

# Tarea 2 Redes de Computadores

Simón Canales Carvajal, [simon.canales@alumnos.uv.cl](mailto:simon.canales@alumnos.uv.cl)

## 1. Introducción

En el ámbito de las redes de computadores, la identificación precisa de los dispositivos conectados es esencial para la gestión y seguridad de la red. Las direcciones MAC (Media Access Control) permiten identificar de manera única a cada dispositivo en una red, pero identificar al fabricante de un dispositivo basándose en su dirección MAC puede ser un proceso manual tedioso.

Este trabajo presenta la herramienta "OUILookup", diseñada para automatizar la consulta de fabricantes mediante la dirección MAC de los dispositivos, utilizando una API REST pública. La implementación de esta herramienta permite a los administradores de redes obtener rápidamente la información del fabricante, lo que facilita la gestión de dispositivos y mejora la seguridad de la red.

El objetivo principal de este trabajo es desarrollar una herramienta de línea de comandos que consulte y devuelva el fabricante asociado a una dirección MAC. Entre las principales conclusiones, destaca la eficiencia y rapidez de la herramienta para resolver este problema, así como su capacidad para manejar errores y excepciones, como la falta de información en la base de datos [1].

## 2. Descripción del problema y diseño de la solución

El desafío principal radica en la necesidad de identificar al fabricante de un dispositivo a través de su dirección MAC en redes de computadoras. El proceso manual de verificar el fabricante a través de bases de datos públicas o locales puede resultar ineficiente y lento. Por lo tanto, se requiere una herramienta automatizada que realice consultas a una API pública de fabricantes de MAC y que, además, muestre los fabricantes de las direcciones MAC presentes en la tabla ARP local.

### 2.1. Requerimientos

**2.1.1. Funcionalidad principal:** Consultar el fabricante de una dirección MAC proporcionada por el usuario.

**2.1.2. Otras funcionalidades:** Mostrar los fabricantes de las direcciones MAC presentes en la tabla ARP del sistema.

**2.1.3. Interfaz de usuario:** Basada en línea de comandos, simple de usar.

**2.1.4. API:** Se utilizará la API pública <https://maclookup.app> para la consulta de direcciones MAC.

**2.1.5. Plataformas soportadas:** La herramienta debe funcionar en sistemas operativos Linux y Windows.

## 2.2. Diseño de la solución

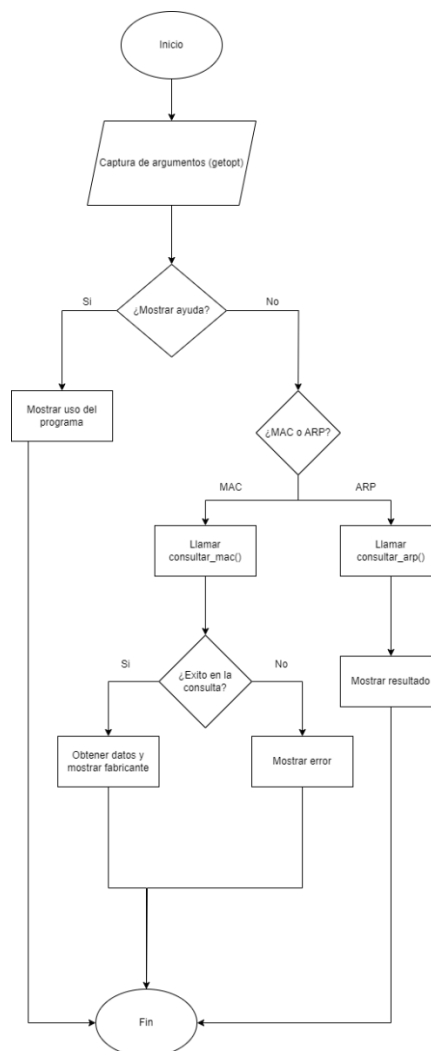
El diseño de la solución se basa en una arquitectura modular compuesta por tres componentes principales:

**2.2.1. Consultor de MAC:** Este módulo interactúa con la API pública para obtener información sobre el fabricante asociado a una dirección MAC.

**2.2.2 ARP Manager:** Este módulo gestiona y muestra las direcciones MAC presentes en la tabla ARP del sistema.

**2.2.3 Interfaz de comandos:** El usuario interactúa con la herramienta mediante argumentos en la línea de comandos para consultar MAC o mostrar la tabla ARP.

## 2.3. Diagrama de flujo:



### 3. Implementación

El código de **OUILookup** se implementó en Python, utilizando la librería requests para interactuar con la API REST. A continuación se describen las partes clave del código:

- 4.1 Consulta de la MAC:** La función consultar\_mac(mac\_address) realiza una solicitud a la API y maneja la respuesta JSON. Si el fabricante no está disponible, la función devuelve "Not found". Se implementó manejo de excepciones con requests.RequestException para gestionar errores de red o fallos en la API.
- 4.2 Simulación de la tabla ARP:** La función consultar\_arp() simula la lectura de la tabla ARP, mostrando las MAC y fabricantes disponibles.
- 4.3 Manejo de argumentos de línea de comandos:** La función main(argv) utiliza la librería getopt para interpretar los argumentos de línea de comandos. El usuario puede ingresar una dirección MAC o elegir mostrar la tabla ARP.

### Desafíos de implementación

Durante el desarrollo, se presentaron desafíos relacionados con el manejo de errores de red y la validación de direcciones MAC. Para resolver estos problemas, se implementaron bloques try-except y se validaron los datos recibidos de la API, asegurando que se manejen adecuadamente las excepciones.

### 4. Pruebas

Las pruebas se realizaron verificando varios casos de uso:

- 4.1. Caso 1: Consulta de una dirección MAC existente en la base de datos de la API [Figura 1]

```
PS C:\Users\canal\OneDrive\Escritorio\Trabajo> python3 OUILookup.py --mac 98:06:3c:92:ff:c5
Argumentos recibidos: [('--mac', '98:06:3c:92:ff:c5')]
MAC address      : 98:06:3c:92:ff:c5
Fabricante       : Samsung Electronics Co.,Ltd
Tiempo de respuesta 618ms
```

Figura 1

- 4.2. Caso 2: Consulta de una dirección MAC no registrada en la API. [Figura 2].

```
PS C:\Users\canal\OneDrive\Escritorio\Trabajo> python3 OUILookup.py --mac 98:06:3f:92:ff:c5
Argumentos recibidos: [('--mac', '98:06:3f:92:ff:c5')]
MAC address      : 98:06:3f:92:ff:c5
Fabricante       : Not found
Tiempo de respuesta 624ms
```

Figura 2

#### 4.3. Caso 3: Visualización de la tabla ARP simulada. [Figura 3].

```
PS C:\Users\canal\OneDrive\Escritorio\Trabajo> python3 OUILookup.py --arp
Argumentos recibidos: [('--arp', '')]
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
```

Figura 3

Cada prueba verificó el correcto funcionamiento de la herramienta, el manejo de excepciones y la correcta interacción con la API.

### 5. Discusión y conclusiones

El desarrollo de la herramienta "OUILookup" permitió automatizar un proceso que, de forma manual, resultaría mucho más tedioso. La herramienta funcionó de manera eficiente en la consulta de fabricantes y la simulación de la tabla ARP. Los resultados fueron precisos, y el manejo de excepciones demostró ser robusto, asegurando que el sistema no fallara ante errores de red o MAC no válidas.

Se aprendió que la interacción con APIs públicas es esencial para obtener información actualizada en tiempo real, y que el manejo adecuado de errores es clave para ofrecer una buena experiencia de usuario. Como posibles mejoras futuras, se podría implementar soporte para otras APIs de consulta de MAC, o agregar una interfaz gráfica para hacerlo más accesible a usuarios menos experimentados.

### 6. Referencias

[1] MacLookup API Documentation. <https://maclookup.app>.