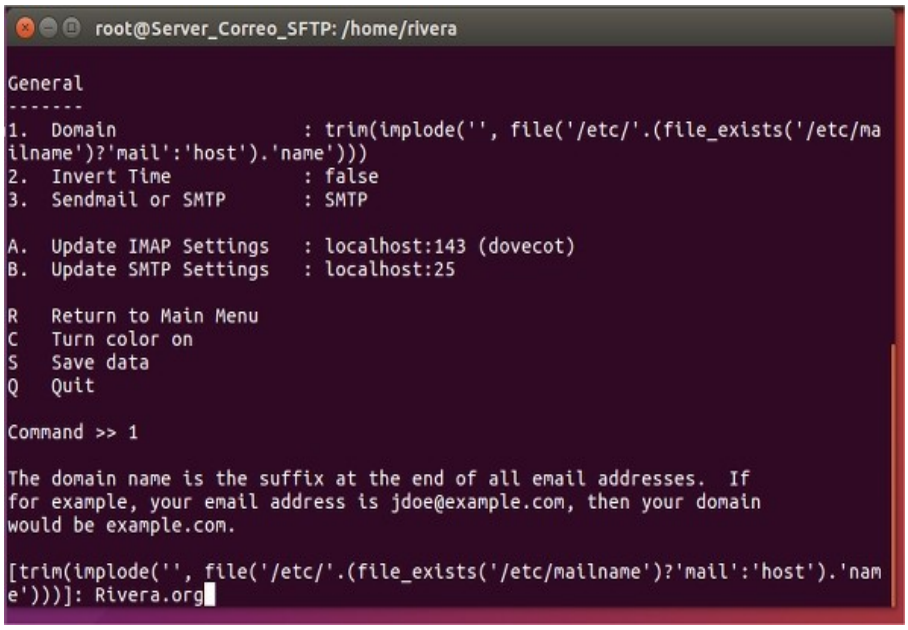


Figura 37: Configuración del dominio

Fuente : Propia

- Después de terminar la configuración, se creará un enlace simbólico en el directorio /*var/*www/html y a continuación se reinicia el servicio de apache para que la carpeta pueda ser vista desde un navegador.

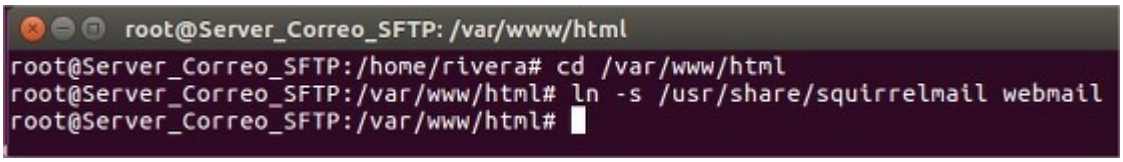


Figura 38: Enlace simbólico

Fuente : Propia

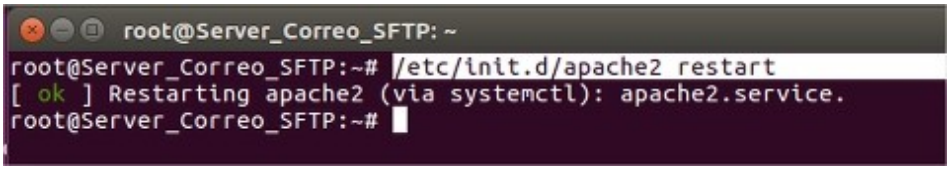


Figura 39: Reinicio Apache

Fuente : Propia

- Por último se comprueba en el navegador ingresando el dominio seguido de /webmail.

Figura 40: Cliente de correo web

Fuente : Propia

6 - Server_LAMP

- LAMP es una plataforma de código abierto para el desarrollo web que usa Linux como sistema operativo, Apache como servidor web, MySQL como base de datos relacional y PHP como lenguaje de scripts orientado a objetos.

- Contiene el software necesario para configurar sitios web o servidores dinámicos con esfuerzo reducido.

- Como se viene haciendo con los anteriores servidores, se configura la única interfaz de red con una IP estática desde el fichero : /*etc/network/*interfaces

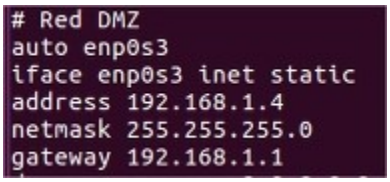


Figura 41: Fichero /etc/network/interfaces

Fuente : Propia

- Su instalación será bastante sencilla y con muy pocos comandos se tendrá el servidor operativo.

- Existen diferentes métodos para instalar LAMP en Ubuntu, pero el más eficiente es utilizando el comando : # apt-get install lamp-server^

- El uso del carácter ^ significa que lo que precede es un metapaquete[13]. Al instalar metapaquetes, también se instalarán otros paquetes. Se agregarán paquetes como apache2-utils, libaprutil1, libhttp-date-perl, php-mysql, php7.3-readline, ssl-cert y más, junto con Apache, MySQL y PHP.

13 https://www.linuxadictos.com/que-son-los-meta-paquetes-de-linux.html

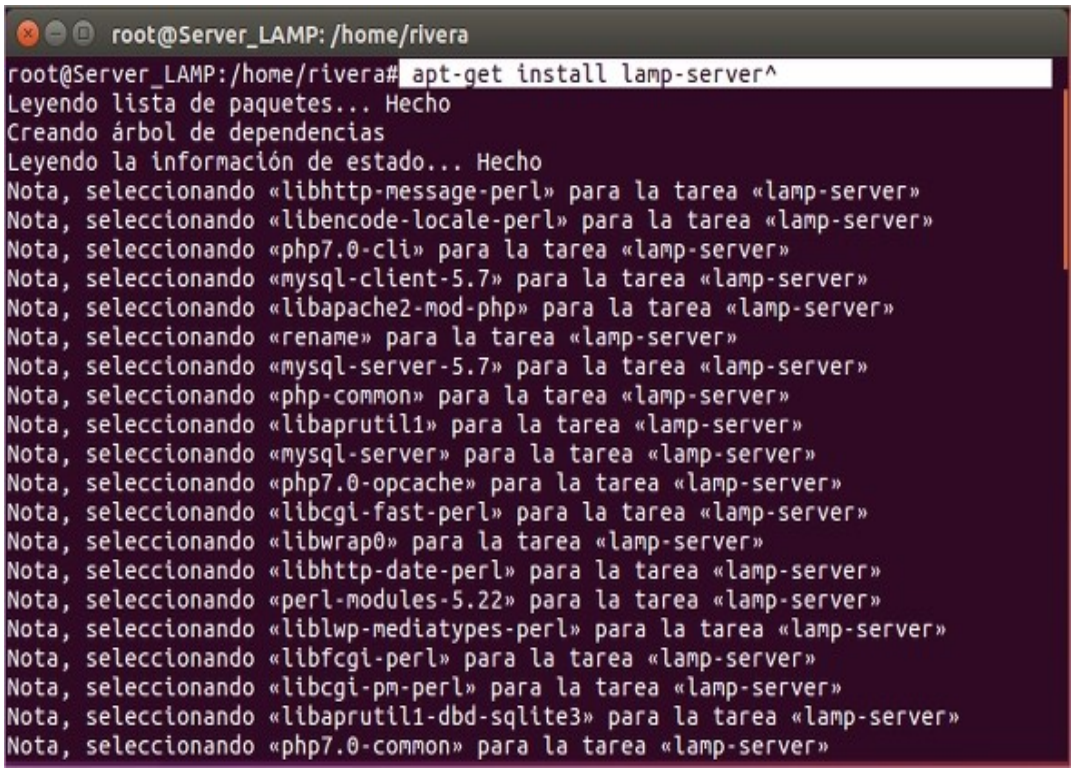


Figura 42: Instalación LAMP metapaquete

Fuente : Propia

- Lo único que hay que hacer después de que se instalen todos estos paquetes será securizar MySQL, para ello se utilizará el comando :

  # mysql_secure_installation

Figura 43: Securizar MySQL

Fuente : Propia

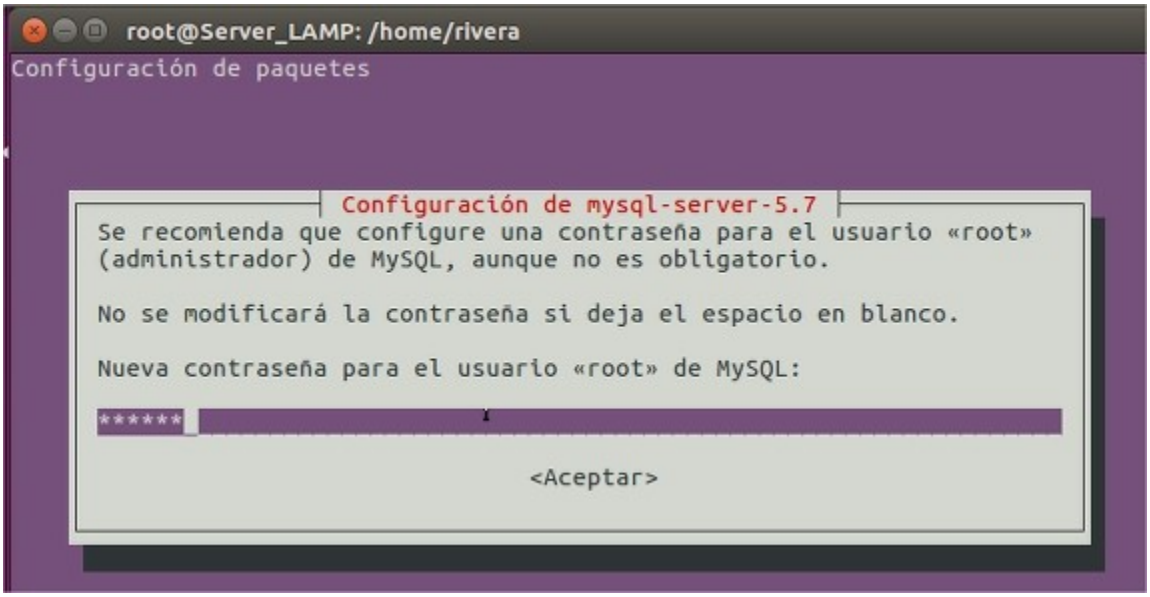


Figura 44: Contraseña MySQL

Fuente : Propia

- En el directorio /*var/www*/ es donde se guardará el proyecto web por defecto, para poder copiar los ficheros se creará un enlace simbólico a una carpeta en "home" , sin necesidad de editar permisos.

Figura 45: Enlace simbólico

Fuente : Propia

- El siguiente paso será instalar phpmyadmin para crear, editar, borrar o realizar consultas en nuestra base de datos. Se utilizará el comando :

```
# apt-get install phpmyadmin
```

- Se configura el servidor web

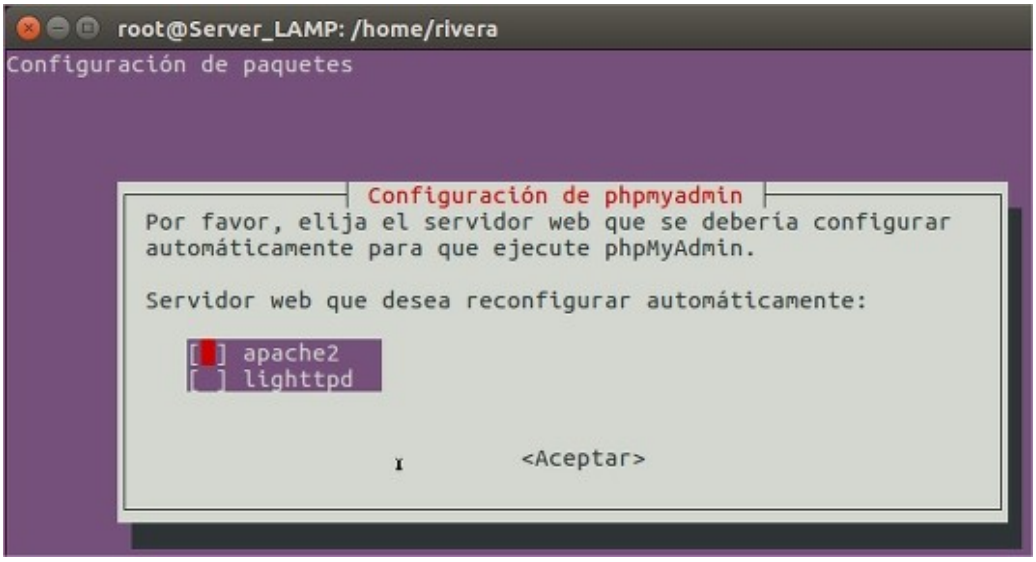


Figura 46: Servidor web

Fuente : Propia

- Después desde el navegador se escribe **http://localhost/phpmyadmin/** para tener acceso a phpmyadmin y se inicia sesión.

Figura 47: Inicio phpmyadmin

Fuente : Propia

- Una vez dentro ya se podrá gestionar la base de datos relacional.

Figura 48: Creación base de datos y tablas

Fuente : Propia

7 - Router Casa

- Al igual que con el router de empresa se crearan dos interfaces de red y se seguirán los mismos pasos, con una interfaz por DHCP que tendrá conexión con Internet haciendo NAT-P y otra estática que hará de puerta de enlace para el cliente.

```
# Red Externa
auto enp0s3
iface enp0s3 inet dhcp

# Red Casa
auto enp0s8
iface enp0s8 inet static
address 10.10.20.1
netmask 255.255.255.0
```

Figura 49: Interfaces router-casa

Fuente : Propia

8 - Server_VPN_WireGuard

- WireGuard está diseñado para poder realizar roaming de manera fácil y rápida. Si nuestro dispositivo cambia de redes, y lógicamente cambia de IP pública, como por ejemplo cuando pasamos de la red Wi-Fi y la red 4G/LTE de nuestro operador, la conexión VPN seguirá levantada porque se volverán a autenticar rápidamente con este servidor VPN, de tal forma que siempre estaremos conectados a la VPN.

- También se podrá habilitar el Kill-Switch en el dispositivo, de esta forma, si la conexión VPN se interrumpe, el propio software también se encargará de interrumpir todo el tráfico de red hasta que se vuelva a restablecer la conexión VPN, con el objetivo de que no se navegue sin la protección que brinda esta VPN.

- Para el servidor VPN se configuran dos interfaces estáticas.

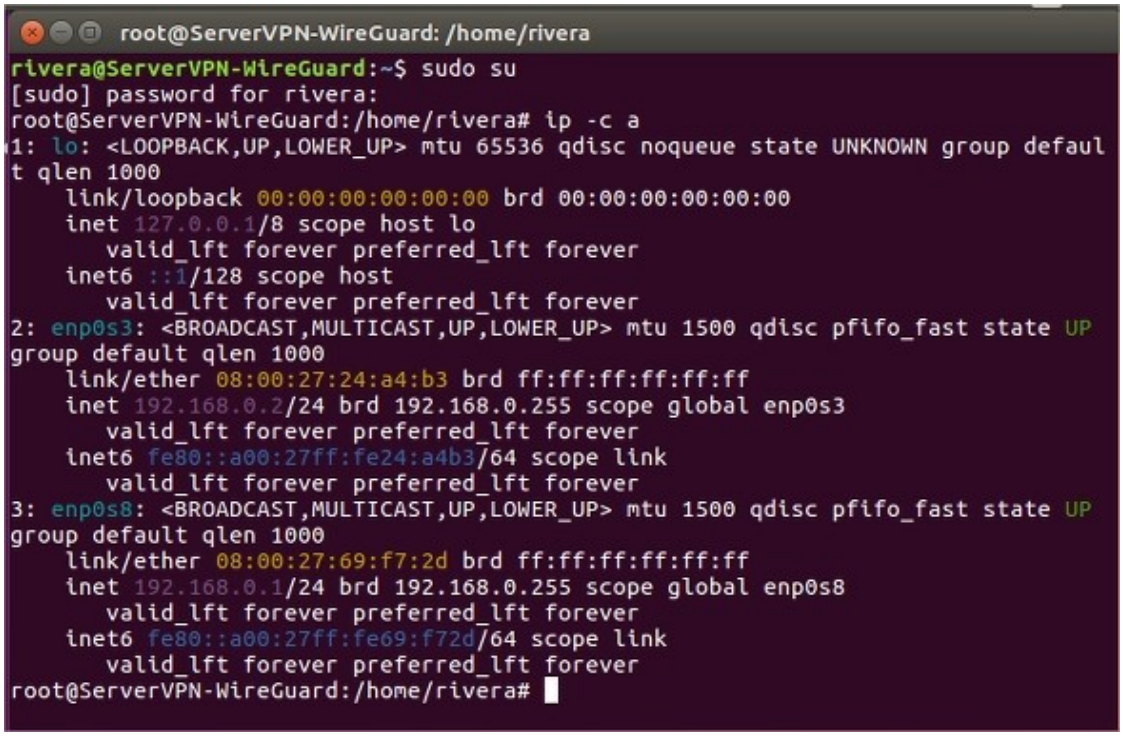


Figura 50: Interfaces Server_VPN

Fuente : Propia

- Se añadirán los repositorios específicos de WireGuard, ya que actualmente por defecto no se encuentra en los paquetes de Ubuntu.

  - Comandos necesarios para ello :

  # sudo add-apt-repository ppa:wireguard/wireguard

  # sudo apt-get update

  # sudo apt-get install wireguard

- Cuando la instalación haya terminado se activará de forma permanente el reenvío de datagramas en el fichero */etc/sysctl.conf* eliminando (#) de la línea *net.ipv4.ip_forward = 1* y se ejecuta el comando *# sysctl -p* para hacer efectivos los cambios, nos posicionaremos en el directorio creado por defecto : # cd /*etc* / wireguard

- En el directorio anterior habrá que generar las claves pública y privada como root.

- El comando para generar las claves será :
  `# wg genkey | tee clave-privada-server | wg pubkey > clave-publica-server`

- Con los comandos " ls " y " cat " se podrá ver si se han generado las claves.

- Después se cambiaran los permisos del usuario y para ello se utilizará el comando :
  `# chmod 600 clave-privada-server clave-publica-server`

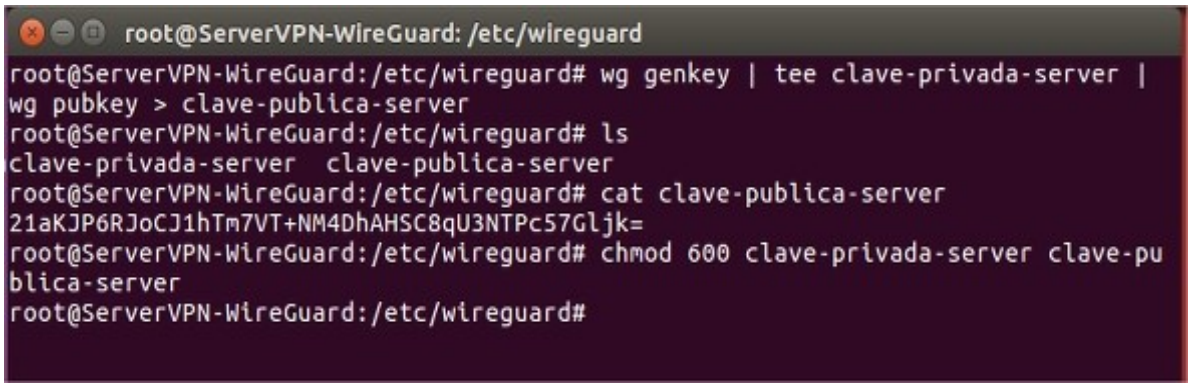


Figura 51: Claves servidor VPN

Fuente : Propia

- Una vez generadas las claves correctamente se pasa a crear el archivo de configuración para el servidor (wg0.conf). En este archivo editaremos los parámetros necesarios para el correcto funcionamiento del servidor. Aunque todavía no se añadirán clientes.

  - Comando de creación:

  `# nano /etc/wireguard/wg0.conf`

[Interfaces]

Address: Define las direcciones IPv4 e IPv6 privadas para el servidor WireGuard.

ListenPort: Especifica qué puerto utilizará WireGuard para las conexiones entrantes.

PostUp y **PostDown:** Define los pasos que se ejecutarán después de encender o apagar la interfaz con respecto a las iptables.

SaveConfig: Ordena al archivo de configuración que se actualice automáticamente cada vez que se agrega un nuevo par.

PrivateKey: Clave privada del servidor.

** Se creará la red virtual 192.168.4.0/24 y el puerto de escucha será el 51820 **

Figura 52: Configuración inicial Wg0.conf

Fuente : Propia

- Para gestionar la interfaz se utilizará el comando : # wg-quick

  # wg-quick up wg0 (habilitar)

Figura 53: Habilitar interfaz Wg0

Fuente : Propia

- Se podrá ver si está activa introduciendo : # wg show

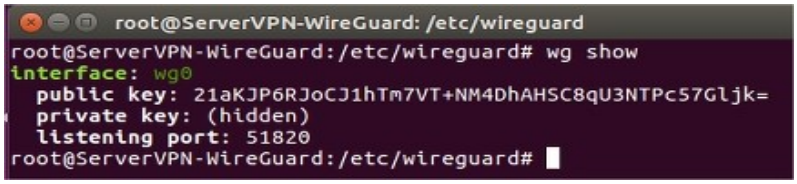


Figura 54: Interfaz Wg0 activa

Fuente : Propia

- Cuando se quiera hacer algún tipo de modificación en el archivo de configuración, como sería introducir un nuevo cliente, se tendrá que deshabilitar la interfaz.

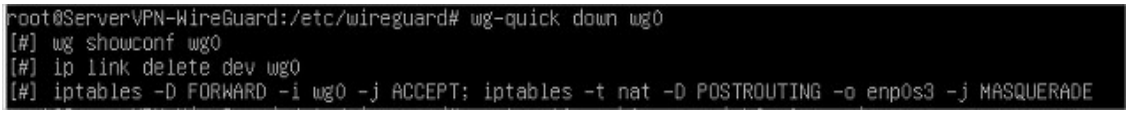


Figura 55: interfaz Wg0 deshabilitada

Fuente : Propia

- Otra forma de gestionar la interfaz es como servicio en Systemd, con ello se logrará que se reinicie automáticamente en el arranque y los comandos necesarios serán :

  # systemctl enable wg-quick@wg0.service

  # systemctl start/stop wg-quick@wg0.service

  # systemctl status wg-quick@wg0.service

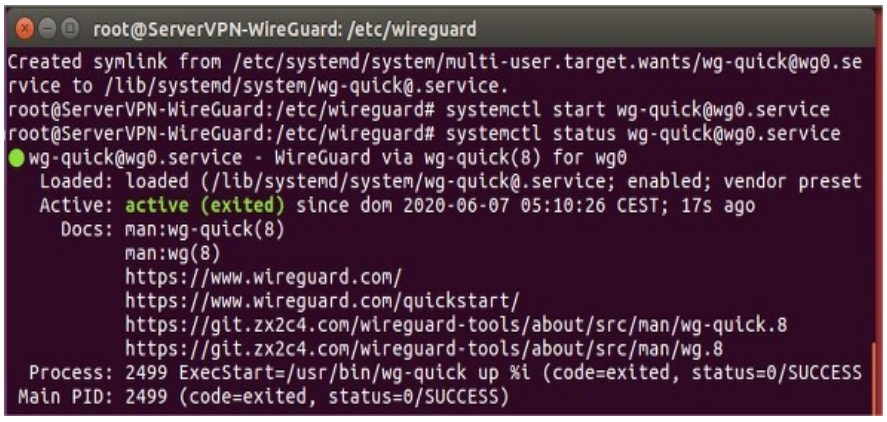


Figura 56: Wg0 en Systemd

Fuente : Propia

- Llegados a este punto, solo quedaría añadir los parámetros de configuración de los clientes.
- Para poder conectarse, se añadirán estos parámetros relacionados y necesarios en el fichero “ wg0.conf “ añadiendo la sección [Peer] tanto en el servidor como en el cliente .

9 - Laptop_Casa

- En este cliente se ha decidido implementar WireGuard en Windows 10 por su facilidad de instalación y por su sencilla configuración. Este se puede conseguir desde la web oficial de WireGuard https://www.wireguard.com/install/

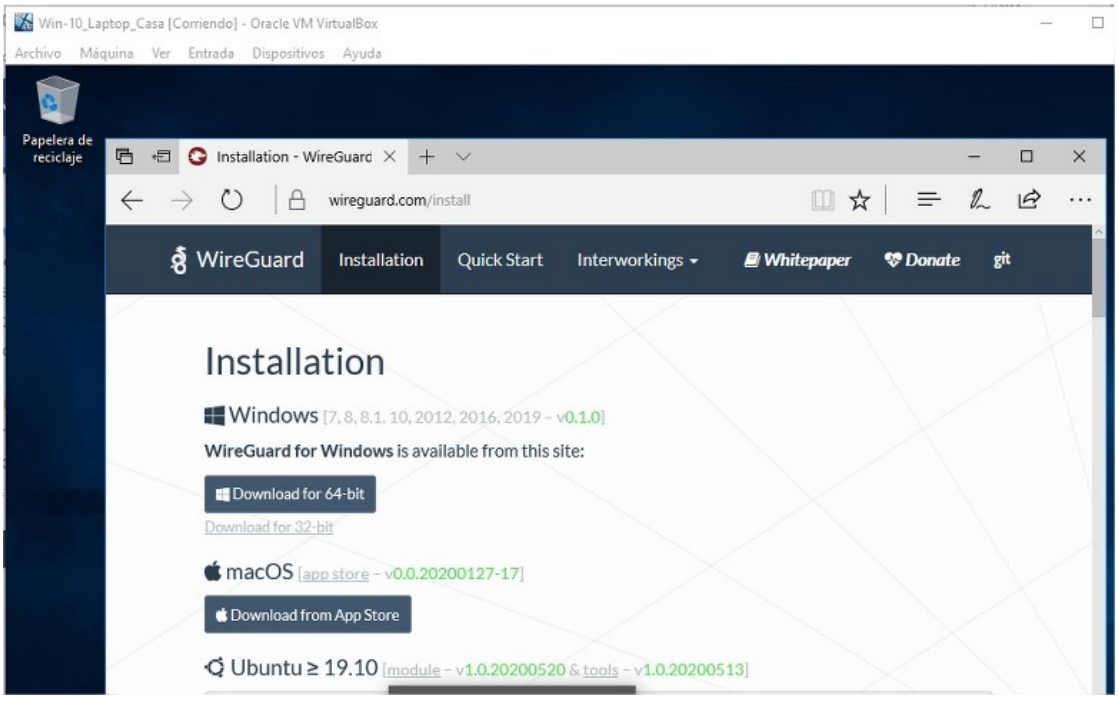


Figura 57: Instalación Windows 10

Fuente : Propia

- Una vez terminada la instalación se ejecutará el software de WireGuard y aparecerá un sencillo e intuitivo entorno gráfico, donde podremos importar un archivo de configuración o crearlo casi desde cero. En la siguiente imagen se puede apreciar su sencillez.

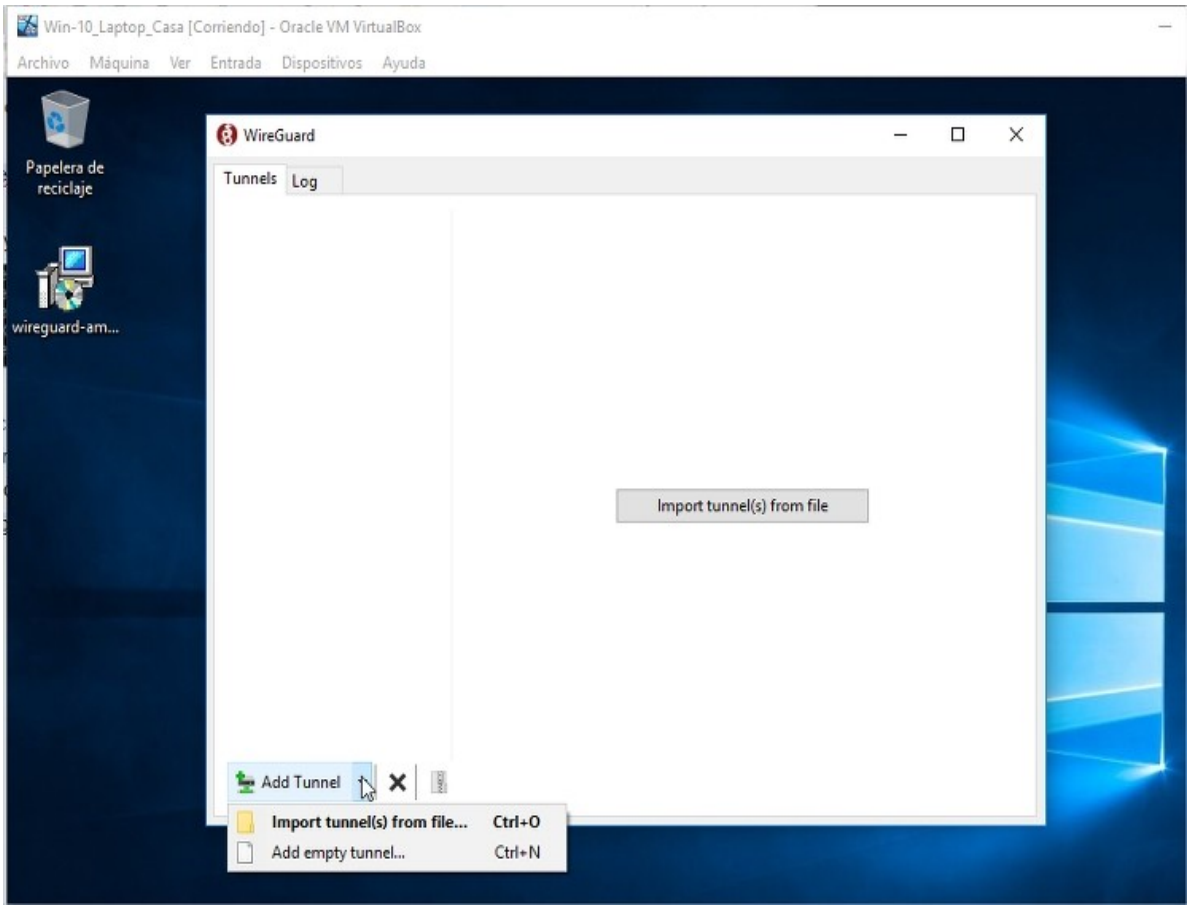


Figura 58: Software WireGuard en Windows 10

Fuente : Propia

- Para crear una configuración completa tanto en el cliente como en el servidor se necesitará intercambiar las claves publicas de ambos. Una forma correcta de hacerlo es utilizando SFTP como se muestra a continuación :

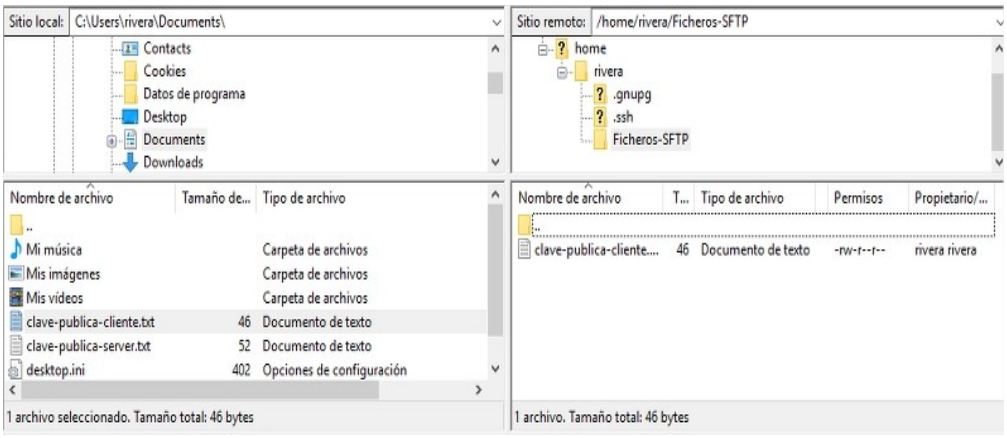


Figura 59: Intercambio de claves en Filezilla

Fuente : Propia

- Es el momento de crear el archivo de configuración en el cliente. Para ello en el menú de selección de " Add Tunnel " en el software de WireGuard se hace clic en :
  " Add empty tunnel "

- Lo primero que se puede ver es que el software ya ha generado la clave publica y la privada y se le trendrá que asignar un nombre al túnel e introducir los parámetros necesarios y su valor. Los parámetros podrán ser los siguientes, aunque algunos no serán obligatorios :

  [Interface]
  Address = < IP del cliente >
  PrivateKey = < Clave privada del cliente >
  DNS = < IP del DNS para el servidor VPN, así se evitan fugas DNS >

  [Peer]
  PublicKey = < Clave publica del servidor >
  AllowedIPs = < Filtrará y enrutará el tráfico de una determinada IP o de todas utilizando 0.0.0.0/0 >
  Endpoint = < IP publica del servidor VPN (:) Puerto de escucha >
  PersistentKeepalive = < Tiempo en segundos para mantener el túnel activo detrás del NAT o Firewall >

- La configuración del cliente con todos los parámetros que se utilizaran quedaría como en la siguiente imagen.

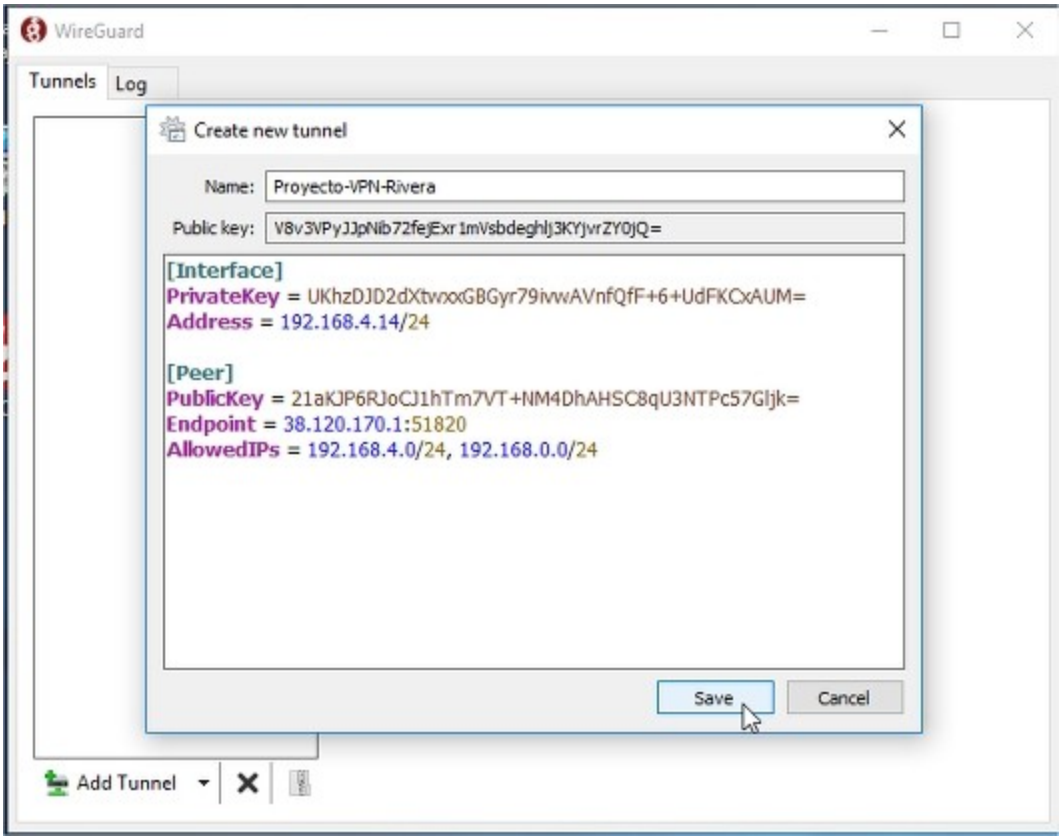


Figura 60: Configuración túnel cliente

Fuente : Propia

- Existe una opción “ Block untunneled traffic “. Esta opción agrega reglas de firewall para eliminar todo el tráfico que no viaja a través de la VPN en Windows.

- Se guarda la configuración y solo faltaría relacionar al cliente con su clave pública en el servidor.

5.1.2 CONEXIÓN DEL CLIENTE CON EL SERVIDOR VPN

- Al conectar un cliente con el servidor, el cliente ya habrá configurado su interfaz con los parámetros del servidor y generado sus claves. Es el momento de incluir al cliente en el archivo de configuración del servidor, para ello se editará el fichero " Wg0.conf ".

- En este fichero se añadirá la sección [Peer], una por cada cliente que se quiera conectar, donde se introducirán dos parámetros esenciales como son la clave publica del cliente (PublicKey) y (AllowedIPs) la IP o rangos de IPs relacionados con dicha clave . Se utilizará el comando : # nano /etc/wiregard/wg0

- El fichero quedaría como la imagen siguiente :

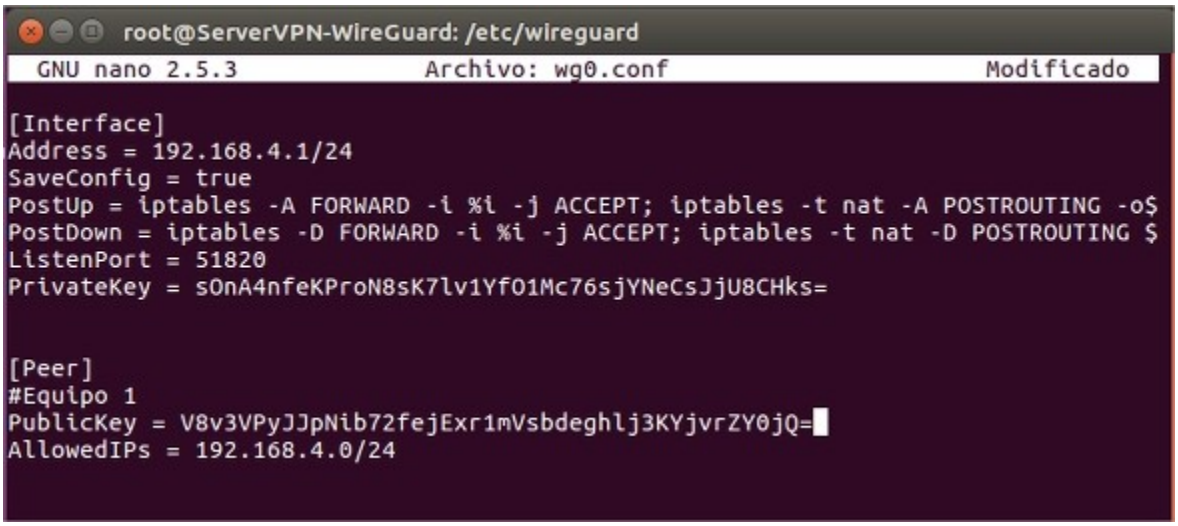


Figura 61: Añadir cliente a Wg0.conf

Fuente : Propia

- Levantaremos la interfaz virtual con el comando : # wg-quick up wg0
- Para ver el estado de la configuración realizada se puede utilizar el comando : # wg showconf wg0

- Si se necesita más información se puede consultar la ayuda :

  # wg –help

```
root@ServerVPN-WireGuard:/etc/wireguard# wg --help
Usage: wg <cmd> [<args>]

Available subcommands:
  show: Shows the current configuration and device information
  showconf: Shows the current configuration of a given WireGuard interface, for use with `setconf'
  set: Change the current configuration, add peers, remove peers, or change peers
  setconf: Applies a configuration file to a WireGuard interface
  addconf: Appends a configuration file to a WireGuard interface
  syncconf: Synchronizes a configuration file to a WireGuard interface
  genkey: Generates a new private key and writes it to stdout
  genpsk: Generates a new preshared key and writes it to stdout
  pubkey: Reads a private key from stdin and writes a public key to stdout
You may pass `--help' to any of these subcommands to view usage.
root@ServerVPN-WireGuard:/etc/wireguard#
```

Figura 62: Ayuda de Wg

Fuente : Propia

- Como se puede ver en la siguiente imagen, en el cliente se tiene que activar la interfaz.

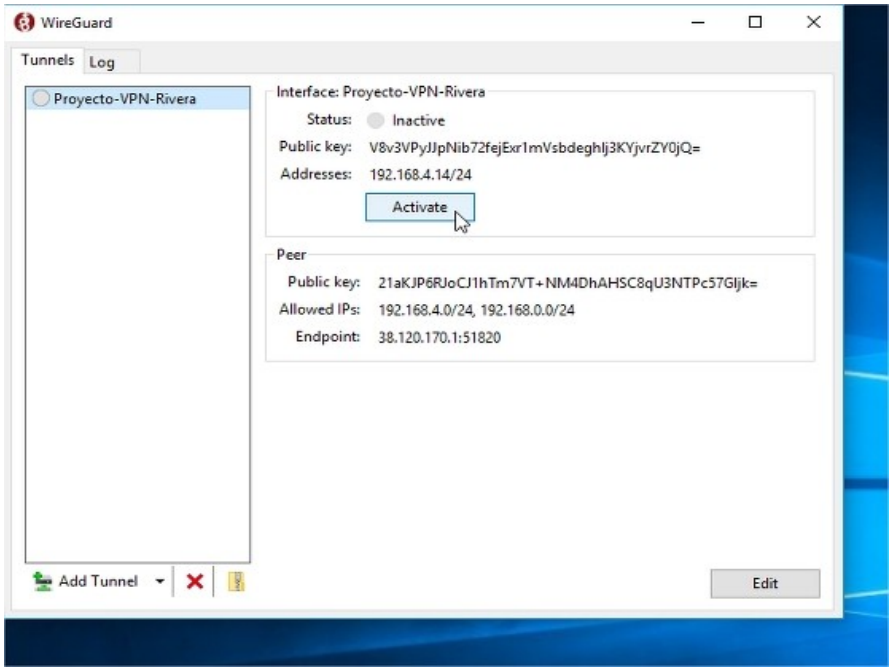


Figura 63: Activar interfaz cliente

Fuente : Propia

- Una vez activada, el túnel se creará comenzando la comunicación con el servidor VPN. Una vez recibido el paquete, lo descifra y comprueba la lista de pares autenticándolo si puede y si es así acepta el paquete.

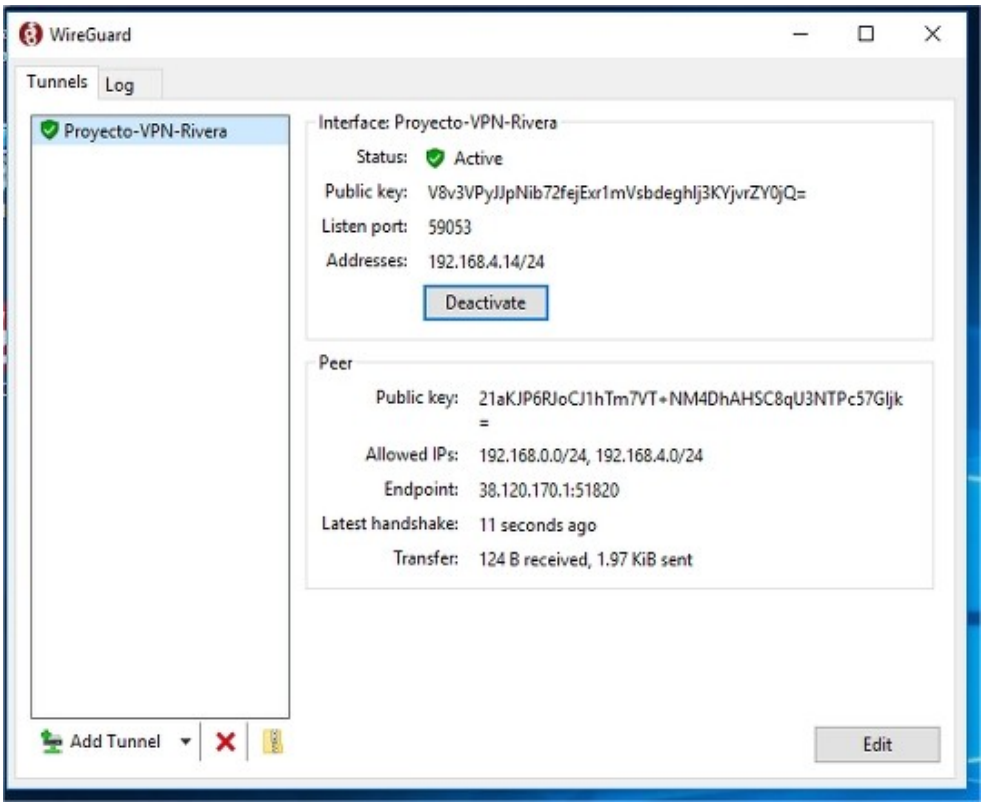


Figura 64: Cliente conectado correctamente

Fuente : Propia

- Para obtener más información en el cliente se dispone de la pestaña de Logs, donde se puede consultar algunos detalles del proceso de conexión.

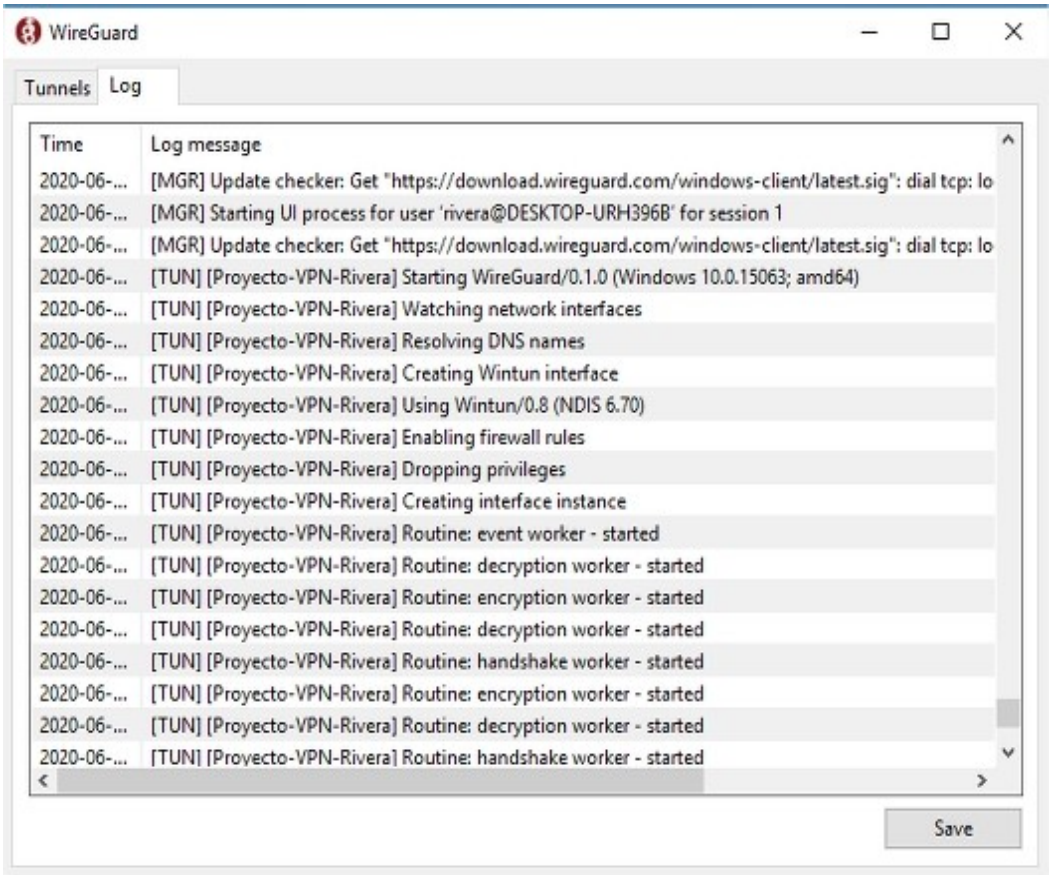


Figura 65: Logs

Fuente : Propia

- Si se tienen problemas con la conexión, y en el apartado de transferencia no hay paquetes recibidos, un detalle a tener en cuenta es la apertura de puertos en el cortafuegos. Tanto en el cliente como en el servidor.

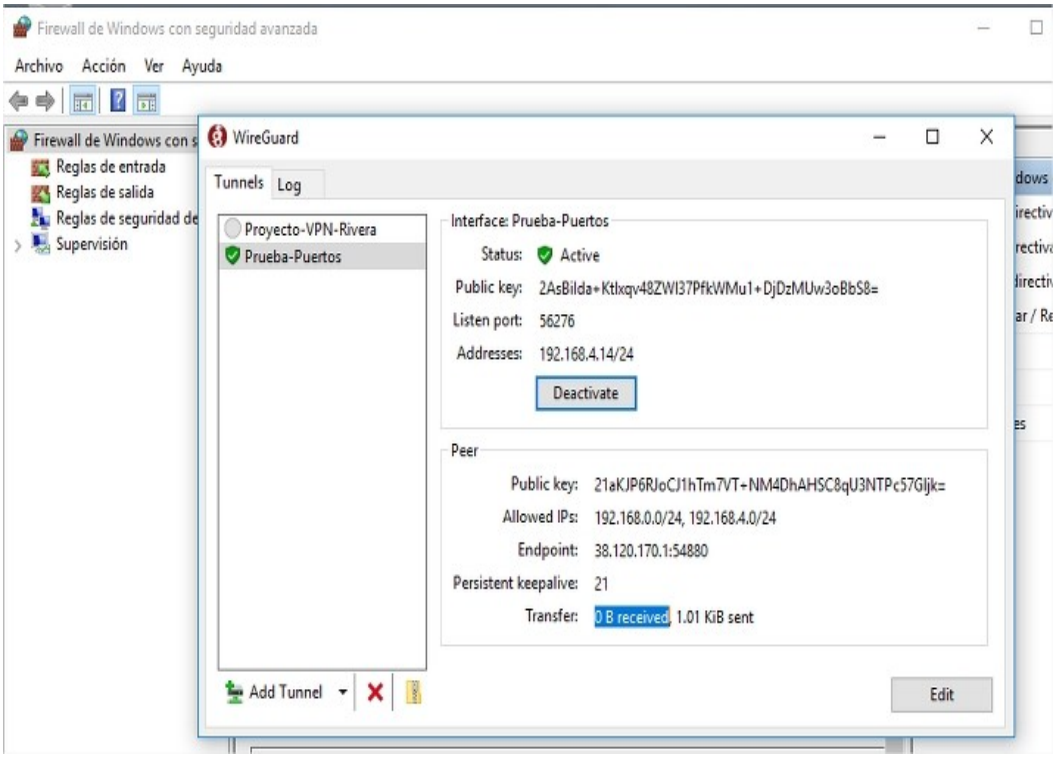


Figura 66: Túnel sin paquetes recibidos

Fuente : Propia

- Al encontrarnos en una situación en la cual el servidor VPN está situado detrás del cortafuegos se tendran que configurar algunas reglas más con iptables. Se añadiran al script de cortafuegos-on las siguientes reglas :

  # Permitimos el tráfico VPN entrante al puerto de escucha.

  iptables -A FORWARD -p udp -m udp - -dport 51820 -j ACCEPT

  # Permitimos el reenvío de paquetes en el túnel VPN

  iptables -A FORWARD -i wg0 -o wg0 -j ACCEPT

- En Windows se tendrá que acceder a la configuración avanzada y crear una regla para poder abrir el puerto UDP necesario para esa conexión.

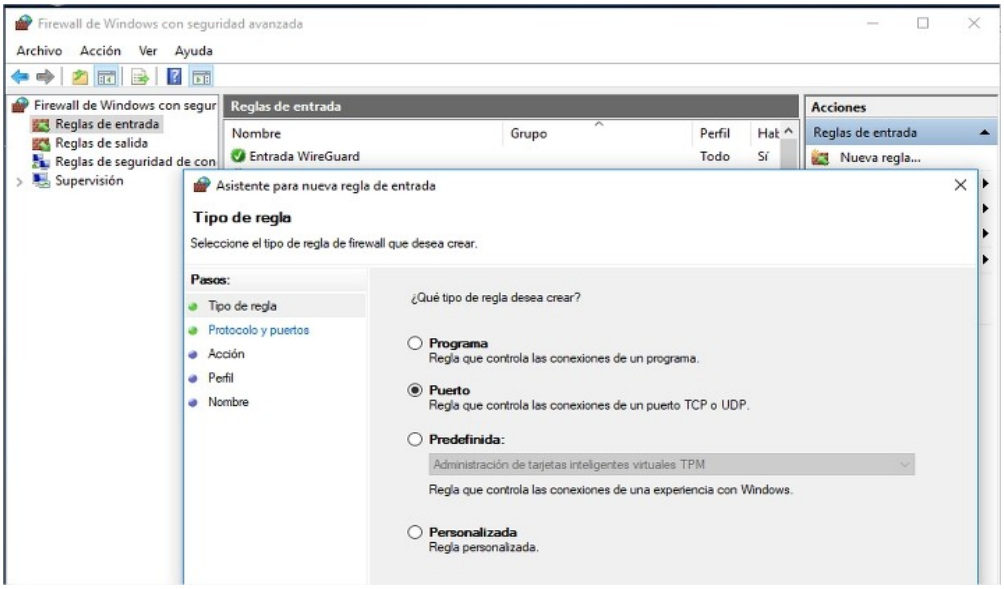


Figura 67: Abrir puertos en Windows

Fuente : Propia

- Un problema importante con muchas configuraciones de VPN es que el DNS tiene fugas. Esto termina filtrando la conexión del cliente y los detalles de ubicación.
  Una buena práctica sería instalar " Unbound[14] ", que es un producto de resolución de DNS de validación, recursivo y de almacenamiento en caché.

- Unbound ha suplantado a Berkeley Internet Name Daemon (BIND) como el servidor de nombres predeterminado del sistema base en FreeBSD y OpenBSD.

14 https://nlnetlabs.nl/projects/unbound/about/

6. CONCLUSIONES

- Debido al rápido avance tecnológico y a las múltiples y continuas mejoras que vive el mundo de la informática, no podría asegurar que WireGuard esté entre nosotros tanto años como llevan OpenVPN o Ipsec, aunque actualmente este protocolo los mejora notablemente. Pienso que las pymes cada vez están más mentalizadas en invertir en tecnología y algunas de ellas ya han comenzado a digitalizarse.

- Con la llegada de la computación cuántica todo lo relacionado con las claves de cifrado pública y privada, como pueden ser los certificados de páginas web, el software de firmas digitales, la comunicación cifrada de la banca electrónica y otros tipos de aplicaciones, tiene un futuro incierto. Mientras un ordenador actual tardaría muchísimos años en descifrar la clave privada, los ordenadores cuánticos lo procesarían en unos pocos minutos. Por este motivo algunas organizaciones importantes como el Instituto Nacional de Estándares y Tecnología están estudiando nuevos algoritmos para remplazar el sistema RSA o el protocolo ECDH.

- A nivel personal este proyecto me ha servido para afianzar los conocimientos adquiridos en los dos años del ciclo, algo que me mueve a mostrar más interés por seguir aprendiendo y hace que me sienta muy satisfecho por haber elegido estos estudios.

6.1 LIMITACIONES ENCONTRADAS

- Al realizar este proyecto he tenido algunos problemas con la virtualización al disponer solo de un ordenador portátil y no disponer de tanta memoria RAM. Ttambién al ser WireGuard un protocolo relativamente nuevo surgieron algunos fallos con los repositorios. Otras complicaciones han sido las relacionadas con la configuración del cortafuegos ASA en Packet Tracer.

7. ÍNDICE DE FIGURAS

Índice de figuras

8. GLOSARIO

- **ACLs:**
  Lista de control de acceso. Las ACL permiten controlar el flujo del tráfico en equipos de redes, tales como enrutadores y conmutadores.
- **BLAKE2:**
  Función hash criptográfica rápida y segura.
- **ChaCha20:**
  Cifrado de flujo. Un cifrado de flujo es un cifrado de clave simétrica.
- **Curve25519:**
  Es una curva elíptica que ofrece 128 bits de seguridad y está diseñada para usarse con el esquema de acuerdo clave de la curva elíptica Diffie-Hellman (ECDH).
- **DMZ:**
  En seguridad informática una zona desmilitarizada o red perimetral es una red local que se ubica entre la red interna de una organización y una red externa.
- **ECDH:**
  Es un protocolo de establecimiento de claves anónimo.
- **HKDF:**
  Función para derivar claves de diferente longitud.
- **IMAP:**
  Protocolo de acceso a mensajes de Internet.
- **NAT-P:**
  Traducción de Direcciones de Red por Puerto.
- **Noise protocol framework:**
  Protocolos criptográficos simples, rápidos y seguros
- **Poly1305:**
  Es un código de autenticación de mensajes de clave secreta de última generación adecuado para una amplia variedad de aplicaciones.
- **POP3:**
  (Protocolo de Oficina Postal) en clientes locales de correo para obtener los mensajes de correo electrónico almacenados en un servidor remoto.

- **Roaming:**

  Concepto utilizado en telecomunicaciones para referirse a la posibilidad de un dispositivo inalámbrico de utilizar una cobertura de red distinta de la principal.
- **SipHash24:**

  Es una familia de funciones pseudoaleatorias (también conocidas como funciones hash con clave) optimizadas para la velocidad en mensajes cortos.
- **UDP:**

  Es un protocolo del nivel de transporte basado en el intercambio de datagramas
- **UNIX:**

  Es un sistema operativo portable, multitarea y multiusuario.
- **VPN:**

  Virtual Private Network, VPN es una tecnología de red de ordenadores que permite una extensión segura de la red de área local (LAN) sobre una red pública o no controlada como Internet.

9. BIBLIOGRAFÍA

- Libro de Redes Locales e Internet (conceptos y práctica)

  Arturo Mora Rioja, M.ª Mercedes Rodríguez Villafáfila
- Libro de Servicios de Red e Internet

  Grupo editorial Garceta
- Apuntes y Prácticas 2º ASIR
- https://www.incibe.es/
- https://www.wireguard.com/ https://www.wireguard.com/performance/
- https://www.cisco.com/c/en/us/products/security/vpn-endpoint-security-clients/what-is-vpn.html
- https://www.opendns.com/
- https://wiki.archlinux.org/index.php/Systemd_(Espa%C3%B1ol)
- https://wiki.archlinux.org/index.php/WireGuard#Manual_WireGuard_setup
- https://www.gamificafp.com/pluginfile.php/8111/mod_resource/content/0/2-Servidor-DHCP-y-DNS.pdf
- http://somebooks.es/administrar-servicios-demonios-de-ubuntu-con-boot-up-manager/
- https://www.ssh.com/ssh/sftp/
- https://www.ecured.cu/RSA
- https://wiki.archlinux.org/index.php/Postfix_(Espa%C3%B1ol)
- https://wiki.archlinux.org/index.php/Dovecot
- https://wiki.archlinux.jp/index.php/Squirrelmail
- https://www.linuxadictos.com/que-son-los-meta-paquetes-de-linux.html
- https://www.wireguard.com/install/

9. ANEXO

- Topología de Red Packet Tracer (Archivo .pkt)
- Script (Cortafuegos-on)