



Actividad | 1 | Configuración de un servidor

Nombre del curso

Ingeniería en Desarrollo de Software



TUTOR: Marco Alonso Rodríguez Tapia.

ALUMNO: Uziel de Jesús López Ornelas.

FECHA: 06 de junio del 2025.

Tabla de Contenido

Introducción	1
Descripción	1
Justificación	1
Desarrollo	2
Configuración del servidor:	2
Conclusión	11
Referencias	12

Introducción

Estamos a punto de comenzar la materia de Sistemas Operativos II, en donde nos pide la actividad número uno la creación de un servidor mediante el sistema operativo de Linux, antes que nada explicaremos términos que son muy importantes para comprender esta actividad, todos conocemos diferentes sistemas operativos que interactúan con nuestros dispositivos día con día, un ejemplo de ello es un iPhone con su característico IOS, o que tal un Oppo con Android corriendo en su base, las computadoras no son la excepción ante esta regla puesto que también cuentan con sus respectivos sistemas, la mayor parte de la gente que ha utilizado una computadora conoce el sistema operativo de Windows, un sistema tan característico que cierta versión fue tan sonada y utilizada por varias empresas en aquel entonces (Windows XP) y también lo son las iMac, computadoras hechas por la empresa de Apple, pero estamos dejando un sistema operativo que también es bastante útil, hablamos de Linux, y más adelante te contaremos por qué es muy importante para esta materia en específico.

Descripción

¿Qué es Linux?, esta podría ser la pregunta principal y necesaria para comprender de lo que va esta materia, pues bien, Linux es un Sistema Operativo (SO) con “open source”, ¿Qué es “Open Source” ?, esto significa que las licencias que se lanzan tienen el código fuente a disposición del usuario final de forma legal, si una persona quiere acceder y jugar o crear interacciones o acciones con dicho código no presentara cargos o tendrá problemas, dicho esto podemos continuar con la explicación, una de las grandes ventajas que tiene este SO es su gran versatilidad, funcionalidad al igual que una seguridad que atrae a una gran base de usuarios que mantienen vivo este sistema por medio de actualizaciones hechas por ellos entre otras cosas interesantes que normalmente lo veras muy difícil en Windows y no solo esto sino que también lo puedes encontrar en varios dispositivos, existen ROMS dedicadas a cambiar el sistema de Android por uno de Linux y disfrutar de la experiencia que te puede brindar este cambio que en ocasiones puede ser significativo . el **Kernel** es una característica importante de Linux ya que sin el este no puede iniciar el sistema, es el encargado de gestionar las tareas que ocurren dentro

Justificación

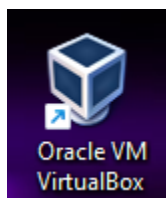
Mencionando lo anterior, ¿Por qué es importante implementar este tipo de ejercicios en la actividad siguiente?, pues bien por que conlleva una serie de pasos para lograr los objetivos y que este salga de una manera correcta y sin errores, que por supuesto, en la práctica se solucionarían y se obtendría una

experiencia ante dichas problemáticas, una “VirtualBox” es como su nombre lo indica una máquina virtual que se aloja en nuestro ordenador para que dentro de este se logre albergar uno o varios sistemas operativos, en el caso de esta actividad, el sistema operativo que tengo en mi computadora es Windows, y con la máquina virtual procederemos a instalar Linux que va de la mano con otro factor importante llamado “Ubuntu”, este es un sistema operativo basado en Linux, que logra ofrecer versatilidad y seguridad a los usuarios que estén en ella, uniendo todos los elementos, “VirtualBox”, “Ubuntu” y Linux tenemos la posibilidad de correr otro sistema operático dentro de nuestra computadora sin necesidad de tocar la BIOS de nuestro ordenador, de alguna manera este es fácil de instalar ya que no presenta complicaciones en su alojamiento. Dicho esto, tenemos la posibilidad de crear un servidor que demuestre la eficiencia de Linux y para la actividad que se presentara a continuación.

Desarrollo

Configuración del servidor:

Iniciaremos primero por dirigirnos a nuestro escritorio donde tenemos instalado nuestra VirtualBox:



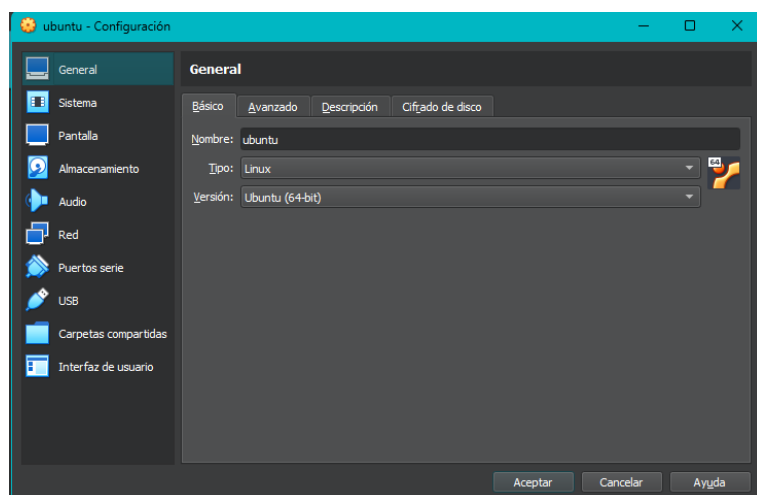
Lo seleccionamos y nos abrirá la siguiente ventana, donde existen diferentes características:



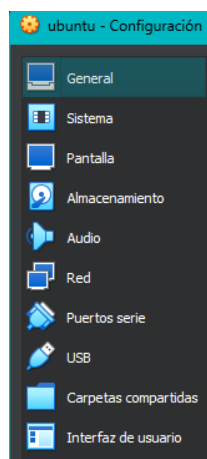
Antes que nada, daremos clic en el botón de configuración que se encuentra en la parte superior:



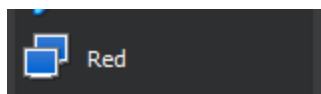
La ventana que nos mostrará será la siguiente:



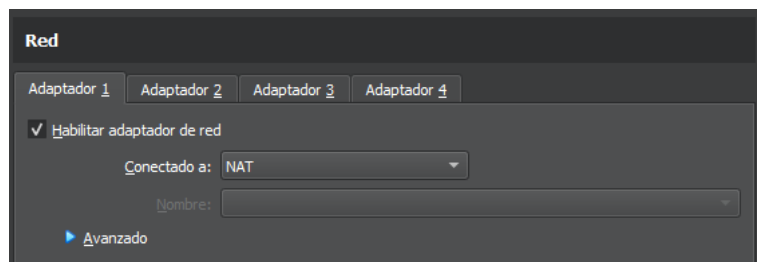
En la parte de la izquierda contaremos con distintas opciones que nos ayudan a configurar nuestra VirtualBox:



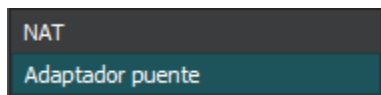
Seleccionaremos el apartado de “Red”:



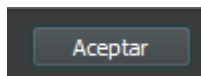
Que por consiguiente nos mostrara la siguiente ventana:



Tenemos el apartado de “NAT”, esto nos dice que usa direcciones de IP públicas para un anfitrión, y lo que queremos es que la VirtualBox se conecte a nuestra misma red, para ello cambiamos la opción a “adaptador puente”.



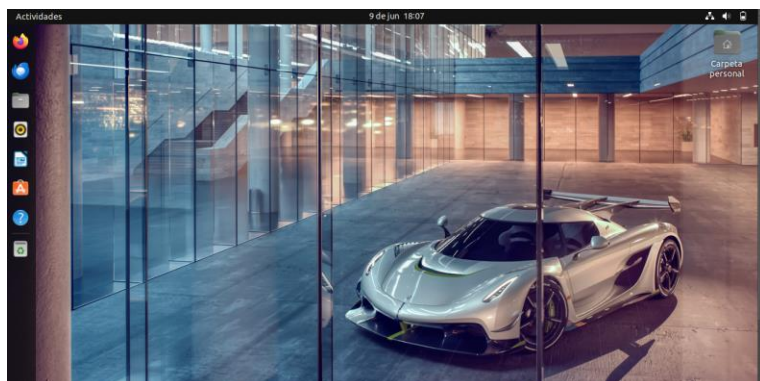
Y por último seleccionamos la opción de “Aceptar”.



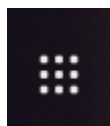
Procedemos a dar inicio y marcha de nuestra VirtualBox para que empecemos con la creación del servidor:



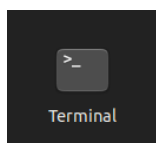
Y tenemos el escritorio de nuestro sistema operativo de Linux:



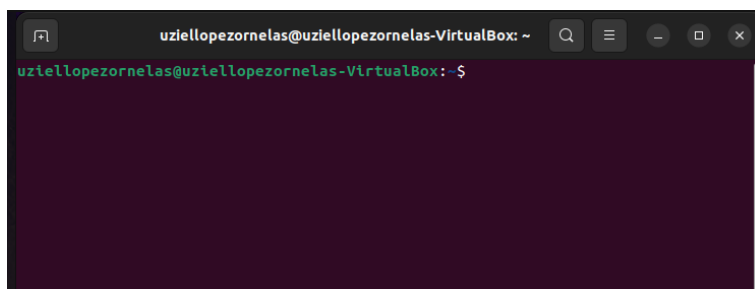
Nos deslizamos en la parte de abajo para que se nos muestren las diferentes aplicaciones que el SO maneja y buscaremos una en específico:



Lo seleccionamos para después ubicar la aplicación de “Terminal”:



Al seleccionarlo nos abrirá nuestra ventana de comandos:



En este lugar es en donde comenzaremos a trabajar en nuestra actividad, colocaremos los pasos necesarios para lograr crear nuestro servidor, empezaremos con el comando “sudo apt update”, este sirve para que actualice la lista de los paquetes que tenemos disponibles en nuestro sistema de Linux:

```
uziellopezornelas@uziellopezornelas-VirtualBox:~$ sudo apt update
[sudo] contraseña para uziellopezornelas:
```

Como se observa en la imagen anterior al colocar el comando nos pide nuestra contraseña de usuario, esto es por cuestiones de seguridad, la colocamos y nos carga la información con la actualización de los paquetes disponibles que tengo en mi SO de Linux:

```
Obj:1 http://mx.archive.ubuntu.com/ubuntu jammy InRelease
Des:2 http://mx.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Des:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Obj:4 http://mx.archive.ubuntu.com/ubuntu jammy-backports InRelease
Descargados 257 kB en 2s (126 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 421 paquetes. Ejecute «apt list --upgradable» para verlos.
```

Consiguiente a ello vamos a colocar el comando que dará inicio a la instalación de “Apache”:

```
uziellopezornelas@uziellopezornelas-VirtualBox:~$ sudo apt install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
Paquetes sugeridos:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
0 actualizados, 8 nuevos se instalarán, 0 para eliminar y 421 no actualizados.
Se necesita descargar 1 922 kB de archivos.
Se utilizarán 7 728 kB de espacio de disco adicional después de esta operación.
```

Al colocar el comando este mostrará los datos de los paquetes que se descargarán, así como la cantidad de memoria que este tendrá:

```
uziellopezornelas@uziellopezornelas-VirtualBox:~$ sudo apt install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
Paquetes sugeridos:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
0 actualizados, 8 nuevos se instalarán, 0 para eliminar y 421 no actualizados.
Se necesita descargar 1 922 kB de archivos.
Se utilizarán 7 728 kB de espacio de disco adicional después de esta operación.
```

Seguido de un mensaje de confirmación para reanudar la descarga:

```
:Desea continuar? [S/n]
```

Damos en confirmar para que este siga ejecutando el proceso:

```
Des:1 http://mx.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapr1 amd64 1.7.0-8ubuntu0.22.04.2 [108 kB]
Des:2 http://mx.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1 amd64 1.6.1-5ubuntu4.22.04.2 [92.8 kB]
Des:3 http://mx.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-5ubuntu4.22.04.2 [11.3 kB]
Des:4 http://mx.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-ldap amd64 1.6.1-5ubuntu4.22.04.2 [9 170 B]
Des:5 http://mx.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-bin amd64 2.4.52-1ubuntu4.14 [1 349 kB]
Des:6 http://mx.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-data all 2.4.52-1ubuntu4.14 [165 kB]
Des:7 http://mx.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-utils amd64 2.4.52-1ubuntu4.14 [89.0 kB]
Des:8 http://mx.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2 amd64 2.4.52-1ubuntu4.14 [97.9 kB]
Descargados 1 922 kB en 2s (1 065 kB/s)
Seleccionando el paquete libapr1:amd64 previamente no seleccionado.
(Leyendo la base de datos ... 204024 ficheros o directorios instalados actualmen
te.)
Preparando para desempaquetar .../0-libapr1_1.7.0-8ubuntu0.22.04.2_amd64.deb ...
Desempaquetando libapr1:amd64 (1.7.0-8ubuntu0.22.04.2) ...
Seleccionando el paquete libaprutil1:amd64 previamente no seleccionado.
Preparando para desempaquetar .../1-libaprutil1_1.6.1-5ubuntu4.22.04.2_amd64.deb
...
Desempaquetando libaprutil1:amd64 (1.6.1-5ubuntu4.22.04.2) ...
Seleccionando el paquete libaprutil1-dbd-sqlite3:amd64 previamente no selecciona
do.
Preparando para desempaquetar .../2-libaprutil1-dbd-sqlite3_1.6.1-5ubuntu4.22.04
.2_amd64.deb ...
Desempaquetando libaprutil1-dbd-sqlite3:amd64 (1.6.1-5ubuntu4.22.04.2) ...
Seleccionando el paquete libaprutil1-ldap:amd64 previamente no seleccionado.
Preparando para desempaquetar .../3-libaprutil1-ldap_1.6.1-5ubuntu4.22.04.2_amd6
4.deb ...
Desempaquetando libaprutil1-ldap:amd64 (1.6.1-5ubuntu4.22.04.2) ...
Seleccionando el paquete apache2-bin previamente no seleccionado.
Preparando para desempaquetar .../4-apache2-bin_2.4.52-1ubuntu4.14_amd64.deb ...
```

```
4.deb ...
Desempaquetando libaprutil1-ldap:amd64 (1.6.1-5ubuntu4.22.04.2) ...
Seleccionando el paquete apache2-bin previamente no seleccionado.
Preparando para desempaquetar .../4-apache2-bin_2.4.52-1ubuntu4.14_amd64.deb ...
Desempaquetando apache2-bin (2.4.52-1ubuntu4.14) ...
Seleccionando el paquete apache2-data previamente no seleccionado.
Preparando para desempaquetar .../5-apache2-data_2.4.52-1ubuntu4.14_all.deb ...
Desempaquetando apache2-data (2.4.52-1ubuntu4.14) ...
Seleccionando el paquete apache2-utils previamente no seleccionado.
Preparando para desempaquetar .../6-apache2-utils_2.4.52-1ubuntu4.14_amd64.deb .
..
Desempaquetando apache2-utils (2.4.52-1ubuntu4.14) ...
Seleccionando el paquete apache2 previamente no seleccionado.
Preparando para desempaquetar .../7-apache2_2.4.52-1ubuntu4.14_amd64.deb ...
Desempaquetando apache2 (2.4.52-1ubuntu4.14) ...
Configurando libapr1:amd64 (1.7.0-8ubuntu0.22.04.2) ...
Configurando apache2-data (2.4.52-1ubuntu4.14) ...
Configurando libaprutil1:amd64 (1.6.1-5ubuntu4.22.04.2) ...
Configurando libaprutil1-ldap:amd64 (1.6.1-5ubuntu4.22.04.2) ...
Configurando libaprutil1-dbd-sqlite3:amd64 (1.6.1-5ubuntu4.22.04.2) ...
Configurando apache2-utils (2.4.52-1ubuntu4.14) ...
Configurando apache2-bin (2.4.52-1ubuntu4.14) ...
Configurando apache2 (2.4.52-1ubuntu4.14) ...
Enabling module mpm_event.
Enabling module authz_core.
Enabling module authz_host.
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
```



```

Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
Procesando disparadores para ufw (0.36.1-4ubuntu0.1) ...
Procesando disparadores para man-db (2.10.2-1) ...
Procesando disparadores para libc-bin (2.35-0ubuntu3.5) ...

```

Todo lo que se vio anteriormente fue el análisis de los paquetes y sus propiedades para ingresarlas e instalarlas en el sistema operativo con el que estamos trabajando, al final no dio ningún error y esto nos puede decir que ya estamos listos para continuar con la creación del servidor, lo que sigue a continuación es corroborar el “Firewall” que se utilizara, tenemos que comprobar que esté instalado en nuestro sistema, para ello el comando es “sudo ufw app list”, Tenemos una lista de los distintos tipos de cifrado que ofrece apache, el primero “Apache” indica un tráfico sin cifrar, es decir una conexión que no es segura, “Apache Full” permite el tráfico de cifrado seguro y el ultimo que es “Apache Secure” nos permite conexiones seguras:

```

uziellopezornelas@uziellopezornelas-VirtualBox:~$ sudo ufw app list

```

```

Aplicaciones disponibles:
  Apache
  Apache Full
  Apache Secure
  CUPS

```

El que nos interesa en esta ocasión es el “Apache” ya que no se estará utilizando una conexión cifrada:

```

uziellopezornelas@uziellopezornelas-VirtualBox:~$ sudo ufw allow "Apache"

```

Revisamos el estatus del comando para verificar que este activado correctamente:

```

uziellopezornelas@uziellopezornelas-VirtualBox:~$ sudo ufw status
Estado: inactivo

```

Como vemos aparece inactivo, esto indica que no está activado correctamente, vamos a activar el “ufw” con el comando “sudo ufw enable” y verificaremos nuevamente el estatus:

```
uziellopezornelas@uziellopezornelas-VirtualBox:~$ sudo ufw status
Estado: activo

Hasta          Acción      Desde
-----          -
Apache         ALLOW      Anywhere
Apache (v6)    ALLOW      Anywhere (v6)
```

Una vez que esto está activo nos dirigiremos al siguiente comando que nos verifique toda la ejecución de “Apache” correctamente:

```
uziellopezornelas@uziellopezornelas-VirtualBox:~$ sudo systemctl status apache2
```

Lo colocamos y revisamos la información que nos arroja:

```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese
   Active: active (running) since Mon 2025-06-09 19:07:31 CST; 28min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 4886 (apache2)
    Tasks: 55 (limit: 2262)
   Memory: 5.1M
      CPU: 107ms
   CGroup: /system.slice/apache2.service
           └─4886 /usr/sbin/apache2 -k start
             └─4888 /usr/sbin/apache2 -k start
               └─4889 /usr/sbin/apache2 -k start

jun 09 19:07:31 uziellopezornelas-VirtualBox systemd[1]: Starting The Apache HT
jun 09 19:07:31 uziellopezornelas-VirtualBox apachectl[4885]: AH00558: apache2:
jun 09 19:07:31 uziellopezornelas-VirtualBox systemd[1]: Started The Apache HT
lines 1-16/16 (FND)
```

Hay un elemento que nos indica si el programa se está ejecutando correctamente y es el siguiente:

```
Active: active (running) since Mon 2025-06-09 19:07:31 CST; 28min ago
```

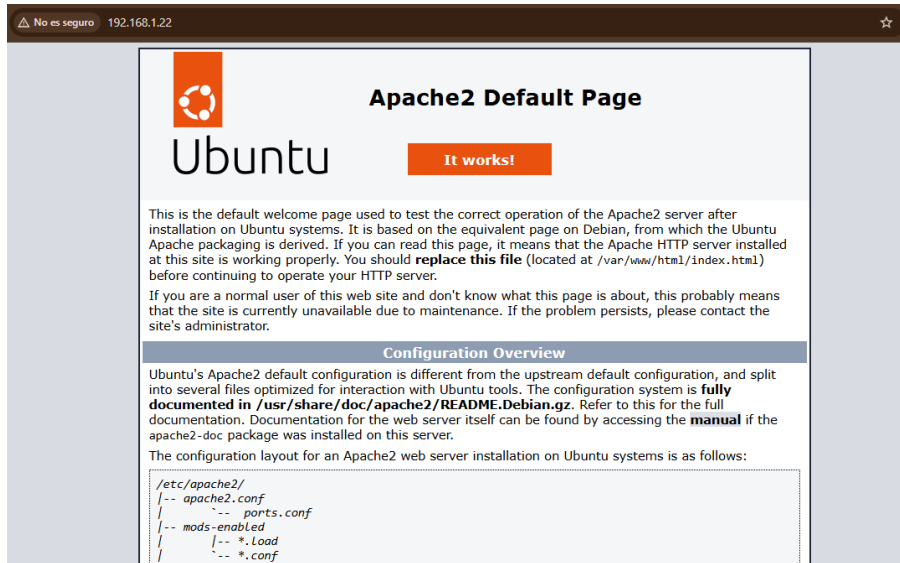
Después de esto escribimos el comando para saber la IP de nuestra “VirtualBox”, clave aclarar que en ocasiones no se tiene descargada la herramienta necesaria y se tiene que instalar para poder ver esa información, una vez que se instala tenemos acceso a revisar la información como la que se muestra en mi ventana:

```
uziellopezornelas@uziellopezornelas-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.22 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 2806:261:49a:cc3:15c3:e5df:b257:6126 prefixlen 64 scopeid 0x0<gl
    bbal>
    inet6 2806:261:49a:cc3:23d3:3b29:90bd:96f7 prefixlen 64 scopeid 0x0<gl
    bbal>
    inet6 fe80::d17a:967a:2fd8:d7e6 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:f6:32:12 txqueuelen 1000 (Ethernet)
    RX packets 31066 bytes 42768759 (42.7 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4638 bytes 431586 (431.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

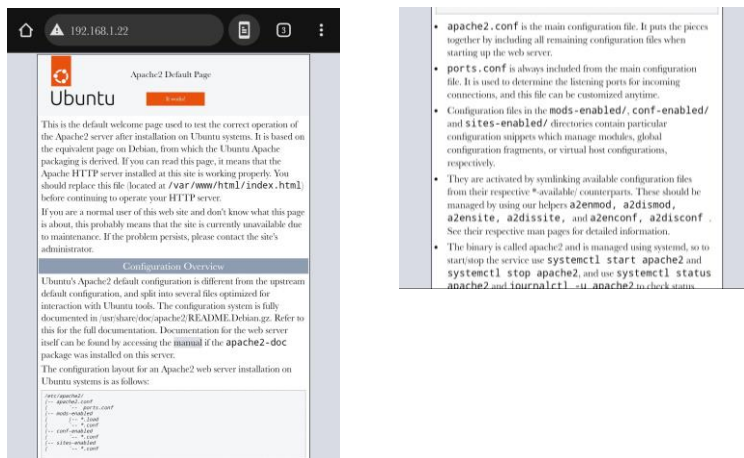
Ahora el elemento importante aquí es la IP que nos arrojó la “VirtualBox”:

inet 192.168.1.22

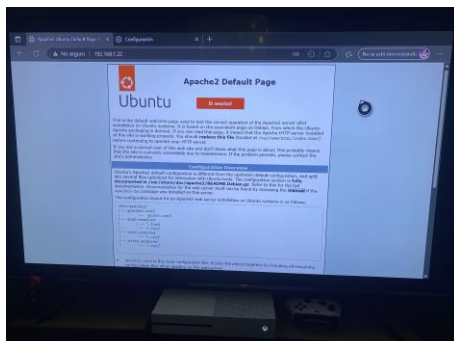
Colocaremos esta IP en el navegador web de la computadora:



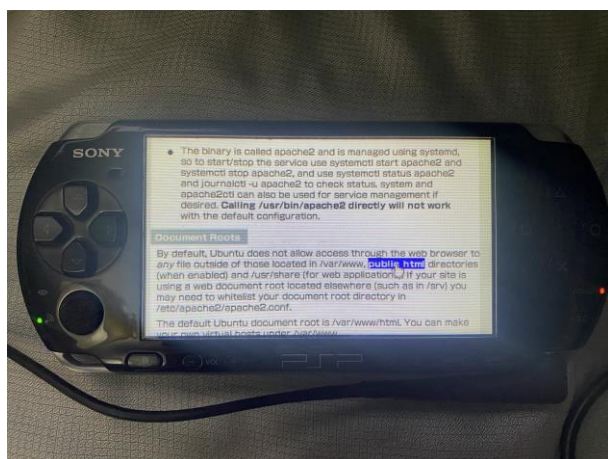
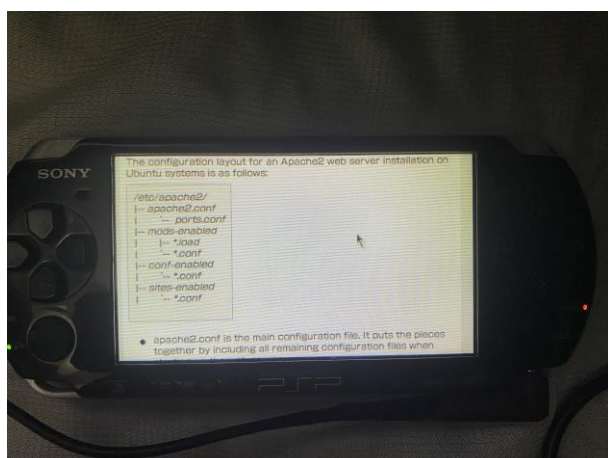
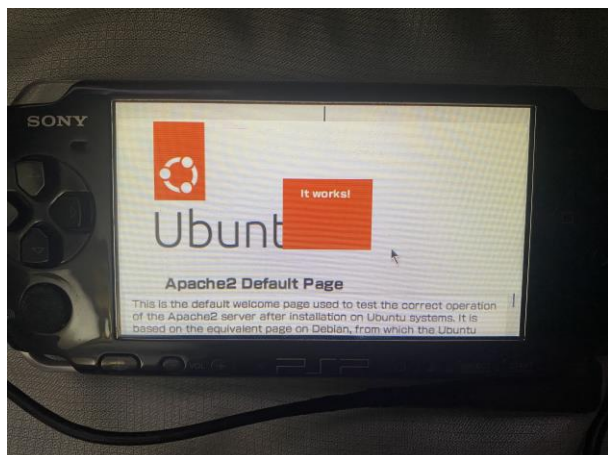
en el celular:



En la consola y la Smart Tv:



Y si se lo preguntan, si, también se puede en un PSP 3000:



Con lo visto anteriormente me doy cuenta de la capacidad que tiene Linux de crear servidores sencillos, pero a la vez útiles que nos muestran la compatibilidad de los diferentes dispositivos, desde un celular hasta un PSP 3000, en definitiva, este SO es bastante versátil y eficiente.

Conclusión

Como logramos observar mediante una “VirtualBox” logramos crear un servidor en donde varios dispositivos pueden reconocer dicha dirección IP, solo es cuestión de descargar los paquetes necesarios para que este se ejecute correctamente y ajustar parámetros, los pasos no se cuentan por pocos, sino que son varios los que nos permite descargar las herramientas, seleccionar el “Firewall” necesario, asegurarnos de que estos sean los correctos y que se ejecute correctamente, esto nos dice la facilidad y la potencia con la que podemos crear cosas, claro está que solo es la punta del iceberg, estamos apunto de descubrir diferentes maneras en las que el sistema operativo Linux nos muestra la potencia, versatilidad y seguridad de la que muchos están hablando y que es la preferida de una gran comunidad de fanáticos por la tecnología que siempre buscan exprimir al máximo las posibilidades que este SO ofrece sin lugar a dudas. Todo esto recordando que se está haciendo en una computadora que alberga de alguna manera dos sistemas operativos.

Link de GitHub

<https://github.com/UZLOP984/Sistemas-Operativos-II.git>

Referencias

Definición de Linux. (s. f.). <https://www.redhat.com/es/topics/linux/what-is-linux>

¿Qué es el software open source? (s. f.). <https://www.redhat.com/es/topics/open-source/what-is-open-source-software>

Fernández, Y. (2020, 1 junio). *VirtualBox: qué es y cómo usarlo para crear una máquina virtual con Windows u otro sistema operativo.* Xataka. <https://www.xataka.com/basics/virtualbox-que-como-usarlo-para-crear-maquina-virtual-windows-u-otro-sistema-operativo>

De Contenidos de GoDaddy, E. (2024, 5 marzo). *¿Qué es Ubuntu y para qué sirve?* GoDaddy Resources - Spain. <https://www.godaddy.com/resources/es/crearweb/que-es-ubuntu-y-para-que-sirve>