

Tarea 2: Implementación de un OUI Lookup Tool.

Integrantes: Franco Contreras – Alexander Farias – Nicolas Flores.

1. Introducción

En este informe, se implementa el desarrollo de la herramienta OUI Lookup Tool, que se trata sobre una aplicación en línea de comandos que se desarrolla en Python.

El propósito de esta herramienta es consultar su fabricante de la tarjeta red dada su dirección Mac o su dirección IP. La base de datos entregada está parametrizada como una variable global y se basa en datos ficticios para fines de demostración. La implantación se ha desarrollado siguiendo los parámetros solicitados en la tarea y puede ser ejecutada en sistema operativo Windows o Linux.

El objetivo principal de este trabajo se centra en la implementación de OUI Lookup Tool, donde se desarrolla una aplicación de línea con comandos en Python que permite a los usuarios consultar el fabricante de una tarjeta red basándose en su dirección MAC o dirección IP.

2. Materiales y Métodos

Para poder implementar la herramienta OUI Lookup, se utilizaron los siguientes materiales y métodos:

Python : Lenguaje de programación que se utilizó.

Base de datos: Base de datos que se entregó .

Librerías de Python: Se utilizaron diversas librerías “Geopt” para procesar las líneas de comandos y “ipaddress” para manejar direcciones IP.

Sistema operativo: Sistema en el cual fue implementado en este caso Windows.

Documentación: Informe en pdf que se describe el trabajo realizado

Readme: Archivo creado en el repositorio para describir integrantes del grupo y correos.

Repositorio Github: Repositorio creado para guardar el proyecto, informe y readme.

Los métodos que se utilizaron fueron los siguientes:

- Obtener datos por IP: Esta función permite obtener los datos del fabricante de una tarjeta de red a partir de una dirección IP.
- Obtener datos por MAC: Esta función permite obtener los datos del fabricante de una tarjeta de red a partir de una dirección MAC.

- Obtener tabla ARP: Esta función muestra los fabricantes de las direcciones MAC disponibles en una tabla de ARP.

3. Resultados

La implementación de OUI Lookup fue realizada con éxito y es capaz de responder a todas las consultas propuestas en la pauta. Donde se utilizó la IP 192.168.1.30 y su máscara entregada fue de 255.255.255.0 (/24).

La implementación se puede ejecutar en sistemas operativos como Windows y Linux, en este caso se utilizó Windows para realizar este trabajo. Se creó un repositorio GitHub para el proyecto, a continuación se presentan ejemplos del código implementado.

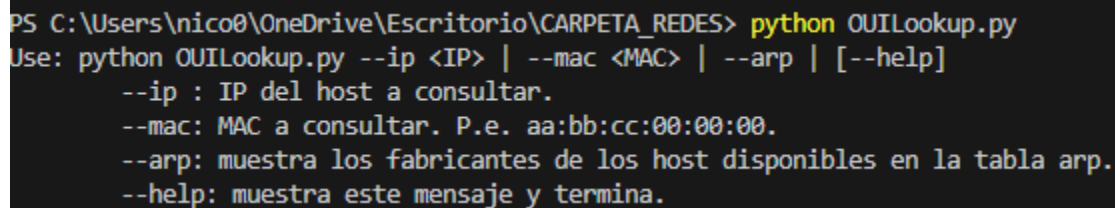
3.1 Ejemplo de uso sin parámetros o con la opción -- help.

La primera consulta entregada fue el uso de la opción help, donde entrega como información cómo utilizar la implementación en óptimas condiciones :

```
# python3 OUILookup.py
Use: python OUILookup.py --ip <IP> | --mac <IP> | --arp | [--help]
--ip : IP del host a consultar.
--mac: MAC a consultar. P.e. aa:bb:cc:00:00:00.
--arp: muestra los fabricantes de los host
      disponibles en la tabla arp.
--help: muestra este mensaje y termina.
```

Figura 1. Esta es una imagen demostrando el ejemplo de implementación con los parámetros entregados.

A continuación se observará la respuesta generada a través de la implementación de nuestro código, cabe destacar que se utilizó el sistema operativo de Windows:



```
PS C:\Users\nico0\OneDrive\Escritorio\CARPETA_REDES> python OUILookup.py
Use: python OUILookup.py --ip <IP> | --mac <MAC> | --arp | [--help]
--ip : IP del host a consultar.
--mac: MAC a consultar. P.e. aa:bb:cc:00:00:00.
--arp: muestra los fabricantes de los host disponibles en la tabla arp.
--help: muestra este mensaje y termina.
```

Figura 1. Esta es una imagen demostrando la respuesta de la consultada entregada.

3.2 Caso IP que pertenezca a su misma red

La segunda consulta entregada fue el uso de que la IP pertenece a su misma red, donde entrega su MAC address y su fabricante correspondiente:

Caso ip que pertenezca a su misma red

```
# python3 OUILookup.py --ip 192.168.1.5  
MAC address : b4:b5:fe:92:ff:c5  
Fabricante   : Hewlett Packard
```

Figura 3. Esta es una imagen demostrando el ejemplo de implementación con los parámetros entregados.

A continuación se observará la respuesta generada a través de la implementación de nuestro código, cabe destacar que se utilizó el sistema operativo de Windows:

```
PS C:\Users\nico0\OneDrive\Escritorio\CARPETA_REDES> python OUILookup.py --ip 192.168.1.5  
MAC address : b4:b5:fe:92:ff:c5  
Fabricante : Hewlett Packard
```

Figura 4. Esta es una imagen demostrando la respuesta de la consulta entregada.

3.3 Caso IP que NO pertenezca a su misma red

La tercera consulta entregada fue el uso de que la IP NO pertenece a su misma red, donde entrega que la IP está afuera de la red del HOST:

Caso ip que NO pertenezca a su misma red

```
# python3 OUILookup.py --ip 192.168.10.5  
Error: ip is outside the host network
```

Figura 5. Esta es una imagen demostrando el ejemplo de implementación con los parámetros entregados.

A continuación se observará la respuesta generada a través de la implementación de nuestro código, cabe destacar que se utilizó el sistema operativo de Windows:

```
PS C:\Users\nico0\OneDrive\Escritorio\CARPETA_REDES> python OUILookup.py --ip 192.168.10.5  
Error: ip is outside the host network
```

Figura 6. Esta es una imagen demostrando la respuesta de la consulta entregada.

3.4 Caso MAC que esté en la base de datos

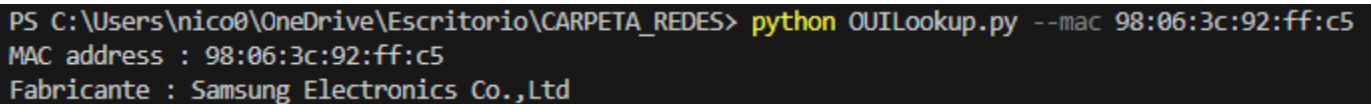
La cuarta consulta entregada fue el uso de la MAC esté en la base de datos, donde entrega la MAC address y su Fabricante correspondiente:

Caso MAC que esté en la base de datos

```
# python3 OUILookup.py --mac 98:06:3c:92:ff:c5  
MAC address : 98:06:3c:92:ff:c5  
Fabricante   : Samsung Electronics Co.,Ltd
```

Figura 7. Esta es una imagen demostrando el ejemplo de implementación con los parámetros entregados.

A continuación se observará la respuesta generada a través de la implementación de nuestro código, cabe destacar que se utilizó el sistema operativo de Windows:



```
PS C:\Users\nico0\OneDrive\Escritorio\CARPETA_REDES> python OUILookup.py --mac 98:06:3c:92:ff:c5  
MAC address : 98:06:3c:92:ff:c5  
Fabricante : Samsung Electronics Co.,Ltd
```

Figura 8. Esta es una imagen demostrando la respuesta de la consulta entregada.

3.5 Caso MAC que NO esté en la base de datos

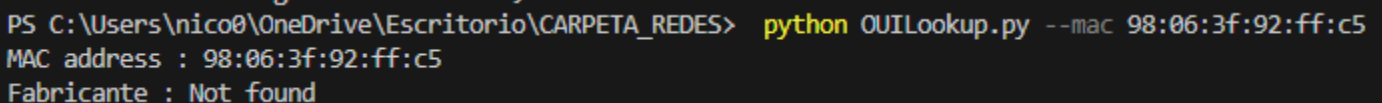
La quinta consulta entregada fue el uso de la MAC NO esté en la base de datos, donde entrega la MAC address y su Fabricante correspondiente en este caso “Not Found”:

Caso MAC que no esté en la base de datos

```
# python3 OUILookup.py --mac 98:06:3f:92:ff:c5  
MAC address : 98:06:3f:92:ff:c5  
Fabricante   : Not found
```

Figura 9. Esta es una imagen demostrando el ejemplo de implementación con los parámetros entregados.

A continuación se observará la respuesta generada a través de la implementación de nuestro código, cabe destacar que se utilizó el sistema operativo de Windows:



```
PS C:\Users\nico0\OneDrive\Escritorio\CARPETA_REDES> python OUILookup.py --mac 98:06:3f:92:ff:c5  
MAC address : 98:06:3f:92:ff:c5  
Fabricante : Not found
```

Figura 10. Esta es una imagen demostrando la respuesta de la consulta entregada.

3.6 Caso fabricantes de las MAC disponibles en la tabla ARP

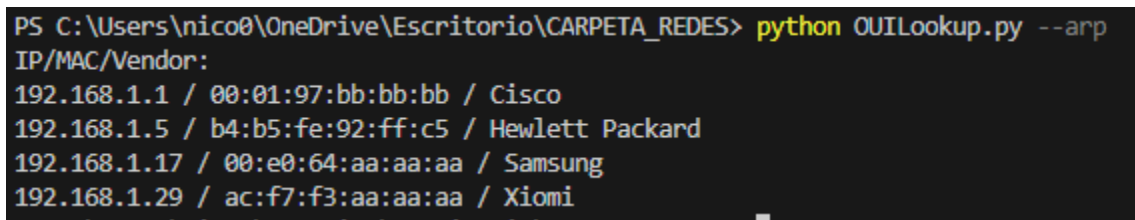
La sexta consulta entregada fue el uso de la MAC disponibles en la tabla arp, en este caso las guardadas en la base de datos, donde entrega todas su IP - MAC- Vendedor :

Caso fabricantes de las MAC disponibles en la tabla arp

```
# python3 OUILookup.py --arp
IP/MAC/Vendor:
192.168.1.1 / 00:01:97:bb:bb:bb / cisco
192.168.1.5 / b4:b5:fe:92:ff:c5 / Hewlett Packard
192.168.1.17 / 00:E0:64:aa:aa:aa / Samsung
192.168.1.29 / AC:F7:F3:aa:aa:aa / Xiaomi
```

Figura 11. Esta es una imagen demostrando el ejemplo de implementación con los parámetros entregados.

A continuación se observará la respuesta generada a través de la implementación de nuestro código, cabe destacar que se utilizó el sistema operativo de Windows:



```
PS C:\Users\nico0\OneDrive\Escritorio\CARPETA_REDES> python OUILookup.py --arp
IP/MAC/Vendor:
192.168.1.1 / 00:01:97:bb:bb:bb / Cisco
192.168.1.5 / b4:b5:fe:92:ff:c5 / Hewlett Packard
192.168.1.17 / 00:e0:64:aa:aa:aa / Samsung
192.168.1.29 / ac:f7:f3:aa:aa:aa / Xiaomi
```

Figura 12. Esta es una imagen demostrando la respuesta de la consulta entregada.

4 Diagrama de flujo

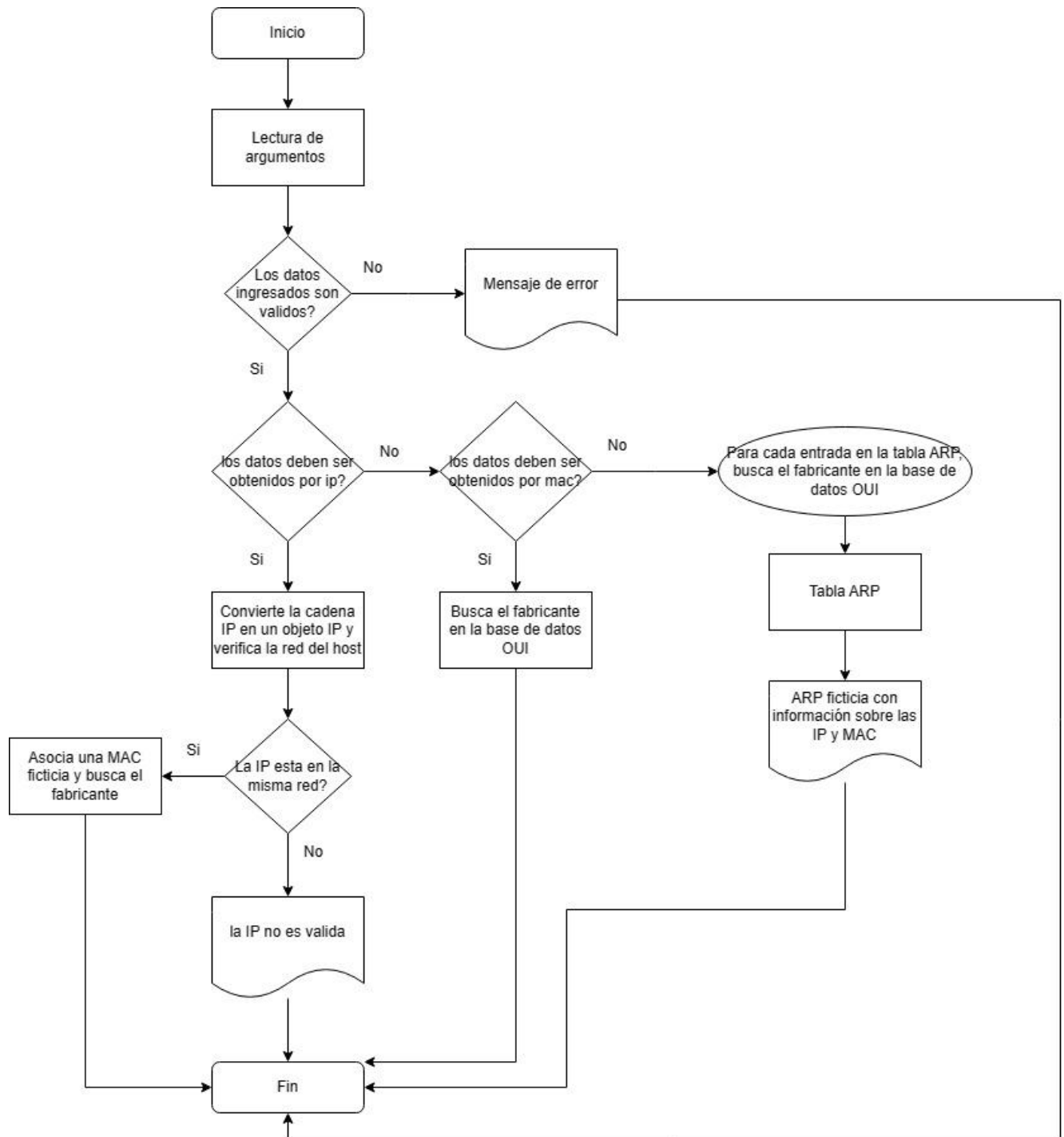


Figura 13. Este modelo se realizó a través de los ejercicios planteados anteriormente, donde se determinó a través del código implementado.

5. Discusión y conclusiones

A modo de conclusión, en este informe se puede comprobar la utilidad de la funcionalidad OUI Lookup, la cual facilita el análisis de datos referentes a las redes de computadoras. En este caso en particular se le dió la tarea de consultar fabricantes con direcciones ip y mac así como utilizar una tabla ARP ficticia para consultar las direcciones MAC existentes.

Para este dato en particular se utilizaron datos ficticios ya que todo esto es en forma de demostración, pero estamos al tanto de que para un ejemplo real se deberían utilizar datos actualizados.

De esta manera, considerando todo lo antes mencionado podemos concluir que la herramienta OUILookUp ha funcionado con total éxito, lo que se traduce en que cumple con los requisitos empleados en su totalidad.

6. Referencias

[1] <https://rico-schmidt.name/pymotw-3/getopt/>

[2] https://documentation.solarwinds.com/en/success_center/ipam/content/ipam-gsg-import-ip-addresses.htm

[3] <https://miro.com/es/diagrama-de-flujo/que-es-diagrama-de-flujo/>