Informe Tarea n° 2

Felipe Robles Naveas, [felipe.roblesn@alumnos.uv.cl](mailto:felipe.roblesn@alumnos.uv.cl)

# Introducción

Para esta tarea, se me ha solicitado realizar un programa en lenguaje Python basado en línea de comandos para consultar el fabricante de cualquier tarjeta red. Para esto, se utilizó una API REST pública como base de datos, (en este caso la API es [1]), la cual entregará el fabricante de cualquier tarjeta usando su dirección MAC fija.

# Desarrollo del programa

Para que este programa funcionara había que lograr acceder a la API sin abrir un navegador externo, (Chrome, FireFox, etc), y lograr almacenar la respuesta de la página. Esta página siempre devuelve una respuesta en formato JSON, por lo que hay que lograr guardar el texto JSON, y traducirlo en un formato legible para Python. Para esto se utilizó la librería “requests” [2], la cual da la capacidad de almacenar la respuesta de la página en una variable, y traducirla en un diccionario legible para Python.

# Implementación

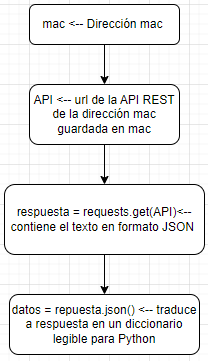
En el código hay 2 funciones importantes para que el programa pueda obtener el fabricante y guardar la mac y su fabricante asociado.

## Buscador\_MAC(mac)

Esta función tiene la tarea de acceder a la API, y guardar en una variable la respuesta donde se encuentra el fabricante.

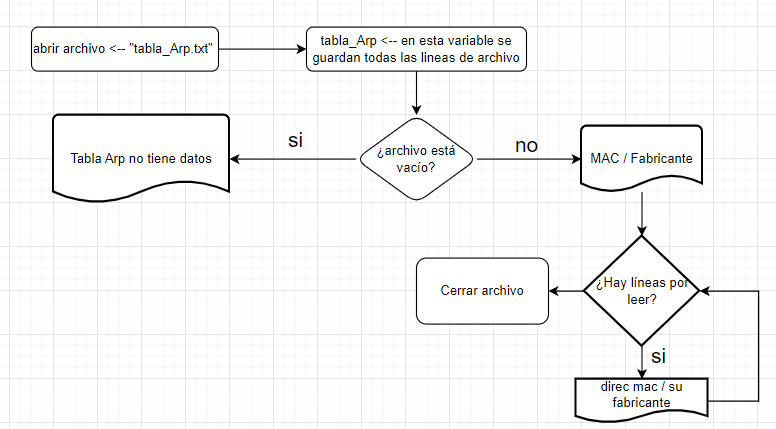
Para esto, usando la librería requests, se guarda una url de la API con la MAC ingresada en la línea de comando en una variable, la cual se usa para hacerle la solicitud a la API obteniendo como respuesta un texto JSON. Para al final traducir el texto JSON en un texto

legible para Python.



**3.2. tabla\_Arp()**

Como se mencionó anteriormente, las direcciones mac y sus fabricantes se dejaban guardadas en un archivo de texto, esto se hizo debido a que el programa se ejecuta una sola vez, por lo tanto la manera mas sencilla de poder almacenar los datos, es en un archivo de texto llamada “tabla\_Arp.txt”. Para escribir y leer en el archivo se usó la librería “os”, si el archivo tiene datos los va a printear, si no va a printear “La tabla Arp no tiene datos”.



# Direcciones MAC aleatorias

En los dispositivos electrónicos una parte primordial es la tarjeta de red la cual tiene una dirección MAC fija [3], la dirección tiene la función de actuar como un identificador único para el dispositivo en una red local. Pero si una persona logra obtener tu dirección MAC, podría generarte problemas como que la falsifiquen para ser usada en otro dispositivo, o la filtración de información de tu equipo.

Para evitar estos problemas existen las direcciones MAC aleatorias [4], las cuales son generadas por el dispositivo cuando realizas ciertos procesos como conectarse a una red WIFI, proporcionándole mayor privacidad y seguridad a tu dispositivo.  
Para diferenciar entre una MAC fija y una aleatoria debes fijarte en el segundo bit más significativo de los primeros dos bytes hexadecimales, (U/L). Si el U/L es un 0 es una MAC fija, pero si es un 1 es una MAC aleatoria.

Por ejemplo:

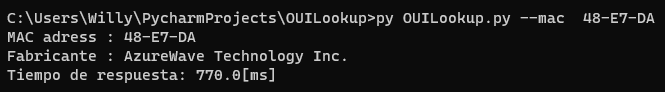
La dirección MAC [98:06:3c:92:ff:c5], los primeros dos bytes son 98, y en binario serían (1001 1000) aquí el segundo bit más significativo es el 0 (1001 1000), por lo es una dirección MAC fija.

La dirección MAC [03:9c:a5:3c:92:ff], los primeros dos bytes son 03, y en binario serían (0000 0011) aquí el segundo bit más significativo es el 1 (0000 0011), por lo es una dirección MAC aleatoria.

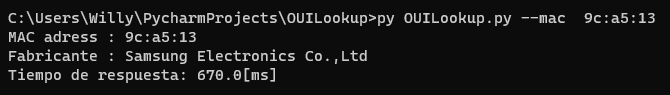
# Pruebas

Para corroborar que el código funcione correctamente, se ejecutó 3 veces para buscar los fabricantes de estas 3 direcciones MAC

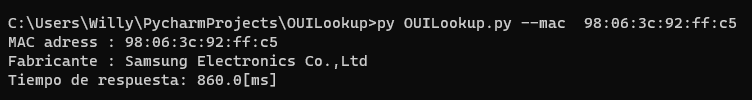
1) 48-E7-DA



2) 9c:a5:13



3) 98:06:3c:92:ff:c5



Aunque ninguna de las 3 direcciones MAC son iguales, el código se ejecutó perfectamente. Esto debido a la estructura de la dirección MAC y al funcionamiento de la API.

Por un lado, una dirección MAC se conforma de 12 caracteres hexadecimales y los primeros 6, (OUI)[5], proporcionan al fabricante del dispositivo. Por lo tanto, la página solo necesita 6 caracteres hexadecimales como mínimo para encontrar al fabricante, igualmente si revisan la documentación de la API en el título secundario “Request” [6], te saldrán los formatos de las direcciones MAC aceptados por la página.

# Discusión y conclusiones

Para ir concluyendo, la elaboración de este proyecto fue algo engorroso, aunque la idea era muy simple cosas tan pequeñas como lograr conectarse a la página o procesarlo en getopt terminó complicando la elaboración de este. Igualmente, habría sido bueno poder guardar las Mac con su respectivo fabricante sin tener que guardarlos en un archivo de texto.

# Referencias

[1] <https://maclookup.app>

[2] <https://www.geeksforgeeks.org/python-requests-tutorial/>

[3]<https://keepcoding.io/blog/que-es-la-direccion-mac-y-para-que-sirve/#%C2%BFPara_que_sirve_la_direccion_MAC>

[4] <https://techsearch.watchguard.com/KB?type=Article&SFDCID=kA16S000000SOFySAO&lang=en_US>

[5]<https://keepcoding.io/blog/que-es-la-direccion-mac-y-para-que-sirve/#%C2%BFQue_es_la_direccion_MAC>

[6] <https://maclookup.app/api-v2/documentation>