

**Instituto Politecnico Nacional**

**ESCOM “ESCUELA SUPERIOR DE CÓMPUTO”**

*ADMINISTRACIÓN DE SERVICIOS EN RED*

*ENRUTAMIENTO MÚLTIPLE*

PROFE: RICARDO MARTÍNEZ ROSALES

ALUMNO: Rojas Alvarado Luis Enrique

GRUPO: 4CM1

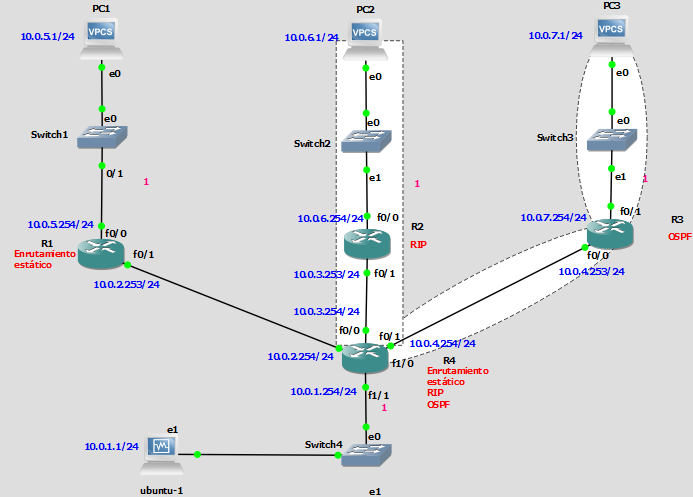
**Objetivo**

Desarrollar un programa en Python que sea capaz de levantar el enrutamiento estático y dinámico de la columna vertebral de una topología con múltiples métodos de enrutamiento.

**Implementación**

Implementar en GNS3 la siguiente topología, donde únicamente se van a configurar las interfaces indicadas en enrutadores, MV y VPCS y un usuario admin con contraseña admin en ssh de los enrutadores.

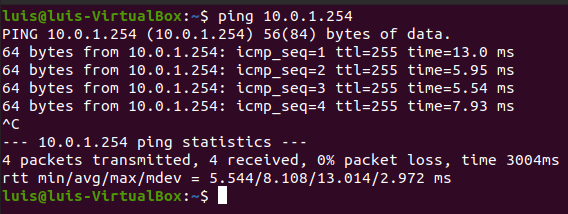
Desarrollar un programa en Python que correrá en la máquina virtual host1 y que sea capaz levantar los distintos métodos de enrutamiento indicados en la topología y permitir que exista conectividad en toda la red.



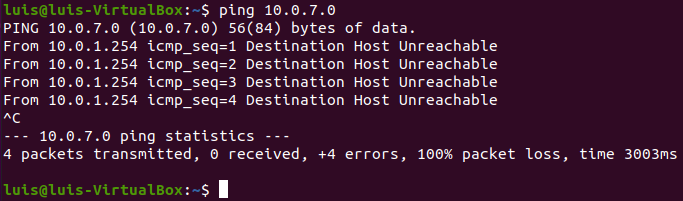
Se configuran los routers R1 con el enrutamiento estático, el R2 con el enrutamiénto dinámico RIP y el R3 con el protocolo de enrutamiento OSPF. Cabe destacar que en el R2 solo se configuran las redes 10.0.6.0 y 10.0.3.0 con el protocolo RIP y para el R3 solo se debe configurar las redes 10.0.7.0 y la 10.0.4.0. El R1 se tendrá que configurar toda la red con 0.0.0.0 para evitar problemas al enrutar el R4. Cabe destacar que se preconfiguró las interfaces antes de levantar ambos protocolos dinámicos. Así como también las interfaces del R4 posteriormente a enrutar con Python y flex.

Puesto que hemos configurado todos los routers con ssh, no deberíamos de poder acceder a los routers R1, R2, R3 debido a que no está enrutado el R4 y la máquina virtual no puede alcanzar esas redes.

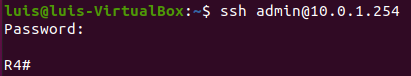
Haciendo ping a la puerta de enlace de la máquina virutal.



Haciendo ping a la red de la máquina PC3 de GNS3. Como se podrá observar, la red es inalcanzable.



Accediendo por la puerta de enlace 10.0.1.254 (R4) a través de SSH.

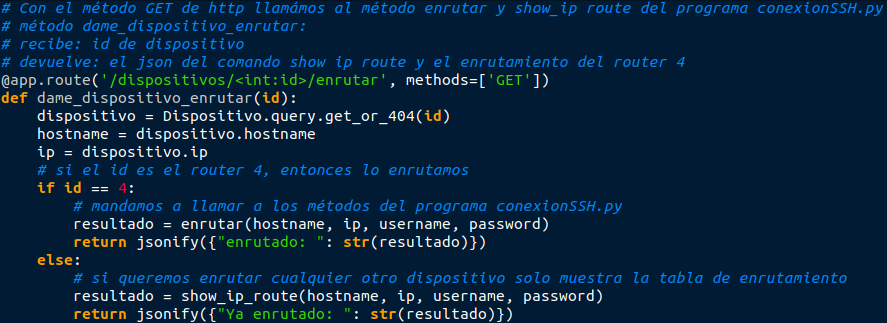


Accediendo por la puerta de enlace 10.0.7.254 (R3) a través de SSH.



Para configurar el R4 tendremos que usar el framework de Python flask para visualizar la información en una página web.

Se creó un método de flask llamado “enrutar”. En el que con un comando GET de HTTP, y colocando la URL en el navegador <http://127.0.0.1:5000/dispositivos/4/enrutar> agregando el número de dispositivo (en este caso el 4), y llamando al método enrutar del programa conexiónSSH.py el cual hace la conexión por medio de SSH que se explica a continuación.



El método de flask muestra en la pantalla del navegador en forma de json, el resultado que devuelve el método enrutar del programa conexionSSH.py

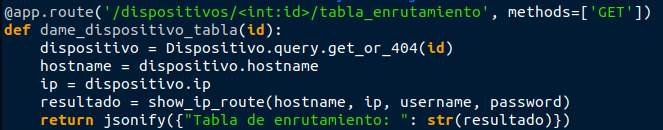
En el programa conexionSSH.py contiene la apertura de un archivo de texto en el cual nosotros colocaremos los comandos para enrutar el R4. Y se guarda en la variable “commands” que es de tipo lista.

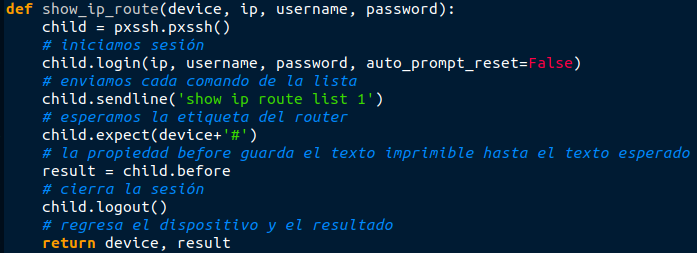


El método enrutar del programa conexionSSH.py recibe el nombre del dispositivo a enrutar, la dirección IP de acceso al router, el usuario y contraseña para ingresar a través de la conexión SSH. Con la ayuda de un bloque try - except se declara una variable en la que se guardará un archivo de texto para comprobar el enrutamiento al final de la ejecución de la función. Y se hace la respectiva conexión SSH con ayuda de la librería paramiko para la administración de redes, después de hacer la conexión se abre el archivo de texto que se mencionó para escribir en él los comandos que se mandarán y que adquirimos en otro archivo de texto, al final se regresa la salida que recibió la conexión de SSH y el nombre del dispositivo que enrutamos.

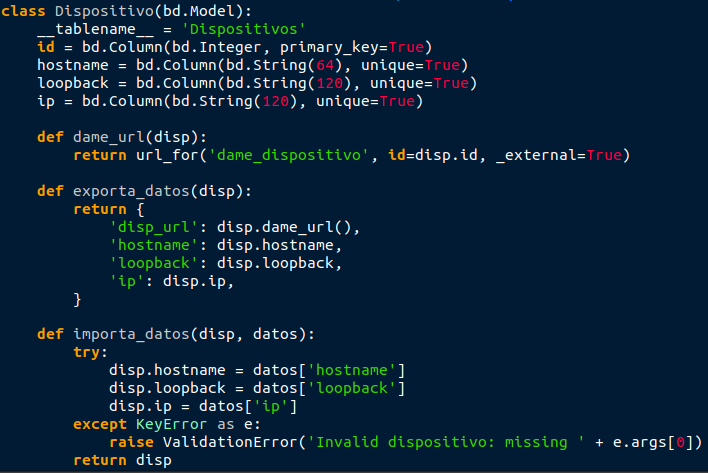


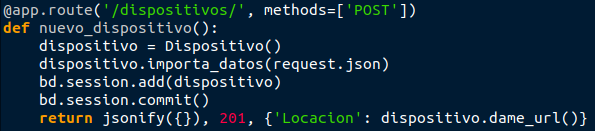
Para comprobar, usamos un método de flask en el cual accedemos con el método GET de HTTP y usando la URL en un navegador <http://127.0.0.1:5000/dispositivos/4/tabla_enrutamiento> en el cual llamaremos a una función del programa conexionSSH.py que muestra la tabla de enrutamiento con el comando “show ip route” por medio de la conexión SSH de pexpect. El método regresa en forma de JSON el texto que se recuperó al ingresar el comando remotamente, y se muestra en la ventana del navegador.



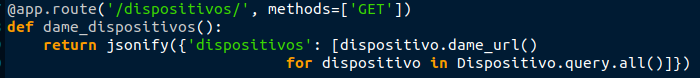


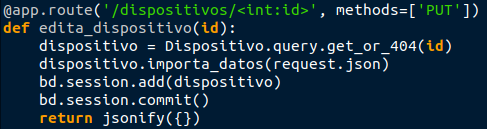
Cabe destacar que los dispositivos y sus direcciones IP se ingresaron por medio de una base de datos, usando SQLAlchemy. Y usando el método POST de HTTP y agregando la URL 127.0.0.1:5000/dispositivos/ y agregando las variables a agregar entre comillas: http POST http://127.0.0.1:5000/dispositivos/ 'hostname'='R1' 'loopback'='192.168.1.0' 'ip'='10.0.5.254'



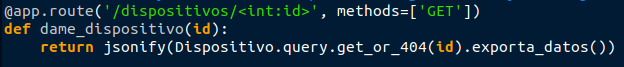


Con el comando GET se obtiene la información de todos los dispositivos ingresados a la base de datos. En formato JSON se visualiza en una página de un navegador.



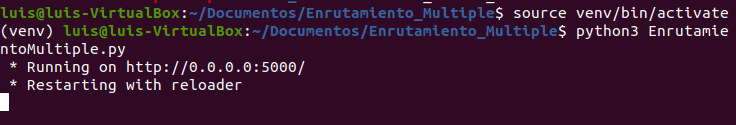
Con el método PUT podemos modificar cualquiera de los datos ingresados en la base de datos.

Finalmente, con el método GET y agregando el número de dispositivo a la URL en el navegador, se puede visualizar la información de dicho dispositivo ingresado.

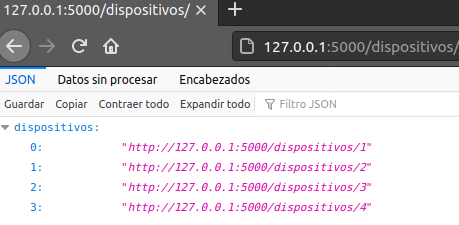


**PRUEBAS:**

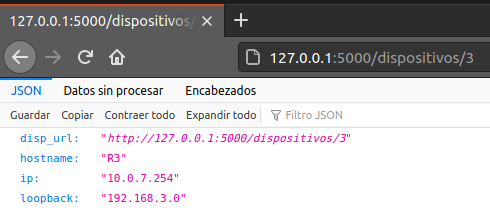
Desde una terminal de Linux, activamos nuestro entorno virtual y ejecutamos el programa de flask (el programa principal) y nos muestra que está ejecutándose en el servidor localhost. Por lo que procedemos a abrir una ventana de navegador.



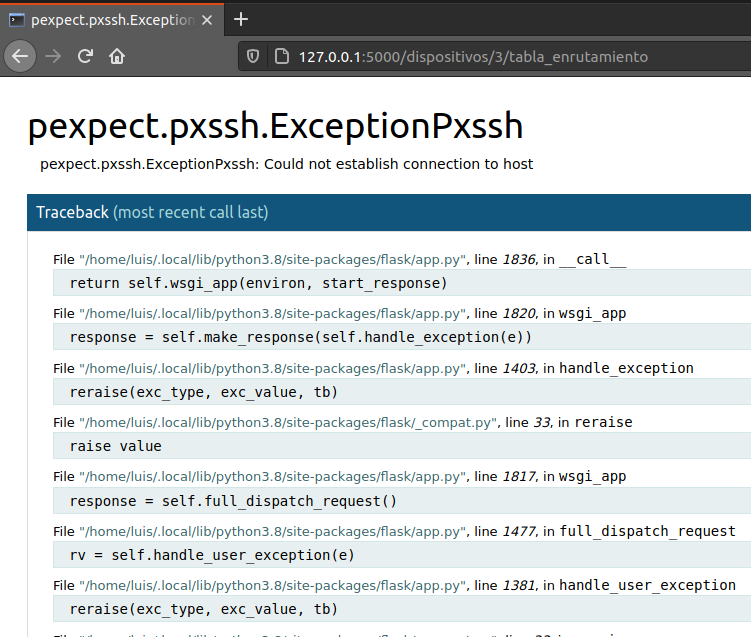
Como nos podemos darnos cuenta, despliega la información contenida en nuestra base de datos llenada con anterioridad. Nos indica la página de flask que tenemos 4 dispositivos.



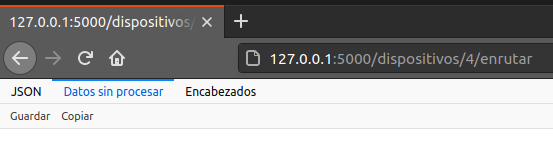
Al agregar el número de dispositivo, podemos visualizar la información que tenemos de él en la base de datos en formato JSON.



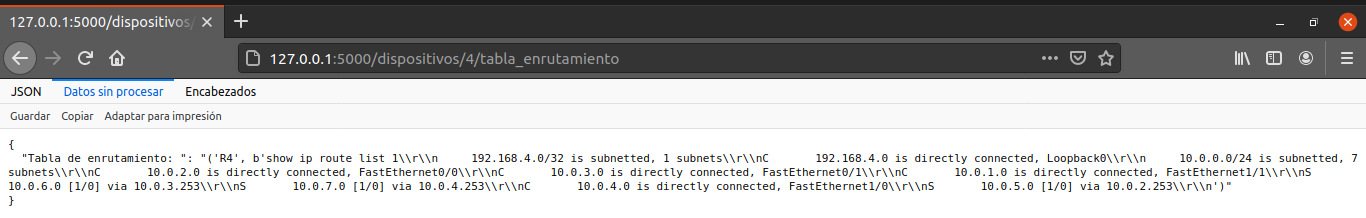
Si queremos ver su tabla de enrutamiento agregamos “tabla\_enrutamiento” al final para ver lo que nos sale.

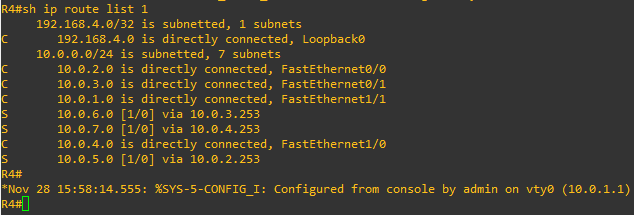


En este caso es una excepción de pexpect, puesto que aún no hemos enrutado el R4 que es el router de frontera y la máquina virtual no puede llegar a ese router. Por lo que procederemos a enrutar el R4.

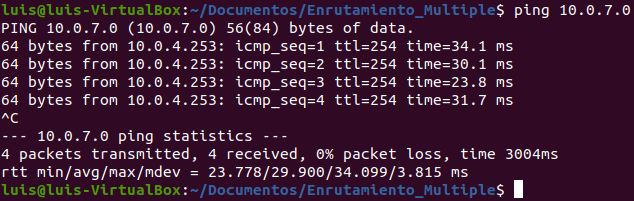


Podemos observar ahora que no nos arroja ningún tipo de excepción o error. Por lo que al comprobar mostrando su tabla de enrutamiento podemos observar que todas las redes ahora son accesibles.

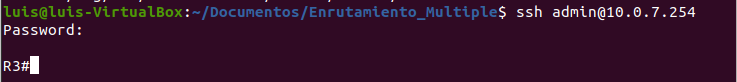




Por lo que si hacemos ping a la misma red de la PC3 de GNS3 ahora podemos llegar.



Y si accedemos por SSH podemos ingresar de igual manera.



Con esto hemos conlcuido la configuración remota de un router usando flask.