

Consultar Fabricantes de tarjetas de red a través de una API

Gabriela Herrera Osorio, gabriela.herrerao@alumnos.uv.cl

1. Introducción

Desde la llegada de Internet, identificar los dispositivos de red se ha vuelto fundamental para la gestión y seguridad de las redes. Este trabajo presenta OUILookup, una herramienta que permite consultar fabricantes de tarjetas de red a través de direcciones MAC utilizando una API REST pública [1]. Su objetivo es facilitar la identificación de dispositivos en una red, lo que es esencial para garantizar la seguridad y el correcto funcionamiento.

2. Descripción del problema y diseño de la solución

Las direcciones MAC, las cuales son únicas para cada dispositivo en la red, permiten determinar el fabricante de la tarjeta de red. Sin embargo, buscar manualmente información sobre el fabricante puede ser poco eficiente, especialmente en redes grandes donde se conectan una gran cantidad de dispositivos. Es por esto por lo que se ha creado OUILookup, la cual es una herramienta creada en Python que permite a través de línea de comandos consultar el fabricante de una tarjeta de red dada su dirección MAC. Para mostrar su funcionalidad serán mostrados casos de prueba.

Funcionalidad:

- Contiene un menú de ayuda que muestra cómo realizar la consulta.
- Acepta una dirección MAC como entrada y devuelve el nombre del fabricante correspondiente y el tiempo de respuesta.
- Muestra una lista de las direcciones MAC y los fabricantes de los dispositivos presentes en la tabla ARP de la red local, incluyendo el tiempo de respuesta.
- En el caso de no encontrar el nombre del fabricante muestra "Not Found".

Uso de la API:

- Se utilizará una API REST pública que permite obtener información sobre los fabricantes a partir de direcciones MAC.

Descripción del problema y diseño de la solución

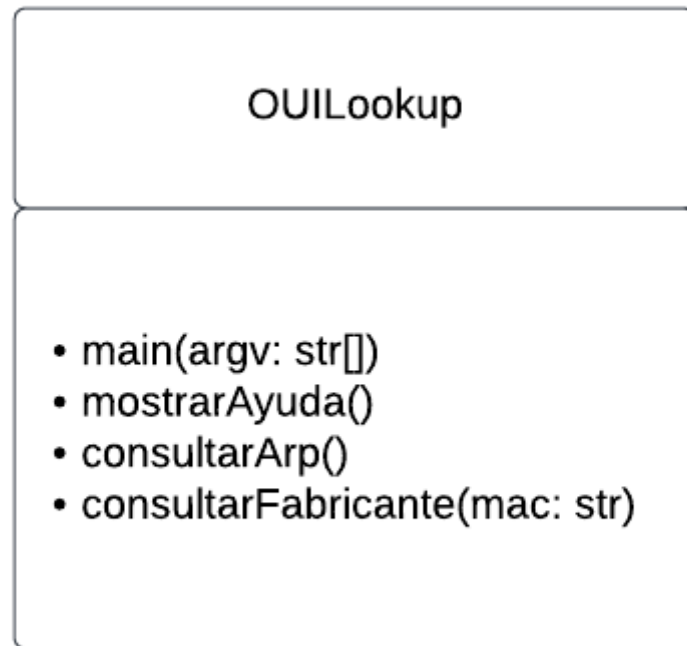


Figura 1: Diagrama del código

3. Implementación

El código contiene tres funciones: `mostrarAyuda`, `consultarFabricante` y `consultarArp`. La primera muestra un menú de ayuda para poder hacer las consultas correctamente. La segunda consulta una API para obtener el fabricante de una tarjeta de red a partir de una dirección MAC entregada. La tercera ejecuta el comando `arp -a`, extrae las direcciones MAC válidas y utiliza la función `consultarFabricante` para cada una. Hubo muchos desafíos a lo largo del proyecto, los principales fueron: manejar las direcciones MAC no válidas, la respuesta de la API en caso de errores de red y como hacer la tabla ARP. Esto se resolvió buscando información en Internet que fue útil para poder resolver las dificultades. El primero se resolvió mediante un formato estándar que permite identificar la MAC [2], el segundo mediante una excepción de la librería `request` y el tercero buscando información en diversas fuentes.

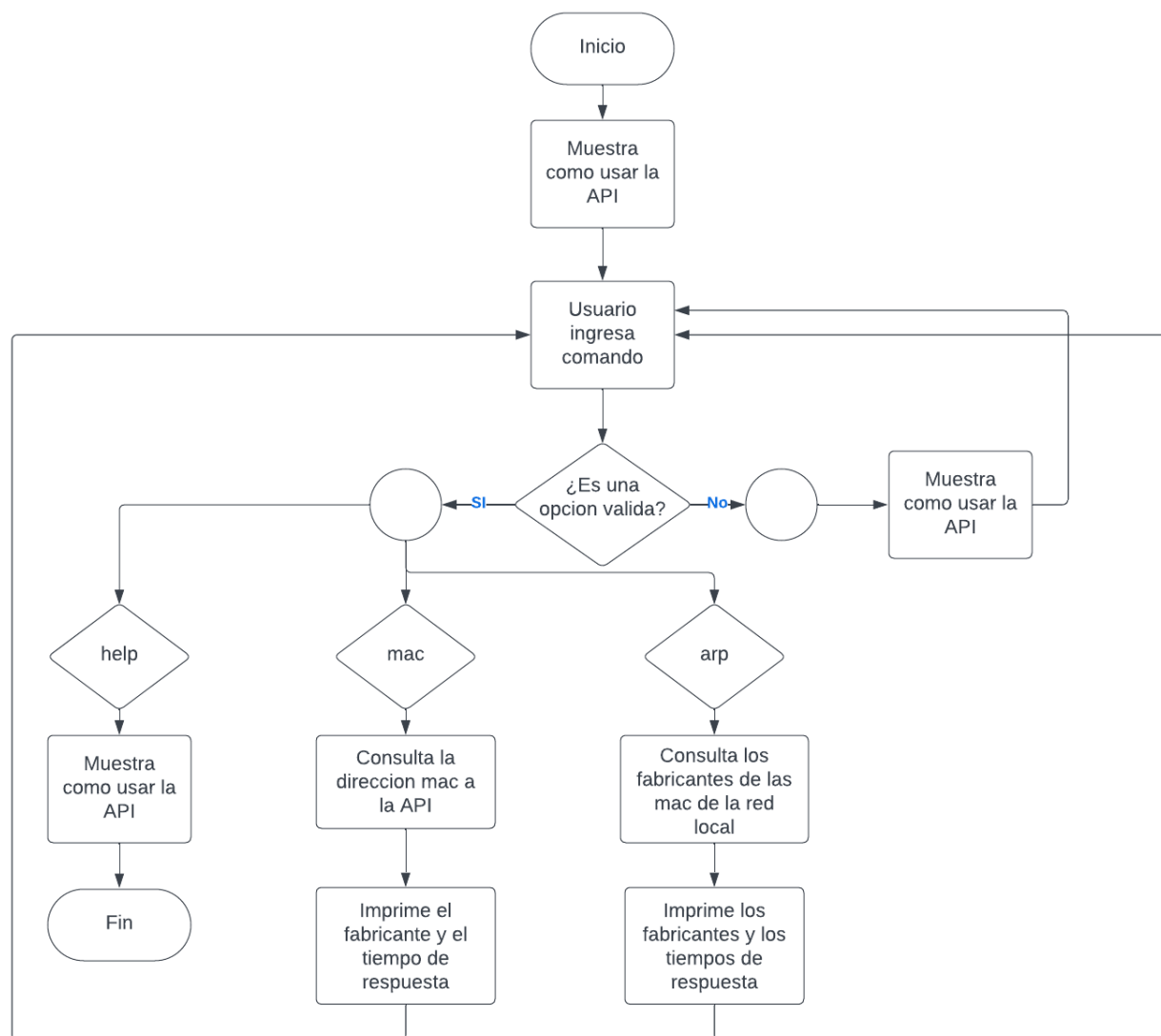


Figura 2: Diagrama de flujo del código

4. Pruebas

Las pruebas del código fueron realizadas a través de línea de comandos en un dispositivo Windows, fueron usados casos de pruebas que verificaron que las repuestas de la API fueran correctas.

```
C:\Users\gherr>python OUILookup.py --help
Uso: python OUILookup.py --mac <mac> | --arp | [--help]
--mac: MAC a consultar. Ej: aa:bb:cc:00:00:00.
--arp: Muestra los fabricantes de los hosts disponibles en la tabla ARP.
--help: Muestra este mensaje y termina.
```

Figura 3: Ejemplo de uso sin parámetros

```
C:\Users\gherr> python3 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: 786 ms
```

Figura 4: Caso MAC que esté en la base de datos uno

```
C:\Users\gherr>python3 OUILookup.py --mac 9c:a5:13
MAC address: 9c:a5:13
Fabricante: Samsung Electronics Co.,Ltd
Tiempo de respuesta: 718 ms
```

Figura 5: Caso MAC que esté en la base de datos dos

```
C:\Users\gherr>python3 OUILookup.py --mac 48-E7-DA
MAC address: 48-E7-DA
Fabricante: AzureWave Technology Inc.
Tiempo de respuesta: 740 ms
```

Figura 6: Caso MAC que esté en la base de datos tres

```
C:\Users\gherr> python3 OUILookup.py --mac 98:06:3f:92:ff:c5
MAC address: 98:06:3f:92:ff:c5
Fabricante: Not Found
Tiempo de respuesta: 842 ms
```

Figura 7: Caso MAC que no esté en la base de datos

```
C:\Users\gherr> python3 OUILookup.py --arp
MAC address: e4-57-40-ad-25-98
Fabricante: ARRIS Group, Inc.
Tiempo de respuesta: 732 ms
MAC address: 00-00-ca-01-02-03
Fabricante: ARRIS Group, Inc.
Tiempo de respuesta: 712 ms
MAC address: ff-ff-ff-ff-ff-ff
Fabricante: Not Found
Tiempo de respuesta: 510 ms
MAC address: 01-00-5e-00-00-16
Fabricante: Not Found
Tiempo de respuesta: 509 ms
MAC address: 01-00-5e-00-00-fb
Fabricante: Not Found
Tiempo de respuesta: 612 ms
MAC address: 01-00-5e-00-00-fc
Fabricante: Not Found
Tiempo de respuesta: 646 ms
MAC address: ff-ff-ff-ff-ff-ff
Fabricante: Not Found
Tiempo de respuesta: 410 ms
```

Figura 8: Caso fabricantes de las MAC disponibles en la tabla arp

4.1. Mac aleatorias

La dirección MAC (Control de Acceso a Medios) es un identificador único de 12 dígitos hexadecimales que permite identificar dispositivos en una red. Cuando un dispositivo se conecta a una red Wi-Fi, el router utiliza esta dirección para identificarlo. Sin embargo, en redes públicas, exponer la dirección MAC original puede comprometer la privacidad del usuario, revelando su identidad y ubicación. Para reducir este riesgo, algunos sistemas operativos generan MAC aleatorias, creando direcciones MAC temporales al conectarse a redes inalámbricas. Esto ayuda a proteger la privacidad del usuario, ya que oculta la dirección MAC real y reduce la posibilidad de seguimiento malicioso [3,4].

5. Discusión y conclusiones

En este proyecto, se creó la herramienta OUILookup para consultar fabricantes de dispositivos a partir de direcciones MAC. A través de distintas pruebas se demostró que

su funcionamiento es correcto, tanto en consultas específicas como en la tabla ARP. En la realización de este proyecto se adquirieron nuevos conocimientos sobre librerías en Python [5-9] y como usarlas, que son las MAC aleatorias, entre otras cosas. Además, se fortalecieron habilidades de programación aplicando nuevos conceptos. Para futuras mejoras se podría realizar una interfaz grafica para facilitar el uso y así presentar la información más ordenada, agregando también una lista donde se puedan ver consultas recientes.

6. Referencias

- [1] MACLookup. Disponible en línea: <https://maclookup.app> (acceso 6 octubre 2024).
- [2] Stack Overflow. What is a regular expression for a MAC address? Disponible en línea: <https://stackoverflow.com/questions/4260467/what-is-a-regular-expression-for-a-mac-address> (acceso 7 octubre 2024).
- [3] Moyens. ¿Qué es la aleatorización de MAC y cómo usarla en sus dispositivos? Disponible en línea: <https://es.moyens.net/como/que-es-la-aleatorizacion-de-mac-y-como-usarla-en-sus-dispositivos/#> (acceso 7 octubre 2024).
- [4] Trend Micro Help Center. What is MAC address randomization? Disponible en línea: <https://helpcenter.trendmicro.com/en-us/article/TMKA-10333> (acceso 8 octubre 2024).
- [5] Stack Overflow. Correct way to try/except using Python requests module. Disponible en línea: <https://stackoverflow.com/questions/16511337/correct-way-to-try-except-using-python-requests-module> acceso 8 octubre 2024)
- [6] Python Software Foundation. sys — System-specific parameters and functions. Disponible en línea: <https://docs.python.org/3/library/sys.html> (acceso 8 octubre 2024).
- [7] Python Software Foundation. getopt — Style parser for command line options. Disponible en línea: <https://docs.python.org/3/library/getopt.html> (acceso 8 octubre 2024).
- [8] Python Software Foundation. re — Regular expression operations. Disponible en línea: <https://docs.python.org/3/library/re.html> (acceso 8 octubre 2024).
- [9] Python Software Foundation. subprocess — Subprocess management. Disponible en línea: <https://docs.python.org/es/3/library/subprocess.html> (acceso 8 octubre 2024).