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](#_Toc43154757)

[Planteamiento del Problema 4](#_Toc43154758)

[Requisitos 4](#_Toc43154759)

[Desarrollo 5](#_Toc43154760)

[Creación de la arquitectura en GNS3 5](#_Toc43154761)

[Instalación del servidor Apache2 6](#_Toc43154762)

[Configuración del servidor Apache2. 7](#_Toc43154763)

[Crear 2 sitios con la misma IP 7](#_Toc43154764)

[Crear sitio web con autenticación 9](#_Toc43154765)

[Pruebas 11](#_Toc43154766)

[Conclusión 13](#_Toc43154767)

# 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 conocidad 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

Captura de pantalla de un videojuego

Descripción generada automáticamente

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

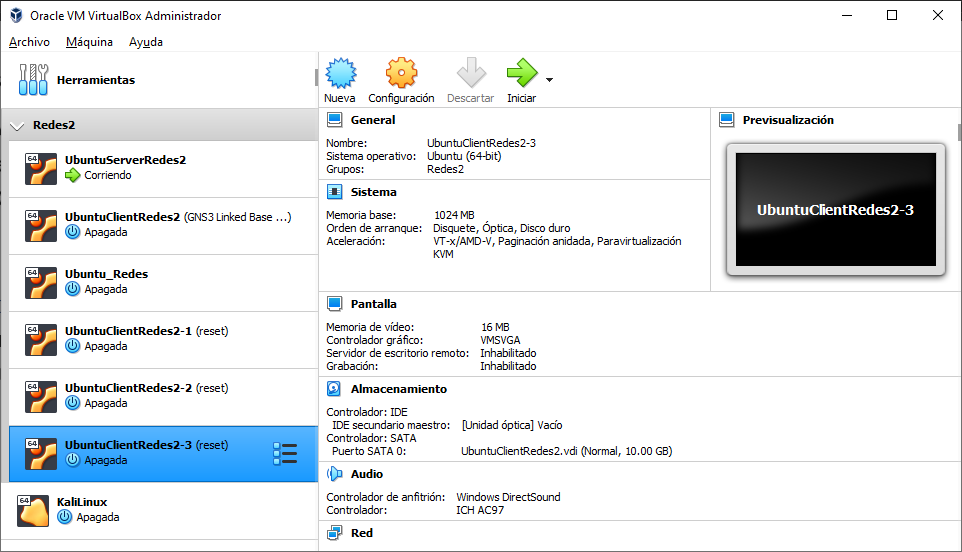


Imagen 2: VirtualBox corriendo la máquina UbuntuServerRedes2

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

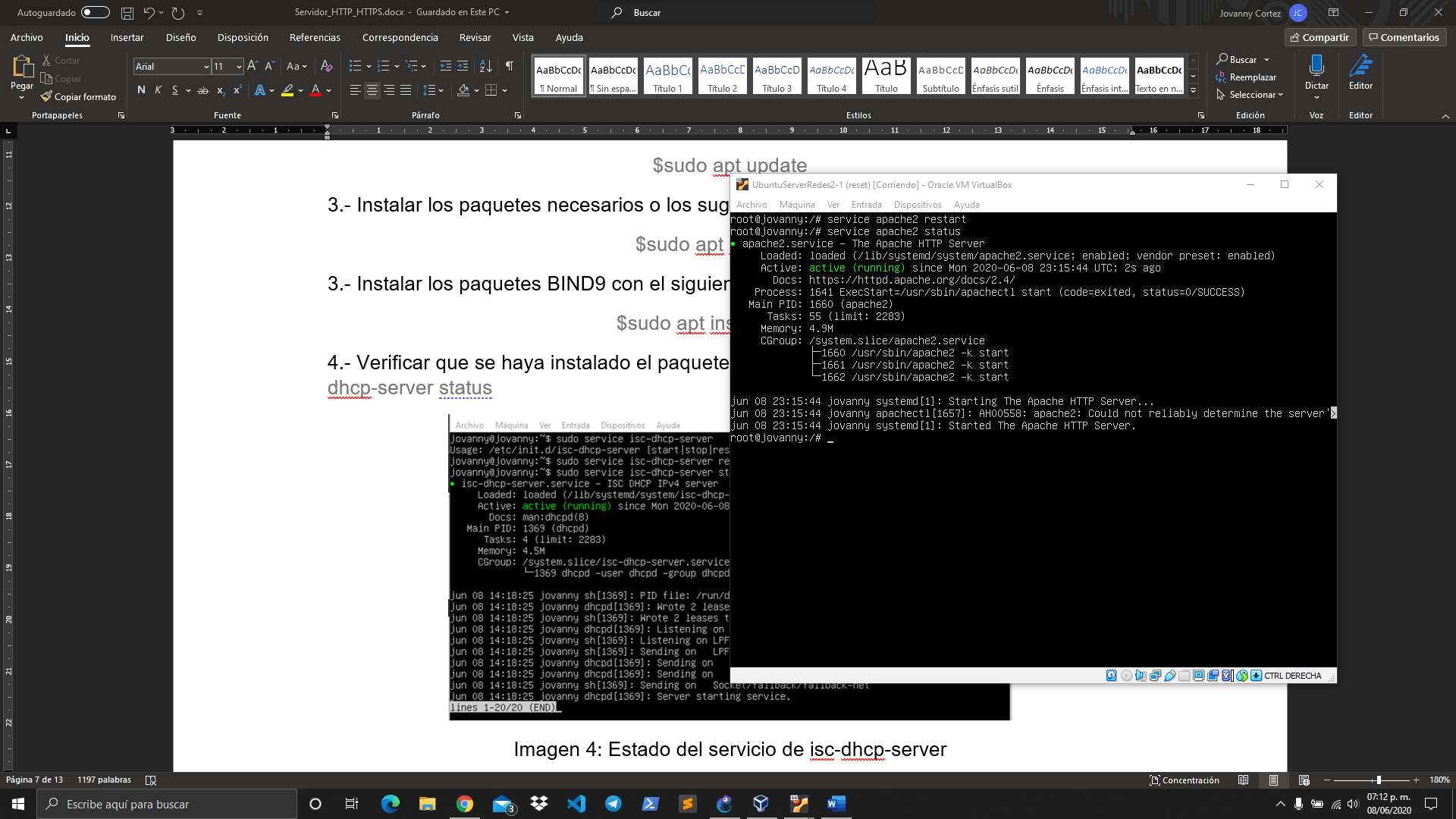


Imagen 3: Estado del servicio de apache2

## 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-enables

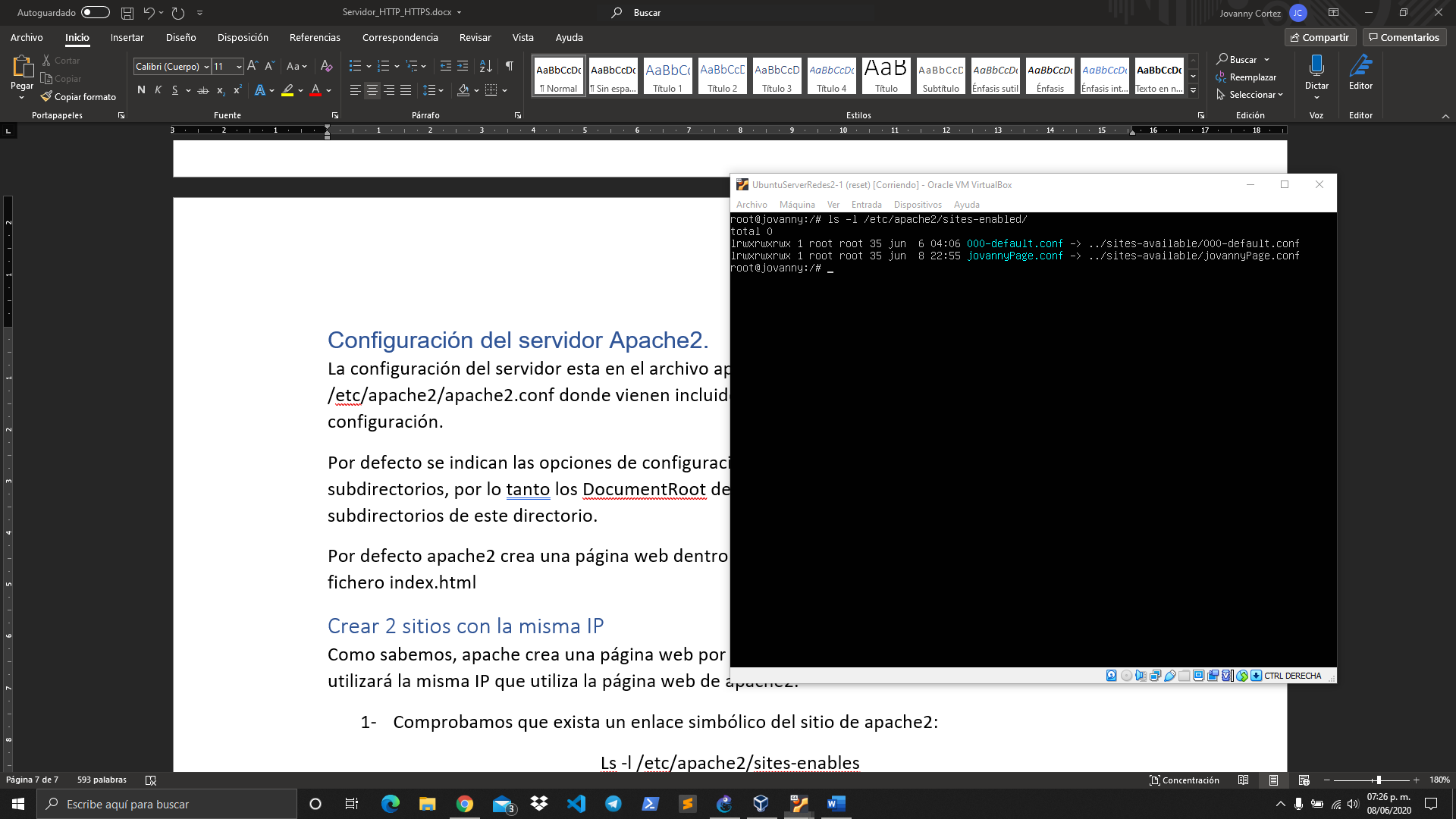


Imagen 4: link simbólico del archivo de configuración

1. 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

1. 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

1. Creamos su enlace simbólico

#a2ensite jovannyPage

1. Creamos el directorio y el fichero index.html

#mkdir /var/www/jovannyPage

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

1. Creamos el contenido html necesario en index.html, como un saludo.
2. Reiniciamos el servicio de http

#service apache2 restart

#service apache2 status

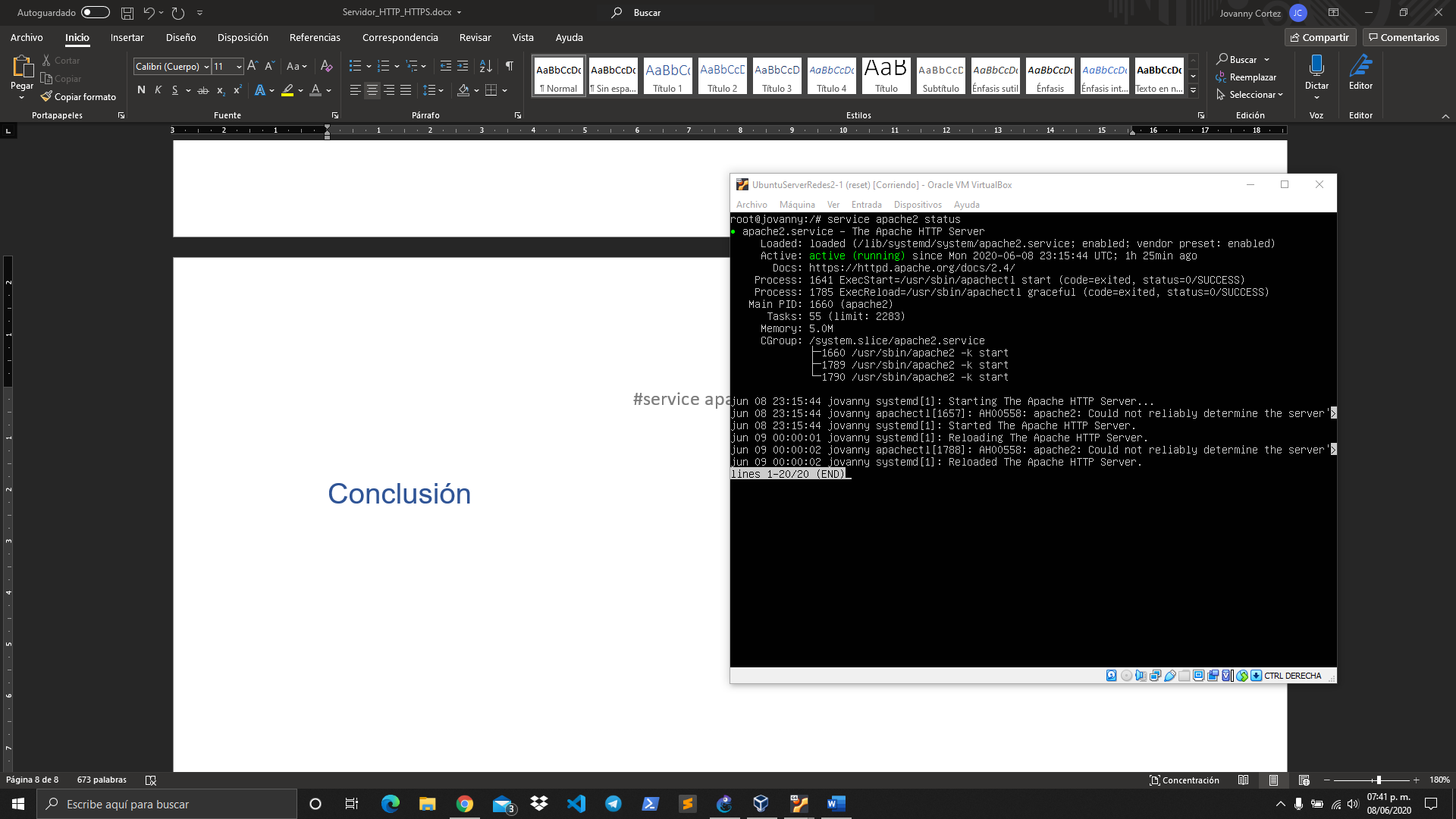


Imagen 5: servicio corriendo.

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

$curl www.jovannyPage.com

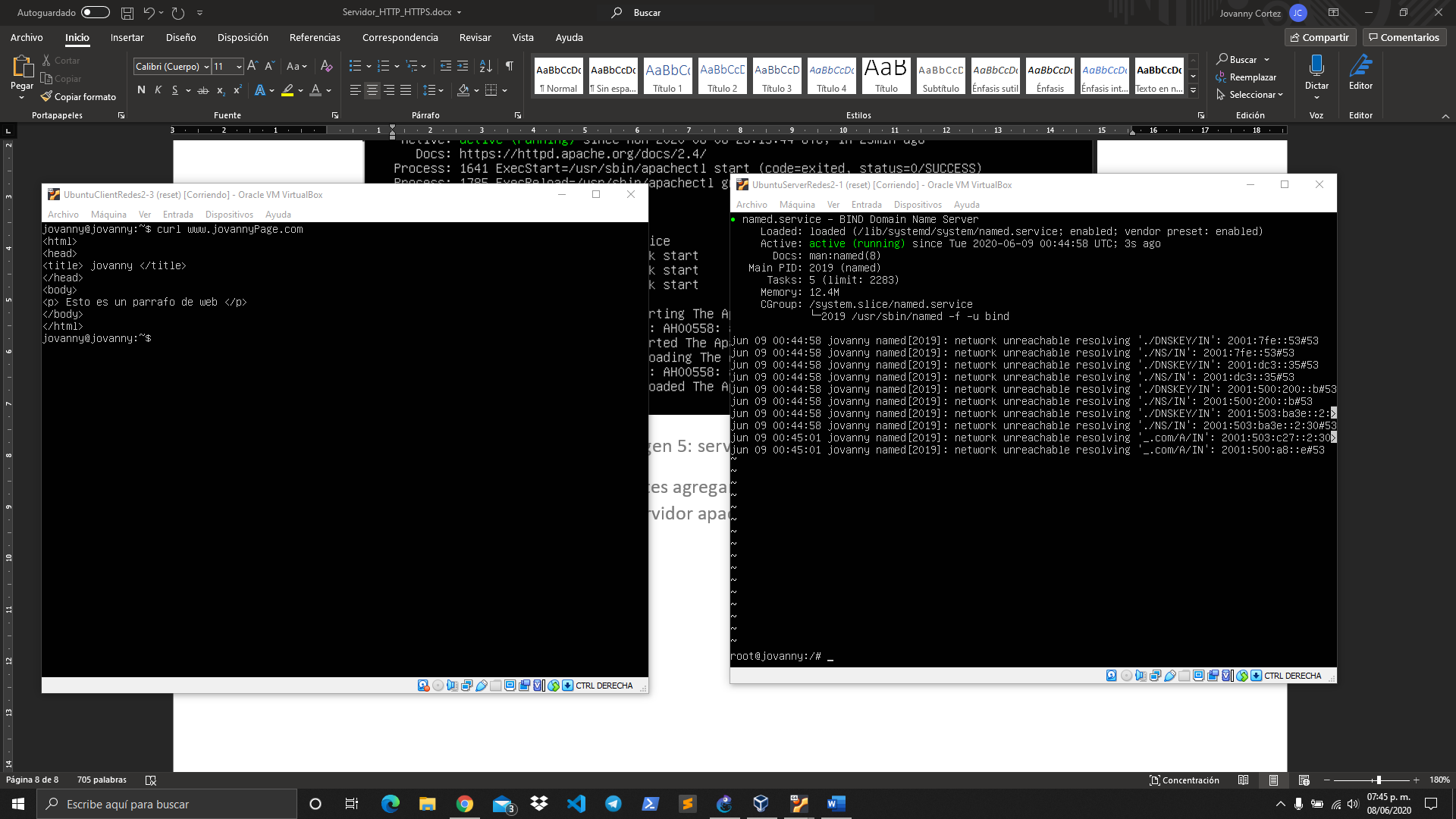


Imagen 6: contenido html del fichero index.html de la página web [www.jovannyPge.com](http://www.jovannyPge.com)

## 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>

1. Creamos el fichero de contraseñas

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

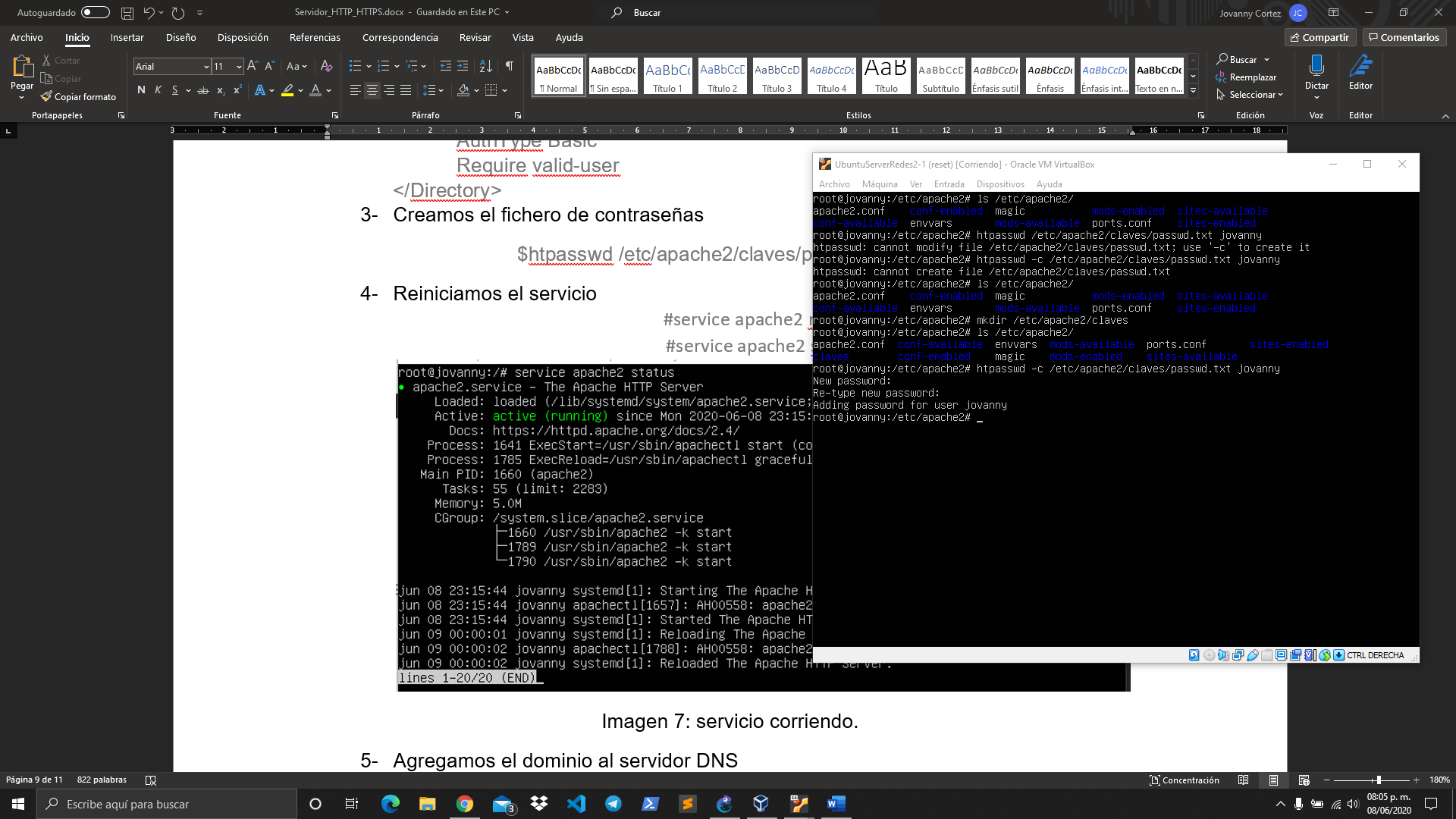


Imagen 7: Creación de usuario con su contraseña

1. Verificamos el fichero creado.

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

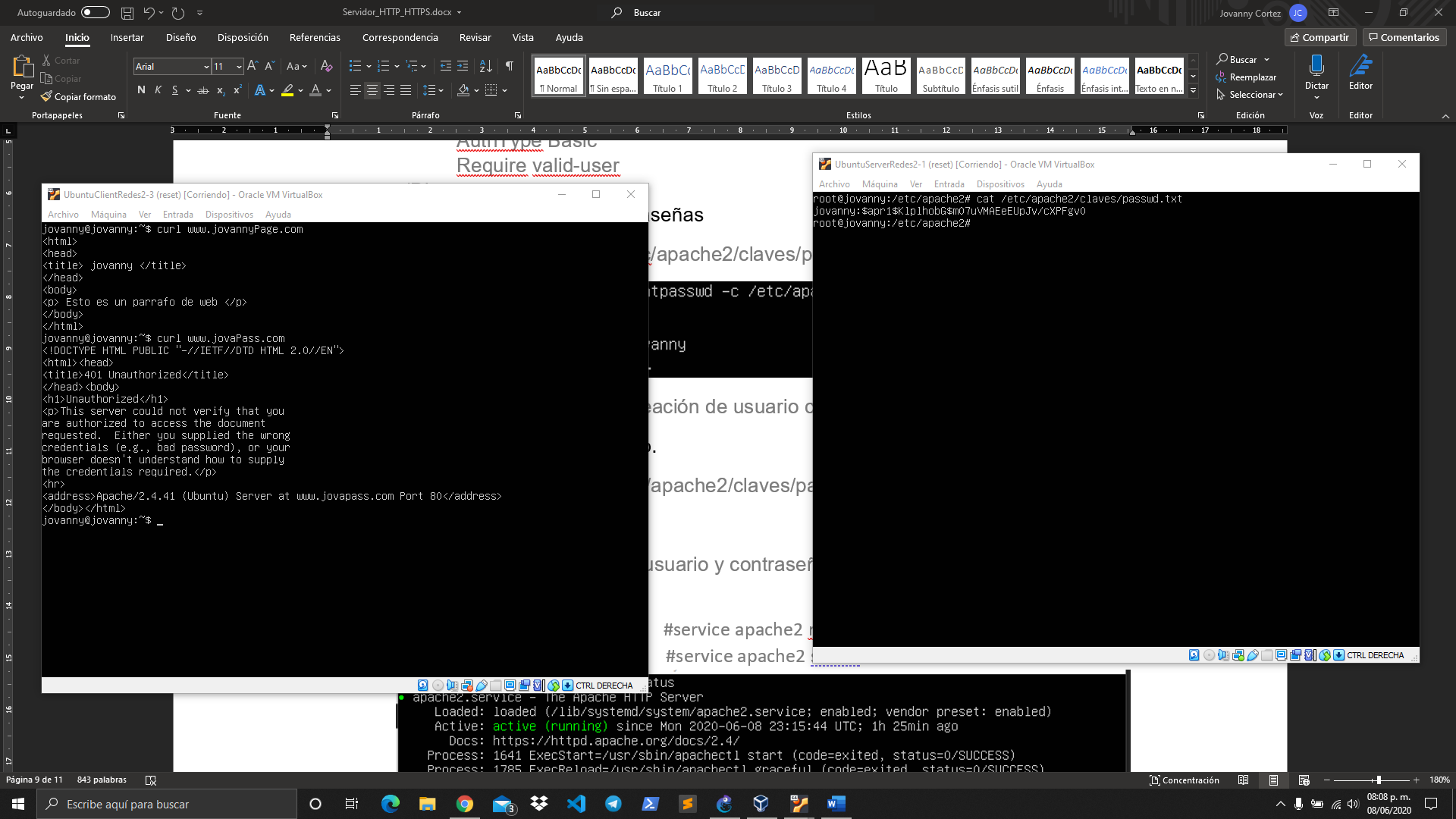


Imagen 8: usuario y contraseña creados.

1. Reiniciamos el servicio

#service apache2 restart

#service apache2 status

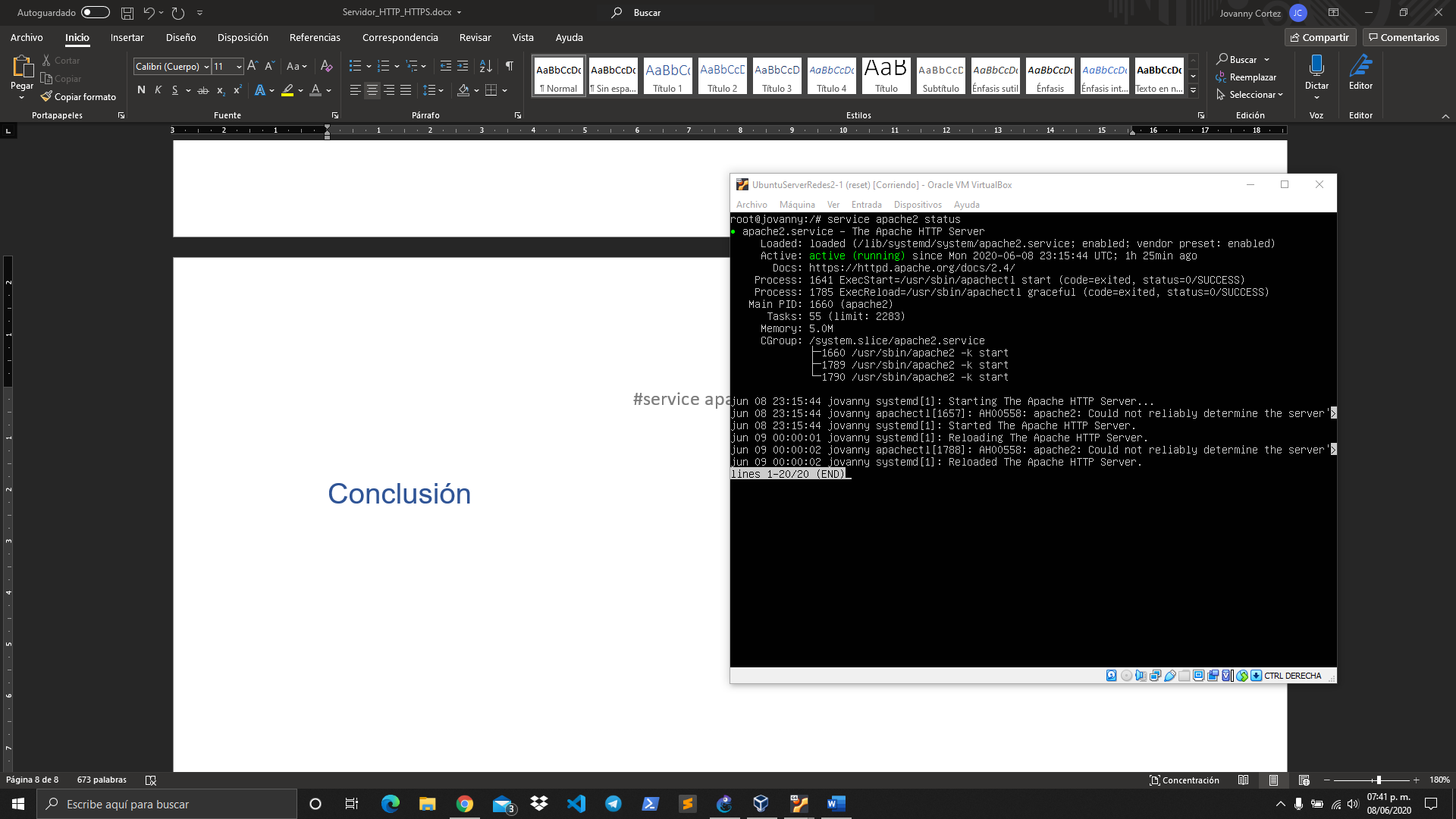


Imagen 9: servicio corriendo.

1. Agregamos el dominio al servidor DNS
2. Probamos desde otra máquina.

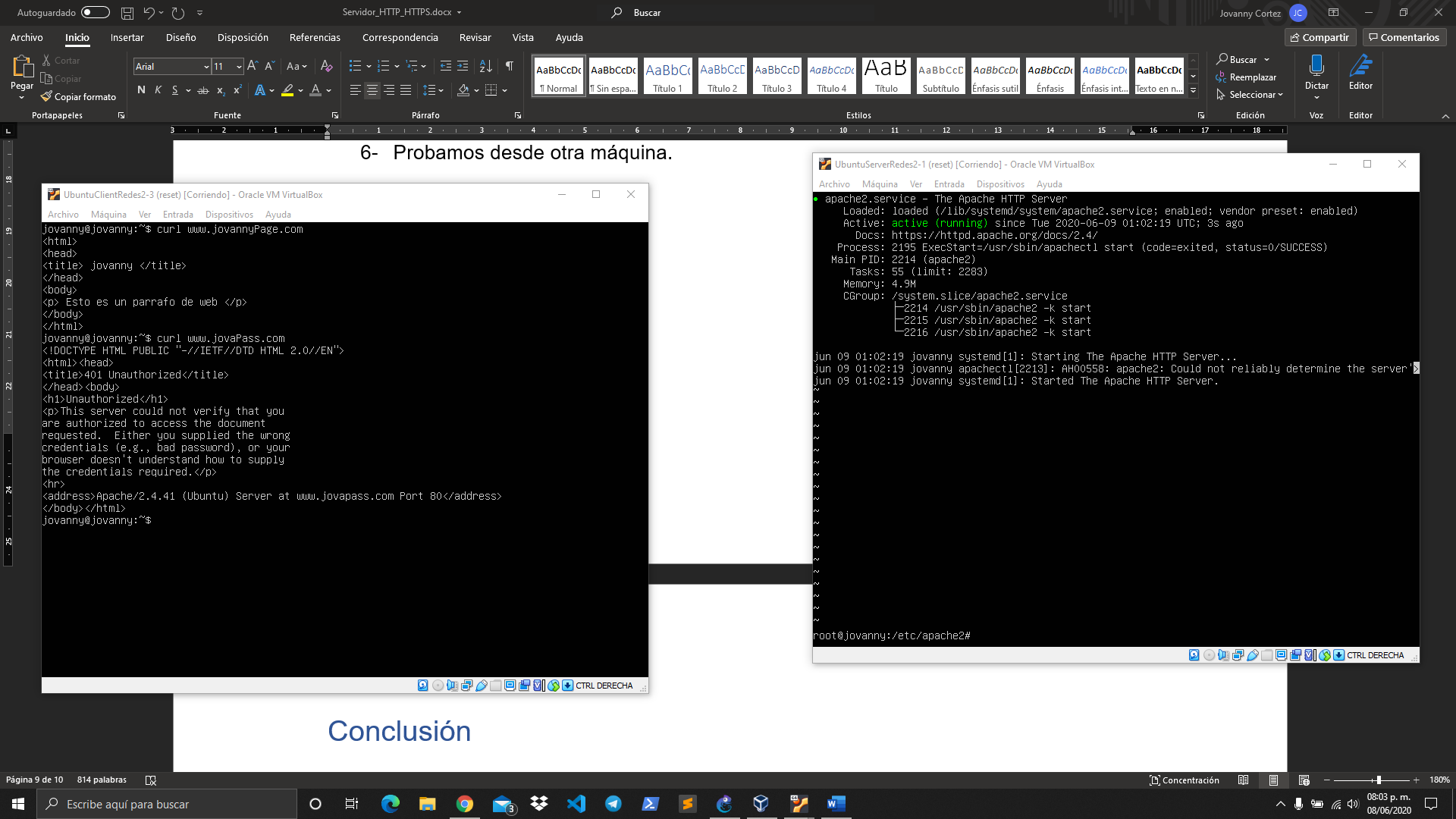


Imagen 10: acceso no autorizado porque requiere autentificación.

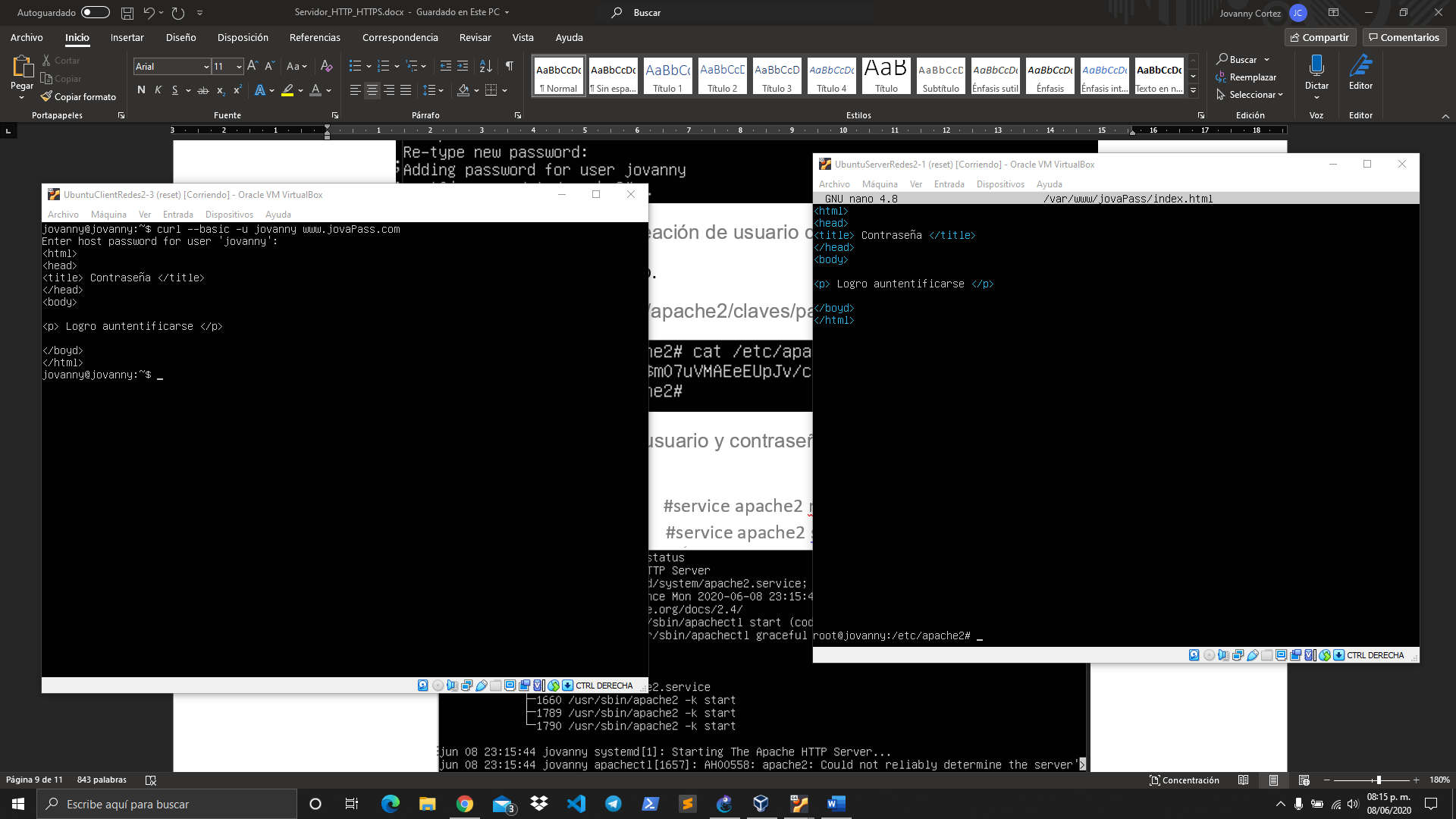
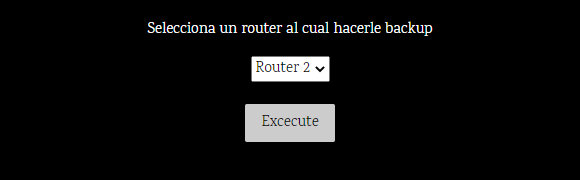


Imagen 11: Autentificación lograda con éxito

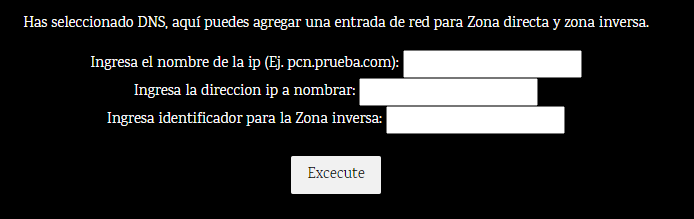
# Pruebas



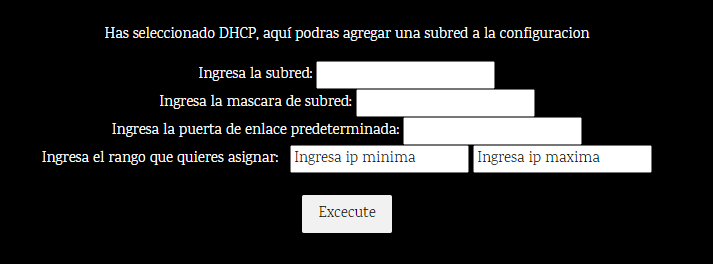
Pagina elaborada



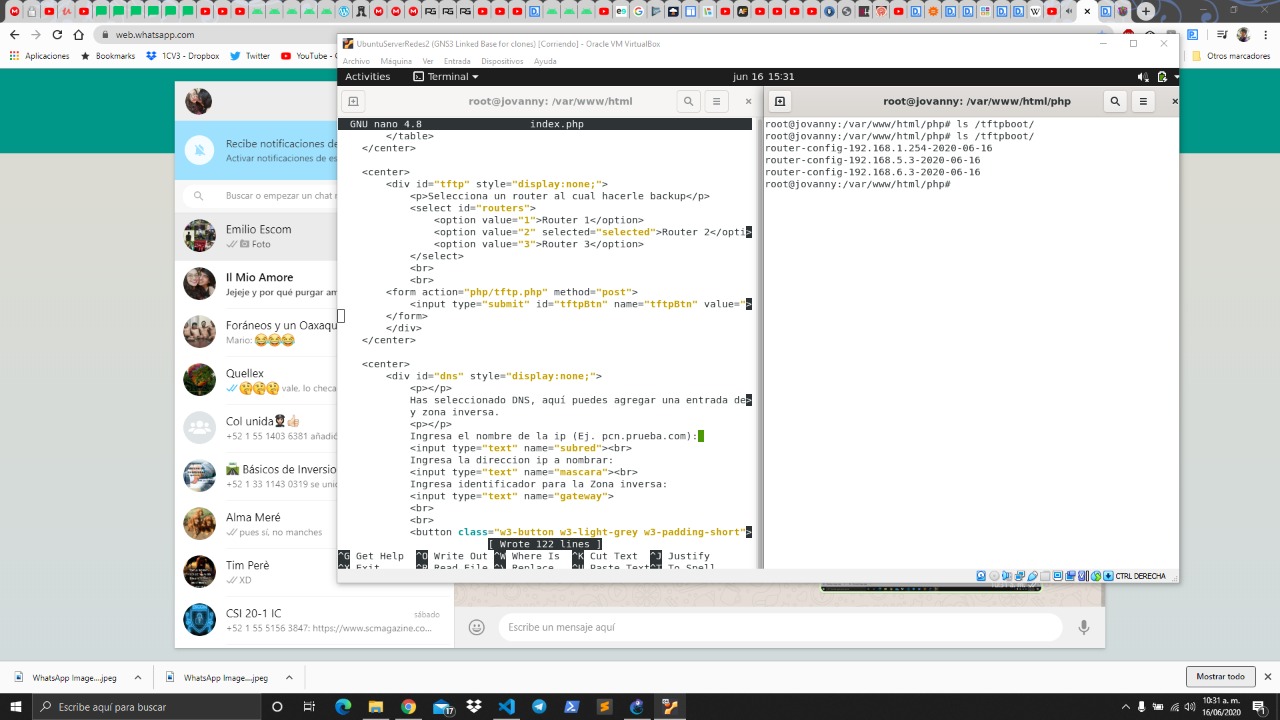
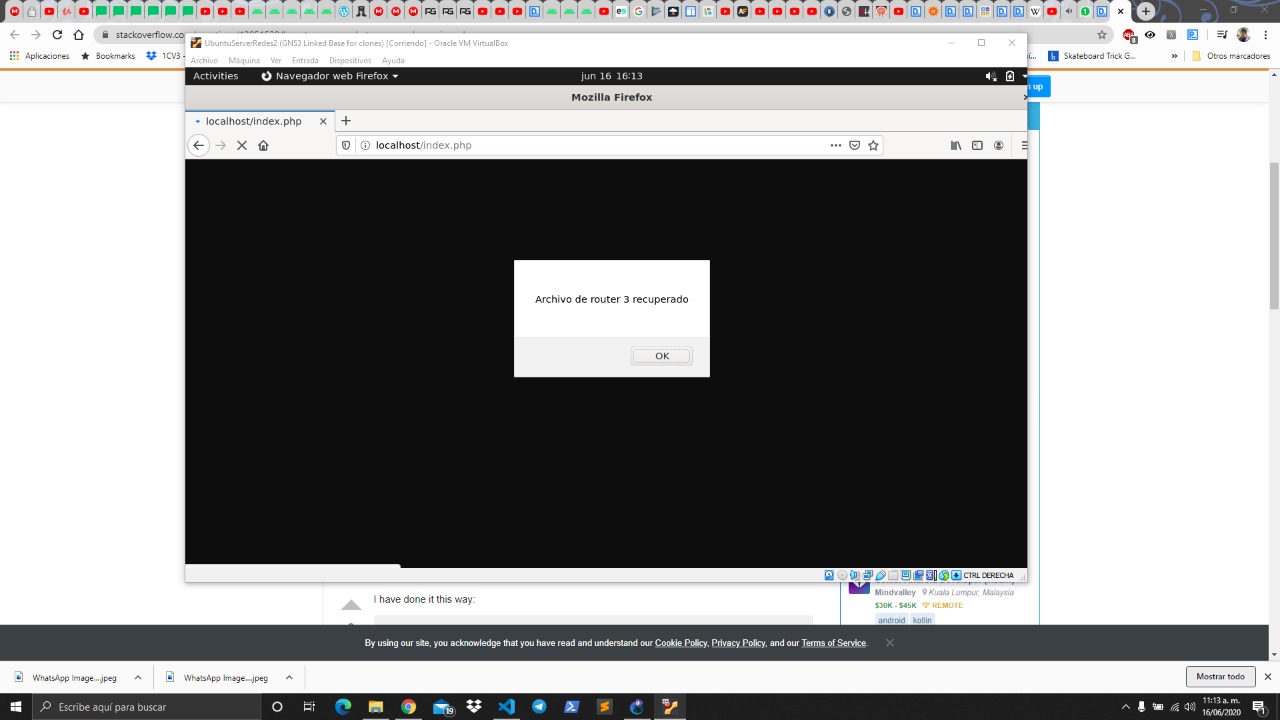
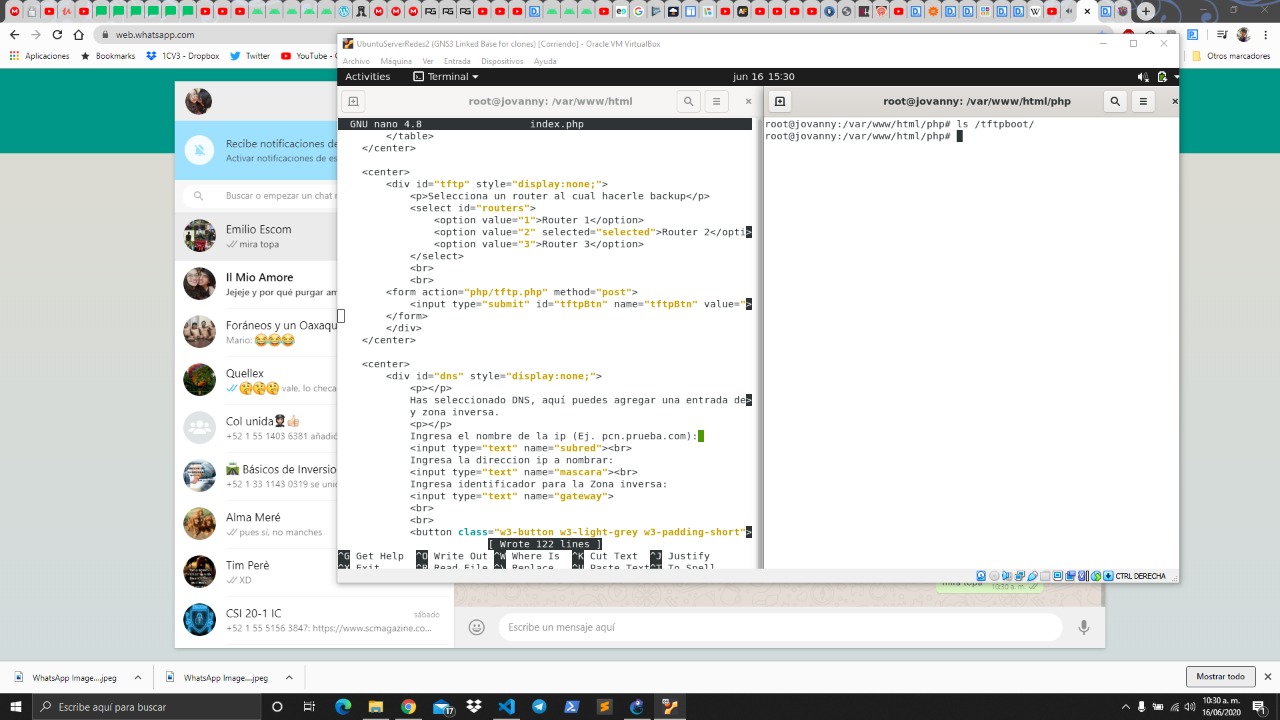
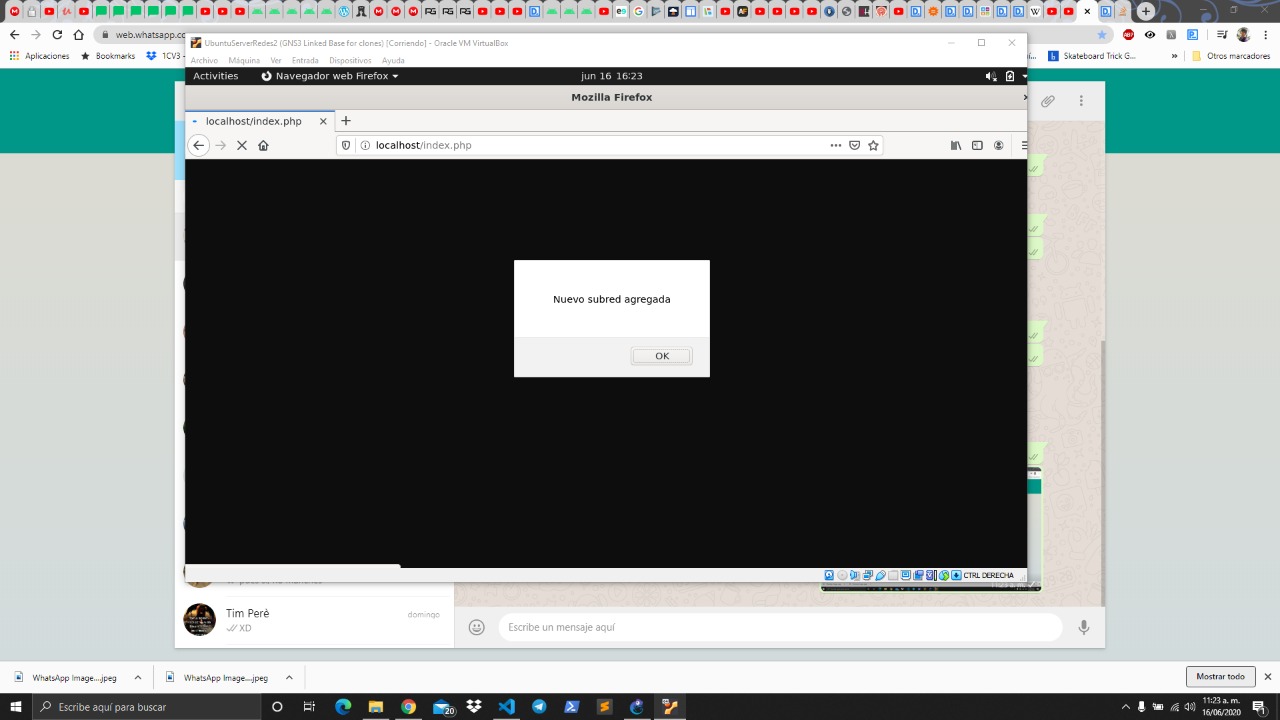
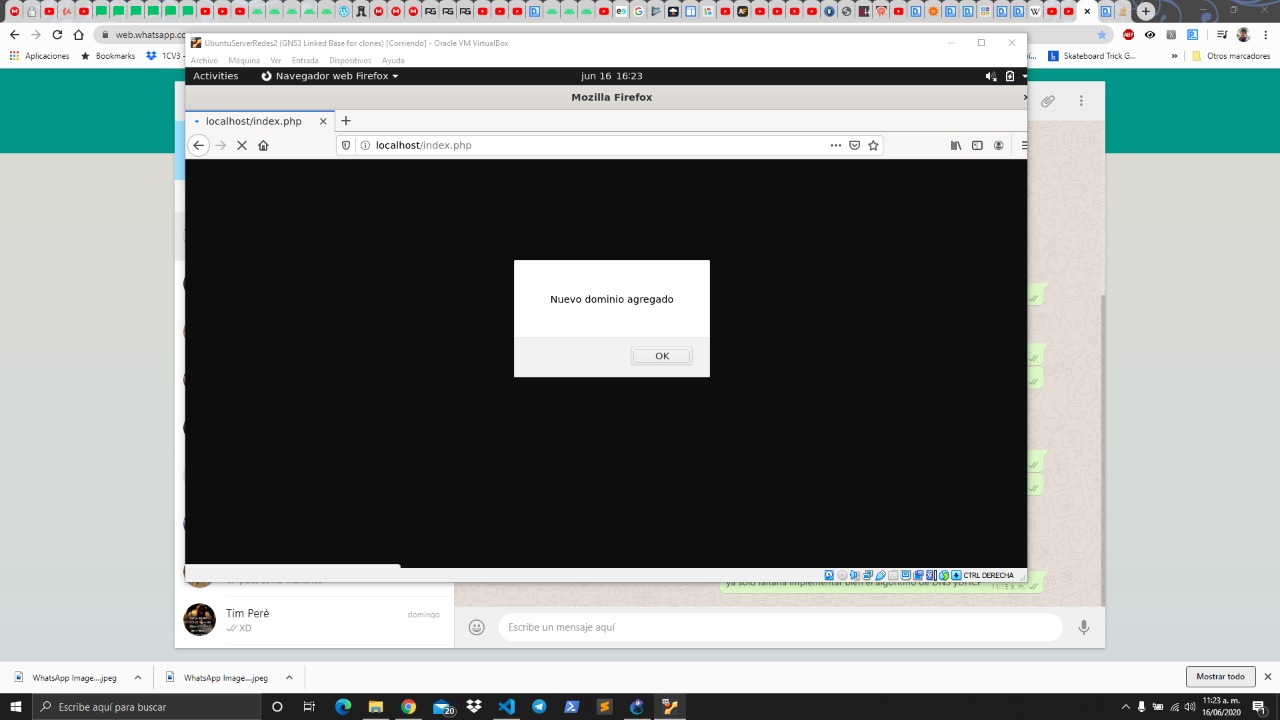
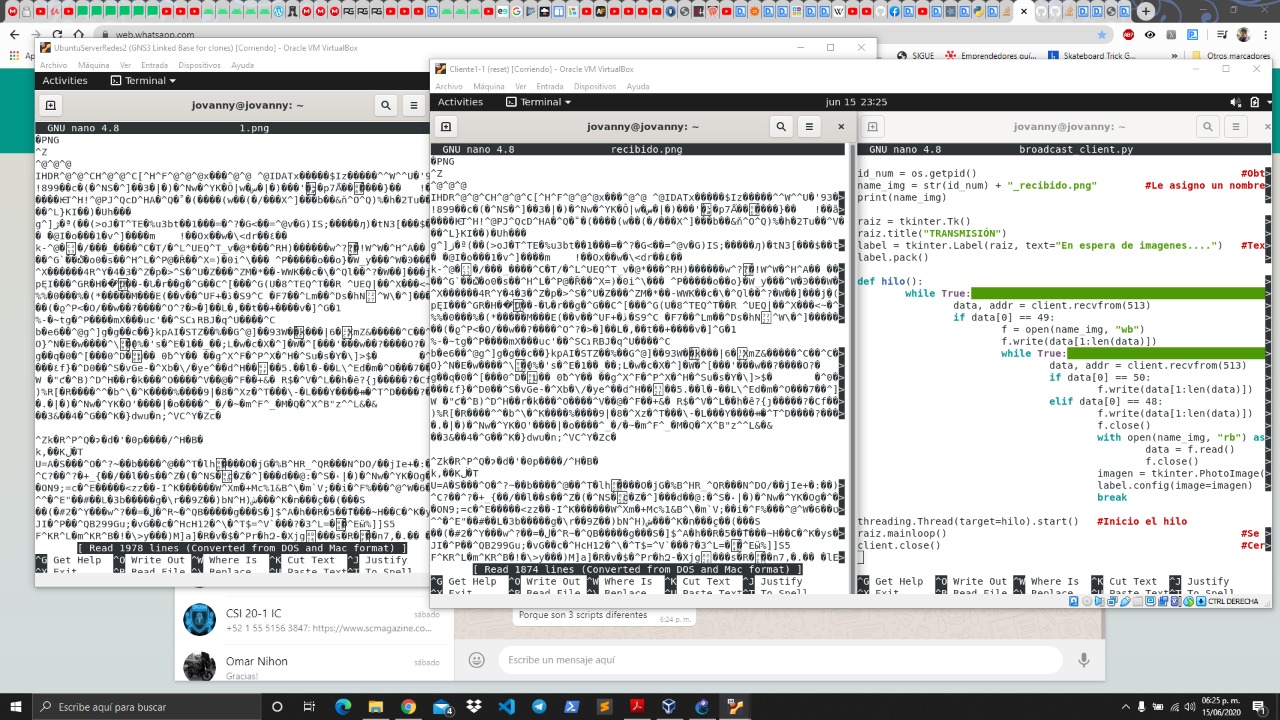
Opción del protocolo TFTP



Opción del protocolo DNS



Opción del protocolo DHCP



# Conclusión

Cortez Enriquez Jovanny Wilver

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.