Instituto Politécnico Nacional

"La Técnica al Servicio de la Patria".

upiita-ipn

UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS AVANZADAS

INGENIERÍA TELEMÁTICA

ADMINISTRACION DE SISTEMAS OPERATIVOS

ESTUDIANTE:

RODOLFO JOSEF ENCANDE NAVARRO

NO° DE BOLETA: 2022640547

GRUPO: 1TM1

CATEDRÁTICO:

MENDOZA CORTÉS HÉCTOR

TOLUCA, ESTADO DE MÉXICO, 17/06/22

Contenido

PROTOCOLO DE SSH

SSH es un protocolo que garantiza que tanto el cliente como el servidor remoto intercambien informaciones de manera segura y dinámica. El proceso es capaz de encriptar los archivos enviados al directorio del servidor, garantizando que las alteraciones y el envío de datos sean realizados de la mejor forma. El desarrollo o mantenimiento de sitios web son demandas comunes en cualquier empresa que trabaje con estos servicios. Mantener la seguridad de los procesos es una de las obligaciones de los profesionales y, para eso, es necesario usar los recursos principales. El protocolo SSH es uno de los parámetros de trabajo que garantizan que las informaciones estarán debidamente protegidas. La comunicación de un computador con un servidor es una actividad recurrente durante estas demandas de gestión de un sitio.

¿Qué es el protocolo SSH?

SSH es una sigla, o acrónimo, para el término secure shell, que significa cápsula segura. En la práctica, el protocolo SSH es un mecanismo de seguridad ofrecido por los servicios de hospedaje. Su función es garantizar que haya una conexión segura entre el computador y el servidor remoto, garantizando la transferencia de datos sin ninguna pérdida de información. El SSH tiene la función de permitir a los usuarios y desarrolladores realizar cualquier modificación en sitios y servidores utilizando una conexión simple.

De esa forma, por medio de un computador conectado a internet, esa persona puede configurar, modificar archivos e incluso trabajar en el desarrollo de una página web. La propuesta de ese protocolo es justamente crear un método seguro, que garantice que no habrá ninguna invasión de esos archivos y de sus códigos.

Por eso, son usadas criptografías que garantizan que solamente dos puntos accedan a las informaciones: el servidor y el computador que envió los datos para ese local remoto.

El funcionamiento del protocolo SSH

En la práctica, el SSH ofrece un mecanismo para que haya la autenticación de ese usuario remoto, garantizando que esa persona tenga autorización para comunicarse con el servidor. De esa manera, es creada la conexión por medio del protocolo y las informaciones son transportadas en ese modelo de secure shell, con la criptografía que protege los datos. El SSH es accedido vía terminal, independientemente del sistema operacional usado, y entonces es desarrollada la criptografía que va a proteger las informaciones. Por medio de la ventana es hecha la conexión con el servidor remoto, y entonces el proceso se desarrolla.

¿Cuándo necesita ser accedido el protocolo SSH?

El SSH es un recurso utilizado en momentos específicos y para trabajos que son de rutina para los programadores y desarrolladores. Desde las pruebas hasta las alteraciones, cuando el sitio ya está listo, hay diversas etapas en que es necesario crear una conexión segura entre el punto de acceso y el servidor remoto.

A continuación, entiende mejor cuándo el protocolo es utilizado y descubre de qué manera se hace relevante en cada una de esas ocasiones.

Programación

La programación es una etapa de trabajo que está relacionada con la creación de un sitio web. Programadores desarrollan códigos, hacen alteraciones y necesitan probar de qué manera todo ese desarrollo se comporta con la aplicación al aire. Para eso, es necesario transferir los datos para el servidor y entonces analizar el

comportamiento de las páginas en línea. Ese procedimiento puede ser hecho incluso durante la instalación de un CMS, como WordPress, así es posible visualizar si los comandos enviados del panel son activados correctamente. Durante todo el proceso, por varias veces, el intercambio de informaciones con el servidor remoto deberá ser hecho, lo que justifica la necesidad de mantener la seguridad de los códigos. esa información es la garantía de que la estructura construida para aquel sitio no será indebidamente desviada. Esto protege la propiedad intelectual del profesional y la inversión del cliente.

Deploy

El deploy es un proceso muy común en la rutina de los desarrolladores y se trata de un trabajo de actualización de sitio, que generalmente es compuesto por cambios o por nuevas aplicaciones implementadas en él.

El deploy es un trabajo más largo y que envuelve, la mayoría de las veces, la trasferencia de un alto nivel de archivos, lo que también requiere un método seguro. Por eso, el uso del SSH es la mejor manera de conducir ese trabajo. El protocolo acostumbra ser una opción recurrente de los profesionales responsables por cuidar de esas tareas.

Por medio del Secure Shell, hay garantías de que la trasferencia de las nuevas aplicaciones y de las alteraciones sea hecha en el tiempo adecuado, sin fallas y sin pérdida de informaciones.

Configuración en Linux del servidor ssh

MicroStrategy usa el SSH incluido con su SO Linux por defecto. Si SSH no se está ejecutando en el momento de la instalación de MicroStrategy, el instalador *no* habilite la función para iniciar y detener servicios en la vista Topología de la estación de trabajo.

Para configurar SSH en Linux:

1. Instale OpenSSH abriendo una terminal y ejecutando los siguientes comandos con permisos de superusuario.

   En Ubuntu/Debian/Linux Mint:

   ```
   # apt-get install openssh-server openssh-client openssh
   ```

   En RHEL/Centos/Fedora:

   ```
   # yum -y instalar openssh-server openssh-clients openssh
   ```

2. Inicie el servicio escribiendo los siguientes comandos en la terminal:

   ```
   # chkconfig sshd on # service sshd start
   ```

3. Si tiene un firewall, abra el puerto SSH en su firewall. Por ejemplo, el puerto 22.
4. Navegue a/opt/MicroStrategy/ServicesRegistration/yaml/y abra el **installation_list.yaml** archivo.
5. Realice las siguientes modificaciones:
   - Modificar **"CommonPath"** al directorio de instalación de MicroStrategy Common Files. Por defecto, es/var/opt/MicroStrategy.
   - Modificar **"InstallType"** a 1.
   - Modificar **"Puerto"** para usar el número de puerto de su servidor SSH.
   - Modificar **"versión"** para usar su número de versión de MicroStrategy.

```
6. --- servicio:
7.   Nombre: ID de &amp;quot;servidor SSH&amp;quot;: CommonPath &amp;quot;SSH-Serve
r&amp;quot;: /var/opt/MicroStrategy InstallType: 1 puerto: 22 Etiquetas:
   &amp;quot;versión&amp;quot;: &amp;quot;11.2.0000.0123&amp;quot;
```

6. Vaya a/opt/MicroStrategy/ServicesRegistration/jar y ejecute el siguiente comando para generar un nuevo archivo JSON para el servidor SSH.

```
# java -jar svcsreg-admin.jar parse SSH-Server
```

El archivo JSON recién generado se encuentra en/opt/MicroStrategy/ServicesRegistration/config.

7. Reinicie el registro de MicroStrategy Services con el siguiente comando:

```
# java -jar<MSTR_INSTALL_PATH> /ServicesRegistration/jar/svcsreg-admin.jar control consul restart
```

8. Abra la vista Topología en Workstation. Ahora puede iniciar y detener servicios.

Configuración mínima para garantizar la seguridad de la conexión

Cuando estamos configurando una infraestructura, hacer que las aplicaciones funcionen bien es generalmente nuestra preocupación principal. Sin embargo, hacer que la aplicación funcione sin tomar en cuenta los aspectos de seguridad podría resultar en consecuencias catastróficas. En este artículo hablaremos sobre algunos conceptos que te ayudarán a agregar seguridad a tu aplicación.

Llaves SSH

Las llaves SSH son un par de llaves criptográficas que son utilizadas para autenticar usuarios en un servidor SSH y se utilizan como una alternativa al inicio de sesión por medio de contraseña. Una sola llave SSH consta de dos archivos, la llave privada y la llave pública. La llave pública puede ser compartida a través de internet en múltiples servidores. La llave privada debe permanecer secreta y asegurada por el usuario. Para configurar la autenticación SSH, primero deberás contar con las llaves públicas y privadas. En el servidor donde deseas autenticarte deberás colocar la llave pública mientras que la llave privada la debes mantener en tu equipo local. El cliente SSH utilizará la llave privada y la comparará con la llave pública y de esta forma determinará si la autenticación es correcta.

Luego de esto se deberá configurar el servidor para utilizar SSH como método de autenticación y una vez configurado tenemos que deshabilitar el acceso por contraseña. Hacer esto incrementará la seguridad de tu servidor evitando que usuarios no autorizados se conecten.

¿Cuál es la diferencia del comando adduser a useradd?

useradd es un comando que ejecuta un binario del sistema, mientras que *adduser* es un script en perl que utiliza el binario *useradd*.

La mayor ventaja del comando *adduser* es que crea el directorio home (*/home/usuario/*) del usuario de manera automática, cosa que no hace *useradd* (hay que usar la opción -m). Sin embargo, como no es un comando del core de GNU/Linux, es posible que no funcione bien en todas las distribuciones que uses. Por ello es recomendable usar el comando *useradd*, porque funciona igual en todas las distribuciones.

Variantes de sintaxis del comando adduser

Unos ejemplos son:

- Crear usuario (incluido directorio home del usuario */home/usuario1*)

```
sudo useradd -m usuario1
```

- Borrar usuario (incluido directorio home del usuario */home/usuario1*)

```
sudo userdel -r usuario1
```

- Crear usuario (sin incluir el directorio home del usuario */home/usuario1*)

```
sudo useradd usuario1
```

- Borrar usuario (sin incluir el directorio home del usuario */home/usuario1*)

```
sudo userdel usuario1
```

- Crear usuario (incluido directorio home del usuario */home/usuario1*)

```
sudo adduser usuario1
```

- Borrar usuario (sin incluir el directorio home del usuario */home/usuario1*)

```
sudo deluser usuario1
```

Protocolo web, servidor apache2, iis

Orientación de red principal para Windows Server

Se aplica a: Windows Server 2022, Windows Server 2019, Windows Server, Windows Server 2016. En este tema se proporciona información general sobre la guía de red principal para Windows Server® 2016 y contiene las secciones siguientes.

Introducción a la red principal de Windows Server

Guía de red principal para Windows Server

Introducción a la red principal de Windows Server

Una red principal es una colección de hardware, dispositivos y software de red que proporciona los servicios fundamentales para satisfacer las necesidades de las tecnologías de la información (TI) de la organización.

Una red principal de Windows Server le ofrece muchas ventajas, entre las que se incluyen las siguientes.

Protocolos principales para la conexión de red entre equipos y otros dispositivos compatibles con Protocolo de control de transmisión/Protocolo de Internet (TCP/IP). TCP/IP es un conjunto de protocolos estándar pensado para conectar equipos y crear redes. TCP/IP es el software de protocolo de redes suministrado con los sistemas operativos de Microsoft® Windows® que implementa y admite el conjunto de protocolos TCP/IP.

Direccionamiento IP automático de servidor de Protocolo de configuración dinámica de host (DHCP). La configuración manual de direcciones IP en todos los equipos de la red es una tarea que consume mucho tiempo y es menos flexible que la opción de proporcionar dinámicamente a equipos y otros dispositivos concesiones de direcciones IP desde un servidor DHCP.

Servicio de resolución de nombres del Sistema de nombres de dominio (DNS). Con DNS, los usuarios, equipos, aplicaciones y servicios pueden usar el nombre de dominio completo de un equipo o un dispositivo para encontrar la dirección IP de dicho equipo o dispositivo.

Un bosque, que es uno o más dominios de Active Directory que comparten las mismas definiciones de clase y atributo (esquema), información de sitio y replicación (configuración) y capacidades de búsqueda para todo el bosque (catálogo global).

Un dominio raíz del bosque, que es el primer dominio creado en un nuevo bosque. Los grupos Administradores de empresas y Administradores de esquema, que son grupos administrativos para todo el bosque, se encuentran en el dominio raíz del

bosque. Además, un dominio raíz del bosque, como los demás dominios, es una colección de objetos de equipo, usuario y grupo definidos por el administrador en Servicios de dominio de Active Directory (AD DS). Estos objetos comparten una base de datos de directorios común y directivas de seguridad. También comparten relaciones de seguridad con otros dominios, si se agregan dominios a medida que la organización crece. El servicio de directorio también almacena datos de directorio y permite que los equipos, aplicaciones y usuarios autorizados tengan acceso a los datos.

Una base de datos de cuentas de usuario y equipo. El servicio de directorio proporciona una base de datos de cuentas de usuario centralizada que le permite crear cuentas de usuario y equipo para las personas y equipos que están autorizados para conectarse a la red y tener acceso a recursos de red, como aplicaciones, bases de datos, carpetas y archivos compartidos e impresoras. Una red principal también le permite escalar la red a medida que crece la organización y cambian los requisitos de TI. Por ejemplo, con una red principal puede agregar dominios, subredes IP, servicios de acceso remoto, servicios inalámbricos y otras características y roles de servidor proporcionados por Windows Server 2016.

Servidor apache2

Apache en Windows, paso por paso

Descarga desde Apache.org la última versión para Windows, puedes utilizar el siguiente enlace: Descargar Apache. Crea dos carpetas en la unidad C, la primera de nombre "Apache" y la segunda "servidor_web".

Descomprime el archivo descargado y ejecútalo, sigue los pasos de la instalación y de los datos que te piden solo escoge el destino de la instalación, que será la carpeta que creaste en C:\Apache, los otros datos déjalos de la forma predeterminada para configurarlos más tarde.

El programa al instalarse crea un icono en el área de notificación que te permitirá: iniciar, detener y reiniciar Apache. Tienes que tener en cuenta que cualquier cambio que hagas en el archivo de configuración no tendrá efecto hasta que reinicies el servidor.

¿Cómo configurar el Servidor Apache

Toda la configuración para el funcionamiento de Apache se guarda en un archivo de texto nombrado: httpd.conf que se encuentra en la ruta C:\Apache\conf, lo podemos editar en cualquier editor de texto como el Bloc de notas pero un programa recomendado es Notepad++, software libre que es inmejorable.

Puedes descargar Notepad++ desde aquí.

Tienes dos opciones a continuación:

1- Primera opción, la más sencilla, descarga en el siguiente link una copia del archivo httpd.conf, descomprímelo, cópialo o muévalo a la carpeta C:\Apache\conf y sustituye el archivo original, ya tendrás listo para funcionar el servidor.

2- La otra opción, más avanzada pero no difícil, abre el archivo httpd.conf y edita manualmente las líneas que se indican:

Todas las líneas que comienzan con el símbolo # son comentarios, explican en cada sección las distintas opciones pero se encuentran en ingles.

La línea 52 Listen indica el puerto y dirección IP por el que el servidor va a recibir las peticiones, puedes usarla de las siguientes maneras:

1- El servidor va recibir peticiones solo de la misma PC: Listen localhost:80

2- Recibirá peticiones de otras máquinas en una red local: Listen 80

En la línea 149 DocumentRoot es necesario especificar la ruta de la carpeta local que contendrá las páginas y archivos a servir, en tu caso será la carpeta que creaste en C:/servidor_web, quedaría de la siguiente forma:

DocumentRoot "C:/servidor_web"

La línea 177 <Directory> establece los permisos necesarios al directorio anterior, quedaría:

<Directory "C:/servidor_web">

Esta es la configuración con los parámetros esenciales para comenzar a utilizar Apache. Guarda los cambios realizados y reinicia el servidor dando clic en el icono del área de notificación.

Configuración en Linux del servidor apache

Edite el archivo de configuración del servidor de Apache

Localice el archivo http-vhosts.conf de Apache en el directorio extras de la instalación del servidor.

cd <apache-dir>/conf/extras/

vi httpd-vhosts.conf

Edite el archivo http-vhosts.conf de Apache. Para añadir información sobre Build Forge a httpd-vhosts.conf, añada las siguientes líneas:

<VirtualHost *:80>

ServerAdmin build@sudominio.com

DocumentRoot /opt/buildforge/webroot/public

ServerName ausbuild01.sudominio.com

ServerAlias build.sudominio.com mc.sudominio.com

ErrorLog logs/ausbuild.error_log

CustomLog logs/ausbuild.access_log common

</VirtualHost>

Modifique el valor de DocumentRoot para que señale la aplicación web de Build Forge. En este ejemplo, el directorio de instalación de Build Forge es /opt/buildforge.

Deje el puerto como 80 o cámbielo al puerto en el que se ejecute Apache HTTP Server localmente.

<VirtualHost *:80>

Importante: No utilice el puerto 8080; es el puerto predeterminado para Apache Tomcat.

Modifique cualquier otro valor de http-vhosts.conf según corresponda para Apache HTTP Server:

ServerAdmin: dirección de correo electrónico del administrador de Build Forge

DocumentRoot: ubicación de la página de entrada para la aplicación Build Forge

ServerName: servidor donde está instalada la aplicación Build Forge

ServerAlias: alias opcionales para el URL ServerName de Build Forge

ErrorLog: registro de errores de Apache para la aplicación Build Forge

CustomLog: registro de errores de Apache para registrar el acceso a la aplicación Build Forge

Instalar y configurar PHP para Apache HTTP Server

PHP no se instala con HTTP Server de Apache. Debe instalar PHP 5.2.4 y configurarlo para que señale el archivo httpd-vhosts.conf para Apache HTTP Server.

Instalar y configurar PHP para la base de datos de Build Forge

Durante la instalación de PHP, seleccione e instale las extensiones de PHP para el tipo de base de datos que utilice como base de datos de Build Forge.

(Opcional) Configure el módulo OpenSSL de PHP para dar soporte al cifrado de contraseña

Para dar soporte a SSL, Build Forge utiliza el módulo OpenSSL de PHP. Este soporte se proporciona con PHP 5.2.4; no se necesita configuración adicional.

Para dar soporte al cifrado de contraseña, se requiere configuración adicional. Se necesita PHP 5.2.4 para dar soporte a esta configuración. Debe localizar los archivos de parche para la extensión OpenSSL, instalarlos en el directorio de OpenSSL y recompilar PHP, de la siguiente forma:

Localice los archivos de parche php_openssl.h y openssl.c en el directorio misc, ubicado en el directorio de instalación de Build Forge, por ejemplo:

UNIX/Linux /opt/buildforge/Platform/misc

Copie los archivos de parche en el directorio openssl, ubicado en el directorio de instalación de Build Forge.

Compile PHP utilizando la opción de configuración --with-openssl=<vía_acceso_a_openssl>, donde <vía_acceso_a_openssl> es el directorio openssl de Build Forge.

Configuración mínima para garantizar la seguridad de la conexión web

Configuración de sshd_config para la máxima seguridad

Cambiar el puerto por defecto del servidor SSH

Por defecto los servidores SSH utilizan el puerto 22 para las conexiones. Es recomendable cambiar este número de puerto, para evitar que bots o cibercriminales puedan intentar iniciar sesión, aunque por sí solo esto no proporciona seguridad, sí podremos pasar desapercibidos a los escaneos masivos desde Internet. Si por ejemplo queremos usar el puerto 22445 debemos poner en el fichero de configuración lo siguiente:

Port 22445

Bloquear el acceso root en las conexiones remotas

Por defecto, cualquier usuario en el sistema operativo que tenga permisos de Shell, podrá iniciar sesión en el servidor. Además, debemos tener en cuenta que si

tenemos activado el usuario root, también podrá conectarse al servidor de forma local o remota, evitando al atacante tener que «adivinar» el nombre de usuario. Por defecto, los bots siempre intentan atacar el puerto 22 y al usuario «root». Desactivando al propio usuario root, y usando «sudo» para elevar a permisos de superusuario, evitaremos esto. Además, OpenSSH también nos permitirá deshabilitar el login del usuario root para dotar al sistema de mayor seguridad:

PermitRootLogin no

De esta manera las conexiones root quedarán bloqueadas evitando que usuarios no autorizados puedan realizar ataques de fuerza bruta contra nuestro servidor SSH para adivinar los credenciales del usuario Root. También tenemos otras opciones en este apartado, como por ejemplo «PermitRootLogin without-password» donde se permite autenticación pero no con usuario y contraseña, sino con claves criptográficas RSA.

Configuraciones de seguridad adicionales

Existen otras configuraciones recomendadas para evitar las conexiones no deseadas a nuestro servidor SSH. Estas conexiones son:

LoginGraceTime: Estableceremos el tiempo necesario para introducir la contraseña, evitando que el atacante tenga que «pensar mucho».

MaxAuthTries: Número de intentos permitidos al introducir la contraseña antes de desconectarnos.

MaxStartups: Número de logins simultáneos desde una IP, para evitar que se pueda utilizar la fuerza bruta con varias sesiones a la vez.

AllowUsers: Es crear una lista blanca de usuario. Este parámetro nos permite configurar los usuarios que podrán conectarse. Una medida muy restrictiva pero a la vez muy segura ya que bloqueará todas las conexiones de los usuarios que no estén en el listado. Los usuarios que tengamos aquí podrán conectarse, y el resto no.

DenyUsers: Parecido al anterior, pero ahora creamos una lista negra. Los usuarios que tengamos aquí no podrán conectarse, y el resto sí.

AllowGroups/DenyUsers: Exactamente igual a lo anterior, pero en lugar de crear una lista blanca/negra de usuarios, es de grupos de usuarios.

Porque utilizamos el puerto 8009 y para qué sirve el puerto 80 y 8080

Puerto 8009: Garantiza la entrega de paquetes de datos en la misma orden, en que fueron mandados. La comunicación garantizada por el puerto TCP 8009 es la diferencia mayor entre TCP y UDP. El puerto UDP no garantizaría la comunicación como TCP.

Puerto 80: Este puerto es el que se usa para la navegación web de forma no segura HTTP.

Puerto 8080: es el puerto alternativo al puerto 80 TCP para servidores web, normalmente se utiliza este puerto en pruebas.

Variantes del servicio apache2

Existen tres perfiles disponibles para Apache:

- **Apache**: este perfil habilita únicamente el puerto 80 (normal, tráfico web sin encriptar).
- **Apache Full**: este perfil habilita dos puertos: puerto 80 (normal, tráfico web sin encriptar) y el puerto 443 (tráfico encriptado mediante TLS/SSL).
- **Apache Secure**: este perfil habilita únicamente el puerto 443 (tráfico encriptado mediante TLS/SSL).

Se recomienda que siempre habilites el perfil con más restricciones dependiendo del tráfico requerido y cómo se ha configurado tu máquina. Como aún no hemos configurado el SSL para nuestro servidor en esta guía, solo permitiremos el tráfico a través del puerto 80:

```
1. sudo ufw allow 'Apache'
2.
```

Se puede verificar el cambio digitando:

```
1. sudo ufw status
2.
```

Se te debería desplegar que el tráfico HTTP se encuentra permitido:

```
Salida

Status: active
```

```
To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
Apache                     ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
Apache (v6)                ALLOW       Anywhere (v6)
```

Como puedes observar, el perfil ha sido activado, y el acceso al servidor web es permitido.

```
entreunosyceros@ubuntu-1804:~$ sudo systemctl start apache2.service
entreunosyceros@ubuntu-1804:~$
```

sudo systemctl start application.service

También podemos **hacer referencia al nombre de la aplicación sin el .service final**. Para **detener el servicio**, el comando a utilizar será algo como:

```
entreunosyceros@ubuntu-1804:~$ sudo systemctl stop apache2.service
entreunosyceros@ubuntu-1804:~$
```

sudo systemctl stop application.service

Porque se creó la carpeta public_html

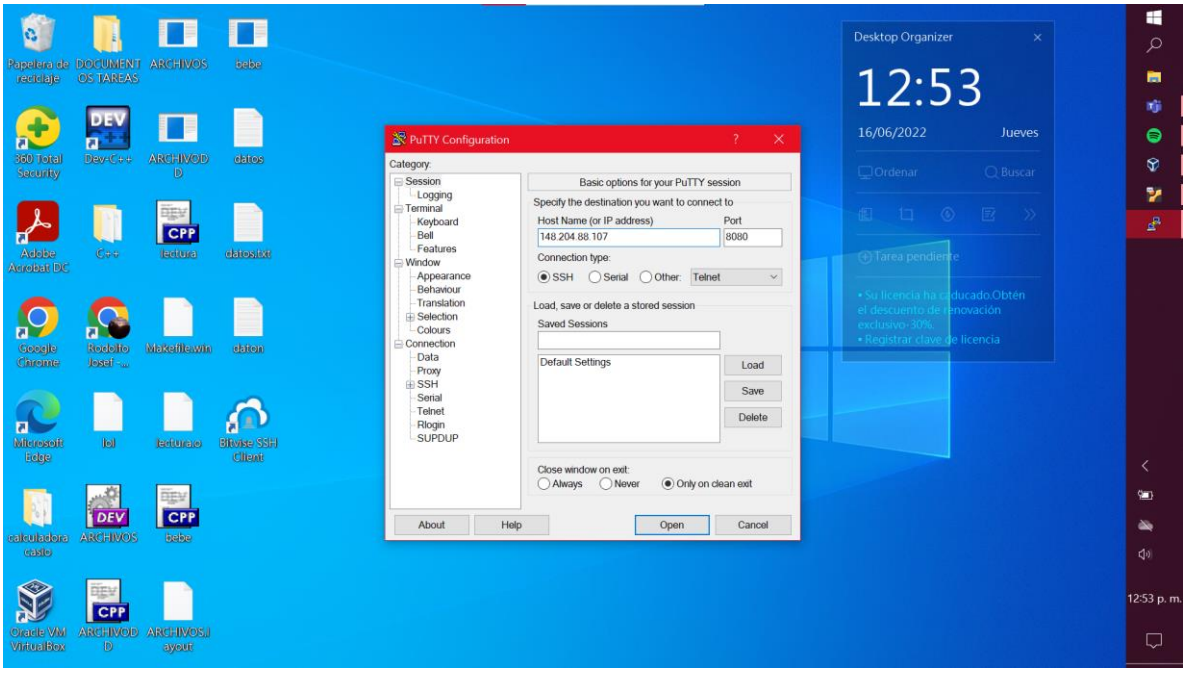
El directorio public_html es la raíz web para el nombre del dominio principal. Esto significa que public_html es la carpeta donde se colocan todos los archivos del sitio web que se desea aparezcan cuando alguien escribe el dominio principal.

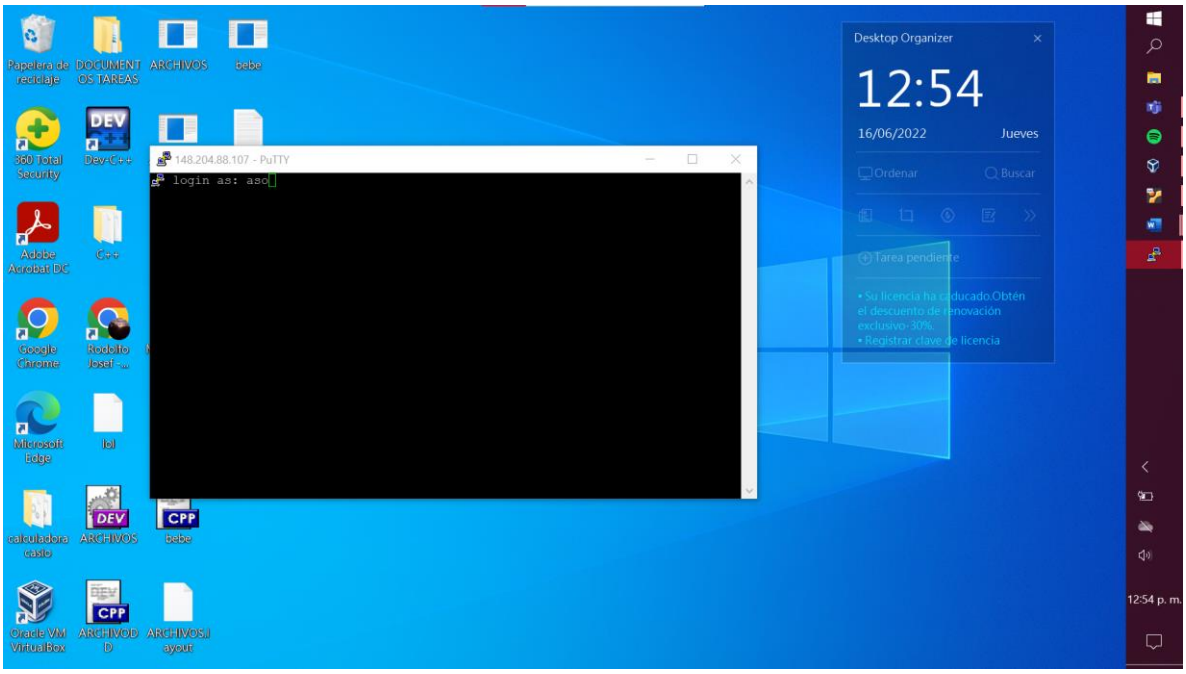
Qué pasa si se modifica el permiso de la carpeta public_html a 750

Cuando se modifica el permiso a 750 no se permite el acceso al servidor apache2, lo hará para el usuario y el grupo en el que se encuentre.

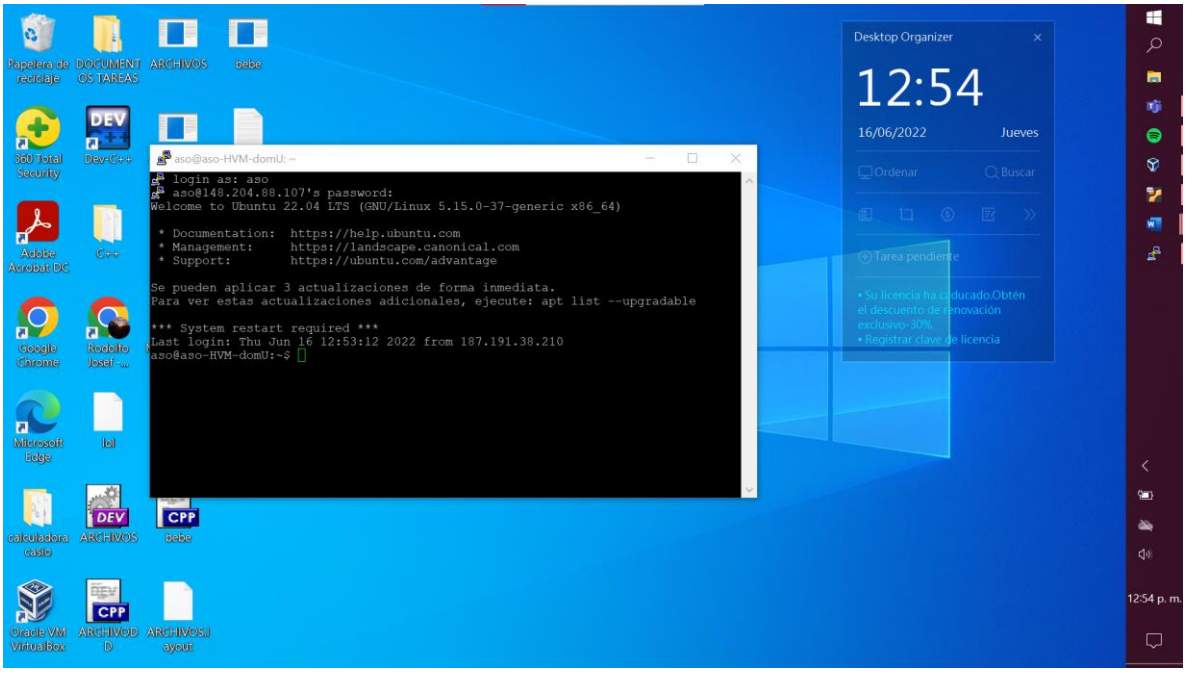
DESARROLLO DEL PROYECTO A TRAVES DEL SERVIDOR DE LA ESCUELA

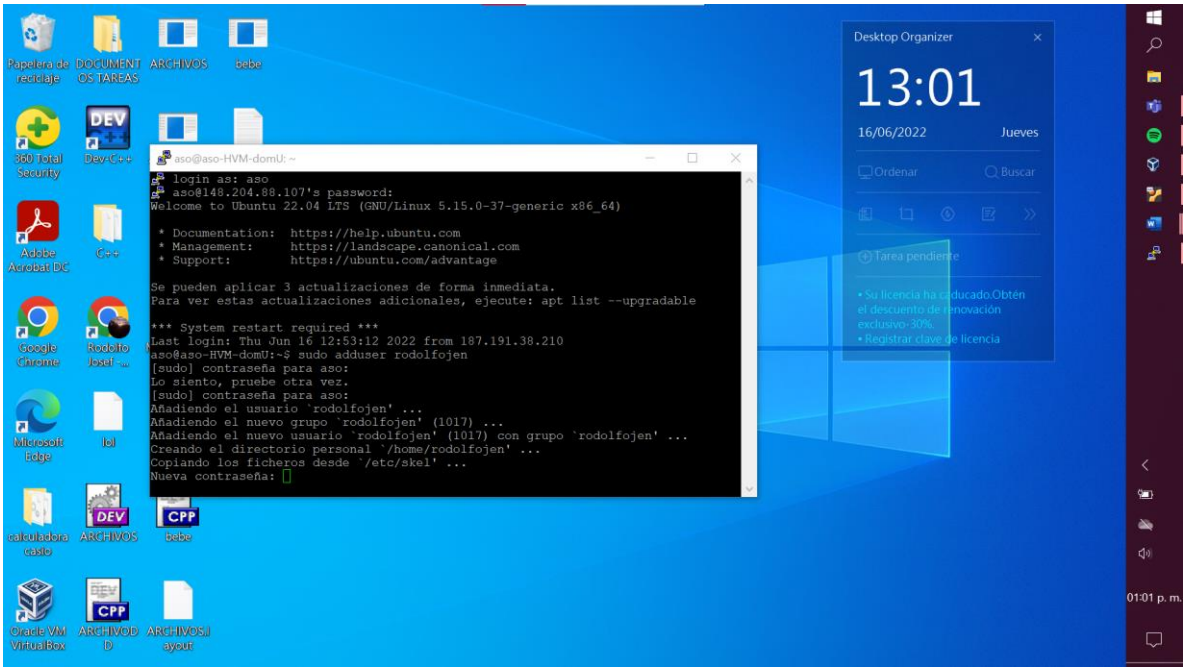
Ingresar al servidor de la escuela a través del ip que nos proporciono el docente



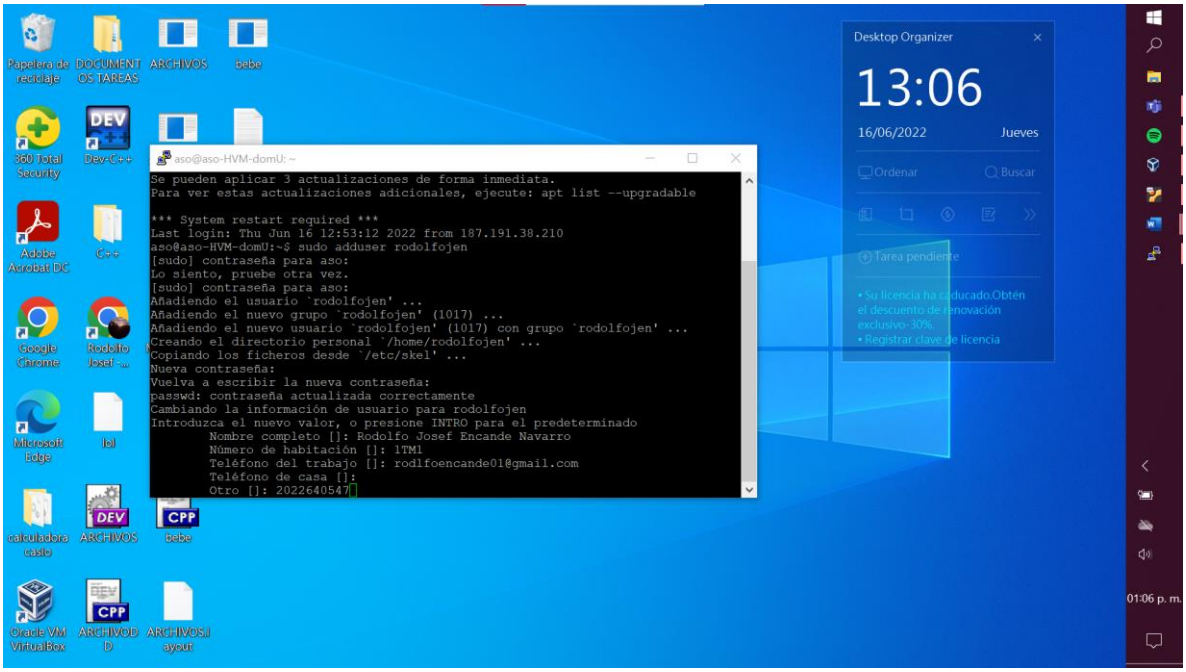


Crear mi usuario y posteriormente ingresar los datos solicitados por el docente

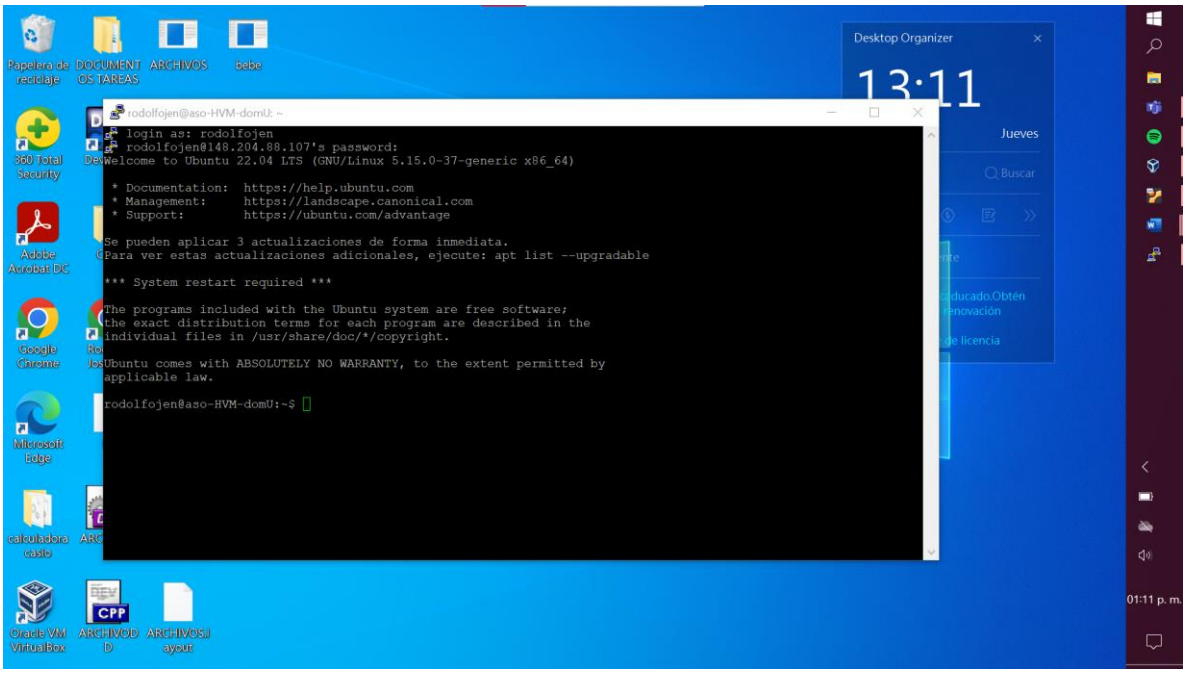


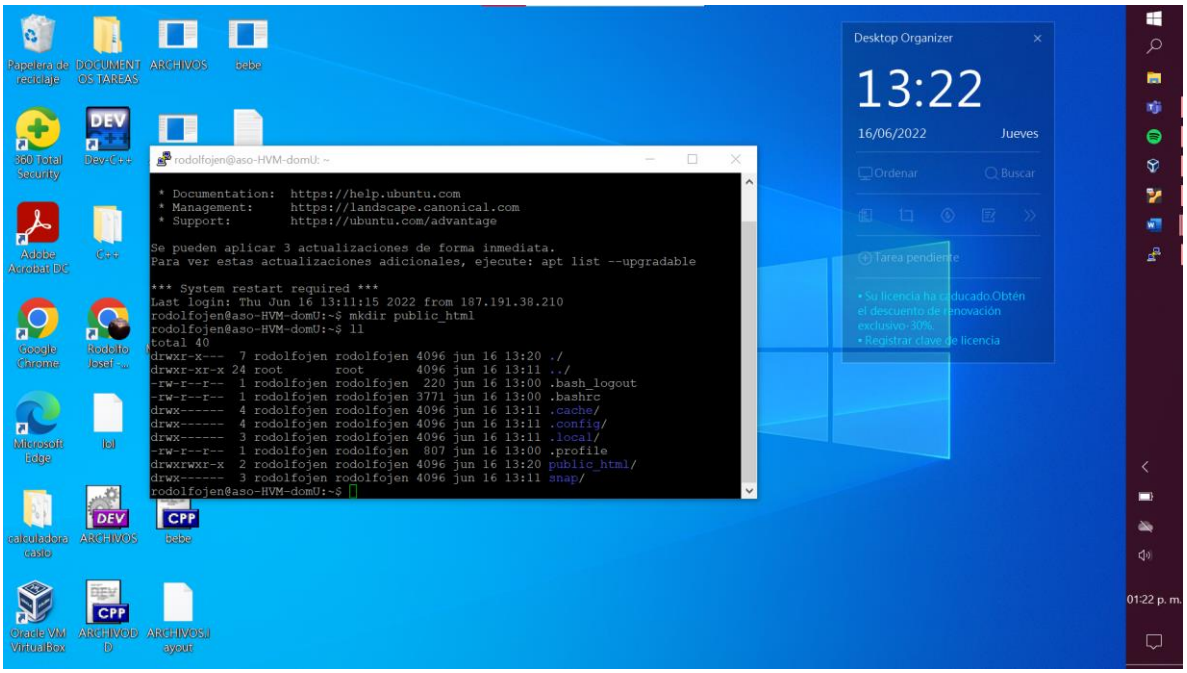


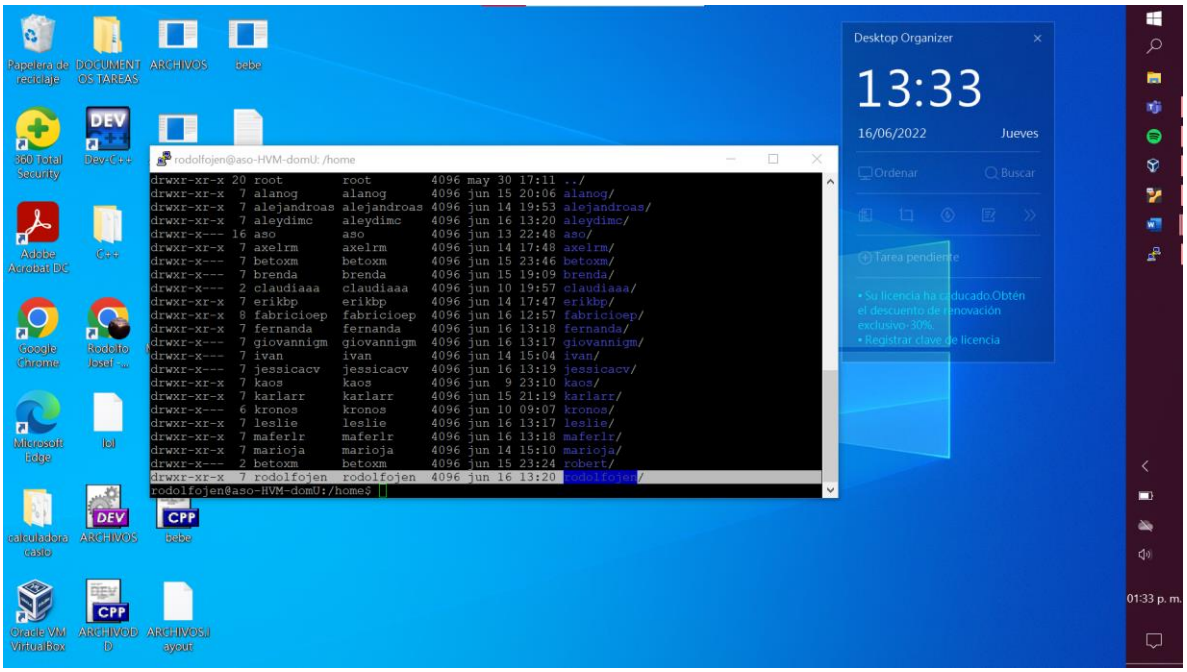
Después, se crea la carpeta public_html y se le dan los permisos solicitados para desarrollar el proyecto

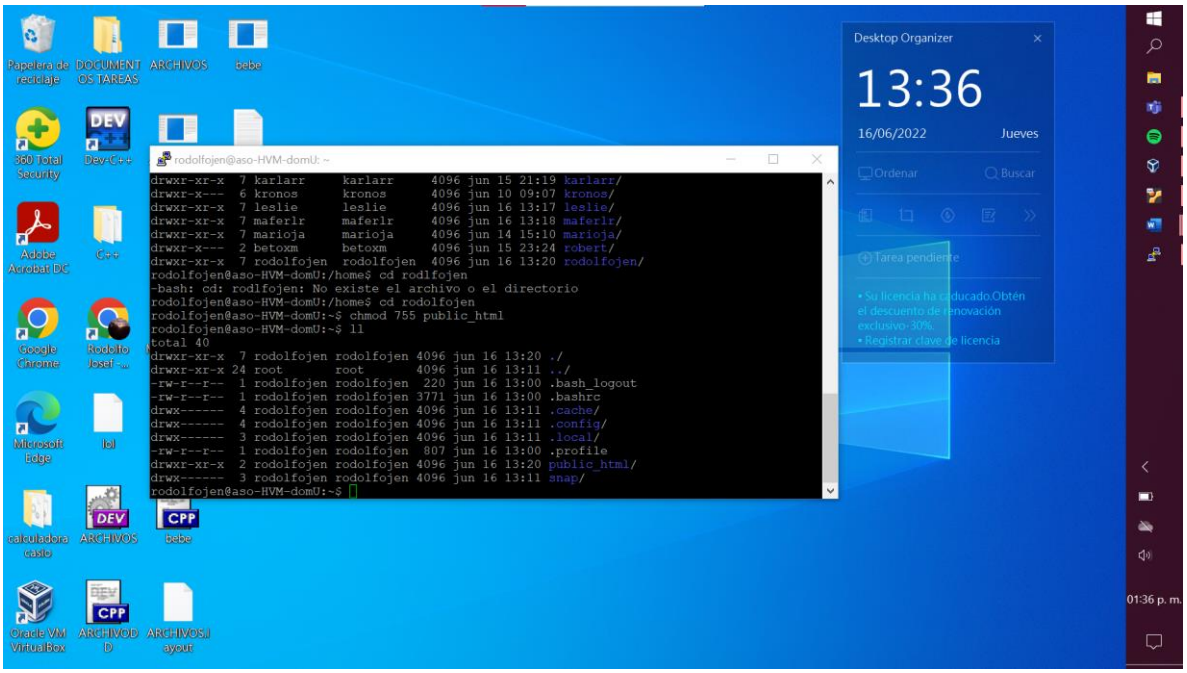


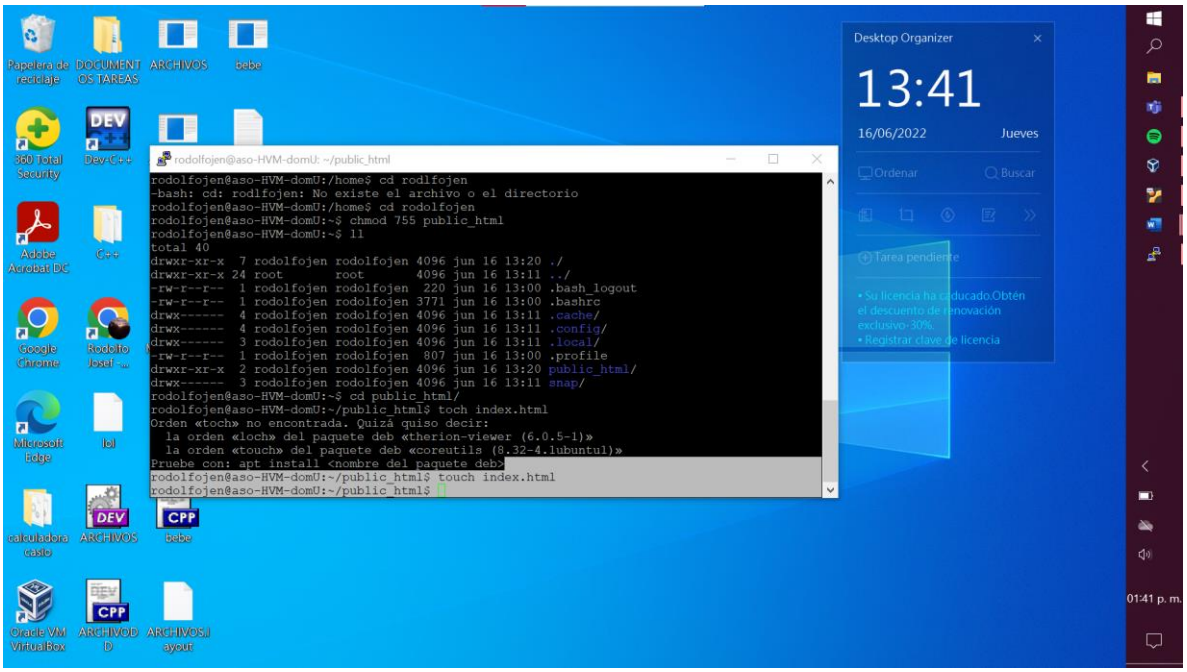
Despues, dentro de la carpeta crear el archivo index.html



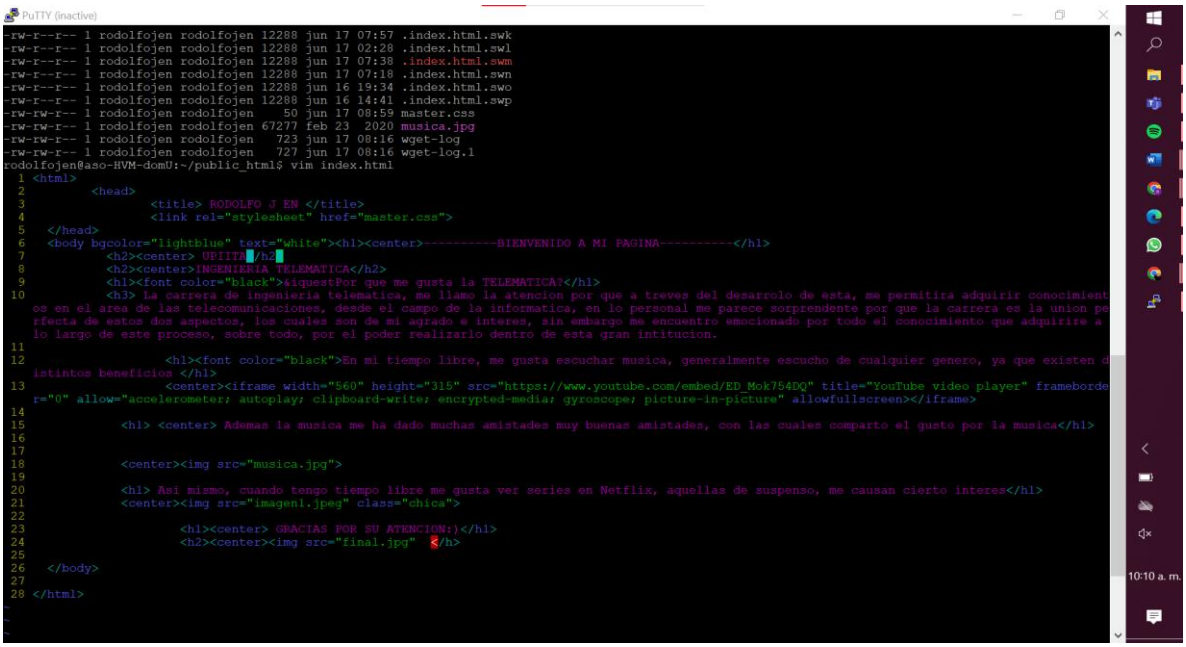

rodolfojen@aso-HVM-domU: ~
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
Se pueden aplicar 3 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable
*** System restart required ***
Last login: Thu Jun 16 13:11:15 2022 from 187.191.38.210
rodolfojen@aso-HVM-domU:~$ mkdir public_html
rodolfojen@aso-HVM-domU:~$ ll
total 40
drwxr-x--- 7 rodolfojen rodolfojen 4096 jun 16 13:20 ./
drwxr-xr-x 24 root root 4096 jun 16 13:11 ../
-rw-r--r-- 1 rodolfojen rodolfojen 220 jun 16 13:00 .bash_logout
-rw-r--r-- 1 rodolfojen rodolfojen 3771 jun 16 13:00 .bashrc
drwx------ 4 rodolfojen rodolfojen 4096 jun 16 13:11 .cache/
drwx------ 4 rodolfojen rodolfojen 4096 jun 16 13:11 .config/
drwx------ 3 rodolfojen rodolfojen 4096 jun 16 13:11 .local/
-rw-r--r-- 1 rodolfojen rodolfojen 807 jun 16 13:00 .profile
drwxrwxr-x 2 rodolfojen rodolfojen 4096 jun 16 13:20 public_html/
drwx------ 3 rodolfojen rodolfojen 4096 jun 16 13:11 snap/
rodolfojen@aso-HVM-domU:~$
Desktop Organizer
13:22
16/06/2022
Jueves
Ordenar
Buscar
Tarea pendiente
• Su licencia ha caducado.Obtén el descuento de renovación exclusivo-30%.
• Registrar clave de licencia
01:22 p. m.


rodolfojen@aso-HVM-domU: /home
drwxr-xr-x 20 root root 4096 may 30 17:11 ../
drwxr-xr-x 7 alanog alanog 4096 jun 15 20:06 alanog/
drwxr-xr-x 7 alejandroas alejandroas 4096 jun 14 19:53 alejandroas/
drwxr-xr-x 7 aleydimc aleydimc 4096 jun 16 13:20 aleydimc/
drwxr-x--- 16 aso aso 4096 jun 13 22:48 aso/
drwxr-xr-x 7 axelrm axelrm 4096 jun 14 17:48 axelrm/
drwxr-x--- 7 betoxm betoxm 4096 jun 15 23:46 betoxm/
drwxr-x--- 7 brenda brenda 4096 jun 15 19:09 brenda/
drwxr-x--- 2 claudiaaa claudiaaa 4096 jun 10 19:57 claudiaaa/
drwxr-xr-x 7 erikbp erikbp 4096 jun 14 17:47 erikbp/
drwxr-xr-x 8 fabricioep fabricioep 4096 jun 16 12:57 fabricioep/
drwxr-xr-x 7 fernanda fernanda 4096 jun 16 13:18 fernanda/
drwxr-x--- 7 giovannigm giovannigm 4096 jun 16 13:17 giovannigm/
drwxr-x--- 7 ivan ivan 4096 jun 14 15:04 ivan/
drwxr-x--- 7 jessicacv jessicacv 4096 jun 16 13:19 jessicacv/
drwxr-xr-x 7 kaos kaos 4096 jun 9 23:10 kaos/
drwxr-xr-x 7 karlarr karlarr 4096 jun 15 21:19 karlarr/
drwxr-x--- 6 kronos kronos 4096 jun 10 09:07 kronos/
drwxr-xr-x 7 leslie leslie 4096 jun 16 13:17 leslie/
drwxr-xr-x 7 maferlr maferlr 4096 jun 16 13:18 maferlr/
drwxr-xr-x 7 marioja marioja 4096 jun 14 15:10 marioja/
drwxr-x--- 2 betoxm betoxm 4096 jun 15 23:24 robert/
drwxr-xr-x 7 rodolfojen rodolfojen 4096 jun 16 13:20 rodolfojen/
rodolfojen@aso-HVM-domU:/home$
Desktop Organizer
13:33
16/06/2022
Jueves
Ordenar
Buscar
Tarea pendiente
• Su licencia ha caducado.Obtén el descuento de renovación exclusivo-30%.
• Registrar clave de licencia
01:33 p. m.





Después de esto mediante el uso de la aplicación de vim, se empieza a programar con html, mediante el cual se desarrolla la pagina web, aplicando conocimientos sobre este lenguaje, y dándole estructura al documento.



Al final, se crea la pagina web, la cual, es básica, pero se que mejorare mis competencias con el tiempo, con determinación y adquiriendo mas conocimiento en este rubro.

RODOLFO J EN

No es seguro | 148.204.88.107/~rodolfojen/

Mirar en YouTube

Ademas la musica me ha dado muchas amistades muy buenas amistades, con las cuales comparto el gusto por la musica

RODOLFO J EN

No es seguro | 148.204.88.107/~rodolfojen/

Asi mismo, cuando tengo tiempo libre me gusta ver series en Netflix, aquellas de suspenso, me causan cierto interes

GRACIAS POR SU ATENCION:)

Conclusiones:

El desarrollo de este proyecto me permitió mejorar y aprender mas aspectos sobre el como se utiliza el servidor y siendo sinceros me sorprendio mucho el como remotamente se hacia una pagina y el desarrollo que tuvimos que tener para poder manipular desde consola.

Se me complico el proyecto pero me gusto poder conocer el nuevo lenguaje sobre html y el saber de las estructuras de las paginas web en general, esto es sorprendente ya que conocí o me di cuenta de que existe cierta estandarización en el desarrollo de paginas web en el mundo, como un leguaje universal, esto inconscientemente nos une como grupo, el cual se encuentra interesado en un mismo objetivo, como lo es la informática y las telecomunicaciones, sin importar el color de piel, las creencias y religiones de cada quien.

Fuentes de información:

- https://forobeta.com/temas/permisos-carpeta-public_html-750-o-755.242653/
- https://c.neolo.com/knowledgebase/36/Sobre-el-directorio-publichtml.html#:~:text=El%20directorio%20public_html%20es%20la,alguien%20escribe%20el%20dominio%20principal.
- https://ubunlog.com/systemctl-trabaja-serviciosterminal/#Ejemplos_de_systemctl
- https://www.digitalocean.com/community/tutorials/como-instalar-el-servidor-web-apache-en-ubuntu-18-04-es
- https://www.ibm.com/docs/es/rational-build-forge/7.1.2?topic=components-apache-http-server-installation-configuration
- https://norfipc.com/internet/instalar-servidor-apache.html