

Redes de Computadores, 2023 Tarea 2: Implementación de un OUI Lookup Tool.

1. Introducción

En el presente trabajo se desarrollará una aplicación llamada OUILookup, la cual será desarrollada en el lenguaje de programación Python, donde se implementará una herramienta la cual tendrá la finalidad de consultar el fabricante de una tarjeta de red dada su dirección MAC o su IP.

El objetivo de este informe es detallar de manera clara como fue desarrollada la herramienta basada en líneas de comandos, la cual fue desarrollada en Python, además de la funcionalidad de esta y su respectivo diagrama de flujo. La herramienta será desarrollada en Windows por ende no se asegura su correcta funcionalidad en Mac Os o en Linux.

2. Materiales y Métodos

2.1. Materiales.

Los materiales que utilizaremos para realizar este trabajo serán un computador con conexión a internet y las herramientas utilizadas son el editor de texto visual studio code en donde programaremos con el lenguaje Python y las librerías a utilizar en este trabajo son **getopt** para procesar argumentos de línea de comandos, **sys**, **get_mac_address** y **subprocess**, **el cmd** y **el draw.io** para realizar el diagrama de flujo.

2.2. Metodos.

2.2.1. Diagrama de flujo.

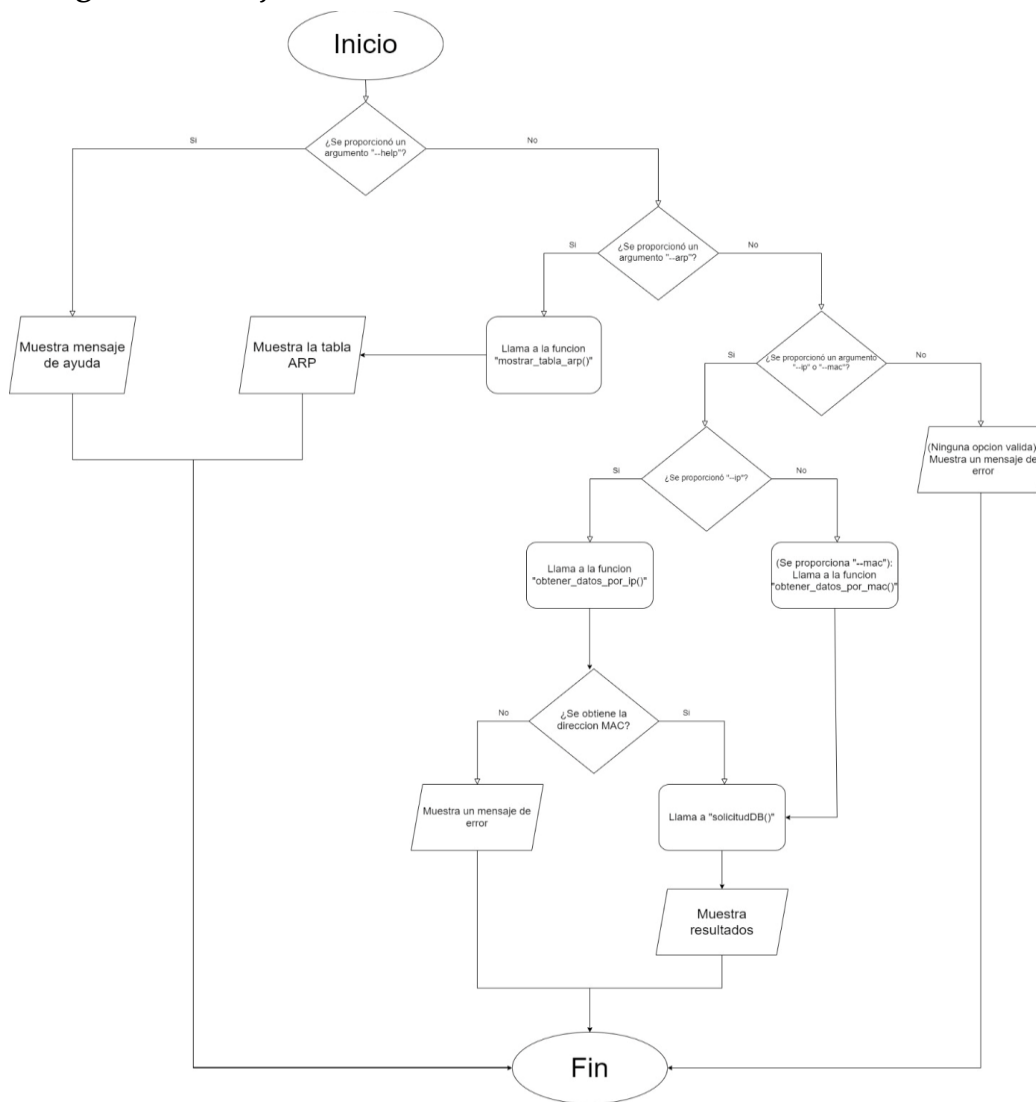


Diagrama de flujo 1

1. Inicio del programa: El programa comienza su ejecución.
2. ¿Se proporciona --help?: En este paso, el programa verifica si se proporcionó el argumento --help al ejecutar el programa.
 - Sí: Si se proporcionó --help, el programa muestra una ayuda que describe cómo usar el programa y luego termina la ejecución.
 - No: Si no se proporcionó --help, el programa continúa a la siguiente verificación.
3. ¿Se proporciona --arp?: El programa verifica si se proporcionó el argumento --arp al ejecutar el programa.
 - Sí: Si se proporcionó --arp, el programa llama a la función mostrar_tabla_arp() que muestra la tabla ARP (una lista de direcciones IP y MAC en la red) y luego termina.

- No: Si no se proporcionó --arp, el programa continúa a la siguiente verificación.
4. ¿Se proporciona --ip o --mac?: El programa verifica si se proporcionó alguno de los argumentos --ip o --mac.
 - Sí: Si se proporcionó al menos uno de los argumentos, el programa continúa a la siguiente verificación.
 - No: Si no se proporcionó ninguno de los argumentos válidos, el programa muestra un mensaje de error y termina.
 5. ¿Se proporciona --ip?: El programa verifica si se proporcionó el argumento --ip.
 - Sí: Si se proporcionó --ip, el programa llama a la función obtener_datos_por_ip() para buscar información basada en la dirección IP.
 - ¿Se obtiene la dirección MAC?: La función obtener_datos_por_ip() intenta obtener la dirección MAC relacionada con la dirección IP.
 - Sí: Si se obtiene la dirección MAC, el programa llama a solicitudDB() para buscar información adicional en la base de datos y muestra los resultados. Luego termina.
 - No: Si no se obtiene la dirección MAC, el programa muestra un mensaje de error y termina.
 - No (se proporciona --mac en su lugar): Si se proporcionó --mac en lugar de --ip, el programa llama a la función obtener_datos_por_mac() para buscar información basada en la dirección MAC.
 - El programa llama a solicitudDB() para buscar información adicional en la base de datos y muestra los resultados. Luego termina.
 - 6. Fin del programa: El programa alcanza este punto y termina su ejecución.

2.2.2. Código.

```
#imports necesarios para el funcionamiento del codigo
import getopt
import sys
from getmac import get_mac_address
import subprocess
```

Figura 1

En la Figura 1 se muestra las librerías utilizadas en nuestra herramienta. La primera que utilizaremos es “getopt” la cual tiene como funcionalidad detener el procesamiento de opciones tan pronto como se encuentra un argumento no-opcional. La segunda que utilizaremos es “sys” esta es un módulo incorporado en Python la cual nos proporciona acceso a algunas variables y funciones que interactúan con el intérprete de Python y con nuestro sistema operativo. El tercer import a utilizar es “get_mac_address” en la cual estamos llamando al módulo “getmac”, cabe destacar que esta no es una biblioteca estándar de Python, en nuestro código la utilizaremos para obtener la dirección MAC de una interfaz de red específica en nuestro sistema. La última función que utilizaremos es “subprocess”, esta nos permite invocar procesos desde Python y comunicarlos entre ellos.

```
7  OUI_DATABASE_FILE = 'manuf.txt'
8
9
10 def obtener_datos_por_ip(ip, database):
11     try:
12         mac = get_mac_address(ip=ip)
13         if mac:
14             return mac, solicitudDB(mac)
15         else:
16             return None, "Error: No se pudo obtener la dirección MAC para la IP: " + ip
17     except Exception as e:
18         return None, str(e)
```

Figura 2

Tal como se nos pidió y como muestra en la Figura 2 nuestra base de datos “manu.txt” está definida como una variable global. La función “obtener_datos_por_ip” tiene la finalidad de obtener la dirección MAC correspondiente a la dirección ip dada, luego utilizaremos la dirección MAC para realizar una solicitud a nuestra base de datos, si la operación se realiza correctamente hace return a la dirección MAC y muestra el resultado de la base de datos, en el caso contrario muestra un mensaje de error.

```
def obtener_datos_por_mac(mac, database):
    return mac, solicitudDB(mac)
```

Figura 3

En la siguiente función de la Figura 3 se utiliza para obtener y relacionar información con una dirección MAC y se encarga de coordinar la llamada a la función solicitudDB para obtener los datos de la base de datos y devolverlos junto con la dirección MAC original.

```
def mostrar_tabla_arp():  
    try:  
        arp_table = subprocess.check_output(["arp", "-a"], universal_newlines=True)  
        print(arp_table)  
    except subprocess.CalledProcessError:  
        print("No se pudo obtener la tabla ARP.")  
    except FileNotFoundError:  
        print("El comando 'arp' no está disponible en tu sistema.")
```

Figura 4

La función mostrada en la Figura 4 tiene la finalidad de obtener y mostrar la tabla ARP de nuestro sistema utilizando el comando ARP, si se ejecuta correctamente muestra la tabla en la consola en el caso contrario si no se ejecuta el comando muestra un mensaje de error.

```
def solicitudDB(mac):  
    mac = mac[:8]  
    with open(OUI_DATABASE_FILE, "r", encoding="utf-8") as f:  
        for line in f:  
            if line.startswith(mac):  
                parts = line.strip().split()  
                if len(parts) > 1:  
                    mac = parts[0]  
                    vendor = ' '.join(parts[1:])  
                    return vendor  
    return "No se encontró información en la base de datos."
```

Figura 5

La función utilizada en la Figura 5 tiene la finalidad de buscar información sobre el fabricante en la base de datos OUI a partir de los ocho primeros caracteres de una dirección Mac. Si se encuentra la información devuelve el nombre del fabricante, sino muestra un mensaje de error.

```
def main(argv):  
    ip = None  
    mac = None  
    show_arp = False  
  
    try:  
        opts, args = getopt.getopt(argv, "hi:m:", ["ip=", "mac=", "arp", "help"])  
  
    except getopt.GetoptError:  
        print("Uso: python OUILookup.py --ip <IP> --mac <MAC> --arp [--help]")  
        sys.exit(2)  
  
    for opt, arg in opts: #ciclo for que muestra las opciones help  
        if opt in ("-h", "--help"):  
            print("Uso: python OUILookup.py --ip <IP> --mac <MAC> --arp [--help]")  
            print("--ip: IP del host a consultar.")  
            print("--mac: MAC address a consultar.")  
            print("--arp: Muestra la tabla ARP.")  
            sys.exit()  
  
        if opt in ("-i", "--ip"):  
            ip = arg  
  
        if opt in ("-m", "--mac"):  
            mac = arg  
  
        if opt == "--arp":  
            show_arp = True
```

Figura 6

En la Figura 6 se inicializan las variables "ip" y "mac" como "none" para posteriormente almacenarlas direcciones ip y mac que se proporcionaran como argumentos además de la variable showarp se inicializa como false para posteriormente determinar si se debe mostrar la tabla arp o no.

Se analizan los argumentos de la línea de comando utilizando getopt.getopt("--ip "--mac" "--arp" "--help")

Se itera con un ciclo for para encontrar una opción que corresponda.

Si es -help muestra que significa cada opción

Si es -ip se asigna el valor del argumento dado a la variable ip llamando a la función obtener_datos_por_ip()

Si es -mac se asigna el valor del argumento dado a la variable mac llamando a la función obtener_datos_por_mac()

Si es -arp muestra la tabla arp y la variable show_arp cambia a true llamando a la función mostrar_tabla_arp()

```
if show_arp:
    mostrar_tabla_arp()
elif ip:
    mac, vendor = obtener_datos_por_ip(ip, OUI_DATABASE_FILE)
    if mac:
        print(f"IP address: {ip}")
        print(f"MAC address: {mac}")
        print(f"Fabricante: {vendor}")
    else:
        print(f"Error: {vendor}")
elif mac:
    mac, vendor = obtener_datos_por_mac(mac, OUI_DATABASE_FILE)
    print(f"MAC address: {mac}")
    print(f"Fabricante: {vendor}")
else:
    print("Debe proporcionar una opción válida (--ip, --mac, --arp, o --help.")

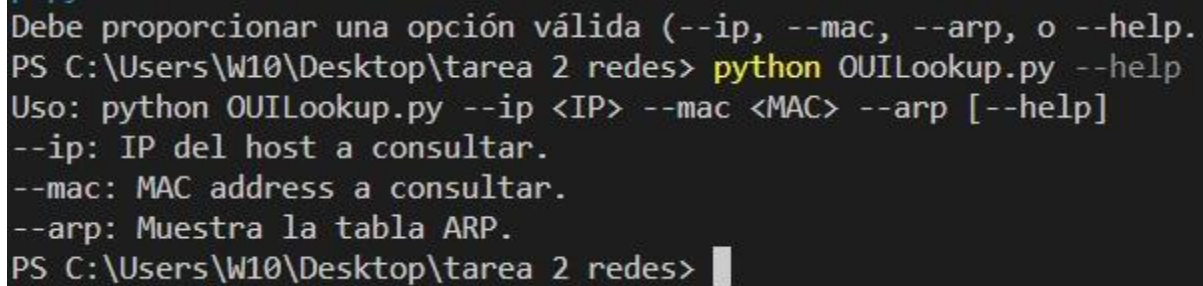
if __name__ == "__main__":
    main(sys.argv[1:])
```

Figura 7

En la Figura 7 está lo que imprimirá al haber escogido una de las opciones disponible (--ip --mac --arp --help)

3. Resultados

3.1. Opción Help



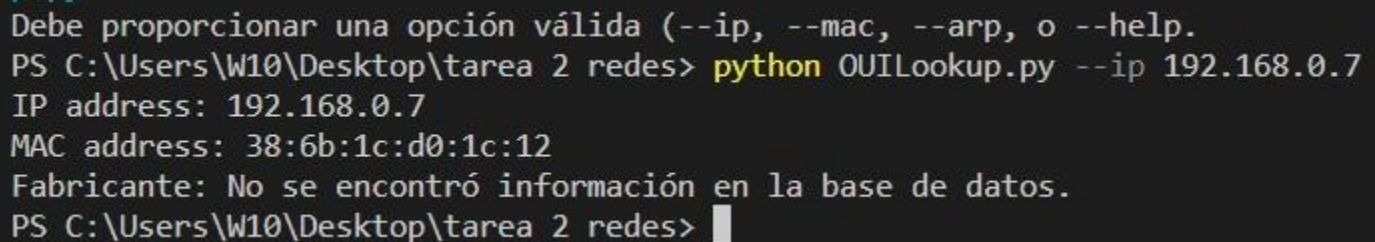
```
Debe proporcionar una opción válida (--ip, --mac, --arp, o --help.  
PS C:\Users\W10\Desktop\tarea 2 redes> python OUILookup.py --help  
Uso: python OUILookup.py --ip <IP> --mac <MAC> --arp [--help]  
--ip: IP del host a consultar.  
--mac: MAC address a consultar.  
--arp: Muestra la tabla ARP.  
PS C:\Users\W10\Desktop\tarea 2 redes> █
```

Figura 8

En la figura 8 muestra cuando se elige la opción help, esto hacer que se imprima lo que significa cada opción disponible

3.2. Caso Ip.

Ejemplo de la ip: 192.168.0.7



```
Debe proporcionar una opción válida (--ip, --mac, --arp, o --help.  
PS C:\Users\W10\Desktop\tarea 2 redes> python OUILookup.py --ip 192.168.0.7  
IP address: 192.168.0.7  
MAC address: 38:6b:1c:d0:1c:12  
Fabricante: No se encontró información en la base de datos.  
PS C:\Users\W10\Desktop\tarea 2 redes> █
```

Figura 9

En la figura 9 se ha ingresado la opción “--ip” junto con una ip(que está dentro de la red de área local) mostrando así la dirección ip, la mac correspondiente y el fabricante

3.3. Caso MAC.

Ejemplo de MAC: FC:A8:41

```
Debe proporcionar una opción válida (--ip, --mac, --arp, o --help.  
PS C:\Users\W10\Desktop\tarea 2 redes> python OUILookup.py --mac FC:A8:41  
MAC address: FC:A8:41  
Fabricante: Avaya # Avaya, Inc  
PS C:\Users\W10\Desktop\tarea 2 redes>
```

Figura 10

En la figura 10 se ha ingresado la opción “--mac” junto con una mac (que esta en el archivo manuf.txt), esta opción muestra la dirección mac y su respectivo fabricante

3.4. Caso fabricantes de las MAC disponibles en la tabla arp.

```
Debe proporcionar una opción válida (--ip, --mac, --arp, o --help.  
PS C:\Users\W10\Desktop\tarea 2 redes> python OUILookup.py --arp  
  
Interfaz: 26.186.45.127 --- 0x7  
Dirección de Internet      Dirección física      Tipo  
26.0.0.1                   02-00-00-00-51-00     din mico  
26.255.255.255             ff-ff-ff-ff-ff-ff     est tico  
224.0.0.22                  01-00-5e-00-00-16     est tico  
224.0.0.250                 01-00-5e-00-00-fa     est tico
```

Figura 11

En la figura 11 se ha ingresado la opción “--arp” esto muestra la tabla arp del computador

4. Discusión y conclusiones.

En este trabajo podemos concluir que el código entrega una utilidad de línea de comandos la cual ayuda a los usuarios a obtener información respecto a las tarjetas de red utilizadas en base a las direcciones IP y MAC, además de permitir una

visualización de la tabla ARP. La cual nos facilita información respecto a la identificación de fabricantes de estas tarjetas basada en sus direcciones MAC.

5. Referencias.

<https://docs.python.org>

<https://www.youtube.com/watch?v=5RQY5Co3kbQ>

<https://www.youtube.com/watch?v=DmHBBBrLealc>

