



INSTITUTO POLITÉCNICO NACIONAL  
ESCUELA SUPERIOR DE CÓMPUTO



Aplicaciones para comunicaciones en Red

***Practica HTTP/HTTPS***

Servidor HTTP/HTTPS

Alumno:

Cortez Enríquez Jovanny Wilver

González Lopez Emiliano

3CV6

## Contenido

Introducción.....	3
Planteamiento del Problema.....	4
Requisitos.....	4
Desarrollo .....	5
Creación de la arquitectura en GNS3.....	5
Instalación del servidor Apache2 .....	6
Configuración del servidor Apache2.....	7
Crear 2 sitios con la misma IP .....	7
Crear sitio web con autenticación .....	9
Pruebas.....	11
Conclusión.....	20

## Introducción

HTTP (Hyper-Text Transfer Protocol) Protocolo de Transferencia de Hiper-Texto es el protocolo de comunicaciones que permite la transferencia de información en la World Wide Web.

HTTP es un protocolo sin estado, es decir, no guarda ninguna información sobre conexiones anteriores.

Su referencia esta en el RFC 2616 y utiliza el puerto 80 por defecto, aunque puede cambiar su puerto según sea el servidor que lo implementa.

Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. El cliente se le suele llamar “agente de usuario”, (en inglés agent user) realiza una petición enviando un mensaje con cierto formato al servidor. El servidor (se le suele llamar servidor web) le envía un mensaje de respuesta. Un ejemplo de cliente son los navegadores web y las arañas web (también conocida por su término en inglés webcrawlers).

HTTPS nace de la versión de HTTP puesto que la información que se manda por HTTP no ésta cifrado por lo que pueden obtener los datos fácilmente cualquier persona que sepa sobre el protocolo. La S significa Seguro o Secure del inglés, el protocolo es la versión segura de HTTP. El cifrado está basado en la seguridad de textos SSL/TLS para crear un canal cifrado más apropiado para el tráfico de información. Utiliza el puerto 443 por defecto, aunque se puede cambiar de igual manera, depende del servidor que lo implementa.

## Planteamiento del Problema

Crear una página web que sirva como interfaz para la configuración de los protocolos previamente configurados ahora de manera gráfica.

Para el protocolo TFTP haremos la copia de seguridad del router seleccionado.

Para el protocolo DNS podremos agregar una nueva entrada para la zona directa y zona inversa.

Para el protocolo DHCP podremos agregar una subred en nuestra lista de subredes.

## Requisitos

- Programa GNS3
- Arquitectura planteada por el profesor
- Sistema Operativo Linux Ubuntu Server 20
- Paquete apache2
- VirtualBox

## Desarrollo

### Creación de la arquitectura en GNS3

Se debe crear la siguiente arquitectura en GNS3 para realizar la práctica del servidor DNS

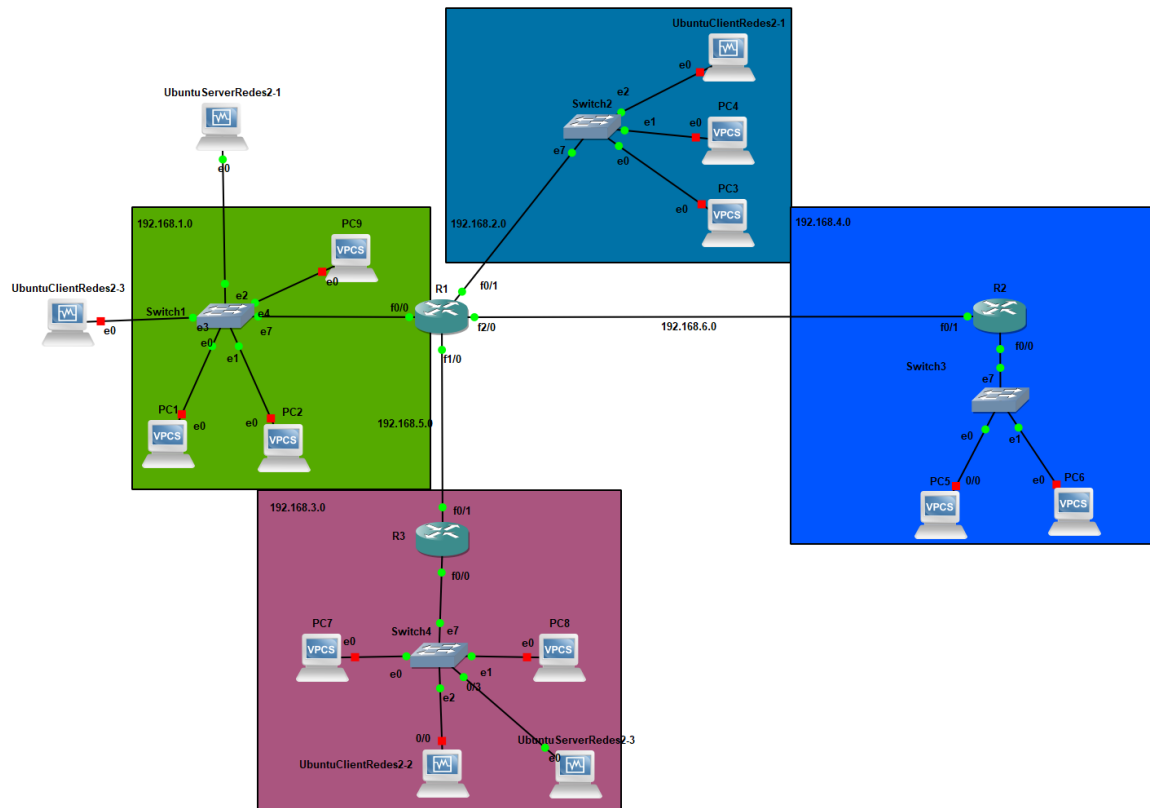


Imagen 1: Arquitectura de red

Se realizan las configuraciones siguientes:

- Configurar la tabla de ruteo de manera dinámica con el protocolo RIP.
- Configurar las IP de manera dinámica con el servidor DHCP
- Configurar el protocolo DNS
- Configurar el protocolo TFTP

## Instalación del servidor Apache2

1.-Correr la PC agregada a la arquitectura de nombre UbuntuServerRedes2 en VirtualBox antes de iniciar la configuración para tener los paquetes necesarios

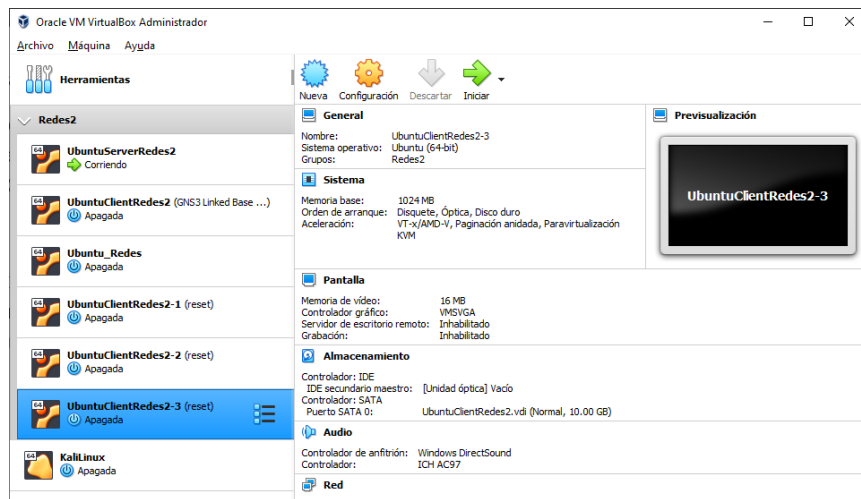


Imagen 2: VirtualBox corriendo la máquina UbuntuServerRedes2

2.- Actualizar los repositorios.

```
$sudo apt update
```

3.- Instalar los paquetes necesarios o los sugeridos por el sistema mismo.

```
$sudo apt upgrade -y
```

3.- Instalar los paquetes BIND9 con el siguiente comando:

```
$sudo apt install apache2
```

4.- Verificar que se haya instalado el paquete tecleando el comando de estado `service apache2 status`

```
root@jovanny:~# service apache2 restart
root@jovanny:~# service apache2 status
• apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2020-06-08 23:15:44 UTC; 2s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 1641 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 1660 (apache2)
    Tasks: 55 (limit: 2283)
   Memory: 4.9M
   CGroup: /system.slice/apache2.service
           └─1660 /usr/sbin/apache2 -k start
             └─1661 /usr/sbin/apache2 -k start
               └─1662 /usr/sbin/apache2 -k start

jun 08 23:15:44 jovanny systemd[1]: Starting The Apache HTTP Server...
jun 08 23:15:44 jovanny apachectl[1657]: AH00558: apache2: Could not reliably determine the server's
jun 08 23:15:44 jovanny systemd[1]: Started The Apache HTTP Server.
root@jovanny:~#
```

Imagen 3: Estado del servicio de apache2

## Configuración del servidor Apache2.

La configuración del servidor esta en el archivo `apache2.conf` ubicado en `/etc/apache2/apache2.conf` donde vienen incluido los ficheros que forman parte de la configuración.

Por defecto se indican las opciones de configuración del directorio `/var/www` y de todos sus subdirectorios, por lo tanto los `DocumentRoot` de los `virtualHost` que se crean deben ser subdirectorios de este directorio.

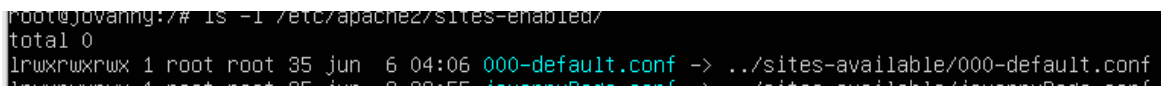
Por defecto `apache2` crea una página web dentro del directorio `/var/www` llamado `html` en el fichero `index.html`

### Crear 2 sitios con la misma IP

Como sabemos, `apache` crea una página web por defecto. Crearemos otra página web que utilizará la misma IP que utiliza la página web de `apache2`.

- 1- Comprobamos que exista un enlace simbólico del sitio de `apache2`:

```
$ls -l /etc/apache2/sites-enabled/
```



```
root@jovanny:~# ls -l /etc/apache2/sites-enabled/
total 0
lrwxrwxrwx 1 root root 35 jun  6 04:06 000-default.conf -> ../sites-available/000-default.conf
```

Imagen 4: link simbólico del archivo de configuración

- 2- Creamos una copia del fichero `000-default.conf` con otro nombre, en este ejemplo `jovannyPage.conf`

```
#cp /etc/apache2/sites-availables/000-defaul.conf /etc/apache2/sites-
availables/jovannyPage.conf
```

- 3- Modificamos su contenido con lo siguiente:

ServerName [www.jovannyPage.com](http://www.jovannyPage.com)

ServerAdmin [jovanny@localhost.com](mailto:jovanny@localhost.com)

DocumentRoot `/var/www/jovannyPage`

- 4- Creamos su enlace simbólico

```
#a2ensite jovannyPage
```

- 5- Creamos el directorio y el fichero `index.html`

```
#mkdir /var/www/jovannyPage
```

```
#touch /var/www/jovannyPage/index.html
```

- 6- Creamos el contenido `html` necesario en `index.html`, como un saludo.

- 7- Reiniciamos el servicio de `http`

```
#service apache2 restart
```

#service apache2 status

```
root@jovanny:/# service apache2 status
• apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2020-06-08 23:15:44 UTC; 1h 25min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 1641 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Process: 1785 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
   Main PID: 1660 (apache2)
     Tasks: 55 (limit: 2283)
    Memory: 5.0M
   CGroup: /system.slice/apache2.service
           └─1660 /usr/sbin/apache2 -k start
             └─1789 /usr/sbin/apache2 -k start
               └─1790 /usr/sbin/apache2 -k start

Jun 08 23:15:44 jovanny systemd[1]: Starting The Apache HTTP Server...
Jun 08 23:15:44 jovanny apachectl[1657]: AH00558: apache2: Could not reliably determine the server's
Jun 08 23:15:44 jovanny systemd[1]: Started The Apache HTTP Server.
Jun 09 00:00:01 jovanny systemd[1]: Reloading The Apache HTTP Server.
Jun 09 00:00:02 jovanny apachectl[1788]: AH00558: apache2: Could not reliably determine the server's
Jun 09 00:00:02 jovanny systemd[1]: Reloaded The Apache HTTP Server.
lines 1-20/20 (END)
```

Imagen 5: servicio corriendo.

- 8- Comprobamos el sitio, pero antes agregamos el nombre de dominio a nuestro servidor DNS y que redirija a la IP del servidor apache.
- 9- Comprobamos el sitio.

\$curl www.jovannyPage.com

```
jovanny@jovanny:~$ curl www.jovannyPage.com
<html>
<head>
<title> jovanny </title>
</head>
<body>
<p> Esto es un parrafo de web </p>
</body>
</html>
jovanny@jovanny:~$
```

Imagen 6: contenido html del fichero index.html de la página web [www.jovannyPage.com](http://www.jovannyPage.com)



## Crear sitio web con autenticación

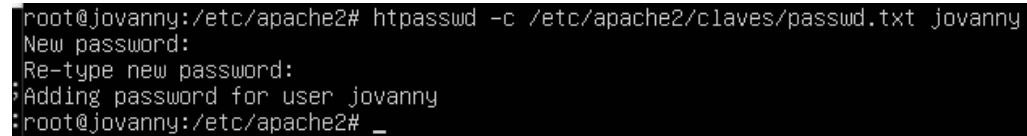
Para crear un sitio que necesite de autenticación básica se debe de hacer lo siguiente.

- 1- Crear un sitio y realizar las configuraciones básicas necesarias como en el punto anterior.
- 2- Configuramos el archivo apache2.conf agregando lo siguiente después del directorio principal.

```
<Directory "/var/www/miweb/privado">
    AuthUserFile "/etc/apache2/claves/passwd.txt"
    AuthName "Palabra de paso"
    AuthType Basic
    Require valid-user
</Directory>
```

- 3- Creamos el fichero de contraseñas

```
$htpasswd /etc/apache2/claves/passwd.txt Jovanny
```

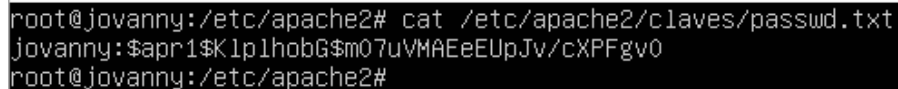


```
root@jovanny:/etc/apache2# htpasswd -c /etc/apache2/claves/passwd.txt jovanny
New password:
Re-type new password:
Adding password for user jovanny
root@jovanny:/etc/apache2# _
```

Imagen 7: Creación de usuario con su contraseña

- 4- Verificamos el fichero creado.

```
$cat /etc/apache2/claves/passwd.txt
```



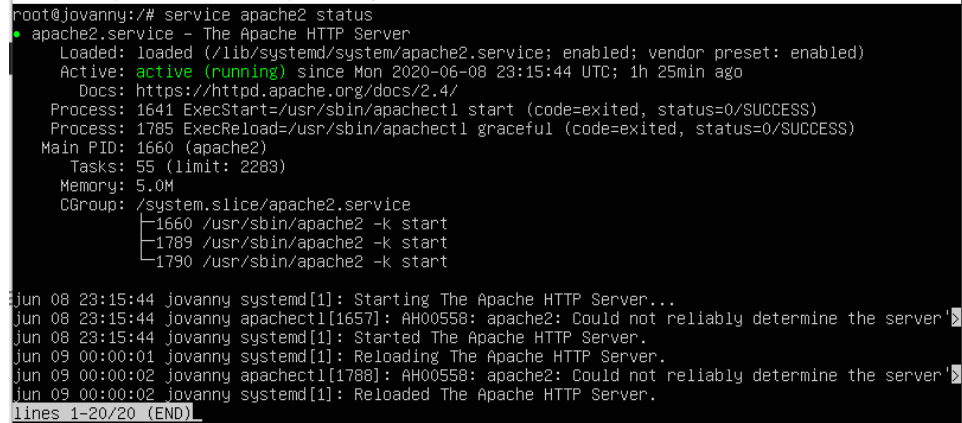
```
root@jovanny:/etc/apache2# cat /etc/apache2/claves/passwd.txt
jovanny:$apr1$K1plhobG$m07uVMAEeEUjV/cXPFgv0
root@jovanny:/etc/apache2#
```

Imagen 8: usuario y contraseña creados.

- 5- Reiniciamos el servicio

```
#service apache2 restart
```

```
#service apache2 status
```



```
root@jovanny:/# service apache2 status
• apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2020-06-08 23:15:44 UTC; 1h 25min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 1641 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Process: 1785 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
   Main PID: 1660 (apache2)
     Tasks: 55 (limit: 2283)
    Memory: 5.0M
   CGroup: /system.slice/apache2.service
           └─1660 /usr/sbin/apache2 -k start
             └─1789 /usr/sbin/apache2 -k start
               └─1790 /usr/sbin/apache2 -k start

Jun 08 23:15:44 jovanny systemd[1]: Starting The Apache HTTP Server...
Jun 08 23:15:44 jovanny apachectl[1657]: AH00558: apache2: Could not reliably determine the server's
Jun 08 23:15:44 jovanny systemd[1]: Started The Apache HTTP Server.
Jun 09 00:00:01 jovanny systemd[1]: Reloading The Apache HTTP Server.
Jun 09 00:00:02 jovanny apachectl[1788]: AH00558: apache2: Could not reliably determine the server's
Jun 09 00:00:02 jovanny systemd[1]: Reloaded The Apache HTTP Server.
lines 1-20/20 (END)
```

Imagen 9: servicio corriendo.

- 6- Agregamos el dominio al servidor DNS
- 7- Probamos desde otra máquina.

```
jovanny@jovanny:~$ curl www.jovaPass.com
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>401 Unauthorized</title>
</head><body>
<h1>Unauthorized</h1>
<p>This server could not verify that you
are authorized to access the document
requested. Either you supplied the wrong
credentials (e.g., bad password), or your
browser doesn't understand how to supply
the credentials required.</p>
<hr>
<address>Apache/2.4.41 (Ubuntu) Server at www.jovapass.com Port 80</address>
</body></html>
jovanny@jovanny:~$
```

Imagen 10: acceso no autorizado porque requiere autenticación.

```
jovanny@jovanny:~$ curl --basic -u jovanny www.jovaPass.com
Enter host password for user 'jovanny':
<html>
<head>
<title> Contraseña </title>
</head>
<body>

<p> Logro auntenticarse </p>

</boyd>
</html>
jovanny@jovanny:~$ _
```

Imagen 11: Autenticación lograda con éxito

## Pruebas

### Gestor de protocolos

TFTP, DNS, DHCP

Las configuraciones posibles serán:

TFTP: Permite hacer la copia de seguridad de los routers.

DNS: Agregar una nueva entrada en la tabla de zona directa e inversa.

DHCP: Agregar una subred gestionando el rango de redes.

Protocolo	
TFTP	<input type="button" value="Seleccionar"/>
DNS	<input type="button" value="Seleccionar"/>
DHCP	<input type="button" value="Seleccionar"/>

Imagen 12 Pagina elaborada

Selecciona un router al cual hacerle backup

Imagen 13 Opción del protocolo TFTP

Has seleccionado DNS, aquí puedes agregar una entrada de red para Zona directa y zona inversa.

Ingresa el nombre de la ip (Ej. pcn.prueba.com):

Ingresa la direccion ip a nombrar:

Ingresa identificador para la Zona inversa:

Imagen 14 Opción del protocolo DNS

Has seleccionado DHCP, aquí podras agregar una subred a la configuracion

Ingresa la subred:

Ingresa la mascara de subred:

Ingresa la puerta de enlace predeterminada:

Ingresa el rango que quieres asignar:

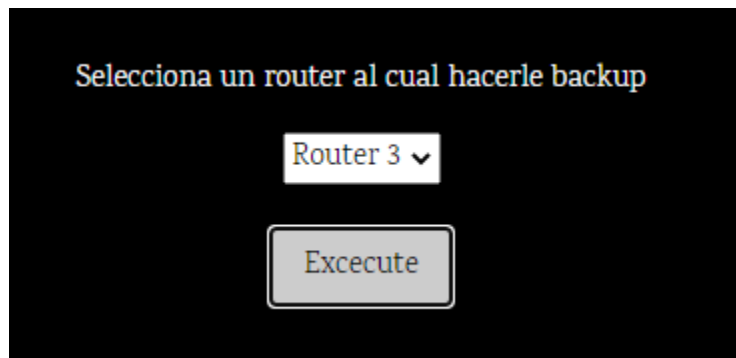
Imagen 15 Opción del protocolo DHCP

## Ejecución de cada una de las opciones

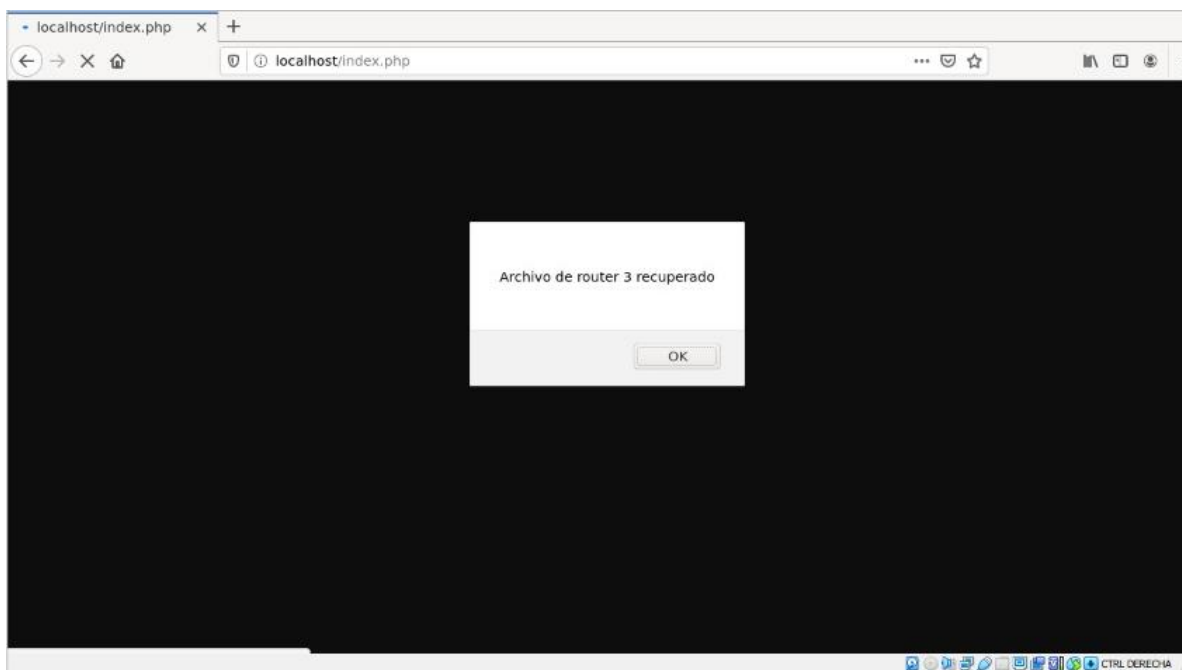
### Ejecución del comando de TFTP

```
root@jovanny:/var/www/html/php# ls /tftpboot/  
root@jovanny:/var/www/html/php#
```

Podemos observar que no tenemos ningún archivo de backup



Seleccionamos el router 3



La pagina nos muestra un mensaje de confirmación de que la el archivo fue recuperado.

```
root@jovanny:/etc/bind# ls /tftpboot/  
router-config-192.168.5.3-2020-06-16  
root@jovanny:/etc/bind#
```

Podemos observar el archivo del router 3 con ip 192.168.5.3

## Ejecución del comando DNS

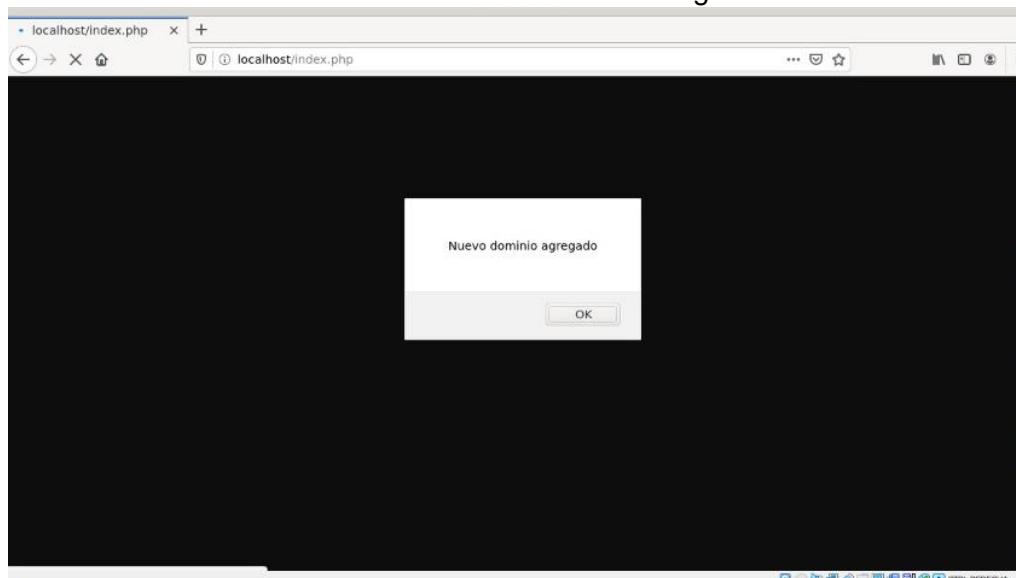
Has seleccionado DNS, aquí puedes agregar una entrada de red para Zona directa y zona Inversa.

Ingresa el nombre de la ip (Ej. pcn.prueba.com):

Ingresa la dirección ip a nombrar:

Ingresa identificador para la Zona Inversa:

Seleccionamos DNS e introducimos los siguientes datos



Mensaje de confirmación

```
GNU nano 4.8 /etc/bind/db.jovanny.local
;
; BIND data file for local loopback interface
;
$TTL 604800
@      IN      SOA     jovanny.local. admin.jovanny.local. (
                        3      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200; Expire
                        604800 ) ; Negative Cache TTL
;
; name servers - NS records
@      IN      NS      jovanny.local.

;name servers - A records
jovanny.local.      IN      A      192.168.1.4

;192.168.0.0/24 - A records
emiliano.jovanny.local. IN      A      192.168.4.3

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit    ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

Podemos observar que se agregó el registro correctamente en zona directa

```

GNU nano 4.8 /etc/bind/db.192.168
;
; BIND reverse data file for local loopback interface
;
$TTL      604800
@         IN      SOA      jovanny.local. root.jovanny.local. (
                        3      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
; name servers - NS records
@         IN      NS       jovanny.local.
;
; PTR Records
4.1       IN      PTR      jovanny.local.
3.4       IN      PTR      emiliano.jovanny.local.

```

[ Read 16 lines ]

```

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line

```

Podemos observar que se agregó el registro correctamente en zona inversa

## Ejecución del comando DHCP

Has seleccionado DHCP, aquí podras agregar una subred a la configuracion

Ingresa la subred: 192.168.4.0

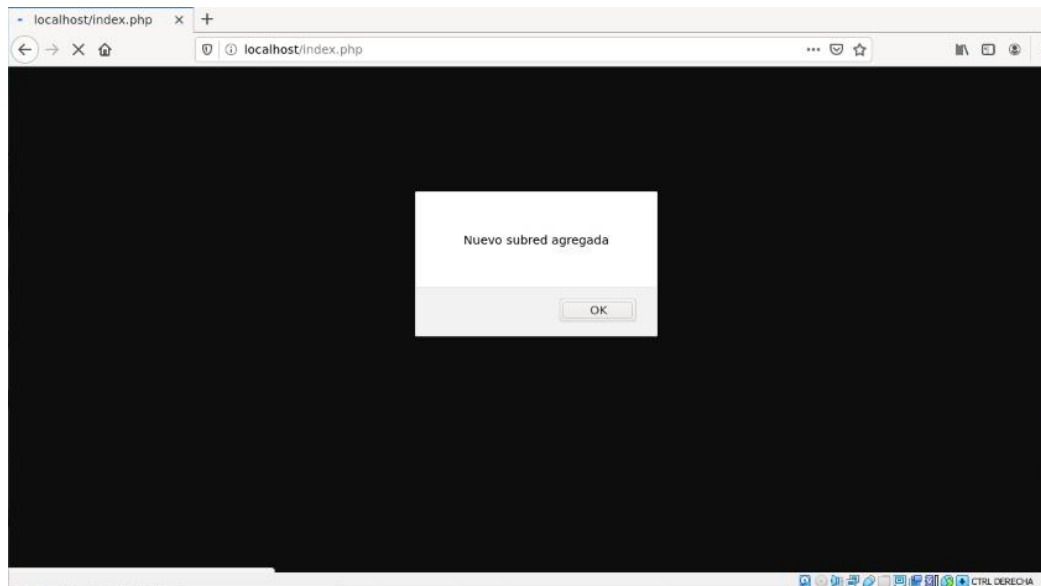
Ingresa la mascara de subred: 255.255.255.0

Ingresa la puerta de enlace predeterminada: 192.168.4.254

Ingresa el rango que quieres asignar: 5 50

Execute

Seleccionamos DHCP e introducimos la información.



Vemos el mensaje de confirmación



```
GNU nano 4.8 /etc/dhcp/dhcpd.conf Modified
# based on that. The example below shows a case where all clients
# in a certain class get addresses on the 10.17.224/24 subnet, and a
# other clients get addresses on the 10.0.29/24 subnet.

#class "foo" {
# match if substring (option vendor-class-identifier, 0, 4) = "SUNW"
#}

#shared-network 224-29 {
# subnet 10.17.224.0 netmask 255.255.255.0 {
# option routers rtr-224.example.org;
# }
# subnet 10.0.29.0 netmask 255.255.255.0 {
# option routers rtr-29.example.org;
# }
# pool {
# allow members of "foo";
# range 10.17.224.10 10.17.224.250;
# }
# pool {
# deny members of "foo";
# range 10.0.29.10 10.0.29.230;
# }
#}

subnet 192.168.4.0 netmask 255.255.255.0 {
range 192.168.4.5 192.168.4.50;
option router 192.168.4.254;
default-lease-time 600;
max-lease-time 7200;
}

^C Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^L Replace ^P Paste Text ^T To Spell
```

Nuestros datos se introducen al final del documento de configuración.

```
Press '?' to get help.

Executing the startup file

DD
Can't find dhcp server

PC6> show ip

NAME       : PC6[1]
IP/MASK    : 0.0.0.0/0
GATEWAY    : 0.0.0.0
DNS        :
MAC        : 00:50:79:66:68:05
LPORT      : 10058
RHOST:PORT : 127.0.0.1:10059
MTU        : 1500

PC6> ip dhcp
DDORA IP 192.168.4.5/24 GW 192.168.4.254

PC6> show ip

NAME       : PC6[1]
IP/MASK    : 192.168.4.5/24
GATEWAY    : 192.168.4.254
DNS        : 192.168.1.4
DHCP SERVER : 192.168.1.4
DHCP LEASE  : 585, 600/300/525
DOMAIN NAME : jovanny.local
MAC        : 00:50:79:66:68:05
LPORT      : 10058
RHOST:PORT : 127.0.0.1:10059
MTU        : 1500

PC6> █
```

Probamos la funcionalidad

## Scripts y comandos de ejecución

```
<?php
if(array_key_exists('tftpBtn',$_POST)){ tftp();}
else if(array_key_exists('dnsBtn',$_POST)){ dns();}
else if(array_key_exists('dhcpBtn',$_POST)){ dhcp();}
function tftp(){
    $router = $_POST['routers'];
    shell_exec("python3 /var/www/html/Script/conectarTelnet.py");
    alert("Archivo de router $router recuperado");
}
function dns(){
    $dominio = $_POST['dominio'];
    $ip = $_POST['ip'];
    $zonei = $_POST['zonei'];
    shell_exec("python3 /var/www/html/Script/DNS.py");
    alert("Nuevo dominio agregado");
}
function dhcp(){
    shell_exec("python3 /var/www/html/Script/DHCP.py");
    alert("Nueva subred agregada");
}
function alert($msg){
    echo "<script type='text/javascript'>alert('$msg');</script>";
}

?>
```

Todos nuestros comandos se ejecutan con un pequeño código de php desde la misma página de index.php, ocupamos php ya que nos permite ejecutar comandos de consola directamente desde la página web, de esta manera podemos ejecutar scripts de Python.

## Script de TFTP

```
import sys
import telnetlib
import datetime
import subprocess
import sys

#Listas
ip = sys.argv[0]

#variables
#ip = ''

#constantes
password = "redes2"
fechaHoy = str(datetime.date.today())
nameConfig = "router-config-"
ipServer = "192.168.1.4"

print("-----")
print("Obteniendo archivo de configuracion")
print("del router: " + ip)
name = nameConfig + ip + '-' + fechaHoy
tn = telnetlib.Telnet(ip)
if password:
    tn.write(password.encode('ascii') + b"\n")
tn.write(b"en\n")
tn.write(password.encode('ascii') + b"\n")
tn.write(b"copy run tftp:\n")
tn.write(ipServer.encode('ascii') + b"\n")
tn.write(name.encode('ascii') + b"\n")
tn.write(b"exit\n")
tn.read_all().decode('ascii')
```

## Conclusión

### Cortez Enriquez Jovanny Wilver

Esta práctica que es de las ultimas estuvo laborioso y complicado debido a que no recordaba como realizar las operaciones necesarias de peticiones web y desarrollo del back para que el servidor realizará los cambios necesarios en el servidor. Ayudo a recordar la programación web y recordar cosas de html necesarias. Fue una práctica entretenida llena de conocimiento y de recuerdo de puntos de web development.

### González Lopez Emiliano

El desarrollo de esta practica mas allá de la recordar los protocolos, me hizo recordar el hacer paginas sencillas usando css, JS y php, todo para poder correr nuestros archivos externos, fuera de esto la realización fue sencilla exceptuando la parte de la ejecución externa de los archivos.