**Tecnológico Nacional de México**

Instituto Tecnológico de Ensenada

Asignatura: Administración de redes.

Docente: Mario Cesar Medina Ortiz

Actividad: C3. Proyecto escaneo de red

Nombre del estudiante: **Jose Manuel Velasco García**

Matrícula: 19760627

Grupo: 8se

Fecha de entrega: 25 de octubre del 2024

**INDICE**

[**INTRODUCCION 3**](#_Toc95941325)

[**DESARROLLO DE LA ACTIVIDAD 4**](#_Toc95941326)

[**CONCLUSIONES 5**](#_Toc95941327)

[**REFERENCIAS BIBLIOGRAFICAS 6**](#_Toc95941328)

# 

# **INTRODUCCION**

La función de escaneo de red en este proyecto se implementa para identificar dispositivos activos en un rango de direcciones IP dentro de una red local específica. Utiliza la herramienta nmap para realizar un escaneo inicial y luego verifica la disponibilidad de cada dispositivo mediante el comando ping. Esto permite distinguir cuáles dispositivos están activos y respondiendo a solicitudes en el momento del escaneo.

### ¿Qué es un escaneo de red?

Un escaneo de red es el proceso de identificar y mapear los dispositivos conectados en una red específica. Generalmente, el escaneo incluye la recopilación de información sobre las direcciones IP y MAC de los dispositivos, además de la identificación de su disponibilidad en el momento del escaneo. Esto es útil para administradores de sistemas y redes, ya que permite monitorear el estado de la red, identificar posibles problemas de conectividad y detectar dispositivos no autorizados.

Escaneo en este Proyecto

Este código emplea nmap, una herramienta de código abierto para escanear redes, que identifica dispositivos y obtiene direcciones IP y MAC, mientras que ping permite verificar la actividad en tiempo real de cada IP detectada. Los resultados de este escaneo se muestran en una página web creada con Flask, donde los dispositivos están clasificados como activos o inactivos según su respuesta al ping.

# 

# **DESARROLLO DE LA ACTIVIDAD**

Para desarrollar esta practica ocupamos varios recursos los cuales son:

* Visual Studio Code.
* Python
* flask
* nmap

Creamos un nuevo proyecto.

El proyecto se llamara redess.

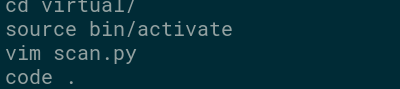


Una vez creado nuestra carpeta entramos en ella y creamos un entorno virtual.

Para crearlo usamos el comando python3 -m venv virtual (si no tenemos python lo descargamos <https://www.python.org/ftp/python/3.13.0/Python-3.13.0.tar.xz>).

Una vez creado nuestro entorno virtual

ahora procedemos a entrar a nuestro entorno virtual



ahora activamos el entorno virtual con el comando source bin/activate y abrimos nuestro proyecto con el comando code . , esto nos abrirá el proyecto en visual studio code (si no lo tenemos lo instalamos <https://code.visualstudio.com/>)

esto nos abrira el proyecto y procedemos a crear nuestro archivo py, el cual se llamara scan.py.

Para crear el scanner ocupamos la librería de flask para renderizar nuestro template de html el cual nos mostrara las ip activas y las inactivas, y la librería de nmap la cual sera necesaria para que todo nuestro proyecto funcione.



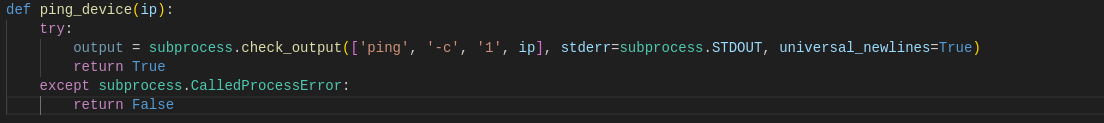
from flask import Flask, render\_template  
Esta línea importa la clase Flask y la función render\_template del módulo flask. Flask se usa para crear la aplicación web, y render\_template para renderizar archivos HTML.

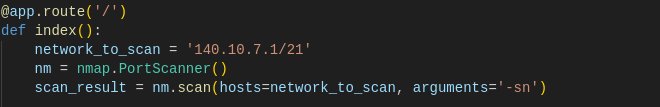
import nmap  
Importa la librería nmap, que permite controlar el programa nmap desde Python para escanear redes.

import subprocess  
Importa el módulo subprocess, que se usa para ejecutar comandos del sistema, en este caso, para realizar "ping" a las IPs.



crea la instancia de la aplicación Flask, que se usará para definir las rutas y ejecutar el servidor web.

Esta función ejecuta el comando ping -c 1 ip, donde -c 1 envía solo un paquete de prueba a la dirección IP proporcionada. Si el dispositivo responde (es decir, el comando ping se ejecuta sin error), retorna True. Si el dispositivo no responde, se produce una excepción (subprocess.CalledProcessError), y la función retorna False.

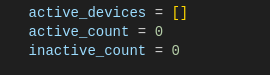


La ruta principal (/) es definida mediante @app.route('/'), vinculándola con la función index.

network\_to\_scan define el rango de red que se va a escanear (140.10.7.1/21).

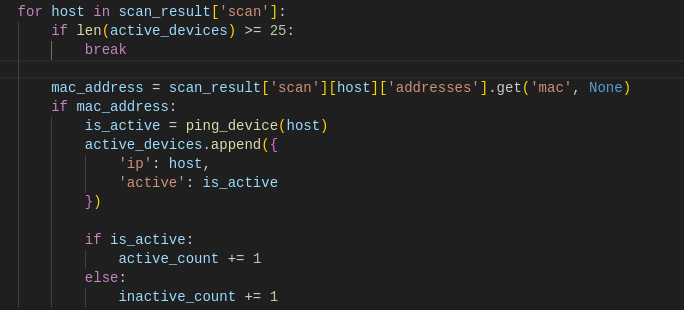
nm = nmap.PortScanner() crea un objeto PortScanner para ejecutar el escaneo de red con nmap.

nm.scan(hosts=network\_to\_scan, arguments='-sn') realiza un escaneo simple sin escanear puertos (-sn), y almacena los resultados en scan\_result.



active\_devices es una lista vacía para almacenar los dispositivos activos encontrados.

active\_count e inactive\_count son contadores que llevarán la cuenta de dispositivos activos e inactivos, respectivamente.

Figura 1: Recorrido del resultado del scaneo

for host in scan\_result['scan']: recorre cada dispositivo escaneado.

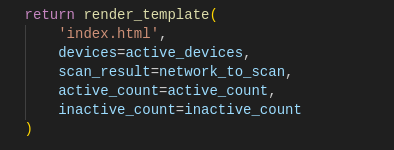
if len(active\_devices) >= 25: break limita la lista active\_devices a un máximo de 25 dispositivos.

mac\_address = scan\_result['scan'][host]['addresses'].get('mac', None) obtiene la dirección MAC del dispositivo (si está disponible).

is\_active = ping\_device(host) verifica si el dispositivo responde al ping.

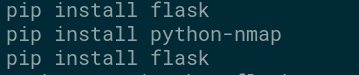
Los datos del dispositivo (ip y active) se agregan a active\_devices en un diccionario, donde 'active' será True o False según el resultado del ping.

active\_count e inactive\_count aumentan según el estado (True o False) de cada dispositivo.

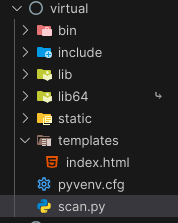


render\_template devuelve la plantilla index.html, pasando active\_devices, scan\_result, active\_count, e inactive\_count como contexto, para que puedan ser accedidos en la página HTML.

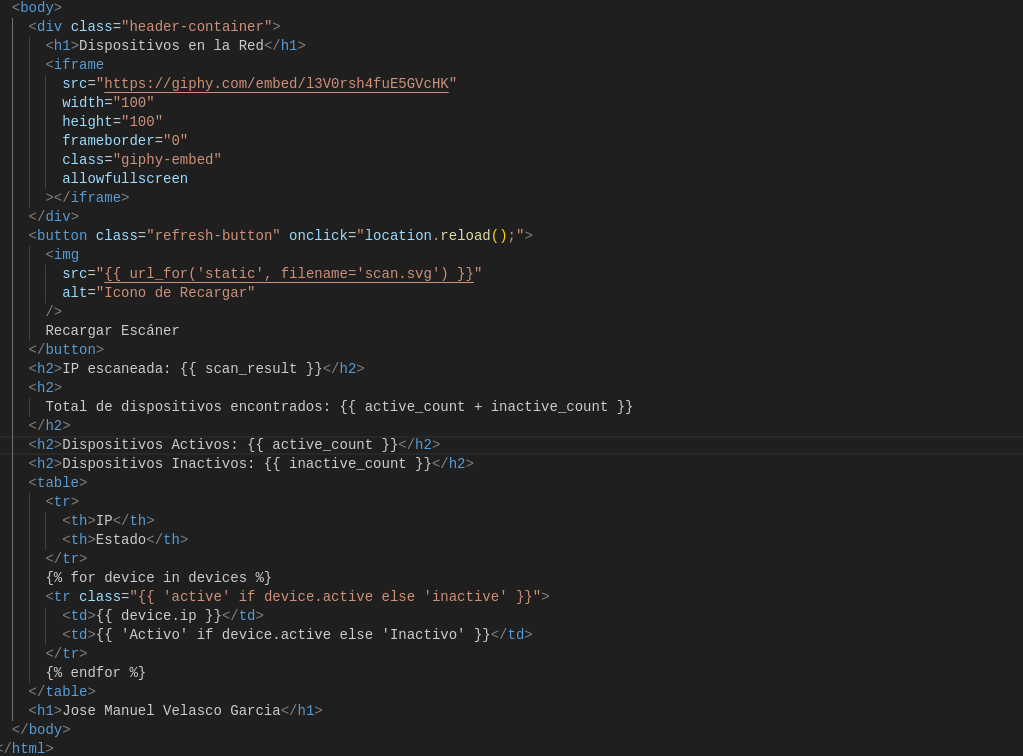
Una vez terminado el codigo procedemos a instalar las librerias que usaremos.



Con estas librerias el programa nos deberia de funcionar, pero antes de eso tenemos que renderizar nuestro template, paro esto creamos una carpeta llamada templates.



Dentro de la carpeta creamos nuestro index.html



ahora procedemos a obtener nuestra ip que escanearemos.

 Con este comando obtenemos nuestra ip, y la ponemos en nuestra variable network\_to\_scan

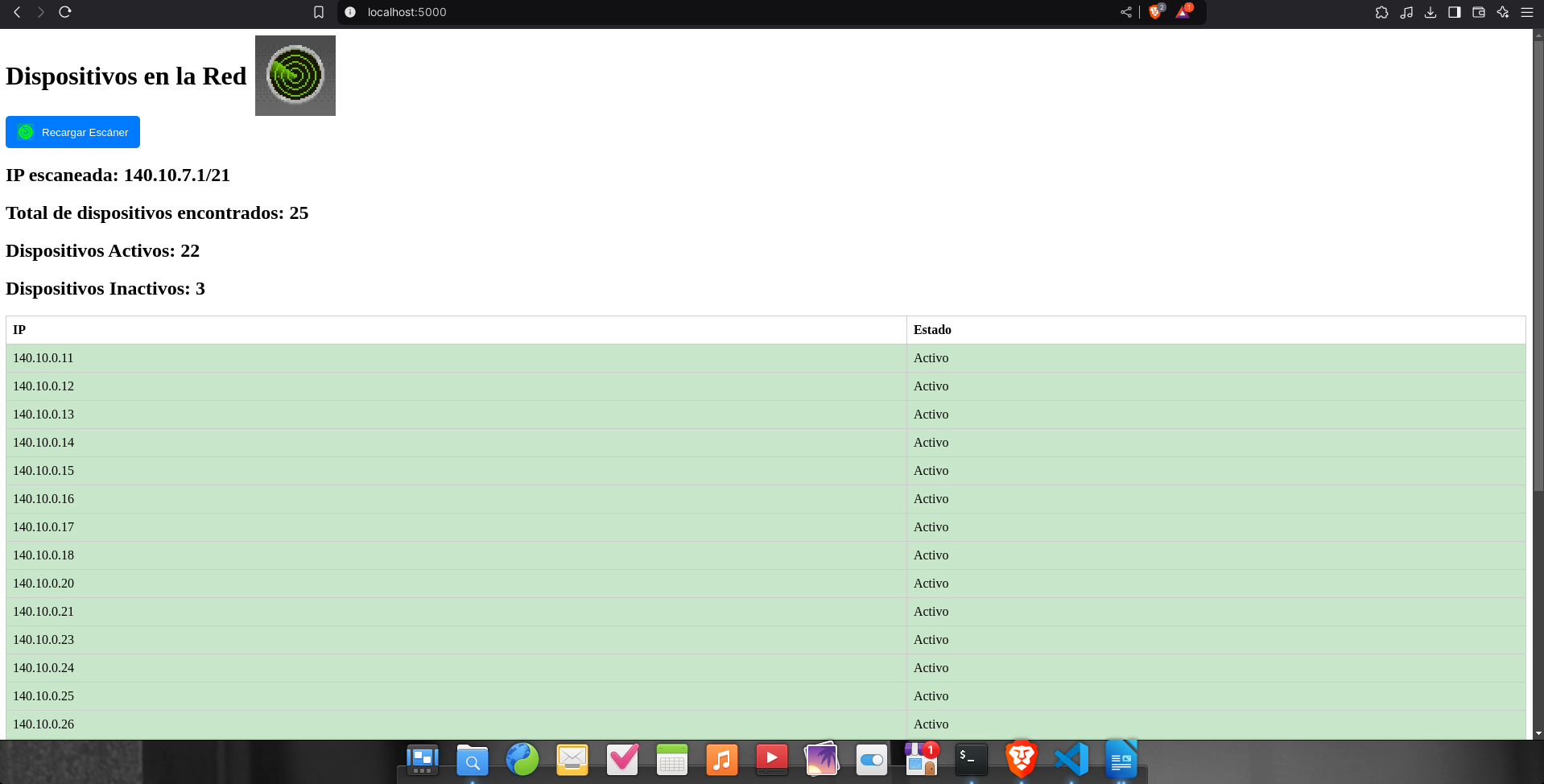
Figura 2: Procedemos a realizar un scan y en este caso el scaneo esta limitado a 26 ip, esto nos mostrara las ip activas y después que hagamos el siguiente scan nos mostrara las ip activas y las inactivas.

Figura 3: Misma red escaneada pero muestra las ip inactivas.

# **CONCLUSIONES**

Al realizar la practica se pensó que seria fácil ya que el primer scaneo se hizo en una ip que solo tenia 10 dispositivos conectados asi que el programa hizo el scan rapido, pero al hacerlo con una red con muchas ip el programa toma demasiado tiempo, la practica estuvo un poco complicada pero salio bien.

# **REFERENCIAS BIBLIOGRAFICAS**

https://code.visualstudio.com/

https://www.python.org/

<https://pypi.org/project/python-nmap/>

https://es.theastrologypage.com/network-scanning