

Consultar Fabricantes de tarjetas de red a través de una API

Brandon Villalobos Espinoza brandon.villalobos@alumnos.uv.cl

David Bombal Gonzalez david.bombal@alumnos.uv.cl

Benjamín Gonzalez Guerra benjamin.gonzalezg@alumnos.uv.cl

1. Introducción

El propósito de este trabajo es desarrollar un programa que permita consultar el fabricante de una tarjeta de red utilizando su dirección MAC. Esto a través de una API REST publica que nos entregara información sobre los fabricantes. Además, el programa tendrá otras funciones como por ejemplo “help”, “arp”, entre otras.

Objetivo del trabajo:

Realizar una solución en código Python donde podamos crear una herramienta en la línea de comandos que nos permita consultar el fabricante de una tarjeta de red dado su MAC, obtener los fabricantes de los dispositivos en la red local mediante la tabla ARP y podamos hacer uso de una API publica para acceder a la base de datos de fabricantes utilizando direcciones MAC.

Resultados de la solución:

El funcionamiento correcto de la solución en Python donde podamos ejecutar cada comando y que entregue de forma correcta los datos sobre consultas de la dirección MAC.

2. Descripción del problema y diseño de la solución

EL problema que trabajar es la creación de un programa donde podamos consultar el fabricante de una tarjeta de red utilizando su dirección MAC a través de un código Python y que pueda entregar la información que se solicitó.

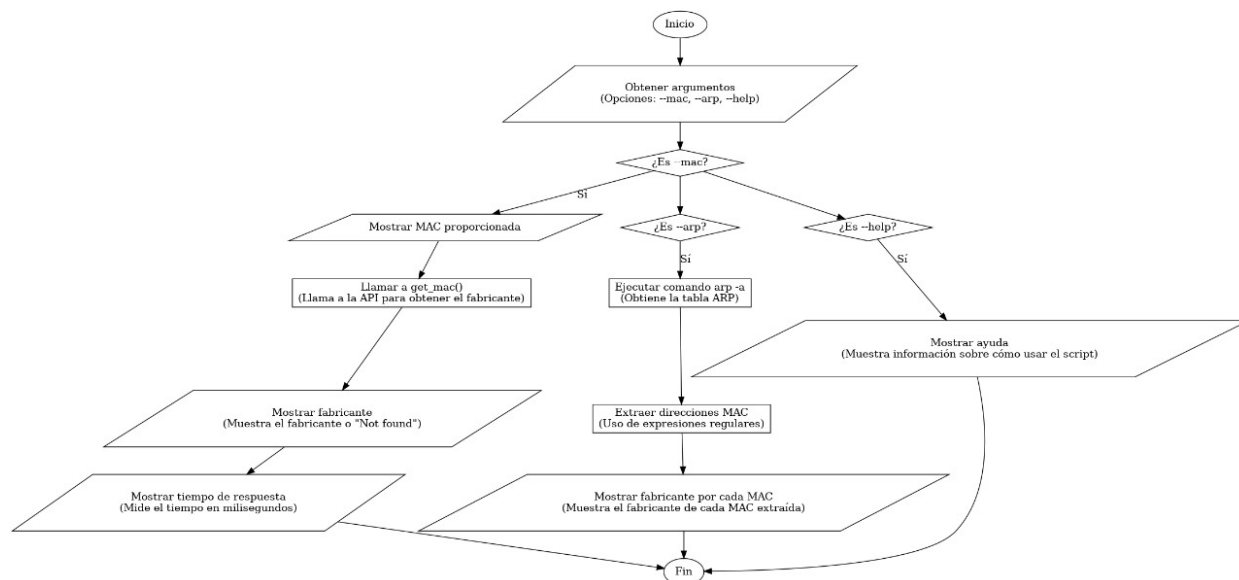


Figura 1: Diagrama de flujo del código de la solución.

3. Implementación

Uso del código

```
9     import sys
10    import requests
11    import getopt
12    import time
13    import re
14    import subprocess
```

Figura 2: Importación de bibliotecas.

- sys: Para interactuar con los parámetros y argumentos de línea de comandos.
- requests: Para realizar solicitudes HTTP a la API que proporciona información del fabricante de la MAC.
- getopt: Para analizar y manejar los argumentos de línea de comandos.
- time: Para medir el tiempo de ejecución de ciertas operaciones.
- re: Para utilizar expresiones regulares en la extracción de direcciones MAC.
- subprocess: Para ejecutar comandos del sistema, en este caso, arp -a.

```
# Obtener el fabricante de la API
def get_mac(address):

    # Extraer los caracteres de interes

    url = f"https://api.maclookup.app/v2/macs/{address}/company/name"

    response = requests.get(url)
    manufacturer = response.text.strip()

    if response.status_code == 200 and manufacturer != "*NO COMPANY*":

        return manufacturer

    else:
        manufacturer = "Not found"

        return manufacturer
```

Figura 3: Definición de función get_mac.

Esta función toma una dirección MAC como entrada y devuelve el nombre del fabricante asociado a esa dirección. Construye una URL para hacer una solicitud a la API de maclookup.app, pasando la dirección MAC como parámetro, luego realiza una solicitud GET a la API. Si la respuesta es exitosa (código 200) y el fabricante es válido (no es "NO COMPANY"), devuelve el nombre del fabricante, si no, devuelve "Not found".

Figura 4: Definición de función arp.

```
# Función de ayuda
def help():
    print("""Use: ./OUILookup --mac <mac> | --arp | [--help]
    --mac: MAC a consultar. P.e. aa:bb:cc:00:00:00.
    --arp: muestra los fabricantes de los hosts disponibles en la tabla arp.
    --help: muestra esta ayuda.
    """)
```

Figura 5: Definición de función help.

```
# Función principal donde se definen los parametros y se mide el tiempo de respuesta
def main():

    try:
        opts, args = getopt.getopt(sys.argv[1:], "", ["mac=", "arp", "help"])
    except getopt.GetoptError as err:
        print(err)
        sys.exit(2)

    for opt, arg in opts:
        if opt == "--mac":
            s_time = time.time()

            print("Dirección MAC: ", arg)
            print("Fabricante: ", get_mac(arg))
            e_time = time.time()
            r_time = (e_time - s_time) * 1000
            print(f'Tiempo de respuesta: {r_time:.2f}ms')

        elif opt == "--arp":
            arp()

        elif opt == "--help":
            help()

if __name__ == "__main__":
    main()
```

Figura 6: Definición de función main.

Maneja los argumentos de línea de comandos y ejecuta las funciones correspondientes según la opción proporcionada. Utiliza getopt para analizar los argumentos de línea de comandos, y luego recorre las opciones y ejecuta acciones basadas en ellas:

--mac: Llama a get_mac(arg) con la dirección MAC proporcionada y muestra el fabricante y el tiempo de respuesta.

--arp: Llama a la función arp() para mostrar los fabricantes de los dispositivos en la tabla ARP.

--help: Muestra la ayuda utilizando la función help().

También desde describir los principales desafíos que se encontraron en esta fase y cómo fueron resueltos.

4. Pruebas

Para probar el funcionamiento del código, fuimos ingresando uno a uno los comandos que se nos pidió para comprobar que los resultados fueran correctos.

En el comando "--help" nos entrega cuales son los comandos que podemos utilizar y la información que nos entregara cada uno.

```
PS C:\Users\david\OneDrive\Escritorio\Tarea2-Villalobos-Espinoza-Brandon-main\Tarea2-Villalobos-Espinoza-Brandon-main> python OUILookup.py --help
Use: ./OUILookup --mac <mac> | --arp | [--help]
      --mac: MAC a consultar. P.e. aa:bb:cc:00:00:00.
      --arp: muestra los fabricantes de los hosts disponibles en la tabla arp.
      --help: muestra esta ayuda.
```

Figura 7: Uso de "--help"

En el comando "--arp" nos muestra la tabla ARP local y muestra las direcciones MAC y sus fabricantes correspondientes.

```
PS C:\Users\david\OneDrive\Escritorio\Tarea2-Villalobos-Espinoza-Brandon-main\Tarea2-Villalobos-Espinoza-Brandon-main> python OUILookup.py --arp
MAC: 18-35-d1-1d-7d-30 || Fabricante: ARRIS Group, Inc.
MAC: 00-00-ca-01-02-03 || Fabricante: ARRIS Group, Inc.
MAC: ff-ff-ff-ff-ff-ff || Fabricante: Not found
MAC: 01-00-5e-00-00-16 || Fabricante: Not found
MAC: 01-00-5e-00-00-fb || Fabricante: Not found
MAC: 01-00-5e-00-00-fc || Fabricante: Not found
MAC: 01-00-5e-7f-ff-fa || Fabricante: Not found
MAC: ff-ff-ff-ff-ff-ff || Fabricante: Not found
```

Figura 8: Uso de "--arp"

En el comando "--mac" nos entrega la dirección MAC, muestra su respectivo fabricante y el tiempo de respuesta.

```
PS C:\Users\david\OneDrive\Escritorio\Tarea2-Villalobos-Espinoza-Brandon-main\Tarea2-Villalobos-Espinoza-Brandon-main> python3 OUILookup.py --mac 98:06:3c:92:ff:c5
Dirección MAC: 98:06:3c:92:ff:c5
Fabricante: Samsung Electronics Co.,Ltd
Tiempo de respuesta: 679.53ms
```

Figura 9: Uso de "--mac" con MAC: 98:06:3f:92:ff:c5

```
PS C:\Users\david\OneDrive\Escritorio\Tarea2-Villalobos-Espinoza-Brandon-main\Tarea2-Villalobos-Espinoza-Brandon-main> python3 OUILookup.py --mac 9c:a5:13
Dirección MAC: 9c:a5:13
Fabricante: Samsung Electronics Co.,Ltd
Tiempo de respuesta: 439.03ms
```

Figura 10: Uso de "--mac" con MAC:9c:a5:13

```
PS C:\Users\david\OneDrive\Escritorio\Tarea2-Villalobos-Espinoza-Brandon-main\Tarea2-Villalobos-Espinoza-Brandon-main> python3 OUILookup.py --mac 48-E7-DA
Dirección MAC: 48-E7-DA
Fabricante: AzureWave Technology Inc.
Tiempo de respuesta: 635.84ms
```

Figura 11: Uso de “--mac” con MAC: 48-E7-DA

5. Discusión y conclusiones

En conclusión, creamos una solución en código Python donde podemos obtener información sobre el fabricante asociado a una dirección MAC o para listar los fabricantes de los dispositivos presentes en la tabla ARP de la red local.

6. Referencias

requests. (2024, 29 mayo). PyPI. <https://pypi.org/project/requests/>

getopt — C-style parser for command line options. (s. f.). Python Documentation. <https://docs.python.org/es/3/library/getopt.html>

re — Regular expression operations. (s. f.). Python Documentation. <https://docs.python.org/es/3/library/re.html>