

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

Amalia Toledo Astudillo, amalia.toledo@alumnos.uv.cl

Cristian Badilla Fuentes, cristian.badilla@alumnos.uv.cl

Benjamín Maldonado Núñez, benjamin.maldonadon@alumnos.uv.cl

1. Introducción

En el contexto de las redes de computadores, cada dispositivo de red está identificado por una dirección MAC (Media Access Control), la cual es única para cada dispositivo. Esta dirección es asignada por los fabricantes de hardware y sigue un formato estandarizado. Conocer el fabricante de un dispositivo de red puede ser útil para diversas aplicaciones, como la gestión de redes, el diagnóstico de problemas de conectividad y la seguridad informática.

El objetivo de esta tarea es desarrollar una herramienta llamada OUILookup que permita consultar el fabricante de un dispositivo de red a partir de su dirección MAC utilizando una API REST pública. La herramienta, implementada en Python, permite interactuar con la base de datos de direcciones MAC a través de una interfaz de línea de comandos. Se utilizará la API pública disponible en <https://maclookup.app>, la cual proporciona información sobre las direcciones MAC y sus fabricantes asociados.

En este informe, se describe el proceso de desarrollo de la herramienta, así como los distintos casos de uso implementados. Además, se detalla la estructura del código, las pruebas realizadas y las conclusiones obtenidas.

2. Descripción del problema y diseño de la solución

2.1 Descripción del problema:

Cada dispositivo en una red se identifica mediante una dirección MAC (Media Access Control), la cual está asociada a su fabricante. Sin embargo, identificar el fabricante a partir de una dirección MAC puede ser complicado sin acceso a una base de datos adecuada. Por ello, se requiere una herramienta que permita a los usuarios consultar rápidamente el fabricante de un dispositivo de red utilizando su dirección MAC.

2.2 Diseño de la solución:

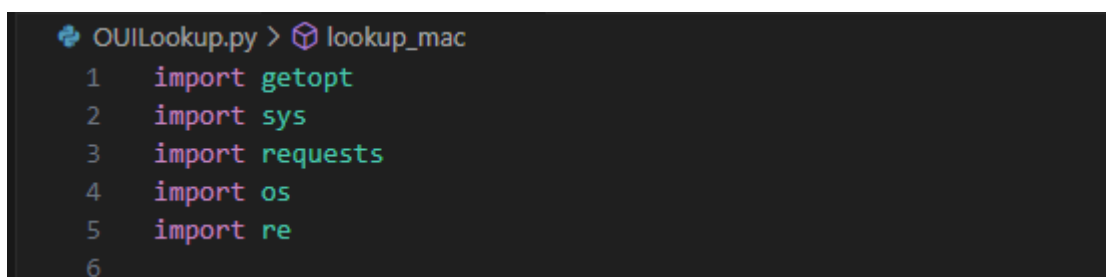
La solución implementada es una herramienta de línea de comandos en Python que permite consultar el fabricante de una dirección MAC o las MACs presentes en la tabla ARP de la red local. Utiliza la API de MacLookup para obtener la información de los fabricantes.

El programa está dividido en tres funciones principales: la primera consulta el fabricante de una MAC específica proporcionada por el usuario, la segunda extrae las direcciones MAC de la tabla ARP y consulta sus fabricantes, y la tercera gestiona los argumentos introducidos en la línea de comandos, como --mac, --arp o --help.

El usuario puede ejecutar el programa con las siguientes opciones: consultar una MAC individual, obtener y consultar todas las MACs de la tabla ARP, o ver un mensaje de ayuda.

3. Explicación del código

a. Imports

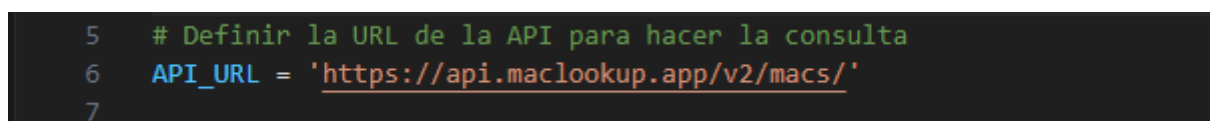


```
OUILookup.py > lookup_mac
1  import getopt
2  import sys
3  import requests
4  import os
5  import re
6
```

Ilustración 1. Se ve parte del Código.

- **getopt:** Se utiliza para manejar argumentos de línea de comandos. Permite definir opciones y procesar los argumentos ingresados por el usuario.
- **sys:** Proporciona acceso a algunas variables y funciones que interactúan con el intérprete de Python, como la gestión de argumentos.
- **requests:** Esta biblioteca permite realizar solicitudes HTTP, lo cual es esencial para consultar la API RES
- **os:** Se usa para interactuar con el sistema operativo, como ejecutar comandos del sistema (e.g., arp -a).
- **re:** Se usa para trabajar con expresiones regulares, permitiendo buscar y extraer patrones de texto (en este caso, direcciones MAC).

b. Definición de la URL de la API



```
5  # Definir la URL de la API para hacer la consulta
6  API_URL = 'https://api.maclookup.app/v2/macs/'
7
```

Ilustración 2. Se ve parte del Código.

- **API_URL:** Define la URL base de la API que se utilizará para consultar el fabricante a partir de una dirección MAC.

c. Función consultar_mac(direccion_mac)

```
10 # Función que realiza la consulta a la API
11 def consultar_mac(direccion_mac):
12     try:
13         # Hacer la solicitud a la API con la dirección MAC
14         respuesta = requests.get(URL_API + direccion_mac)
15
16         # Verificar si la solicitud fue exitosa
17         if respuesta.status_code == 200:
18             datos = respuesta.json()
19
20             # Verificar si la respuesta contiene información del fabricante
21             if datos and 'company' in datos and datos['company']:
22                 print(f"Dirección MAC: {direccion_mac}")
23                 print(f"Fabricante : {datos['company']}")
24             else:
25                 # Si no se encuentra fabricante
26                 print(f"Dirección MAC: {direccion_mac}")
27                 print("Fabricante : No encontrado")
28
29             # Imprimir el tiempo de respuesta
30             print(f"Tiempo de respuesta: {respuesta.elapsed.total_seconds() * 1000:.2f}ms")
31
32         else:
33             # Imprimir el código de estado y el texto de respuesta para diagnósticos
34             print(f"Error: La API devolvió un estado {respuesta.status_code}. Respuesta: {respuesta.text}")
35
36     except Exception as e:
37         print(f"Error al consultar la API: {e}")
```

Ilustración 3. Se ve parte del Código.

Esta función se encarga de realizar la consulta a la API y obtener el fabricante de una dirección MAC específica.

- **Función principal:** Realiza la consulta a la API utilizando la dirección MAC.
- **Manejo de errores:** Utiliza un bloque try para capturar cualquier excepción que pueda ocurrir durante la solicitud.
- **Verificación de respuesta:** Comprueba si la respuesta de la API es exitosa. Luego verifica si la información del fabricante está presente en la respuesta JSON.
- **Impresión de resultados:** Muestra la dirección MAC, el fabricante encontrado, y el tiempo que tardó la solicitud en completarse.

d. Función obtener_tabla_arp()

```
39 # Función para obtener las direcciones MAC de la tabla ARP
40 def obtener_tabla_arp():
41     try:
42         # Ejecutar el comando "arp -a" y capturar la salida
43         salida = os.popen('arp -a').read()
44
45         # Expresión regular para extraer las direcciones MAC en el formato correcto
46         direcciones_mac = re.findall(r'([0-9a-fA-F]{2}[:-]){5}([0-9a-fA-F]{2})', salida)
47
48         # Unir los fragmentos de la dirección MAC y asegurarse de que el formato sea correcto
49         return ['.'.join(mac) for mac in direcciones_mac]
50
51     except Exception as e:
52         print(f"Error al obtener la tabla ARP: {e}")
53     return []
```

Ilustración 4. Se ve parte del Código.

Esta función obtiene las direcciones MAC de la tabla ARP del sistema.

- Ejecuta el comando arp -a mediante os.popen() para obtener la tabla ARP, que contiene información de los dispositivos conectados a la red local.
- Forma correctamente las MACs al unir los fragmentos de las direcciones MAC encontradas con dos puntos :.
- Devuelve la lista de direcciones MAC.

e. Función main()

```
55 # Función principal para manejar los argumentos y ejecutar la lógica
56 def main():
57     try:
58         opciones, argumentos = getopt.getopt(sys.argv[1:], "hm:a", ["help", "mac=", "arp"])
59     except getopt.GetoptError as error:
60         print(error)
61         uso()
62         sys.exit(2)
63
64     direccion_mac = None
65     arp = False
66
67     for opcion, argumento in opciones:
68         if opcion in ("-h", "--help"):
69             uso()
70             sys.exit()
71         elif opcion in ("-m", "--mac"):
72             direccion_mac = argumento
73         elif opcion in ("-a", "--arp"):
74             arp = True
75         else:
76             assert False, "Opción no válida"
77
78     if direccion_mac:
79         direccion_mac = direccion_mac.replace('-', ':').lower() # Asegurarse del formato correcto
80         print(f"Consultando fabricante para la MAC: {direccion_mac}")
81         # Llamar a la función consultar_mac para consultar el fabricante
82         consultar_mac(direccion_mac)
83     elif arp:
84         print("Obteniendo tabla ARP...")
85         macs = obtener_tabla_arp()
86         if macs:
87             for mac in macs:
88                 mac_str = mac.lower() # Convertir todas las direcciones a minúsculas
89                 print(f"Consultando fabricante para la MAC: {mac_str}")
90                 consultar_mac(mac_str)
91         else:
92             print("No se encontraron entradas en la tabla ARP.")
93     else:
94         uso()
```

Ilustración 5. Se ve parte del Código.

Esta función gestiona la lógica principal del programa, incluyendo el manejo de argumentos de línea de comandos y la llamada a las funciones necesarias.

getopt.getopt() se utiliza para procesar los argumentos de línea de comandos. Aquí los argumentos pueden ser:

- --mac: Para consultar un fabricante a partir de una dirección MAC específica.
- --arp: Para consultar los fabricantes de todas las direcciones MAC presentes en la tabla ARP.
- --help: Para mostrar el mensaje de ayuda.

Dependiendo de la opción seleccionada, el programa consulta un fabricante específico o bien la tabla ARP.

f. Función uso ()

```
96 def uso():
97     print("Uso: OUILookup.py --mac <mac> | --arp | [--help]")
98     print("--mac: MAC a consultar. Ej: aa:bb:cc:00:00:00.")
99     print("--arp: Muestra los fabricantes de los hosts disponibles en la tabla ARP.")
100    print("--help: Muestra este mensaje y termina.")
101
```

Ilustración 6. Se ve parte del Código.

- **Instrucciones:** Proporciona una guía sobre cómo usar el programa, incluyendo los parámetros disponibles y ejemplos.

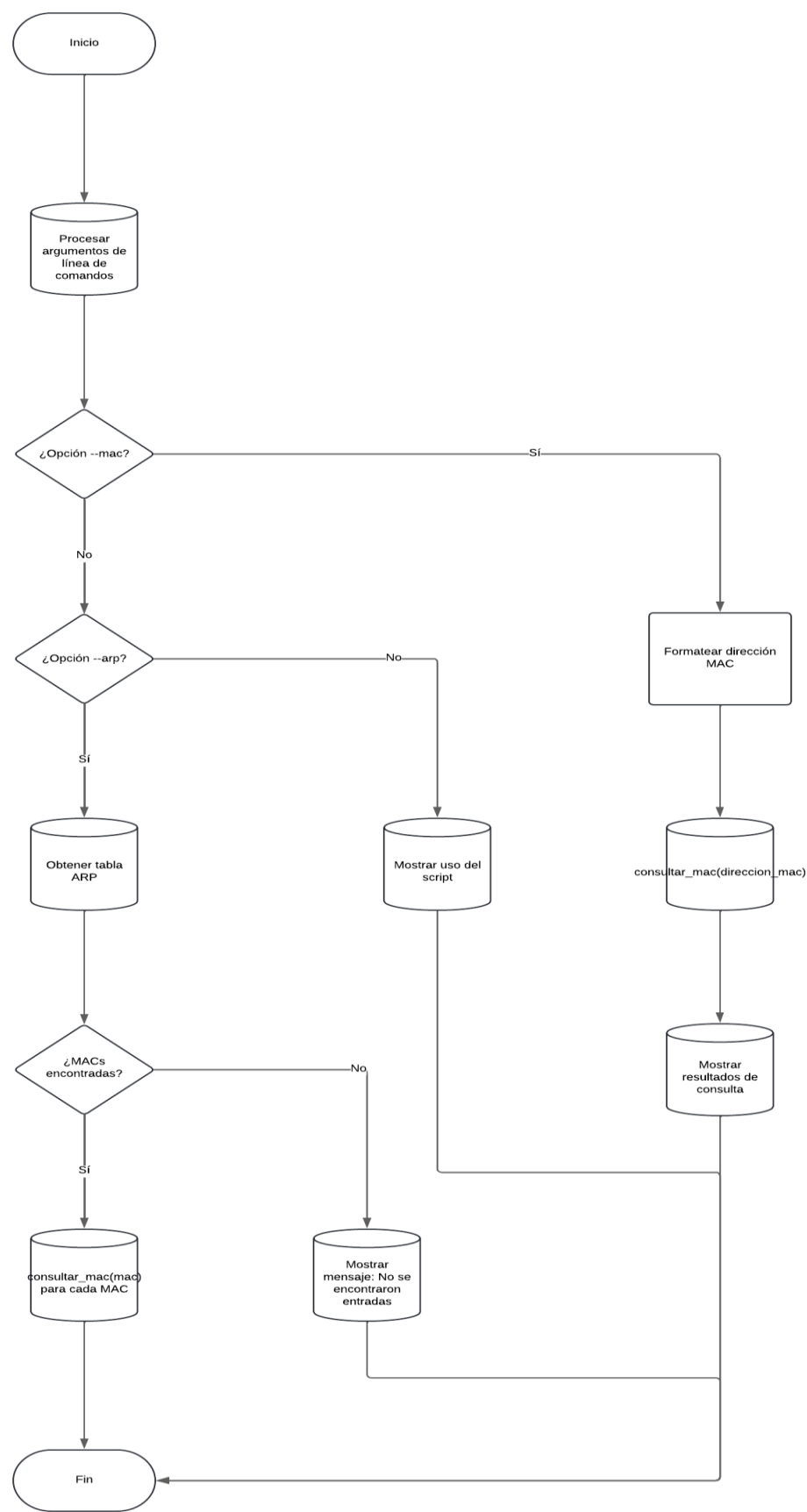
g. Bloque if __name__ == "__main__"

```
102 if __name__ == "__main__":
103     main()
104
```

Ilustración 7. Se ve parte del Código.

- **Ejecución del programa:** Este bloque asegura que la función main() se ejecute cuando se ejecuta el archivo directamente.

Diagrama de flujo de la solución:



4. Pruebas

Casos MAC que están en la base de datos

```
● PS C:\Users\crisb\OneDrive\Desktop\tarea> python3 OUILookup.py --mac 98:06:3c:92:ff:c5
Consultando fabricante para la MAC: 98:06:3c:92:ff:c5
MAC address   : 98:06:3c:92:ff:c5
Fabricante    : Samsung Electronics Co.,Ltd
Tiempo de respuesta: 448.91ms
```

Ilustración 8. Primera prueba del Código

```
● PS C:\Users\crisb\OneDrive\Desktop\tarea> python3 OUILookup.py --mac 9c:a5:13
Consultando fabricante para la MAC: 9c:a5:13
MAC address   : 9c:a5:13
Fabricante    : Samsung Electronics Co.,Ltd
Tiempo de respuesta: 399.60ms
```

Ilustración 9. Segunda prueba del Código.

```
● PS C:\Users\crisb\OneDrive\Desktop\tarea> python3 OUILookup.py --mac 48-E7-DA
Consultando fabricante para la MAC: 48:e7:da
MAC address   : 48:e7:da
Fabricante    : AzureWave Technology Inc.
Tiempo de respuesta: 478.34ms
```

Ilustración 10. Tercera prueba del Código.

Caso MAC que no está en la base de datos

```
● PS C:\Users\crisb\OneDrive\Desktop\tarea> python3 OUILookup.py --mac 98:06:3f:92:ff:c5
Consultando fabricante para la MAC: 98:06:3f:92:ff:c5
MAC address   : 98:06:3f:92:ff:c5
Fabricante    : Not found
Tiempo de respuesta: 427.78ms
```

Ilustración 11. Cuarta prueba del Código.

5. Direcciones MAC aleatorias

Como ya se mencionó anteriormente, las direcciones MAC (Media Access Control) son identificadores únicos que se asignan a los dispositivos que se conectan a redes, como laptops, celulares o tablets. Sirven para que el dispositivo sea identificado en la red y que los datos sean enviados al lugar correcto, o sea, que cuando envías o recibes información por una red Wi-Fi, tu dispositivo tenga una dirección identificable. Antes, estas direcciones eran fijas, es decir, no cambiaban nunca, lo cual podía ser un problema para la privacidad porque cualquier red podría seguir el rastro de un dispositivo solo por su dirección MAC.

Para solucionar ese problema, los dispositivos modernos ahora usan direcciones MAC aleatorias. Esto significa que el dispositivo puede generar una dirección nueva, sobre todo

cuando busca una red Wi-Fi a la que conectarse o cuando está en redes públicas. Así, se evita que sea tan fácil rastrear el dispositivo.

¿Cómo funcionan las MAC aleatorias?

Los sistemas operativos actuales, como iOS, Android y Windows 10, ya incluyen la opción de usar direcciones MAC aleatorias. Cada vez que un dispositivo busca una red Wi-Fi, genera una nueva dirección, haciendo que sea más difícil para alguien asociar esas conexiones con un mismo dispositivo.

Además, este concepto también se ha aplicado a tecnologías como el Bluetooth, donde el dispositivo genera una dirección MAC aleatoria cuando se conecta o busca dispositivos cercanos. Esto ayuda a proteger la privacidad, haciendo más difícil identificar el dispositivo de manera constante.

Beneficios y problemas

El principal beneficio de usar direcciones MAC aleatorias es que protegen la privacidad, porque evita que el dispositivo sea rastreado fácilmente. Sin embargo, también puede traer algunos problemas. Por ejemplo, en redes empresariales o corporativas, donde se necesita una dirección MAC fija para autenticar los dispositivos. También, algunos servicios requieren que la dirección MAC no cambie para mantener una conexión estable y confiable.

6. Discusión y conclusiones

En conclusión, el desarrollo de la herramienta OUILookup ha demostrado ser una solución para obtener información sobre el fabricante de dispositivos de red a partir de su dirección MAC. A través del uso de la API REST pública de <https://maclookup.app> se puede consultar datos relevantes para la gestión y seguridad de redes. Al ocupar Python permite que la herramienta sea accesible desde la línea de comandos, lo que la hace ideal para administradores de redes y profesionales de la ciberseguridad. Las pruebas realizadas confirmaron la precisión de la herramienta en distintos casos de uso, demostrando su capacidad para adaptarse a diferente consulta de fabricantes de las direcciones Mac. Esta solución contribuye significativamente a la gestión de inventarios de dispositivos y al diagnóstico de problemas de conectividad, mejorando la eficiencia en la administración de redes.

7. Referencias

[1] Xataka. "¿Qué es la dirección MAC de tu ordenador, del móvil o de cualquier dispositivo?". Disponible en: <https://www.xataka.com/basics/que-es-la-direccion-mac-de-tu-ordenador-del-movil-o-de-cualquier-dispositivo>.

[2] KeepCoding. "¿Qué es la dirección MAC y para qué sirve?". Disponible en: <https://keepcoding.io/blog/que-es-la-direccion-mac-y-para-que-sirve/>.

[3] Tarlogic. "MAC aleatoria en Bluetooth" Disponible en:
<https://www.tarlogic.com/bsam/es/controles/mac-aleatoria-bluetooth/>.