

Tarea 2: Consultar Fabricantes de tarjetas de red a través de una API

Reinaldo Berríos García, reinaldo.berrios@alumnos.uv.cl

Juan Durán Aracena, juan.durana@alumnos.uv.cl

José Hernández Olea, jose.hernandez@alumnos.uv.cl

1. Introducción

A día de hoy, es sumamente común la existencia y el uso de dispositivos inteligentes o aparatos con acceso a la red, por esto, es igualmente relevante su identificación en el medio, o en otras palabras, su control de acceso al medio, mayormente entendido como MAC, este se comprende como la dirección física de cada tarjeta de red. En relación a su utilidad, la existencia de dicha dirección es crucial para la comunicación de dispositivos dentro de una red local, puesto que, permite identificar a cada uno, con lo que se establece un punto de partida y destino a la hora de transmitir información, sin dicha implementación, un computador, por ejemplo, no podría comunicarse con otro al no saber con quién se comunica en cuestión. Además, dentro de cada MAC está, en lenguaje hexadecimal, una sección en la que se identifica al fabricante de dicha tarjeta de red, lo cual facilita el reconocimiento del dispositivo o tarjeta en cuestión.

En el presente trabajo, se busca poder consultar el fabricante de una tarjeta de red en base a su dirección MAC mediante el desarrollo de una herramienta codificada en Python utilizando una API REST, la cual se entiende como una interfaz de comunicación entre sistemas de información, dicha interfaz es a la cual se le realizarán las consultas en base a las direcciones MAC, con esto, se puede profundizar en la comprensión del funcionamiento de una red, puesto que lo trabajado representa una parte relevante de su total de características; la comunicación entre dispositivos.

2. Descripción del problema y diseño de la solución

Para el presente trabajo, se solicita el desarrollo de una herramienta a modo de ejecutable usando el lenguaje de programación Python, con esta, se busca obtener el nombre del fabricante asociado a una determinada dirección MAC que se le brinde a la herramienta, junto con el tiempo de espera asociado a la solicitud. Además, se solicita investigar sobre el funcionamiento de una dirección MAC aleatoria.

Para cumplir con lo solicitado, se plantea el desarrollo de un programa en Python el cual se ejecute por medio del CMD, en este, el usuario posee las opciones de pedir ayuda sobre el funcionamiento de la herramienta, obtener el nombre del fabricante en base a una determinada dirección MAC y generar una tabla arp de fabricantes conocidos. Para el desarrollo del código, se establece utilizar a getop para procesar las solicitudes por consola. Para la investigación asociada a las direcciones Mac aleatorias, se plantea utilizar

páginas de fabricantes conocidos para acceder a la información y generar un posterior resumen en la sección 5., dónde se plantea explicar su funcionamiento y descripción general.

3. Implementación

Para el desarrollo del programa, en primera instancia se importan las librerías(Requests, time, getopt, sys. las cuales tiene la función de hacer solicitudes HTTP a la API, medir el tiempo de ejecución, manejar los argumentos de la línea de comandos, interactuar con el sistema, respectivamente.).

La función principal del sistema es “get_fabricante”, que recibe como parámetro la dirección MAC proporcionada por el usuario. Esta función se encarga de buscar el fabricante correspondiente a dicha dirección MAC mediante una consulta a la API. Además, dentro de la función, se calcula el tiempo que tarda en realizar esta solicitud, lo que permite medir la eficiencia de la consulta y obtener información sobre la latencia de la respuesta. (Ver **Figura Funcion get_fabricante**).

```
def get_fabricante(mac_address):
    # URL de la API con la dirección MAC a consultar
    url = f"https://api.maclookup.app/v2/mac/{mac_address}"
    try:
        # registrar el tiempo antes de realizar la solicitud para medir el tiempo de respuesta
        tiempo = time.time()

        response = requests.get(url)

        # calcular el tiempo que se hizo la solicitud
        tiempo_ms = int((time.time() - tiempo) * 1000)

        if response.status_code == 200:
            data = response.json()
            # Obtener el campo 'company', que es el fabricante asociado a la MAC
            fabricante = data.get("company", "Desconocido")
            return fabricante, tiempo_ms
        else:
            # Si la respuesta no es exitosa, devolver "Not found" y el tiempo de respuesta
            return "Not found", tiempo_ms

    except Exception as e:
        # En caso de error, devolver el mensaje del error y 0 en el tiempo de respuesta
        return str(e), 0
```

Figura Funcion get_fabricante

Para iniciar el programa, el usuario debe ingresar comandos específicos, que son --help o --h, --arp y --mac [MAC a buscar], lo que genera tres posibles rutas de acción. La opción --help muestra los comandos disponibles para utilizar en el programa, mientras que el comando --arp presenta una lista de direcciones MAC que se pueden emplear. Por otro lado, el comando --mac permite al usuario realizar una búsqueda del fabricante asociado a una dirección MAC específica.(Ver **Figura Comandos**).

El código completo, ya iniciada su ejecución, cumple con retornar al usuario lo solicitado mediante CMD, lo que puede ser; pedir ayuda, recibir el nombre del fabricante o una tabla ARP, según lo que él ingrese al sistema (Ver **Diagrama.1_Programa**).

```
for o, a in opts:
    # si el usuario pidió la ayuda (-h o --help), mostrar el mensaje de ayuda y salir
    if o in ("-h", "--help"):
        print("Uso: OUILookup.py --mac <mac> | --arp | --help")
        print("--mac: MAC a consultar. P.e. aa:bb:cc:00:00:00.")
        print("--arp: muestra los fabricantes de los hosts disponibles en la tabla ARP.")
        print("--help: muestra este mensaje y termina.")
        sys.exit() # Salir del programa
    #proporcionar una dirección MAC con la opción -m o --mac, almacenarla
    elif o in ("-m", "--mac"):
        mac_address = a

    elif o in ("--arp"):
        show_arp = True

#consultar la API y mostrar el fabricante
if mac_address:
    fabricante, response_time = get_fabricante(mac_address) # Obtener fabricante y tiempo
    print(f"MAC address: {mac_address}")
    print(f"Fabricante: {fabricante}")
    print(f"Tiempo de respuesta: {response_time} ms")

# si el usuario utiliza --arp, mostrar una lista fija de direcciones MAC y fabricantes
elif show_arp:
    print("IP/MAC/Vendor:")
    print("00:01:97:BB:BB:BB / cisco")
    print("B4:B5:FE:92:FF:C5 / Hewlett Packard")
    print("00:E0:64:AA:AA:AA / Samsung")
    print("AC:F7:F3:AA:AA:AA / Xiaomi")
```

Figura Comandos

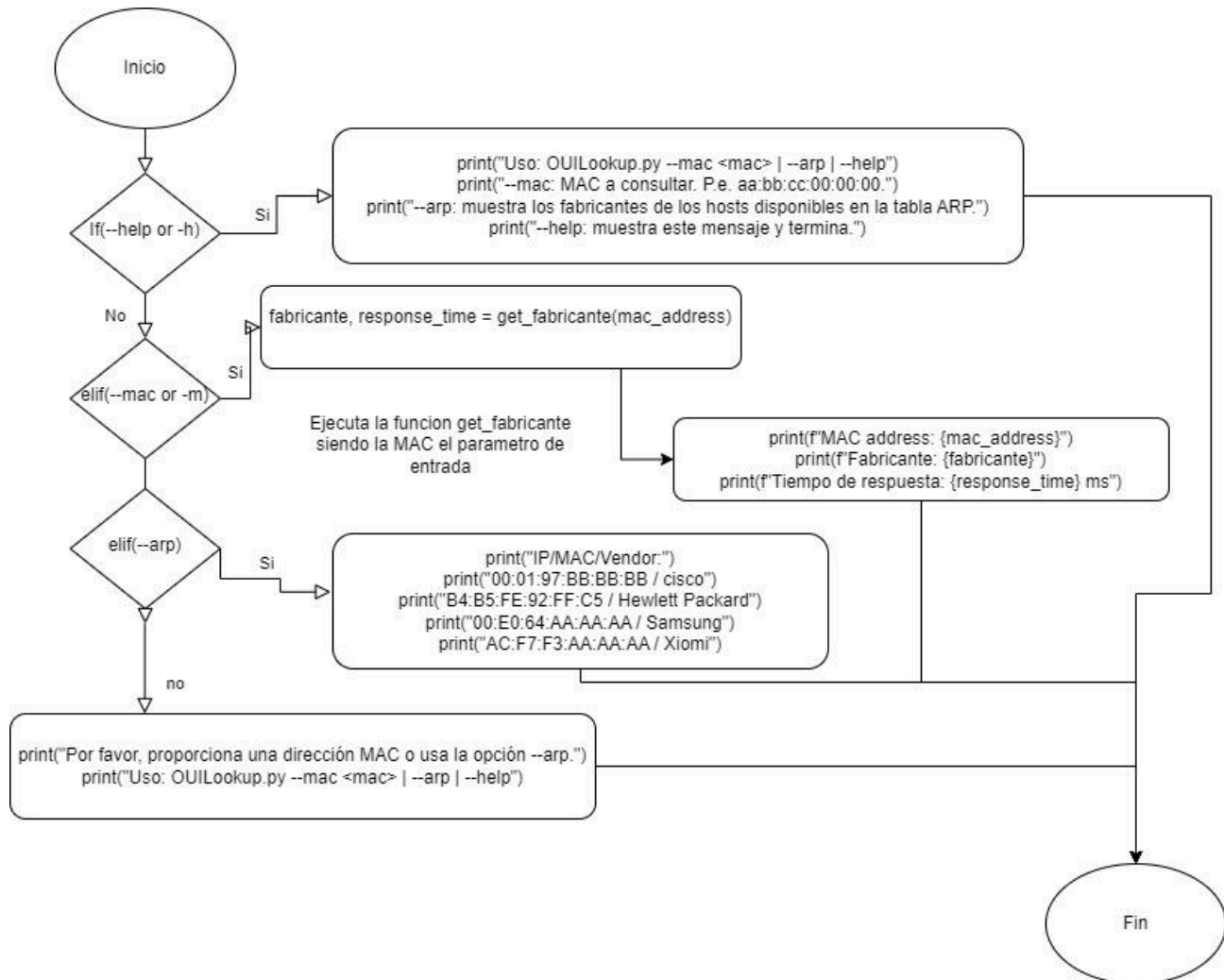


Diagrama.1_Programa

```
def get_fabricante(mac_address):  
    tiempo = time.time()response = requests.get(url)  
    tiempo_ms = int((time.time() - tiempo) * 1000)  
    if response.status_code == 200:  
        data = response.json()  
        fabricante = data.get("company", "Desconocido")  
        return fabricante, tiempo_mselse:  
    return "Not found", tiempo_msexcept Exception as  
        e:  
        return str(e), 0
```

Función.1

4. Pruebas

Para realizar pruebas en el código de Python, primero debemos abrir la consola de comandos (cmd) en nuestro sistema operativo. Una vez abierta la consola, es necesario navegar hasta la carpeta donde se encuentra el archivo OUILookup.py.

El comando `python OUILookup.py` iniciará el programa, pero su comportamiento dependerá de los parámetros que proporcionamos. El programa acepta diferentes opciones de línea de comandos que permiten realizar distintas acciones.

1. Para obtener ayuda: si se desea ver una lista con todas las funciones disponibles y la forma correcta de usarlas, podemos ejecutar el comando con la opción `--help`

```
C:\Users\lSkipl\Downloads>python OUILookup.py --help  
Uso: OUILookup.py --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 este mensaje y termina.
```

Imagen 1 Prueba (--help)

Esto mostrará una descripción de las funciones y parámetros que podemos utilizar con el programa. (ver **Imagen 1 prueba (--help)**).

2. Consulta ARP: Si deseamos realizar una búsqueda relacionada con ARP, podemos ejecutar el comando con la opción --arp

```
C:\Users\lSkipl\Downloads>python OUILookup.py --arp
IP/MAC/Vendor:
00:01:97:BB:BB:BB / cisco
B4:B5:FE:92:FF:C5 / Hewlett Packard
00:E0:64:AA:AA:AA / Samsung
AC:F7:F3:AA:AA:AA / Xiaomi
```

Imagen 2 Prueba (--arp)

Este comando mostrará la información de los fabricantes asociados a los hosts disponibles en la tabla ARP.(ver **Imagen 2 prueba (--arp)**)

3. Consulta de dirección MAC: Para realizar una consulta específica sobre una dirección MAC, podemos ejecutar el comando con la opción --mac [Dirección Mac]

```
C:\Users\lSkipl\Downloads>python OUILookup.py --mac 98:06:3c:92:ff:c5
MAC address: 98:06:3c:92:ff:c5
Fabricante: Samsung Electronics Co.,Ltd
Tiempo de respuesta: 574 ms

C:\Users\lSkipl\Downloads>python OUILookup.py --mac 9c:a5:13
MAC address: 9c:a5:13
Fabricante: Samsung Electronics Co.,Ltd
Tiempo de respuesta: 575 ms

C:\Users\lSkipl\Downloads>python OUILookup.py --mac 48-E7-DA
MAC address: 48-E7-DA
Fabricante: AzureWave Technology Inc.
Tiempo de respuesta: 347 ms
```

Imagen 3 Prueba (--mac [Dirección Mac])

Mostrando en pantalla la dirección Mac, el fabricante y el tiempo de respuesta de cada consulta (ver **Imágen 3 prueba (--mac [Dirección Mac])**).

5. Mac aleatorias

Una dirección MAC (Media Access Control) aleatoria es un identificador único que se asigna a la interfaz de red de un dispositivo para facilitar la comunicación en redes locales. Estas direcciones pueden generarse aleatoriamente por diversas razones, siendo las más destacadas la privacidad y la seguridad. En términos de privacidad, el uso de direcciones MAC aleatorias en entornos donde los dispositivos se conectan a múltiples redes, como las redes Wi-Fi públicas, ayuda a proteger la identidad del dispositivo y reduce el riesgo de ser rastreado a lo largo del tiempo. En cuanto a la seguridad, al cambiar frecuentemente la dirección MAC, se dificulta la identificación de un dispositivo específico por parte de terceros, lo que a su vez protege tanto la información del usuario como la integridad de su conexión. Se utilizan fuentes de aleatoriedad para asegurar que las direcciones generadas sean únicas y no predecibles, lo que ayuda a prevenir el rastreo de dispositivos en redes.

6. Discusión y conclusiones

Una vez finalizado el trabajo, se puede concluir que la existencia de una dirección MAC es vital para la comunicación entre dispositivos, permite su distinción y reconocimiento entre ellos dentro de una misma red local, lo que a su vez que permiten entregar y recibir datos desde o hacia una red global, distinguen en todo momento el origen y destino de la información. Se entiende que cada MAC es único y permite el correcto tráfico a través de la red. Además, se destacan aspectos de seguridad a la hora de indagar sobre las Mac aleatorias, puesto que permiten un tráfico más seguro a través de la red, dificultando el rastreo de terceros, dando así una defensa a la hora de privacidad y ciberseguridad.

Junto con la conclusión, aspectos relevantes a destacar son la dificultad del trabajo realizado, la mayor de estas fue el propio desarrollo de la herramienta, esto debido a requerir indagar sobre conceptos y funciones para la correcta aplicación de esta, destacando el principalmente acceder a la API REST para buscar al fabricante asociado a su dirección MAC. Finalmente, se establece como posible mejora adaptar la interfaz para hacerla más accesible al usuario, facilitar su uso, en otras palabras, que desde que se ejecuta el programa el usuario pueda observar sus opciones disponibles, además, aumentar el rango de búsqueda en otras API REST públicas, de tal manera que se pueda evitar retornar el mensaje de “not found” que aparece cada vez que se busca una mac que no esté dentro de la búsqueda.

7. Referencias

- [1]<https://support.microsoft.com/es-es/windows/-por-qué-usar-direcciones-de-hardware-aleatorias-060ad2e9-526e-4f1c-9f3d-fe6a842640ed>
- [2]<https://support.microsoft.com/es-es/windows/cómo-usar-direcciones-de-hardware-aleatorias-en-windows-ac58de34-35fc-31ff-c650-823fc48eb1bc>
- [3]<https://support.apple.com/es-lamr/guide/security/secb9cb3140c/web#:~:text=Aleatorización%20de%20la%20dirección%20MA>
- [4]<https://support.apple.com/es-lamr/guide/security/secb9cb3140c/web#:~:text=Aleatorización%20de%20la%20dirección%20MAC,con%20una%20red%20Wi-Fi>.