

# Requisitos generales del proyecto final de Administración de Servicios en Red (primera presentación).

## Objetivo

Desarrollar una API que ofrezca un conjunto de herramientas para la gestión de una red de cómputo mediante Python en una arquitectura de transferencia de estado representacional o REST.

## Características y funciones

### Características generales

1. Comunicarse con los dispositivos de red de forma segura usando SSH.
2. Cuando se hace una solicitud vía URL a un dispositivo, interfaz o cualquier otro elemento que depende de su presencia en la topología inexistente, debe de devolver un código de HTTP 404.
3. Los datos de los comandos HTTP y sus respuestas se harán con formato Json.
4. La identificación de las interfaces deberá ser por el nombre (o inicial), slot, guion bajo y número, por ejemplo: fastethernet1\_0 o f1\_0.
5. La información necesaria de la topología y otros elementos se deberá de guardar en una base de datos.
6. Los roles de los routers corresponden a Edge como frontera, R1 y R2 como núcleo y el resto como hoja.
7. La asignación de Ip administrativas será .1 para Edge, .2 y .3 para R1 y R2 respectivamente y el resto para los routers TOR.

### Funciones

1. Generar una representación gráfica de la topología de una red, detectándola de forma dinámica.
2. CRUD de usuarios en los dispositivos de red, globales y por dispositivo, con gestión de los permisos y accesos vía SSH.
3. Obtener la información general de los enrutadores en la red mediante comandos SNMP.
4. Obtener la información general por interfaz del enrutador mediante comandos SNMP.
5. Generar y controlar un proceso de monitoreo de octetos de entrada por interfaz.
6. Generar y controlar la trampa de linkup y linkdown por interfaz.
7. Generar un gráfico que muestre, el monitoreo (punto 5) y la activación de trampas (punto 6) en una interfaz.

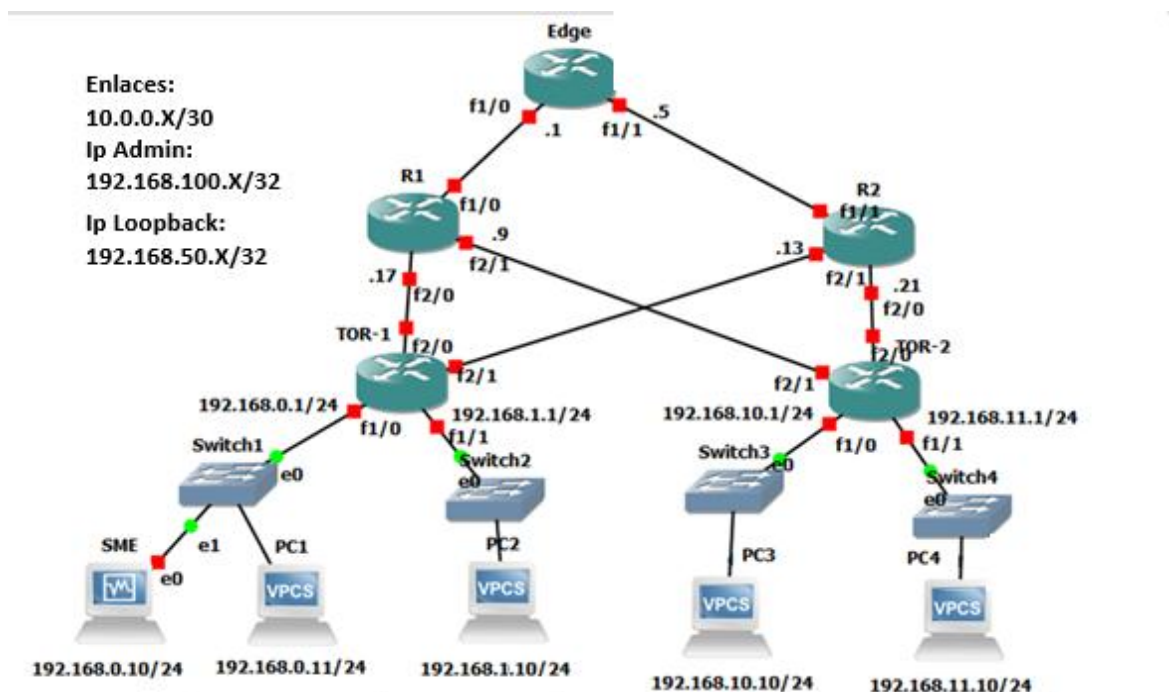
Este monitoreo debe de poderse realizar en cualesquiera de las interfaces de la topología de muestra y deberá de poderse activar tanto el de octetos como el de linkup y linkdown de forma independiente y autónoma.

En el monitoreo de octetos deberá de poderse indicar el intervalo de tiempo entre muestra y muestra.

La gráfica deberá de poder mostrar los octetos de entrada por intervalo de tiempo de muestreo o el estado de la interfaz. En el caso de que los dos muestreos estén representados en la misma gráfica hay que guardar las proporciones para que ambas se vean con facilidad.

### Características de la revisión

La topología sobre la que se revisará el software es dinámica, es decir, que se pedirá que cambie durante la revisión. Basado en la siguiente topología:



Los dispositivos de la red de prueba tendrán únicamente definido las direcciones IP de las interfaces, algún enrutamiento dinámico y un usuario con permisos de administración y acceso por ssh.

### Primera presentación.

Implementar las siguientes funciones de la API-REST

Función	Ruta	Comando HTTP	
CRUD usuarios	/usuarios	GET	POST
		Regresar json con todos los usuarios existentes en los routers, incluyendo nombre, permisos y dispositivos donde existe (URL a routers donde exista cada usuario).	Agregar un nuevo usuario a todos los routers, regresar json con la misma información de GET pero solo del usuario agregado.
		PUT	DELETE
		Actualizar un usuario en todos los routers, regresar json con la misma información de GET pero del usuario actualizado.	Eliminar usuario común a todos los routers, recuperar json con la misma información de GET, pero del usuario eliminado.
Enrutadores	/routers	GET	
		Regresa la información general de todos los routers de la topología. Incluyendo Nombre, IP loopback, IP administrativa, rol, empresa, Sistema operativo y ligas a las interfaces activas.	
	/routers/<hostname>/	GET	
		Regresa en formato json la información general del router definido. Incluyendo Nombre, IP loopback, IP administrativa, rol, empresa, Sistema operativo y ligas a las interfaces activas.	
Interfaces por router	/routers/<hostname>/interfaces	GET	
		Regresa en formato json la información general de la interfaz del router definido. Incluyendo tipo, número, IP, mascara de rubred, estado y la liga al router que está conectado, si es el caso.	
CRUD usuarios por enrutador	/routers/<hostname>/usuarios/	GET	POST
		Regresar json con los usuarios existentes en el router específico, incluyendo nombre y permisos.	Agregar un nuevo usuario al router específico, regresar json con la misma información de GET, pero del usuario agregado.
		PUT	DELETE
		Actualizar un nuevo usuario al router específico, regresar json con la misma información de GET, pero del usuario actualizado.	Eliminar usuario del router, recuperar json con la misma información de GET, pero del usuario eliminado.

Detectar topología	/topologia	GET	POST
		Regresar json con los routers existentes en la topología y ligas a sus routers vecinos	Activa un demonio que cada 5 minutos explora la red para detectar cambios en la misma.
		PUT	DELETE
		Permite cambiar el intervalo de tiempo en el que el demonio explora la topología.	Detiene el demonio que explora la topología.
Gráfica de topología	/topologia/grafica	GET	
		Regresa un archivo en algún formato gráfico donde se pueda visualizar la topología existente.	
Monitoreo de interfaz	/routers/<hostname>/interfaces/<Interfaz>/octetos/<tiempo>	GET	POST
		Regresar json con todas las muestras que se tengan hasta el momento del monitoreo de la interfaz.	Activa el monitoreo de octetos de entrada en la interfaz respectiva, por el tiempo indicado. Regrese un json con el estado del proceso de monitoreo.
		DELETE	
		Para el proceso de monitoreo de octetos de interfaz. Regresa un json con el estado del proceso de monitoreo.	
Gestión de traps de linkup y linkdown	/routers/<hostname>/interfaces/<Interfaz>/estado	GET	POST
		Regresa un json con el estado de la interfaz.	Activa la captura de trampas linkup y linkdown de una interfaz en específico. Regresa un json con el estado de la interfaz.
		DELETE	
		Para la captura de trampas linkup y linkdown de una interfaz en específico. Regresa un json con el estado de la interfaz.	
Gráfica de monitoreo	/routers/<hostname>/interfaces/<Interfaz>/grafica	GET	
		Regresa un archivo jpg o equivalente que contiene la gráfica de los monitoreos que se tengan activos en dicha interfaz.	

## Entrega de la primera presentación

Subir a Moodle un archivo pdf con las diapositivas usadas para la presentación.

### Rúbrica primera presentación

Concepto	Funcionamiento	
	Regular	Correcto
<b>Presentación del proyecto en clase.</b> El equipo presenta en clase de forma clara las características principales del diseño de su API-REST, usando apoyo visual para su explicación.	0.0	0.5
<b>Funcionamiento del CRUD de usuarios en los dispositivos de red.</b> El equipo presenta en clase con una topología de muestra, como puede realizar altas, bajas y cambios de las cuentas de usuario de los dispositivos.	0.0	0.5
<b>Muestra la información más relevante de todos los enrutadores, tanto individual como general.</b> El equipo presenta en clase con una topología de muestra la información que, indicada de los enrutadores de la topología, tanto individual como general.	0.0	0.5
<b>Muestra la información de las interfaces.</b> El equipo presenta en clase con una topología de muestra la información de las interfaces de cada uno de los enrutadores.	0.0	0.5
<b>Gráfico de la topología.</b> Se muestra la gráfica de la topología dinámicamente.	1.0	2.0
<b>Activación o paro del muestreo de octetos de entrada.</b> Las URL respectivas muestran el inicio, y para del muestro de los octetos de entrada en cualquiera de las interfaces usadas en la topología.	0.5	1.0
<b>Recuperación de los datos muestreados de octetos de entrada y salida.</b> La URL correspondiente nos regresa los muestreos obtenidos hasta el momento en el intervalo de tiempo solicitado.	1.0	2.0
<b>Activación o para de la captura de traps en interfaz.</b> La URL correspondiente activa o desactiva la captura de traps en cualesquiera de las interfaces usadas en la topología.	1.0	2.0
<b>Gráfico del muestreo.</b> Se muestra la gráfica del muestreo realizado hasta el momento en la interfaz, en una o dos imágenes.	0.5	1.0
<b>TOTAL</b>		10.0