MANUAL DE USUARIO

Equipo Los Enredados

7CM2
Integrantes: Carrillo Estrada Héctor Enrique
Paniagua Arroyo Sergio Iván

Administración de Servicios en Red

Objetivos

Objetivo General

Diseñar e implementar una topología de red utilizando una máquina virtual que permita la gestión y verificación de routers conectados a través de SSH o Telnet, empleando scripts desarrollados en Python para la configuración y monitoreo de interfaces, listas de acceso, DHCP y NAT.

Objetivos Específicos

1. **Configurar la Máquina Virtual:**
   - Instalar y configurar una máquina virtual que servirá como punto de acceso central para la administración de la red.
   - Configurar las herramientas necesarias (SSH, Telnet, Python) en la máquina virtual.
2. **Desarrollar Scripts en Python:**
   - Crear scripts en Python que permitan la conexión a los routers a través de SSH o Telnet.
   - Implementar funcionalidades en los scripts para verificar y mostrar la configuración de interfaces, listas de acceso, DHCP y NAT en los routers conectados.
3. **Configurar Routers:**
   - Configurar las interfaces de los routers para permitir el acceso remoto seguro a través de SSH o Telnet.
   - Configurar listas de acceso para controlar el tráfico de red según políticas definidas.
   - Configurar el servicio DHCP en los routers para asignar direcciones IP dinámicamente a los dispositivos conectados.
   - Configurar NAT (Network Address Translation) en los routers para permitir la comunicación entre diferentes redes.
4. **Implementar y Verificar la Topología de Red:**
   - Crear una topología de red que conecte la máquina virtual con los routers configurados.
   - Probar la conectividad entre la máquina virtual y los routers para asegurar que el acceso remoto sea funcional.
   - Utilizar los scripts desarrollados en Python para verificar la configuración y operación de las interfaces, listas de acceso, DHCP y NAT en los routers.
5. **Documentar el Proceso y Resultados:**
   - Documentar detalladamente el proceso de configuración de la máquina virtual, el desarrollo de los scripts, la configuración de los routers y la implementación de la topología de red.
   - Presentar un informe con los resultados obtenidos, incluyendo las configuraciones realizadas y las pruebas de verificación ejecutadas.

Introducción

En la era digital actual, la administración eficiente de redes es fundamental para el correcto funcionamiento de cualquier organización. Las redes deben ser configuradas, monitoreadas y gestionadas de manera precisa para asegurar que todos los dispositivos interconectados operen de manera óptima y segura.

Este proyecto tiene como objetivo diseñar e implementar una topología de red que permita la administración centralizada de routers utilizando una máquina virtual como punto de acceso. A través de la configuración de servicios como SSH (Secure Shell) y Telnet, y el desarrollo de scripts en Python, se facilitará la conexión remota y la verificación de configuraciones críticas en los routers. Estos scripts permitirán la supervisión y gestión de interfaces de red, listas de acceso (ACL), configuraciones de DHCP (Dynamic Host Configuration Protocol) y NAT (Network Address Translation).

La máquina virtual configurada será el núcleo de nuestra red, desde donde se podrán ejecutar comandos y scripts para administrar y monitorear los routers conectados. La configuración de los routers incluirá la habilitación de acceso remoto seguro, la implementación de políticas de listas de acceso para el control de tráfico, la configuración de servicios DHCP para la asignación dinámica de direcciones IP y la configuración de NAT para la traducción de direcciones de red.

Este enfoque práctico no solo reforzará los conceptos teóricos aprendidos en la materia, sino que también proporcionará una experiencia práctica valiosa en la configuración y administración de redes reales.

Configuración en GNS3

1. Configuración de routers y VPCs: En GNS3, puedes agregar routers virtuales, como Cisco IOSv o Cisco CSR1000v, así como VPCs (Virtual PC) para emular dispositivos finales. Puedes arrastrar y soltar estos dispositivos desde la paleta de dispositivos de GNS3 a tu topología de red. Luego, puedes configurar la interfaz, las rutas, los protocolos de enrutamiento y otros aspectos utilizando la interfaz de configuración del router o VPC dentro de GNS3.

2. Conectividad SSH y Telnet: GNS3 proporciona la capacidad de acceder a los routers y VPCs a través de conexiones SSH y Telnet. Para configurar la conectividad SSH, debes asegurarte de que los routers y VPCs tengan configurado un nombre de host y una clave SSH. Para la conectividad Telnet, debes configurar una contraseña Telnet en los dispositivos. Luego, puedes utilizar herramientas de línea de comandos, como PuTTY o el propio terminal de GNS3, para establecer las conexiones SSH o Telnet a los dispositivos.

3. Visualización de la configuración y los protocolos de enrutamiento: GNS3 proporciona una interfaz gráfica intuitiva donde puedes ver y editar la configuración de los routers y VPCs. Puedes acceder a la configuración de cada dispositivo haciendo clic derecho sobre él y seleccionando "Configuración". Además, GNS3 te permite visualizar las tablas de enrutamiento y los protocolos de enrutamiento configurados en los routers, lo que facilita la comprensión de la topología de la red y el flujo de tráfico.

Configuración de Python en VM

1. Preparar la máquina virtual: Crea una nueva máquina virtual en GNS3 utilizando una imagen de sistema operativo compatible con Python, como Ubuntu, CentOS o Windows.

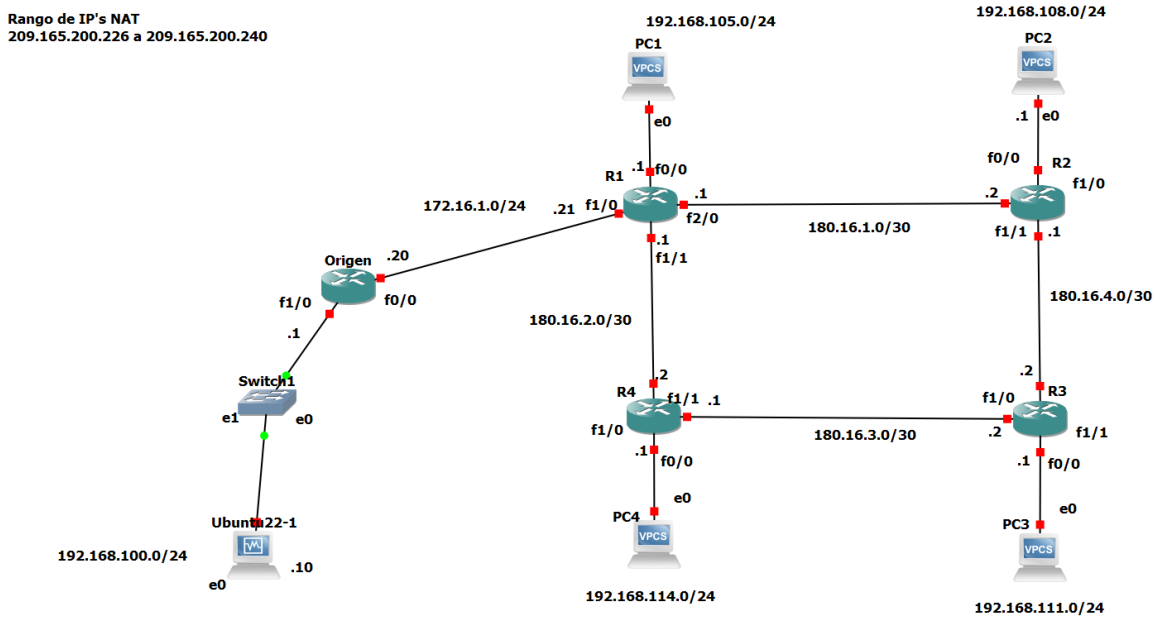
2. Instalar Python: Una vez que la máquina virtual esté lista, accede a ella y abre una terminal o línea de comandos. Utiliza el administrador de paquetes correspondiente de tu sistema operativo para instalar Python. Por ejemplo, en Ubuntu, puedes usar el comando sudo apt install python3 para instalar Python 3.

3. Instalar bibliotecas y módulos necesarios: Dependiendo de los requerimientos del programa de Python que estás desarrollando, es posible que necesites instalar bibliotecas adicionales. Utiliza el administrador de paquetes o el gestor de paquetes de Python, como pip, para instalar las bibliotecas necesarias. Por ejemplo, si necesitas instalar la biblioteca SNMP para interactuar con SNMP, puedes usar el comando pip install pysnmp para instalarla.

4. Desarrollar y ejecutar el programa: Utiliza tu editor de código favorito para desarrollar el programa de Python según los objetivos establecidos. Asegúrate de importar las bibliotecas necesarias y escribir el código para gestionar y visualizar la configuración de la red, los protocolos de enrutamiento y otros aspectos mencionados en el objetivo general.

5. Ejecutar el programa: Una vez que hayas terminado de desarrollar el programa, guárdalo y ejecútalo en la máquina virtual. Puedes ejecutar el programa desde la terminal utilizando el intérprete de Python.

Topología



La topología de GNS3 creada para abordar el objetivo general mencionado puede incluir varios componentes y configuraciones de red.

1. Routers virtuales: En la topología, puedes incluir routers virtuales, como Cisco IOSv o Cisco CSR1000v, que actúan como dispositivos de enrutamiento en la red simulada. Estos routers se configuran con interfaces, rutas y protocolos de enrutamiento, según los requerimientos de tu red.

2. VPCs: Además de los routers, puedes incluir VPCs (Virtual PC) en la topología. Estas son máquinas virtuales que actúan como dispositivos finales en la red, como computadoras o servidores. Puedes configurar las interfaces y ajustar la configuración de red en cada VPC.

3. Conectividad entre dispositivos: Para establecer la conectividad entre los routers y VPCs, debes configurar las interfaces de red en cada dispositivo y asignarles direcciones IP. Esto permite que los dispositivos se comuniquen entre sí a través de la red simulada.

4. Protocolos de enrutamiento: Configura los protocolos de enrutamiento necesarios, como OSPF (Open Shortest Path First) o BGP (Border Gateway Protocol), en los routers virtuales. Estos protocolos permiten que los routers compartan información de enrutamiento y tomen decisiones sobre cómo dirigir el tráfico en la red simulada.

5. Servicios de red: Puedes incluir servicios de red importantes, como un servidor DHCP (Dynamic Host Configuration Protocol) para asignar automáticamente direcciones IP a los dispositivos en la red, un servidor DNS (Domain Name System) para resolver nombres de dominio y un servidor NAT (Network Address Translation) para traducir direcciones IP entre redes.

6. Monitoreo de red: Para implementar la aplicación de SNMP y monitorear la red, configura los routers virtuales para admitir SNMP y generar notificaciones de caída de enlaces cuando se detecte un cambio en el estado de una interfaz. Utiliza una herramienta de monitoreo de SNMP para recibir y visualizar estas notificaciones. Para esta implementación de dicha topología es la culminación de cada practica que se realizó durante el semestre impartido por el profesor, en donde se abarco cada tema visto, así como cada comando usado, todo unido en una sola topología para una pequeña muestra a través de un proyecto de semestre final.

Librerías Utilizadas

Tkinter

Tkinter es una librería del lenguaje de programación Python y funciona para la creación y el desarrollo de aplicaciones de escritorio. Esta librería facilita el posicionamiento y desarrollo de una interfaz gráfica de escritorio con Python. Tkinter es el paquete estándar de Python para interactuar con Tkt.[7] De acuerdo con la documentación de Python, TK se describe a sí mismo como el único toolkit o kit de herramientas para el desarrollo de una interfaz gráfica de usuario (GUI) que funciona en todos los sistemas operativos, es decir, funciona en Windows, Mac OS y Linux. Además, está diseñado y preparado para lenguajes dinámicos con un alto nivel, como pueden ser Tcl, Ruby, Perl o python aplicaciones de escritorio entre otros.[7]

tkinter.ttk

El módulo tkinter.ttk proporciona acceso al conjunto de widgets temáticos Tk, introducido en Tk 8.5. Si Python no se ha compilado con Tk 8.5, todavía se puede acceder a este módulo si se ha instalado Tile. El método anterior que utiliza Tk 8.5 proporciona ventajas adicionales, incluida la representación de fuentes suavizada en X11 y la transparencia de ventanas [8]

tkinter.messagebox

El módulo tkinter.messagebox proporciona una clase base de plantilla, así como una variedad de métodos convenientes para configuraciones de uso común. Los cuadros de mensaje son modales y devolverán un subconjunto de (Verdadero, Falso, OK, Ninguno, Sí, No) según la selección del usuario.

tkinter.font

El módulo tkinter.font proporciona la clase Font para crear y usar fuentes con nombre.[10] Los diferentes pesos e inclinaciones de la fuente son:

- tkinter.font.NORMAL
- tkinter.font.BOLD
- tkinter.font.ITALIC
- tkinter.font.ROMAN [10]

telnetlib

El módulo telnetlib proporciona una clase Telnet que implementa el protocolo Telnet. Además, proporciona constantes simbólicas para los caracteres de protocolo (ver más abajo) y para las opciones telnet. Los nombres simbólicos de las opciones de telnet siguen las definiciones de arpa/telnet.h, conel TELOPT_ principal eliminado. Para conocer los nombres simbólicos de las opciones que tradicionalmente no se incluyen en arpa/telnet.h, consulte la propia fuente del módulo.[11]

time

Este módulo proporciona varias funciones relacionadas con el tiempo. Aunque este módulo siempre está disponible, no todas las funciones están disponibles en todas las plataformas. La mayoría de las funciones definidas en este módulo llaman a funciones C de la biblioteca de la plataforma con el mismo nombre. A veces puede ser útil consultar la documentación de la plataforma, ya que la semántica de estas funciones varía entre plataformas.[12]

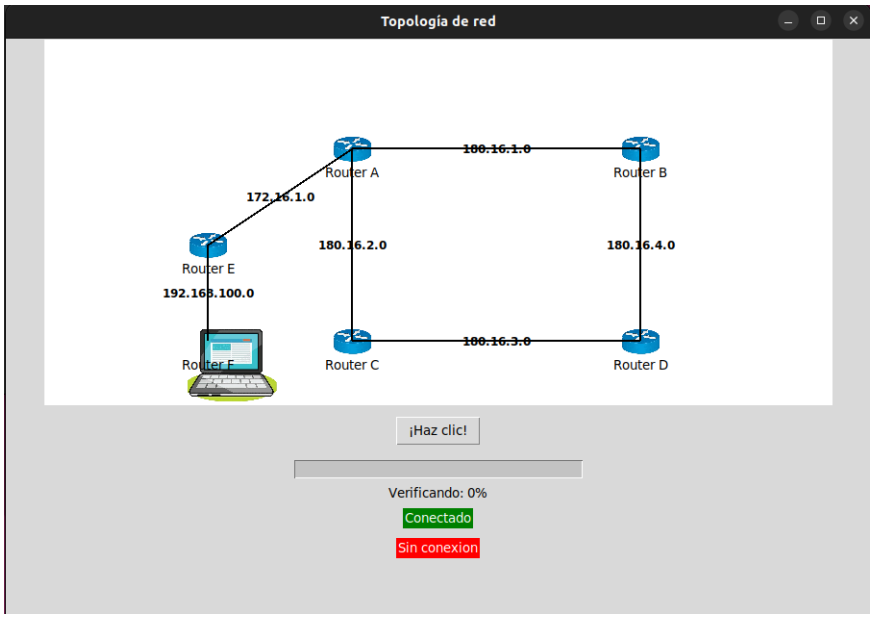
Asyncio

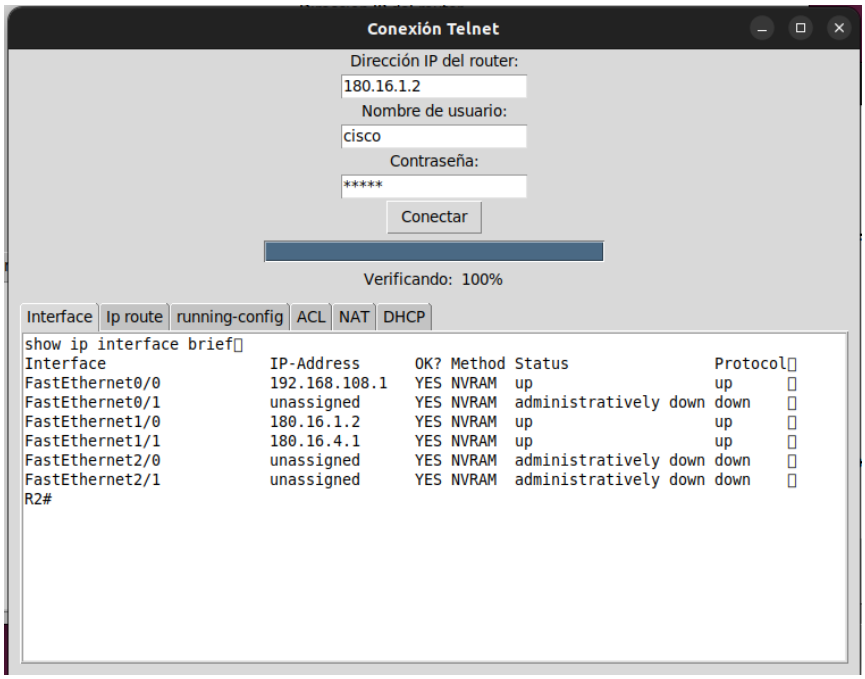
asyncio es una biblioteca para escribir código concurrente utilizando la sintaxis async/await. [13]

asyncio es utilizado como base en múltiples frameworks asíncronos de Python y provee un alto rendimiento en redes y servidores web, bibliotecas de conexión de base de datos, colas de tareas distribuidas, etc.[13]

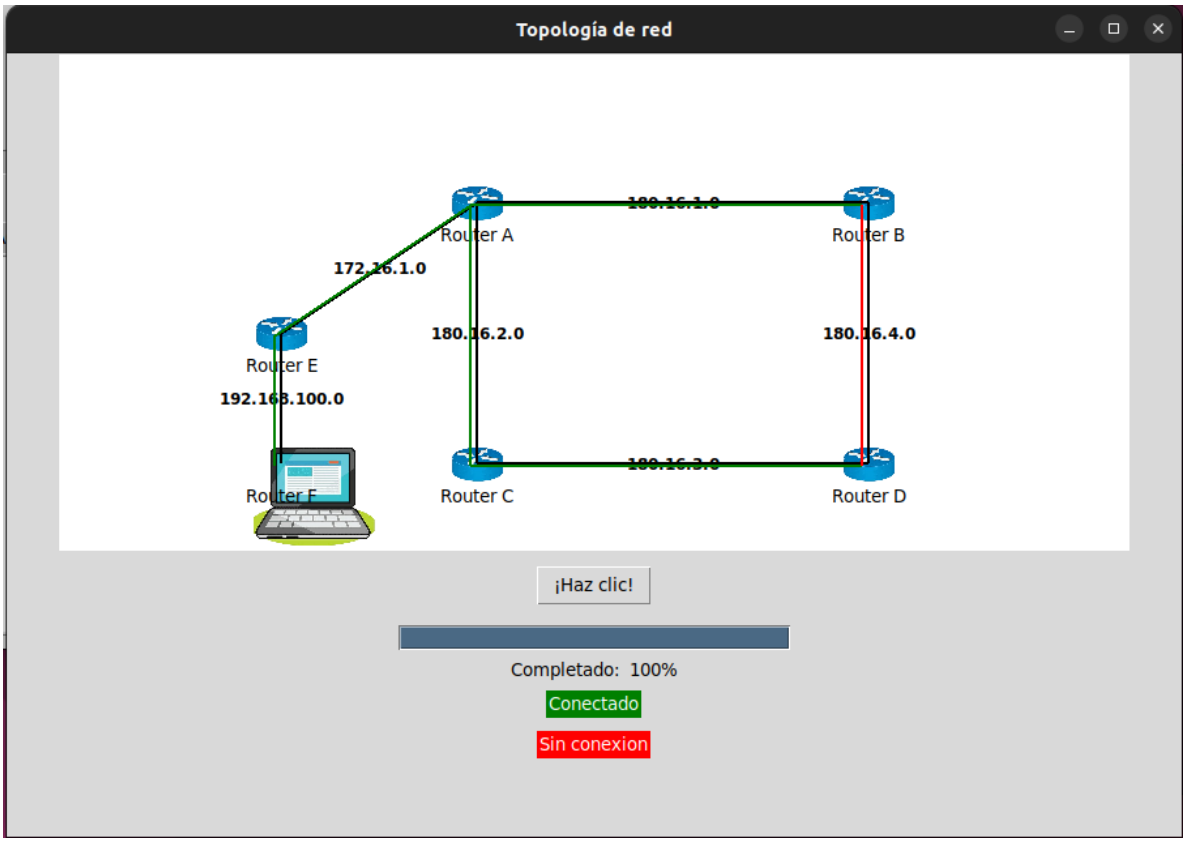
Funcionamiento del Proyecto

A través del código en Python, se generó un programa útil para poder interactuar de manera intuitiva con cada router configurado previamente dentro de la topología seleccionada para el proyecto, de manera que, a través de ejecutarlo en la Máquina Virtual, este pueda contar con una simulación física de las conexiones y al dar clic en cualquier router, se pueda acceder a través de telnet, al router y revisar las configuraciones hechas, como las interfaces, configuración de arranque, dhcp, nat, dns , listas de acceso, etc.





Además, se simularon las conexiones entre routers para que, si una comunicación fue suspendida, se pueda revisar que parte de la conexión falló.



Conclusiones

Carrillo Estrada Héctor Enrique: La realización de este proyecto ha demostrado la importancia de una administración de redes eficiente y segura mediante la implementación de servicios esenciales como DHCP, NAT, DNS, listas de acceso, y la conexión remota a través de Telnet y SSH. La configuración de DHCP y NAT es fundamental para garantizar que todos los dispositivos de una red puedan comunicarse correctamente y tener acceso a recursos externos, optimizando el uso de direcciones IP. La implementación de listas de acceso permite un control granular del tráfico de red, asegurando que solo el tráfico autorizado tenga acceso a los recursos críticos.

El uso de SSH y Telnet ha mostrado ser indispensable para la administración remota segura y eficiente de los routers. Mientras que Telnet ofrece una solución básica, SSH proporciona una capa adicional de seguridad necesaria en entornos de producción. La automatización de tareas a través de scripts en Python ha mejorado la eficiencia y precisión en la gestión de la red, reduciendo el tiempo y esfuerzo requeridos para tareas repetitivas.

Paniagua Arroyo Sergio Iván: Este proyecto ha proporcionado una comprensión profunda de la administración de redes mediante la configuración y gestión de servicios clave como DHCP, NAT, DNS, listas de acceso, y el acceso remoto a través de Telnet y SSH. El protocolo DHCP ha demostrado ser esencial para la asignación dinámica y eficiente de direcciones IP, simplificando la gestión de direcciones en redes complejas. NAT ha permitido la conexión segura y eficiente entre diferentes redes, facilitando el acceso a recursos externos sin comprometer la seguridad interna.

La configuración de listas de acceso ha sido crucial para la implementación de políticas de seguridad que protegen la red contra accesos no autorizados y tráfico malicioso. La conexión remota mediante Telnet y SSH ha resaltado la necesidad de herramientas robustas para la administración remota, con SSH proporcionando una capa de seguridad adicional indispensable para proteger las comunicaciones de administración.

El uso de scripts en Python ha demostrado ser una herramienta poderosa para la automatización de tareas de administración, mejorando la eficiencia operativa y reduciendo el riesgo de errores humanos. Este proyecto ha subrayado la importancia de integrar servicios de red clave y herramientas de automatización para construir y mantener una infraestructura de red eficiente, segura y resiliente, preparada para enfrentar los desafíos actuales y futuros en el ámbito de la administración de servicios en red.

Bibliografías

[1] (S/f). Unican.es. Recuperado el 20 de junio de 2023, de https://repositorio.unican.es/xmlui/bitstream/handle/10902/16949/419449.pdf?sequence=1

[2] Walton, A. (2017, diciembre 27). Listas de Control de Acceso (ACL): Funcionamiento y Creación. CCNA desde Cero. https://ccnadesdecero.es/listascontrol-acceso-acl-router-cisco/

[3] De Luz, S. (2021, noviembre 12). Qué es el DHCP, funcionamiento y ejemplos de configuración. RedesZone. https://www.redeszone.net/tutoriales/internet/que-esprotocolo-dhcp/

[4] Pérez, E. (2020, mayo 28). Como configurar NAT en un router Cisco. Estudia Redes. https://estudiaredes.com/cisco/como-configurar-nat-en-un-router-cisco/

[5] Telnet: ¿qué es y cómo se activa? (2022, abril 25). IONOS Digital Guide; IONOS. https://www.ionos.mx/digitalguide/servidores/herramientas/telnet-el-protocolo-paracualquier-plataforma/

[6] de Zúñiga, F. G. (2022, agosto 17). SSH: qué es y cómo funciona este protocolo. Blog de arsys.es; Arsys Internet. https://www.arsys.es/blog/ssh

[7] ¿Qué es Tkinter? (2022, junio 10). KeepCoding Bootcamps. https://keepcoding.io/blog/que-es-tkinter/

[8] tkinter.ttk — Tk widgets temáticos — documentación de Python - 3.9.16. (s/f). Python.org. Recuperado el 20 de junio de 2023, de https://docs.python.org/es/3.9/library/tkinter.ttk.html

[9] tkinter.messagebox — Indicadores de mensajes de Tkinter. (s/f). Python documentation. Recuperado el 20 de junio de 2023, de https://docs.python.org/es/3/library/tkinter.messagebox.html

[10] tkinter.font — Envoltorio de fuente Tkinter — documentación de Python - 3.9.16. (s/f). Python.org. Recuperado el 20 de junio de 2023, de https://docs.python.org/es/3.9/library/tkinter.font.html

[11] telnetlib — cliente Telnet. (s/f). Python documentation. Recuperado el 20 de junio de 2023, de https://docs.python.org/es/3/library/telnetlib.html

[12] time — Acceso a tiempo y conversiones. (s/f). Python documentation. Recuperado el 20 de junio de 2023, de https://docs.python.org/es/3/library/time.html

[13] asyncio — E/S Asíncrona. (s/f). Python documentation. Recuperado el 20 de junio de 2023, de https://docs.python.org/es/3/library/asyncio.html