Sistema de gestión de redes - Proyecto – CyberTech Solutions

Javier Alejandro Avalos Galindo

Universidad Mariano Gálvez de Guatemala
Ingeniería en sistemas de información y ciencias de la computación
Simulacro de examen privado

Tabla de Contenidos

Análisis y diseño de la red

Diseño de la topología

Para el diseño de la red de CyberTech Solutions se ha implementado una topología jerárquica de tres capas (Core, Distribución y Acceso), la cual es el modelo más utilizado en redes empresariales debido a su organización, escalabilidad, facilidad de administración y tolerancia a fallos.

La red se estructuró considerando los diferentes niveles de comunicación dentro de la empresa, separando las funciones críticas. De esta manera, se logra una arquitectura modular que permite ampliar o modificar la red sin afectar al resto de la infraestructura.

Capa 1 – CORE

En la capa Core se encuentra el router (R1 Core), encargado del enrutamiento principal y de proporcionar salida hacia Internet. Esta capa representa el punto de mayor jerarquía en la red, ya que centraliza el tráfico entre las distintas VLANs y hacia redes externas. Su función principal es ofrecer un tránsito rápido, estable y confiable entre los diferentes segmentos internos de la empresa y el ISP.

Capa 2 – Distribución

La capa de Distribución está compuesta por dos switches, los cuales se encargan de agregar el tráfico proveniente de los switches de acceso y segmentar la red mediante VLANs. En esta capa también se ubica la conexión con el servidor principal (SRV1), el cual pertenece a la VLAN 40 – SERVERS. Estos dispositivos actúan como intermediarios entre el Core y la capa de Acceso, aplicando políticas de control y priorización de tráfico.

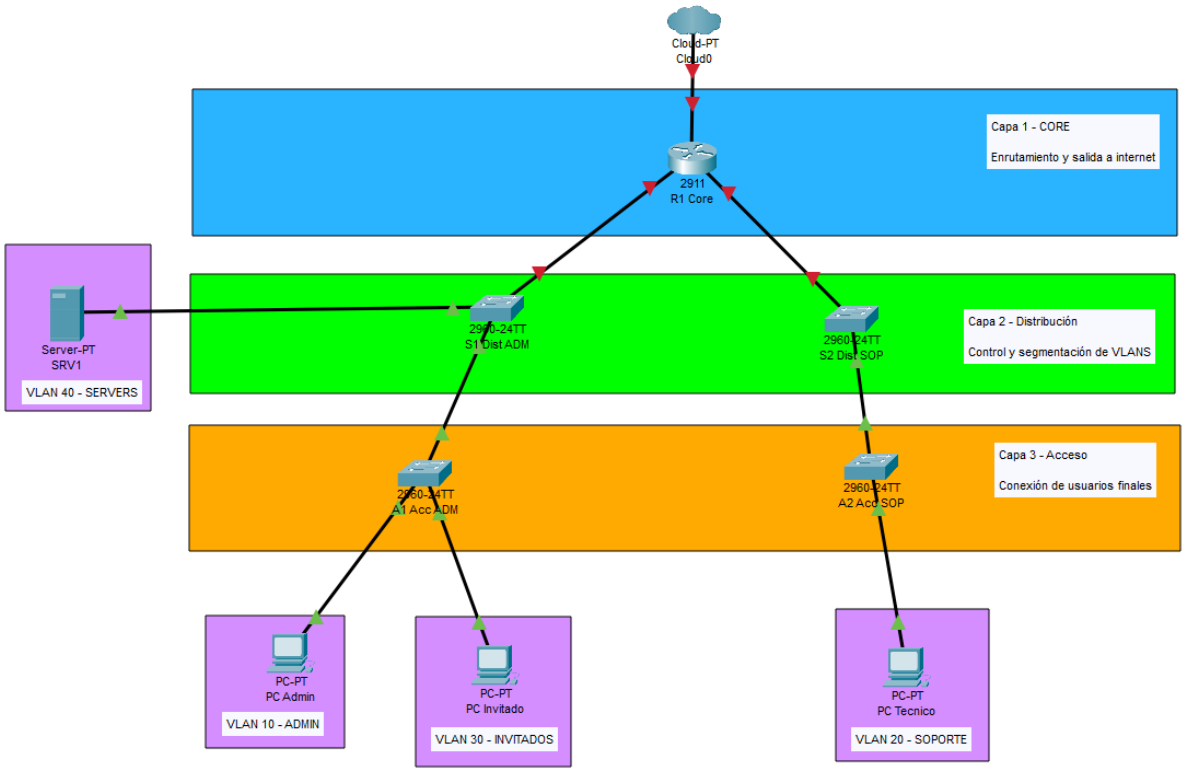
Capa 3 – Acceso

La capa de Acceso está conformada por dos switches, donde se conectan directamente los usuarios finales de la empresa.

- El switch A1 gestiona las VLAN 10 – ADMIN y 30 – INVITADOS, donde se conectan los equipos de los empleados administrativos y los dispositivos de visitantes que requieren acceso limitado a Internet.
- El switch A2 pertenece a la VLAN 20 – SOPORTE, la cual agrupa los equipos de los técnicos y personal encargado del mantenimiento y monitoreo.

Esta separación por VLANs permite un mejor control de seguridad, evitando que los invitados o usuarios no autorizados accedan a los servicios internos de la empresa.

Diseño de la red



Modelo OSI y TCP/IP

En la red diseñada para CyberTech Solutions, se aplican los principios del modelo OSI y TCP/IP, los cuales permiten entender cómo se comunican los dispositivos desde el nivel físico hasta las aplicaciones que utilizan los usuarios.

Capas físicas / Enlace (OSI) – Acceso a la red (TCP/IP)

Se utilizan cables de cobre para enlazar el router R1-Core con los switches de distribución (S1 y S2), y de estos hacia los switches de acceso (A1 y A2). Los puertos Gigabit se destinan a los enlaces troncales entre equipos, mientras que los puertos FastEthernet se emplean para conectar los usuarios y servidores. El control de acceso a la red se realiza mediante VLANs, que separan a los diferentes grupos de trabajo.

Capa red (OSI) – internet (TCP/IP)

En esta capa se maneja la dirección IP y el enrutamiento del tráfico. El router R1-Core funciona como puerta de enlace para todas las subredes de la empresa, permitiendo la comunicación entre las distintas VLANs y la salida a Internet. Esta función asegura que cada departamento tenga su propio segmento de red, manteniendo la organización y el control del flujo de información.

Capa transporte (OSI y TCP/IP)

Aquí se garantiza que los datos lleguen de forma segura y ordenada entre los dispositivos. Los servicios internos, como las pruebas de conectividad, la navegación web (HTTP), la transferencia de archivos (FTP) y la asignación de direcciones IP (DHCP), utilizan protocolos de esta capa para enviar y recibir información correctamente. Si la red creciera, en esta capa también podrían aplicarse políticas de priorización de tráfico.

Capa sesión / presentación / aplicación (OSI) – Aplicación (TCP/IP)

En estas capas se encuentran los servicios visibles para el usuario. El servidor SRV1 aloja las aplicaciones principales de la empresa:

- Un sitio web informativo,
- Un servidor FTP para compartir archivos,
- Un servidor DHCP que asigna direcciones IP automáticas,
- Y un agente de monitoreo e inteligencia artificial para analizar el tráfico de la red.

Los usuarios administrativos y de soporte pueden acceder a estos servicios internos, mientras que los invitados (VLAN 30) únicamente tienen acceso a Internet, manteniendo la seguridad de la red interna.

Subnetting

Para el ejercicio de Subneteo se solicita utilizar VLSM, el cual es un método de Subneteo que permite dividir una red grande en varias subredes mas pequeñas, asignando mas direcciones IP a las áreas que lo necesitan y menos a las que usan pocas. Esto permite optimizar el uso de direcciones IP.

Ejercicio con 100 dispositivos

A modo de ejercicio, se realiza el ejercicio con la misma base que se indicó en el punto “Diseño de la topología” con valores inventados para cada área, los valores son:

VLAN	**Nombre**	**Tipo de equipos**	**Dispositivos**
10	Administración	PCs de oficina	25
20	Soporte Técnico	PCs y equipos de mantenimiento	15

30	Invitados	Laptops y visitantes	50
40	Servidores	Servidores	10

En total son 100 dispositivos (lo que solicita el documento), como red base privada se utilizará: 192.168.0.0/24. Esta red brinda 254 hosts disponibles, con VLSM se procede a dividir en bloques más pequeños según el tamaño de cada VLAN.

Paso 1 – ordenar por tamaño

Se ordenan de mayor a menor la cantidad de hosts.

1. VLAN 30 – Invitados (50 hosts)
2. VLAN 10 – Administración (25 hosts)
3. VLAN 20 – Soporte técnico (15 hosts)
4. VLAN 40 – Servidores (10 hosts)

Paso 2 – Calculo de bits necesarios

VLAN	Hosts necesarios	Hosts reales	Máscara	CIDR
30	50	62	255.255.255.192	/26
10	25	30	255.255.255.224	/27
20	15	30	255.255.255.224	/27
40	10	14	255.255.255.240	/28

Paso 3 – Asignar rangos

VLAN	Red asignada	Rango de hosts	Broadcast	Máscara
30	192.168.0.0/26	192.168.0.1 – 192.168.0.62	192.168.0.63	255.255.255.192
10	192.168.0.64/27	192.168.0.65 – 192.168.0.94	192.168.0.95	255.255.255.224
20	192.168.0.96/27	192.168.0.97 – 192.168.0.126	192.168.0.127	255.255.255.224
40	192.168.0.128/28	192.168.0.129 – 192.168.0.142	192.168.0.143	255.255.255.240

Ojo: Observamos que solo utilizamos hasta la IP 192.168.0.143, por lo que aun se tienen libres desde las IP 192.168.0.144 a 192.168.0.255 para futuras expansiones de la red.

Paso 4 – Definición de Gateway por VLAN

Estas puertas de enlace serán utilizadas luego en las subinterfaces del router.

VLAN	Gateway
10 – Administración	192.168.0.65
20 – Soporte	192.168.0.97
30 – Invitados	192.168.0.1
40 – Servidores	192.168.0.129

Paso 5 – Resumen

El esquema asigna subredes de distinto tamaño según la necesidad real de cada VLAN, priorizando primero la más grande que es Invitados y luego Administración, Soporte y Servidores. Con VLSM se optimiza el espacio IP: cada área recibe una máscara justa, evitando desperdicio. Para cada VLAN se define su Gateway en la primera IP utilizable y quedan rangos libres para crecimiento futuro sin reconfigurar toda la red. Este plan mantiene la red ordenada, escalable y fácil de administrar.

VLAN	Nombre	IP de red	Máscara	Rango de Hosts	Gateway	Broadcast
10	Administración	192.168.0.64	255.255.255.224	192.168.0.65 – 192.168.0.94	192.168.0.65	192.168.0.95
20	Soporte Técnico	192.168.0.96	255.255.255.224	192.168.0.97 – 192.168.0.126	192.168.0.97	192.168.0.127
30	Invitados	192.168.0.0	255.255.255.192	192.168.0.1 – 192.168.0.62	192.168.0.1	192.168.0.63
40	Servidores	192.168.0.128	255.255.255.240	192.168.0.129 – 192.168.0.142	192.168.0.129	192.168.0.143

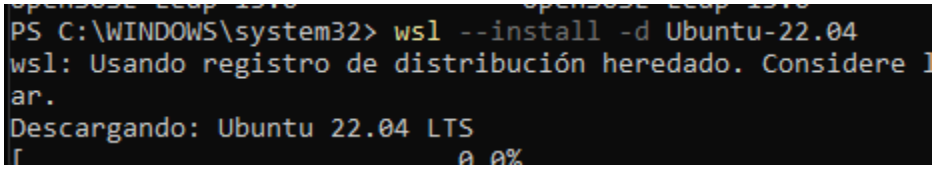
Configuración de sistemas operativos

Configuración de servidor Linux – Ubuntu

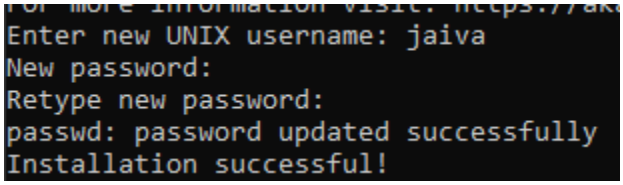
Para la fase de sistemas operativos, se utilizará la herramienta de WSL, la cual es una herramienta que permite ejecutar un entorno Linux en el propio sistema operativo de Windows. Se realiza de esta manera debido a la facilidad de creación y rapidez, obviando por un momento la configuración de la máquina virtual y ahorrando tiempo.

Tener en cuenta que hacerlo en una maquina virtual utiliza los mismos comando y configuraciones.

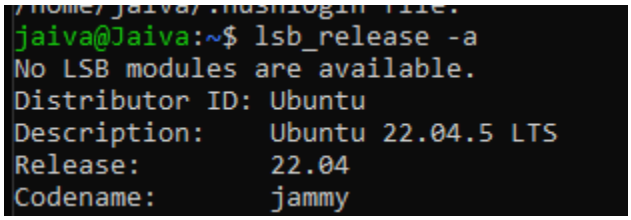
Para la práctica se utilizará la distribución de Ubuntu



Se configura el usuario principal



Una vez instalado, se debe ver la versión de Linux que se tiene



Al ser un sistema operativo de código abierto, siempre se recomienda mantenerlo actualizado a la última versión, para ello se ejecutan los siguientes comandos.

```
jaiva@Jaiva:~$ sudo apt update && sudo apt upgrade -y
[sudo] password for jaiva:
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
```

Automáticamente inicia la actualización del sistema operativo. Se espera a que termine.

A continuación, se realiza la instalación de las herramientas de red y monitoreo

```
jaiva@Jaiva:~$ sudo apt install net-tools curl wget nano htop -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
htop is already the newest version (3.0.5-7build2).
curl is already the newest version (7.81.0-1ubuntu1.21).
```

Una vez instaladas las herramientas de red, se puede configurar el hostname del servidor, en este caso se utiliza: cybertech-server

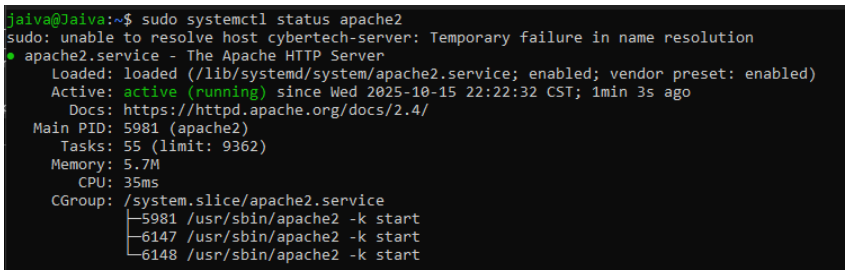
```
jaiva@Jaiva:~$ sudo hostnamectl set-hostname cybertech-server
jaiva@Jaiva:~$ hostnamectl
 Static hostname: cybertech-server
       Icon name: computer-container
         Chassis: container
      Machine ID: 1d03896bb1c94f64970f6edf0d432697
         Boot ID: 24ceef1548654ffbaba6329164781597
  Virtualization: wsl
Operating System: Ubuntu 22.04.5 LTS
          Kernel: Linux 6.6.87.2-microsoft-standard-WSL2
    Architecture: x86-64
```

Además, se instalan servicios de red básicos, demostrando que el servidor puede gestionar funciones de red internas.

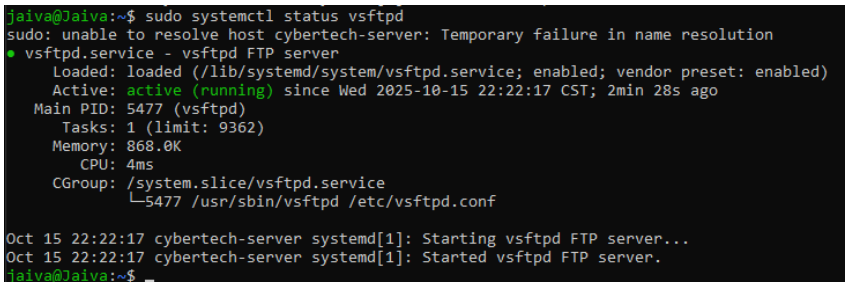
```
jaiva@Jaiva:~$ sudo apt install apache2 vsftpd isc-dhcp-server -y
sudo: unable to resolve host cybertech-server: Temporary failure in name resolution
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Se validan que los servicios estén activos.

Apache:

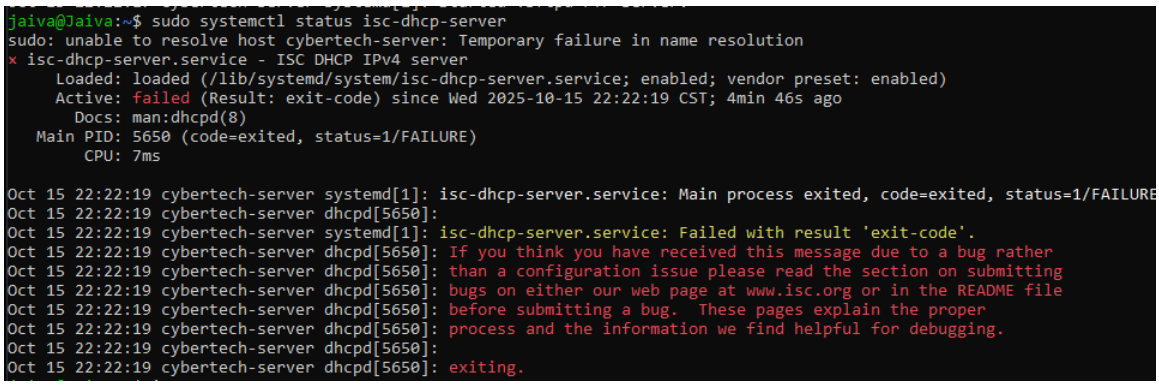


FTP:



Dhcp server:

Durante la configuración del servicio DHCP en Ubuntu sobre WSL, se presentó un error de inicio (status=FAILURE) debido a que WSL no posee acceso directo a las interfaces de red físicas del host. Este comportamiento es normal, ya que el subsistema funciona en modo virtualizado.



Sin embargo, la instalación y los archivos de configuración del servicio se completaron correctamente, demostrando el proceso de implementación que se realizaría en un entorno real.

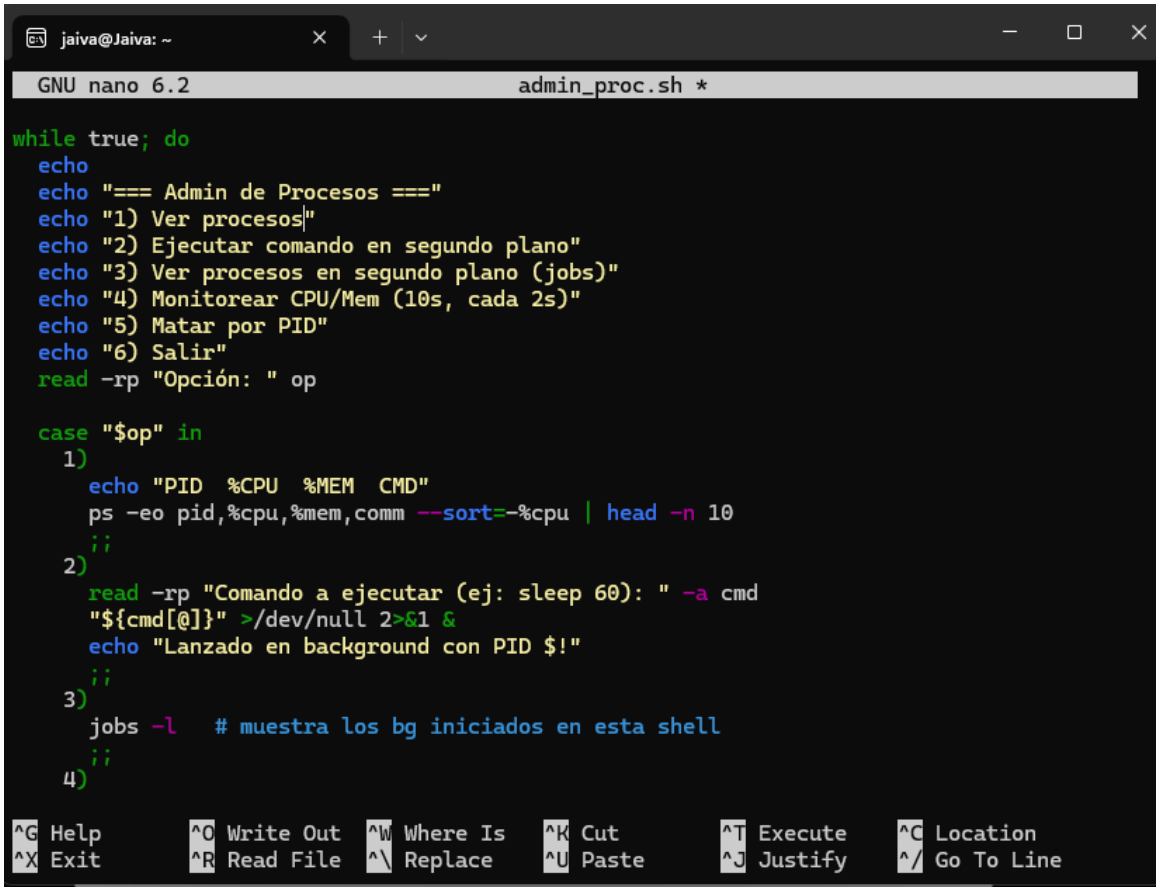
```
jaiva@Jaiva:~$ cat /etc/default/isc-dhcp-server
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPDv4_PID=/var/run/dhcpd.pid
#DHCPDv6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
#       Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4=""
INTERFACESv6=""
```

Por último, se valida que se tiene conexión a internet.

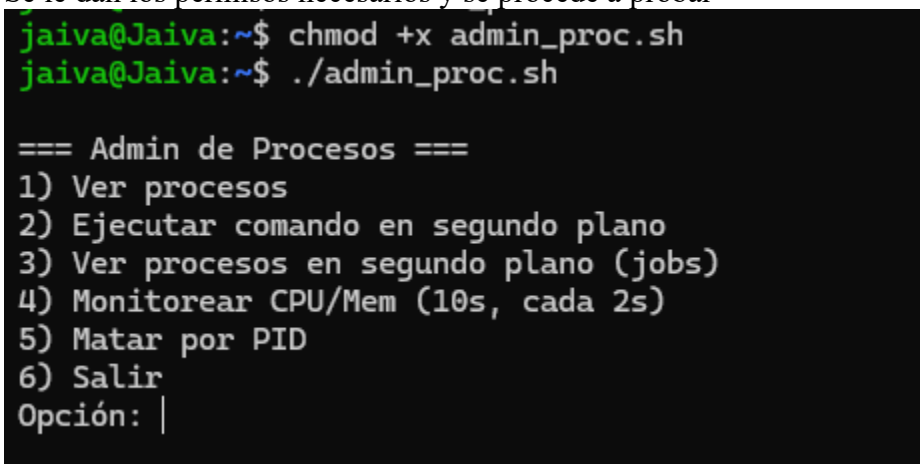
```
jaiva@Jaiva:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet 10.255.255.254/32 brd 10.255.255.254 scope global lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:a2:33:40 brd ff:ff:ff:ff:ff:ff
    inet 172.24.157.110/20 brd 172.24.159.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fea2:3340/64 scope link
       valid_lft forever preferred_lft forever
jaiva@Jaiva:~$ ping -c 4 google.com
PING google.com (172.217.15.206) 56(84) bytes of data.
64 bytes from mia09s20-in-f14.1e100.net (172.217.15.206): icmp_seq=1 ttl=114 time=43.8 ms
64 bytes from mia09s20-in-f14.1e100.net (172.217.15.206): icmp_seq=2 ttl=114 time=43.7 ms
64 bytes from mia09s20-in-f14.1e100.net (172.217.15.206): icmp_seq=3 ttl=114 time=41.3 ms
64 bytes from mia09s20-in-f14.1e100.net (172.217.15.206): icmp_seq=4 ttl=114 time=39.7 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3045ms
rtt min/avg/max/mdev = 39.684/42.123/43.838/1.725 ms
```

Script en BASH

En la misma consola de Ubuntu de WSL, se realiza la creación de un archivo .sh, esto para poder administrar de forma sencilla los procesos que tiene el sistema operativo.



Se le dan los permisos necesarios y se procede a probar



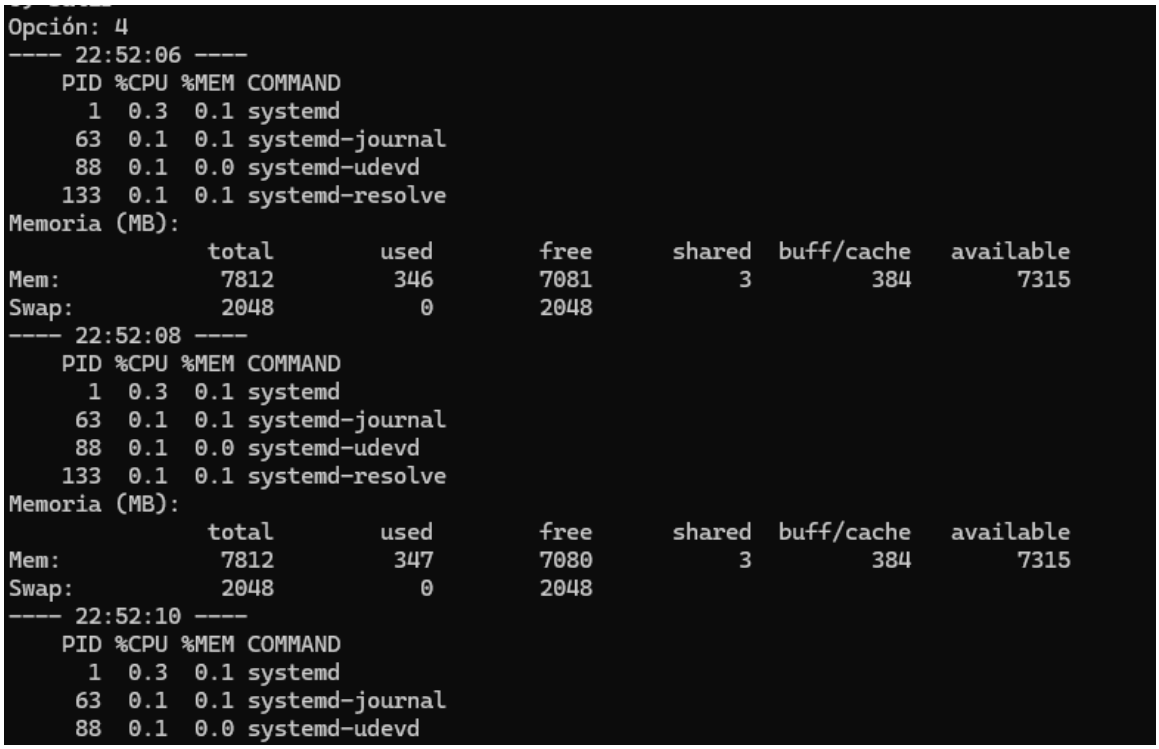
Con la opción 1 se ven los procesos en ese momento

```
Opción: 1
PID  %CPU  %MEM  CMD
    PID %CPU %MEM COMMAND
      1  0.6  0.1 systemd
     63  0.1  0.1 systemd-journal
     88  0.1  0.0 systemd-udevd
    133  0.1  0.1 systemd-resolve
    165  0.1  0.0 systemd-timesyn
    212  0.1  0.2 networkd-dispat
    311  0.1  0.2 unattended-upgr
      2  0.0  0.0 init-systemd(Ub
      6  0.0  0.0 init
```

Con la opción 2 se ejecuta un proceso en segundo plano, luego con la opción 3 se pueden validar esos procesos en segundo plano

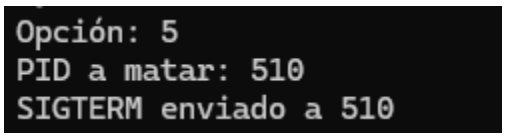
```
=== Admin de Procesos ===
1) Ver procesos
2) Ejecutar comando en segundo plano
3) Ver procesos en segundo plano (jobs)
4) Monitorear CPU/Mem (10s, cada 2s)
5) Matar por PID
6) Salir
Opción: 2
Comando a ejecutar (ej: sleep 60): sleep 120
Lanzado en background con PID 480

=== Admin de Procesos ===
1) Ver procesos
2) Ejecutar comando en segundo plano
3) Ver procesos en segundo plano (jobs)
4) Monitorear CPU/Mem (10s, cada 2s)
5) Matar por PID
6) Salir
Opción: 3
[1]+   480 Running                 "${cmd[@]}" > /dev/null 2>&1 &
```

Con la opción 4 se pueden monitorear los top procesos y memoria



```
Opción: 4
---- 22:52:06 ----
    PID %CPU %MEM COMMAND
      1  0.3  0.1 systemd
     63  0.1  0.1 systemd-journal
     88  0.1  0.0 systemd-udevd
    133  0.1  0.1 systemd-resolve
Memoria (MB):
               total        used        free      shared  buff/cache   available
Mem:            7812         346        7081           3         384        7315
Swap:           2048           0        2048
---- 22:52:08 ----
    PID %CPU %MEM COMMAND
      1  0.3  0.1 systemd
     63  0.1  0.1 systemd-journal
     88  0.1  0.0 systemd-udevd
    133  0.1  0.1 systemd-resolve
Memoria (MB):
               total        used        free      shared  buff/cache   available
Mem:            7812         347        7080           3         384        7315
Swap:           2048           0        2048
---- 22:52:10 ----
    PID %CPU %MEM COMMAND
      1  0.3  0.1 systemd
     63  0.1  0.1 systemd-journal
     88  0.1  0.0 systemd-udevd
```

Por último, con la opción 5 se pueden matar procesos solo con conocer su PID



```
Opción: 5
PID a matar: 510
SIGTERM enviado a 510
```

Simulación de interbloqueo

Un interbloqueo no es mas que el resultado de dos procesos que intentan tomar dos recursos en distinto orden, esto genera un bloqueo entre los dos procesos por el recurso. Para simularlo, se generarán dos scripts, los cuales intentarán tomar dos recursos simulados.

Se crea el primer proceso

```
# Proceso A: primero R1, luego R2
(
  flock -x 200
  echo "A: obtuvo R1"
  sleep 2
  echo "A: intentando R2..."
  flock -x 201
  echo "A: obtuvo R2 (fin)"
) 200>"$R1" 201>"$R2"
```

Se crea el segundo proceso

```
# Proceso B: primero R2, luego R1 (orden inverso)
(
  flock -x 201
  echo "B: obtuvo R2"
  sleep 2
  echo "B: intentando R1..."
  flock -x 200
  echo "B: obtuvo R1 (fin)"
) 200>"$R1" 201>"$R2"
```

Se brindan los permisos necesarios

```
jaiva@Jaiva:~$ chmod +x procA.sh procB.sh
```

Se ejecutan los procesos intentando hacerlo al mismo tiempo

```
jaiva@Jaiva:~$ ./procA.sh & ./procB.sh & sleep 5 jobs -1
[1] 884
[2] 885
```

Como se nota, A tiene R1 y quiere R2; B tiene R2 y quiere R1. Esto es un interbloqueo.

```
jaiva@Jaiva:~$ A: obtuvo R1
B: obtuvo R2
A: intentando R2...
B: intentando R1...
```

Soluciones para no caer en interbloqueos

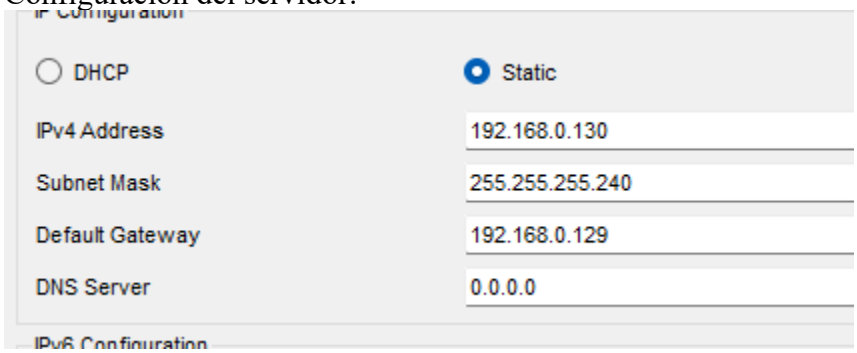
1. Aplicar un orden único para la toma de recursos, por ejemplo, siempre usar Recurso 1 primero y luego el recurso 2.
2. Usar timeouts y reintentos para evitar esperas indefinidas, esto hace que los procesos intenten nuevamente tomar el recurso, el mismo ya no estará en uso y así se evita el interbloqueo.

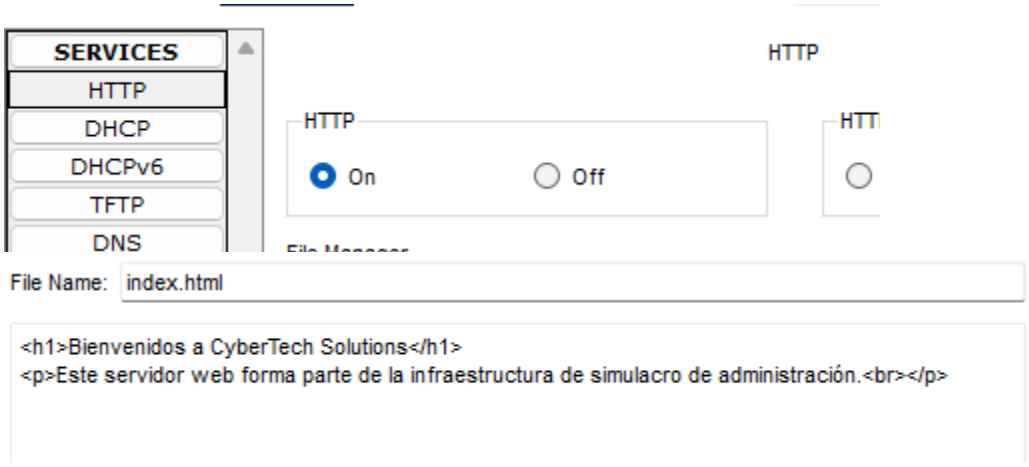
Implementación de servicios de red

Antes de implementar todos los servicios de red solicitados, se debe de configurar la red, con sus vlans, ips y demás.

- Se utiliza el método router on a stick, con subinterfaces para cada VLAN
- Se crearon las VLANs correspondientes en los switches de acceso y distribución.
- Los puertos hacia los equipos finales se configuraron como *access*, y los enlaces entre switches y hacia el router se configuraron como *trunk* usando encapsulación 802.1Q.
- Se habilitó el servicio HTTP en el servidor SRV1 con dirección IP 192.168.0.130, configurando una página informativa que muestra el mensaje “Bienvenidos a CyberTech Solutions”.
- Los equipos de las diferentes VLANs acceden mediante la URL http://192.168.0.130.

Configuración del servidor:





Ingreso al servidor desde el navegador de los otros equipos de la red