

# Tarea 2

Vicente Cruces Collao, vicente.cruces@alumnos.uv.cl

## 1. Introducción

Se desarrollo una herramienta de línea de comandos en Python con el nombre de "OUILookup" para consultar el nombre del fabricante de red a través del uso de una dirección MAC con una API pública.

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

Este proyecto se centra en la utilización de las direcciones MAC (Media Access Control) que es una dirección única asignada a la interfaz de red para la comunicación de la capa de enlace de datos del modelo OSI

El objetivo del programa OUILookup es consultar el nombre del fabricante de red, obteniendo su OUI y la utilización de una API publica que provee esta información.

## 3. Implementación

El programa OUILookup fue creado e implementado en Python para poder ejecutarse desde una línea de comando (CMD) a través de uso de librerías para gestionar los parámetros de entrada y para consultar a la API publica que se ha utilizado en maclookup.app.

## 4. Pruebas

### 4.1. Código Python

A continuación, la explicación del código Python

```
1 import sys
2 import getopt
3 import requests
4 import time
5
6 def lookup_mac(mac_address):
7
8     start_time = time.time()
9
10
11     url = f"https://api.maclookup.app/v2/mac/{mac_address}"
12     response = requests.get(url)
13
14
15     end_time = time.time()
16     response_time = (end_time - start_time) * 1000
17
18     if response.status_code == 200:
19         data = response.json()
20         fabricante = data.get('company', 'Fabricante no encontrado')
21     else:
22         fabricante = "Error al consultar la API."
23
24     return fabricante, response_time
25
26 def main(argv):
27     mac_address = ''
28     try:
29         opts, args = getopt.getopt(argv, "hm:", ["help", "mac="])
30     except getopt.GetoptError:
31         print('Uso: OUILookup.py --mac <direccion_MAC>')
32         sys.exit(2)
33     for opt, arg in opts:
34         if opt in ("-h", "--help"):
35             print('Uso: OUILookup.py --mac <direccion_MAC>')
36             sys.exit()
37         elif opt in ("-m", "--mac"):
38             mac_address = arg
39
40     if mac_address:
41         fabricante, response_time = lookup_mac(mac_address)
42         print(f"MAC address: {mac_address}\nFabricante: {fabricante}\nTiempo de respuesta: {response_time:.2f} ms")
43     else:
44         print('Uso: OUILookup.py --mac <direccion_MAC>')
45
46 if __name__ == "__main__":
47     main(sys.argv[1:])
48
```

Figura 1

Librerías:

```
1 import sys
2 import getopt
3 import requests
4 import time
```

Figura 2

En la figura 2 importaremos estas librerías para el uso de esta tarea

-sys: captura los argumentos de la línea de comandos para finalizar el programa.

- getopt: Gestiona las opciones [--mac, --help] para permitir que el usuario pase la dirección MAC o solicite ayuda.
- requests: Realiza una solicitud HTTP a una API para obtener información sobre la dirección MAC.
- time: calcula cuanto tiempo se tarda en ejecutar el programa en milisegundos.

Funciones:

```
def lookup_mac(mac_address):  
  
    start_time = time.time()  
  
    url = f"https://api.maclookup.app/v2/macs/{mac_address}"  
    response = requests.get(url)  
  
    end_time = time.time()  
    response_time = (end_time - start_time) * 1000  
  
    if response.status_code == 200:  
        data = response.json()  
        fabricante = data.get('company', 'Fabricante no encontrado')  
    else:  
        fabricante = "Error al consultar la API."  
  
    return fabricante, response_time
```

Figura 3.

El funcionamiento de esta función `lookup_mac(mac_address)` es:

- realizar una solicitud a la API `maclookup.app` utilizando una dirección MAC ingresada como parámetro en el CMD
- mide el tiempo de la respuesta
- retornar el nombre del fabricante de la dirección MAC y su tiempo de respuesta

```
def main(argv):
    mac_address = ''
    try:
        opts, args = getopt.getopt(argv, "hm:", ["help", "mac="])
    except getopt.GetoptError:
        print('Uso: OUILookup.py --mac <direccion_MAC>')
        sys.exit(2)
    for opt, arg in opts:
        if opt in ("-h", "--help"):
            print('Uso: OUILookup.py --mac <direccion_MAC>')
            sys.exit()
        elif opt in ("-m", "--mac"):
            mac_address = arg

    if mac_address:
        fabricante, response_time = lookup_mac(mac_address)
        print(f"MAC address: {mac_address}\nFabricante: {fabricante}\nTiempo de respuesta: {response_time:.2f} ms")
    else:
        print('Uso: OUILookup.py --mac <direccion_MAC>')
```

Figura 4.

El funcionamiento de la función main(argv):

- se encarga de gestionar los argumentos de la línea de comando
- llama a la función lookup\_mac() cuando se ingresa una dirección MAC con el argumento -mac.
- muestra el mensaje de ayuda si ingresa con -help.(ejemplo en la figura 5)

```
PS C:\Users\vicru\Desktop\codigo> python OUILookup.py --help
>>
Uso: OUILookup.py --mac <direccion_MAC>
```

Figura 5.

#### 4.2) Caso de pruebas

Se realizó distintas pruebas utilizando direcciones MAC ya conocidas, se mostrarán algunos ejemplos de ejecución.

Ejemplo con la MAC 98:06:3c:92:ff:c5

```
PS C:\Users\vicru\Desktop\codigo> 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: 900.43 ms
```

Figura 6.

Ejemplo con la MAC 9c:a5:13

```
PS C:\Users\vicru\Desktop\codigo> python OUILookup.py --mac 9c:a5:13
MAC address: 9c:a5:13
Fabricante: Samsung Electronics Co.,Ltd
Tiempo de respuesta: 609.71 ms
```

Figura 7.

### Ejemplo con la MAC 48-E7-DA

```
PS C:\Users\vicru\Desktop\codigo> python OUILookup.py --mac 48-E7-DA
MAC address: 48-E7-DA
Fabricante: AzureWave Technology Inc.
Tiempo de respuesta: 627.59 ms
```

Figura 8.

### 4.3) Pruebas de Mac no existentes

Para comprobar si un MAC no existe tomaremos la MAC de la figura 6 (98:06:3c:92:ff:c5) y cambiaremos el 3c a 3f

```
Uso: OUILookup.py --mac <direccion_MAC>
PS C:\Users\vicru\Desktop\codigo> python OUILookup.py --mac 98:06:3f:92:ff:c5
>>
MAC address: 98:06:3f:92:ff:c5
Fabricante:
Tiempo de respuesta: 576.37 ms
```

Figura 9.

Como se da cuenta en la casilla del fabricante se muestra en blanco a lo que significa que no se pudo encontrar el fabricante a lo que implica que es dirección de MAC no existe.

## 5. Discusión y conclusiones

El desarrollo de OUILookup ha permitido poner en práctica conceptos importantes relacionados con las redes de computadores, como las direcciones MAC, su formato, y la consulta de fabricantes a través de APIs públicas.

-El programa fue capaz de realizar consultas de manera eficiente, mostrando resultados en tiempos de respuesta razonables (entre 576.37 ms y 900 ms, dependiendo de la red).

-Se identificaron algunos factores que podrían mejorar el rendimiento, como la optimización de la conexión de red o el uso de caché para evitar múltiples solicitudes a la API en un corto período de tiempo.

En conclusión, el objetivo de la tarea funcionó y tuvo un rendimiento excelente.