

Tarea 2

Bruno Osega Soza, bruno.osega@alumnos.uv.cl

Joaquin Astudillo Castro, joaquin.astudillo@alumnos.uv.cl

1. Introducción

- Antes de introducir el trabajo debemos saber que es la dirección MAC, la dirección MAC es una identificación única asignada a la interfaz de red de un dispositivo , cada dispositivo conectado a una red ya sea un ordenador , un teléfono inteligente , una impresora o incluso un dispositivo IoT tiene una propia dirección MAC. Este tiene una serie de 12 caracteres hexadecimales dividido en 6 pares las cuales son
- **Primeros seis caracteres (OUI):** Estos identifican al fabricante del dispositivo y se conocen como [Organizational Unique Identifier](#) (OUI).
- **Últimos seis caracteres (NIC):** Estos identifican de manera única al dispositivo dentro de la red y se conocen como Número de Identificación de Componente de Red (NIC).[1]
- Ya sabiendo el contexto sobre lo que se basa la dirección MAC este trabajo se basa en realizar una herramienta llamada OUILookup que este permitirá consultar los fabricantes de las tarjetas de redes a través de una dirección MAC y que este se permitirá al utilizar una API REST y que mostrará el nombre del fabricante de cada tarjeta de red que será analizado.

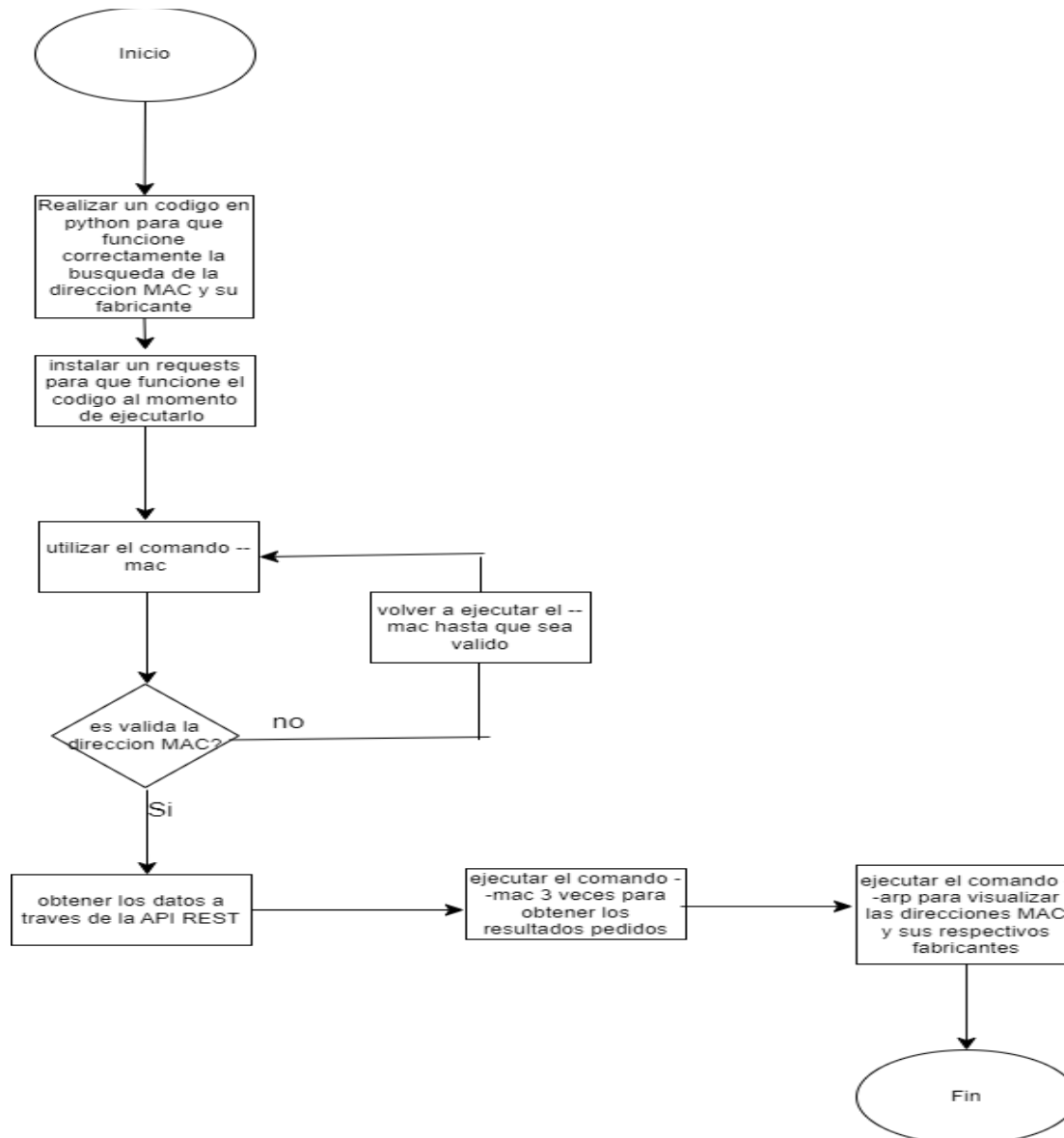
2. Descripción del problema y diseño de la solución

- El problema por realizar se debe implementar una herramienta que su función es consultar el fabricante de las tarjetas de redes en base a su dirección MAC y esta herramienta es desarrollada en el lenguaje Python. Y además contara con una API REST que permita tener los fabricantes de la tarjeta de red a partir de la dirección MAC para poder conocer a los fabricantes.
- La solución de este problema es crear una herramienta a partir del lenguaje Python que permita el uso de comandos y que esta haga consultas a la API REST respectiva para obtener más información sobre los fabricantes

3. Implementación

- La implementación de la herramienta OUILookup se basa en hacer que se pueda consultar el fabricante de las tarjetas de red a través de las direcciones MAC y además se utiliza la tabla ARP y además este se basa en mostrar las direcciones MAC que están en una API REST publica, en nuestro caso se aplicó la herramienta en Windows.

Diagrama de flujos:



4. Pruebas

-Este es para que el código funcione correctamente

```
Símbolo del sistema
C:\Users\bruno\Desktop>pip install requests
Collecting requests
  Downloading requests-2.32.3-py3-none-any.whl.metadata (4.6 kB)
Collecting charset-normalizer<4,>=2 (from requests)
  Downloading charset_normalizer-3.3.2-cp311-cp311-win_amd64.whl.metadata (34 kB)
Collecting idna<4,>=2.5 (from requests)
  Downloading idna-3.10-py3-none-any.whl.metadata (10 kB)
Collecting urllib3<3,>=1.21.1 (from requests)
  Downloading urllib3-2.2.3-py3-none-any.whl.metadata (6.5 kB)
Collecting certifi<2017.4.17 (from requests)
  Downloading certifi-2024.8.30-py3-none-any.whl.metadata (2.2 kB)
Downloading requests-2.32.3-py3-none-any.whl (64 kB)
----- 64.9/64.9 kB 1.8 MB/s eta 0:00:00
Downloading certifi-2024.8.30-py3-none-any.whl (167 kB)
----- 167.3/167.3 kB 3.4 MB/s eta 0:00:00
Downloading charset_normalizer-3.3.2-cp311-cp311-win_amd64.whl (99 kB)
----- 99.9/99.9 kB 6.0 MB/s eta 0:00:00
Downloading idna-3.10-py3-none-any.whl (70 kB)
----- 70.4/70.4 kB 4.0 MB/s eta 0:00:00
Downloading urllib3-2.2.3-py3-none-any.whl (126 kB)
----- 126.3/126.3 kB 7.3 MB/s eta 0:00:00
Installing collected packages: urllib3, idna, charset-normalizer, certifi, requests
WARNING: The script normalizer.exe is installed in 'C:\Users\bruno\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed certifi-2024.8.30 charset-normalizer-3.3.2 idna-3.10 requests-2.32.3 urllib3-2.2.3

[notice] A new release of pip is available: 24.0 -> 24.2
[notice] To update, run: C:\Users\bruno\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\python.exe -m pip install --upgrade pip
```

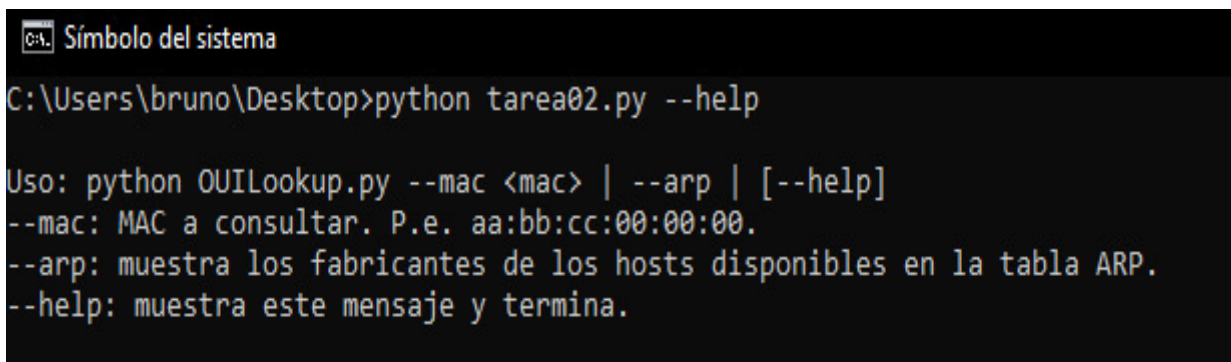
Esto es el resultado de utilizar el comando `--mac` que hace que este envíe una solicitud a la API REST para poder obtener la información necesario para obtener los resultados que se piden

```
Símbolo del sistema
C:\Users\bruno\Desktop>python tarea02.py --mac 9c:a5:13
MAC address : 9c:a5:13
Fabricante : Samsung Electronics Co.,Ltd
Tiempo de respuesta: 1031ms
```

```
Símbolo del sistema
C:\Users\bruno\Desktop>python tarea02.py --mac 48-E7-DA
MAC address : 48-E7-DA
Fabricante : AzureWave Technology Inc.
Tiempo de respuesta: 1198ms
```

```
Símbolo del sistema
C:\Users\bruno\Desktop>python tarea02.py --mac 98:06:3f:92:ff:c5
MAC address : 98:06:3f:92:ff:c5
Fabricante :
Tiempo de respuesta: 1178ms
```

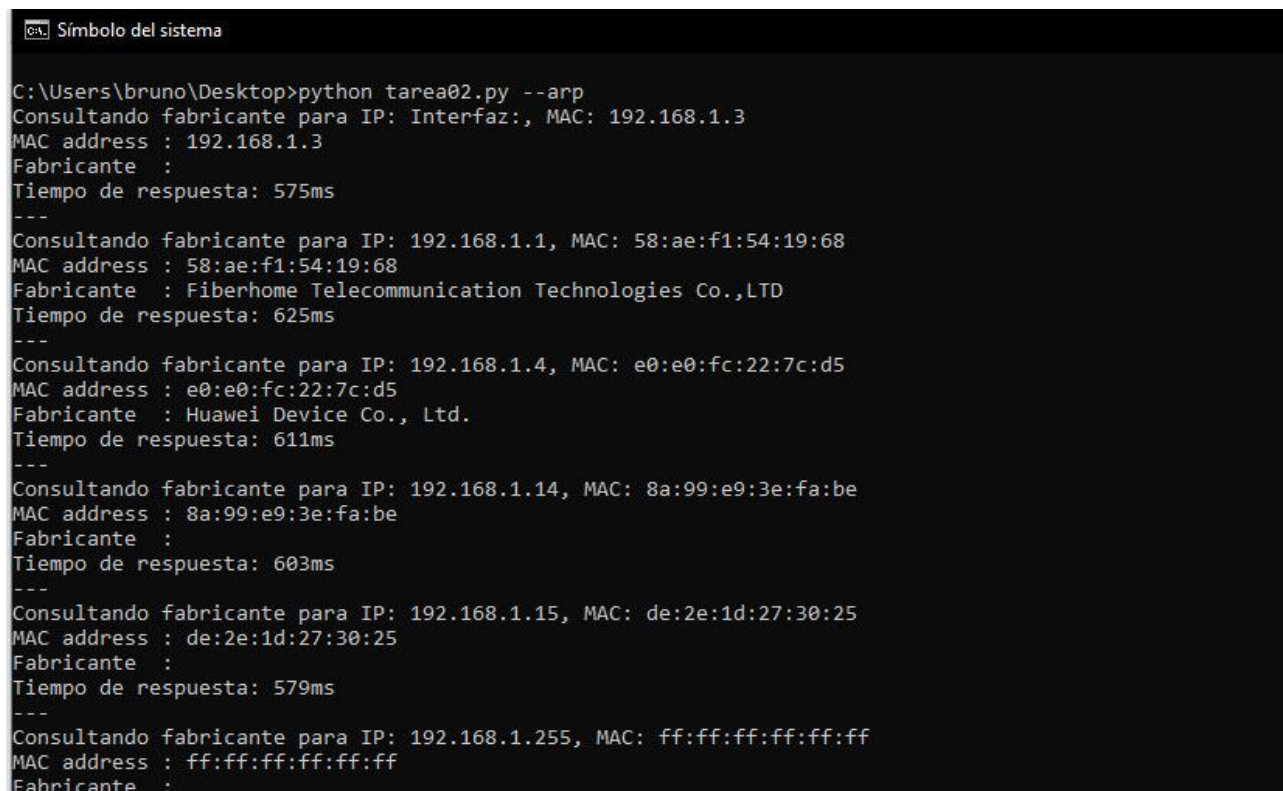
El resultado de utilizar el comando `--help` que visualiza la funcionalidad de cada comando y algún ejemplo en este caso de como utilizar el comando `--mac`



```
C:\Users\bruno\Desktop>python tarea02.py --help

Uso: python 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.
```

El resultado de aplicar el comando `--arp` que hace que muestre los resultados de los fabricantes, sus direcciones MAC y su respectivo tiempo de ejecución que se encuentran en la tabla arp



```
C:\Users\bruno\Desktop>python tarea02.py --arp
Consultando fabricante para IP: Interfaz:, MAC: 192.168.1.3
MAC address : 192.168.1.3
Fabricante :
Tiempo de respuesta: 575ms
---
Consultando fabricante para IP: 192.168.1.1, MAC: 58:ae:f1:54:19:68
MAC address : 58:ae:f1:54:19:68
Fabricante : Fiberhome Telecommunication Technologies Co.,LTD
Tiempo de respuesta: 625ms
---
Consultando fabricante para IP: 192.168.1.4, MAC: e0:e0:fc:22:7c:d5
MAC address : e0:e0:fc:22:7c:d5
Fabricante : Huawei Device Co., Ltd.
Tiempo de respuesta: 611ms
---
Consultando fabricante para IP: 192.168.1.14, MAC: 8a:99:e9:3e:fa:be
MAC address : 8a:99:e9:3e:fa:be
Fabricante :
Tiempo de respuesta: 603ms
---
Consultando fabricante para IP: 192.168.1.15, MAC: de:2e:1d:27:30:25
MAC address : de:2e:1d:27:30:25
Fabricante :
Tiempo de respuesta: 579ms
---
Consultando fabricante para IP: 192.168.1.255, MAC: ff:ff:ff:ff:ff:ff
MAC address : ff:ff:ff:ff:ff:ff
Fabricante :
```

- Las fotos son el resultado de realizar la herramienta de OUILookup realizado en Python y al momento de ejecutarlo después de aplica en el terminal y aplicar los comandos necesario para poder encontrar la MAC de cada fabricante y se aseguró el resultado optimo.

5. Discusión y conclusiones

- Una dirección MAC puede ser usada para identificar pasivamente un dispositivo como teléfono, tableta o computador a través de los puntos de acceso de la red; y debido a esto, los proveedores de sistemas operativos comenzaron a introducir la aleatorización de la dirección MAC, para reforzar la seguridad de los usuarios de los dispositivos. Esto funciona cambiando la dirección MAC anunciada en el dispositivo cada poco minutos, lo cual implica que, aunque sea trivial determinar si una dirección dada es real o aleatorizada, es imposible determinar si una dirección MAC aleatorizada dada pertenece al mismo dispositivo que la última dirección MAC aleatorizada vista. Por ello, a menos que un dispositivo esté asociado a una red de Wifi, será casi imposible relacionarlo con un usuario o determinar si el dispositivo ya se ha conectado antes. También por ello se ha dificultado considerablemente el conteo de dispositivos únicos en una ubicación, puesto que un mismo dispositivo puede ser representado por múltiples direcciones MAC aleatorizadas.[2]
- En el caso de Android, la generación aleatoria de direcciones MACs empieza en Marshmallow, la versión 6 del sistema operativo de Google. Según la nota de cambios en Android 6, para mejorar la privacidad de los dispositivos los desarrolladores no tendrán acceso a la dirección MAC del hardware bluetooth y WI-Fi para usar en sus aplicaciones, y a la hora de intentar obtenerla se consigue el valor constante de 02:00:00:00:00:00. [3]
- **Conclusión:** Al ejecutar se analiza los fabricantes que se encuentra a partir de las direcciones MAC se mostraran los resultados desde un terminal que mostrara el MAC adrees respectivo , el fabricante de la tarjeta de red y el tiempo en que se demora en realizar la ejecución y además se aprendió a como interactuar con una API REST que en el futuro nos servirá para poder realizar con mayor facilidad paginas/servicios web que estén con una base con API REST y además se aprendió a utilizar comandos para poder obtener información sobre las MAC de distintas tarjetas de red que al momento de realizar un mayor estudio podremos encontrar la MAC de distintos dispositivos electrónicos.

6. Referencias

[1] <https://keepcoding.io/blog/que-es-la-direccion-mac-y-para-que-sirve/>

[2]<https://support.purple.ai/hc/es/articles/7330834299805-Aleatorizaci%C3%B3n-de-la-direcci%C3%B3n-MAC>

[3] <https://telefonicatech.com/blog/mac-aleaatorias-y-privacidad-parte2>