Redes de Computadores, 2020
Informe de Tarea 1: Implementación de un OUI Lookup Tool.

Escuela de Ingeniería Civil Informática
Universidad de Valparaíso

Autor: *Marco Vivar De La Cruz*

1. Recursos utilizados:

- Lenguaje(s) de programación utilizado(s):
    a) Python 3.8.5
- Sistema(s) operativo(s) usado(s):
    a) Windows 10 Home Single Language
- Librería(s):
    a) getopt
    b) sys
    c) subprocess
    d) requests
    e) getmac

Las librerías en **negrito** vienen por defecto en el lenguaje de programación ***Python 3***, debido a que son parte de las librerías estándar de este lenguaje.

2. Lógica del software:

Direcciones IP en la red local del usuario:

Para hacer esta implementación base del algoritmo requerido en la tarea, se hizo uso de la librería ***getmac*** para poder trabajar con la **IP** que pueda ingresar el usuario por consola, la cual es una librería disponible para el lenguaje de programación ***Python*** que permite obtener la dirección ***MAC*** de las interfaces de red y los hosts en la ***red local del usuario.*** Esta librería proporciona una función llamada ***get_mac_address()*** la cual puede buscar por el nombre de una interfaz de red, por el nombre de un host o por la **IP** de un host en específico para responder(en caso de que se encuentre en la red local del usuario) con la dirección **MAC** del dispositivo, todo dependiendo de que se le pase por parámetro a esta función, para este caso se le pasara por parámetro la **IP** de un host específico para verificar si está en la red del usuario y poder así entregar la dirección **MAC**.

Direcciones MAC:

Para realizar esta implementación, se consideró trabajar(para efectos prácticos) con los ***6 dígitos hexadecimales más significativos*** de la dirección **MAC** ingresada por el usuario o la dirección **MAC** obtenida mediante el ingreso de una dirección **IP**, debido a que los 6 primeros dígitos de una dirección **MAC** representan el ***OUI(Identificador único organizacional)*** el cual identifica al fabricante o en este caso al vendor. Para ello lo que se hizo fue recortar el *string* de la dirección **MAC** ingresada o resultante hasta dichos 6 dígitos más significativos y con esto se ejecuta la función para identificar si existe el fabricante o el vendor en la base de datos(un archivo o mediante el sitio web en donde se encuentra este documento) de direcciones **MAC** entregadas en la tarea por medio de este identificador.

Base de datos(archivo):

Para utilizar la base datos que contiene las direcciones **MAC** con el nombre de su fabricante, se implemento ***el uso del sitio web*** en donde se encuentre este archivo y también se realizo una ***descarga de este archivo*** en el directorio en donde se ejecuta el software, ***esto para no generar una dependencia total*** del uso del sitio web o de una conexión a internet para poder ejecutar el software.

El ***uso del sitio web*** se realizó mediante un *requests*, en el cual fue necesario utilizar la librería llamada ***requests*** disponible en el lenguaje de programación ***Python***, con esto se realizó una solicitud del tipo ***get()*** al sitio web en concreto:

https://gitlab.com/wireshark/wireshark/-/raw/master/manuf

En donde el sitio responde con el contenido del archivo de direcciones ***MAC*** y fabricantes el cual se procede a guardar en una variable local para ser usado en la búsqueda de fabricantes mediante el paso de una dirección ***MAC*** especifica.

Para el caso de la ***descarga del archivo*** que funciona como base de datos, se tiene como ***condicional*** que este debe estar en el directorio de ejecución del software para realizar la ejecución completa de este, esto debido al punto mencionado anterior mente de evitar dependencias. Para realizar esta descarga primero se evalúa si existe un archivo llamado ***"OUILookup.txt"*** en el directorio en donde se ejecuta el software, si no se detecta este archivo se procede a hacer un ***request al sitio web*** del contenido para poder guardarlo en una variable y reescribir dicho contenido en un archivo nuevo creado, llamado ***"OUILookup.txt"***.

Con esto realizado se permite el uso normal del software en donde utiliza principalmente el sitio web para hacer la búsqueda del fabricante o vendor, en caso de que suceda algún problema con el ***request al sitio web*** se utilizara la base de datos

guardada en el archivo ***"OUILookup.txt"*** para poder ejecutar esta búsqueda.

En los dos casos se utiliza un ciclo ***for*** para recorrer el contenido de la base de datos y comparar las direcciones **MAC** existentes en ella con la dirección **MAC** introducida u obtenida mediante una dirección **IP** ingresada por el usuario.

Funcionamiento del algoritmo:

El código comienza verificando si las librerías requeridas por el software están instaladas en el computador del usuario, si están no están se procede a utilizar la librería ***subprocess*** para ejecutar el comando ***"pip install -r requeriments.txt"*** en donde requeriments.txt es el archivo en donde se listan las librerías necesarias para utilizar el programa. Luego de esto hay que ejecutar el programa nuevamente para su uso

```
try:
    import requests
    from getmac import get_mac_address
except:
    print("Error when importing necessary libraries  to make the program work, it will proceed to start the intallation
    import subprocess
    subprocess.call(['pip', 'install', "-r","requeriments.txt"])
    print("Libraries were installed to execute the program\n")
```

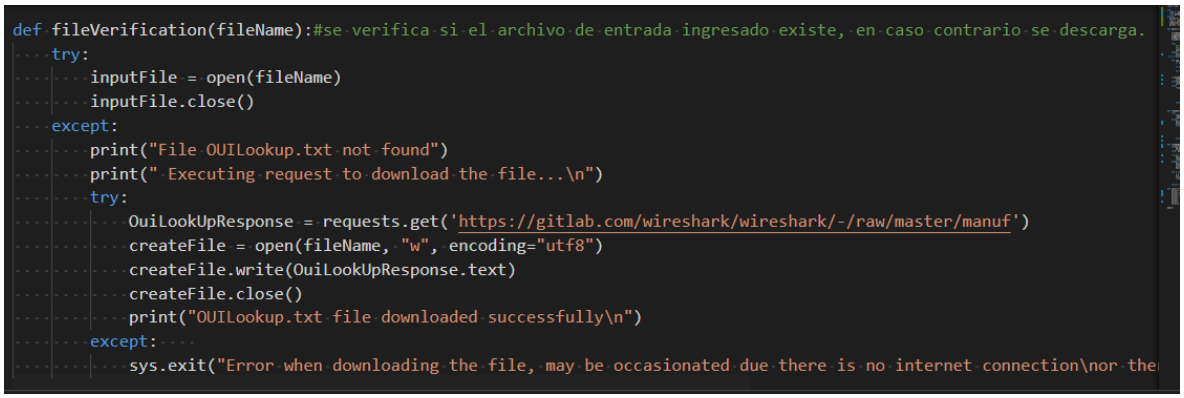
Ilustración 1.instalación de librerías

Luego se definen las variables principales a usar y se ejecuta la función fileVerification(), la cual le pasamos por parámetros el nombre de nuestro a archivo a modificar y verifica si el archivo pasado por parámetros a la función existe en el directorio de ejecución del usuario, si este archivo no existe se la hace un request al sitio web que tiene el contenido dela archivo para poder guardarlo en una variable y después crear un archivo con el contenido que se requiere para continuar el programa.

```
def main():#Funcion main en donde se iniciara toda la logica del codigo

    fileName = "OUILookup.txt"
    argIpInput = None
    argMacInput = None
    fileVerification(fileName)
```

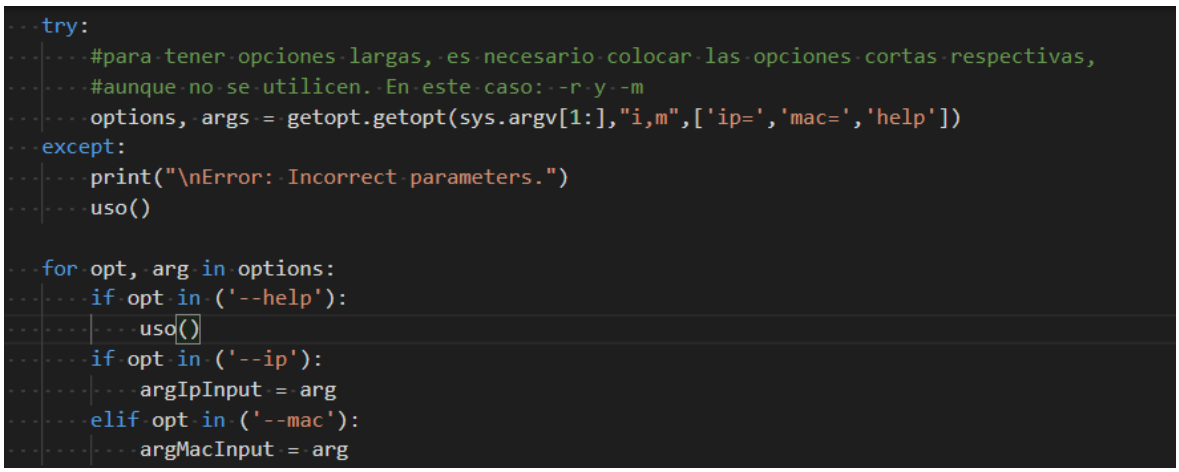
Ilustración 2.definición de variables y verificación del archivo.



```
def fileVerification(fileName):#se verifica si el archivo de entrada ingresado existe, en caso contrario se descarga.
    try:
        inputFile = open(fileName)
        inputFile.close()
    except:
        print("File OUILookup.txt not found")
        print(" Executing request to download the file...\n")
        try:
            OuiLookUpResponse = requests.get('https://gitlab.com/wireshark/wireshark/-/raw/master/manuf')
            createFile = open(fileName, "w", encoding="utf8")
            createFile.write(OuiLookUpResponse.text)
            createFile.close()
            print("OUILookup.txt file downloaded successfully\n")
        except:
            sys.exit("Error when downloading the file, may be occasionated due there is no internet connection\nor the
```

Ilustración 3.función para verificar sin el archivo existe en el directorio del usuario

Después de esto se verifican si los parámetros de entrada son correctos y no se ingresan parámetros que no estén definidos en el software, si esto ocurre, se hace uso de la función ***uso()*** para mostrarle al usuario por pantalla los comandos y como debería ser la sintaxis de ingresos de estos por consola. Posterior a esto se crea un ciclo for para poder recorrer los parámetros y los argumentos ingresados, guardando los argumentos de los parámetros "***--ip***" y "***—mac***" en variables. Hay que destacar que si se detecta el parámetro "***—help***" se llamará a la función ***uso()*** nuevamente y se pondrá termino a la ejecución del programa



```
try:
    #para tener opciones largas, es necesario colocar las opciones cortas respectivas,
    #aunque no se utilicen. En este caso: -r y -m
    options, args = getopt.getopt(sys.argv[1:],"i,m",['ip=','mac=','help'])
except:
    print("\nError: Incorrect parameters.")
    uso()

for opt, arg in options:
    if opt in ('--help'):
        uso()
    if opt in ('--ip'):
        argIpInput = arg
    elif opt in ('--mac'):
        argMacInput = arg
```

Ilustración 4.verificación de los parámetros de entrada ingresados por el usuario.

```
def uso():#Funcion para usar con el comando --help y mostrar como funcionan los parametros
    print("\nUse in Windowns: OUILookup.py" + " --ip <ip address> | --mac <mac address> [--help] ")
    print("\nUse in Linux/Mac: ./OUILookup.py" + " --ip <ip address> | --mac <mac address> [--help] ")
    print("\nParameters:")
    print("      --ip: specify the IP of the host to query(ej: OUILookup.py --ip 192.168.0.6)")
    print("      --mac: specify the MAC address to query(ej: OUILookup.py --mac f8:28:19:46:2f:b9)")
    print("      --help: Display these instructions and finish")
    exit(1)
```

Ilustración 5.función use(), la cual muestra las instrucciones de uso de los parámetros existentes del programa.

Luego si se ingresaron parámetros son válidos, se procede a hacer verificaciones de si se ingresaron dos parámetros validos***("--ip", "--mac")*** a la vez o no se ingreso ninguno, si es que los parámetros no cumplen las condiciones anteriores se procede a diferenciar si la entrada es una dirección ***IP*** o una ***MAC***, ya que dependiendo de qué caso(y debido a que solo se puede ingresar una a la vez) se hará una acción extra para obtener la dirección ***MAC***(caso en el que se ingrese una ***IP***).

Independiente del caso que ocurra, se hará uso de la función ***findByMac(),*** a la cual se le pasa por parámetro la dirección ***MAC*** obtenida o ingresada y realiza el proceso de comparar la ***MAC*** ingresada(*primeros 6 dígitos hexadecimales*) con las ***MACS*** existentes en la base de datos(en donde está la opción de utilizar el contenido tanto del sitio web, como del archivo ***OUILookup.txt***) esto mediante un ciclo for que recorre cada línea de la base de datos y separa o elimina sus caracteres especiales, como lo son sus salto de línea ***("\n")*** o sus tabulaciones ***("\t")***, para llegar a un arreglo que en su posición 0 contengan solo la dirección ***MAC*** de la línea y en la posición 2 el nombre del fabricante, terminando así el proceso del software.

```
if(argIpInput and argMacInput):
    print(f"Please, enter only one IP or MAC parameter")
elif(argIpInput == None and argMacInput == None):
    uso()
else:
    if(argIpInput):
        mac_address_output = get_mac_address(ip = argIpInput)
        if(mac_address_output):
            vendorName = findByMac(mac_address_output)
            print(f"MAC address : {mac_address_output}\nVendor : {vendorName}")#BUSCAN EL VENDOR EN ARCHIVO MAC
        else:
            print(f"Error: ip({argIpInput}) is outside the host network")
    elif(argMacInput):
        vendorName = findByMac(argMacInput)
        print(f"MAC address : {argMacInput}\nVendor : {vendorName}")
```

Ilustración 6.verificación de los parámetros validos de entrada y ejecución de la búsqueda de una dirección mac para los dos casos existentes.

```
def findByMac(macAddress):#Funcion para buscar una macAddress en el archivo o en el sitio web(o base de datos) de direcciones mac
    if(len(macAddress) > 8):#Se ve si la longitud de la mac es mayor a 8 debido a que tendria mas de 6 digitos(contando los separadores ":")
        macAddressNetId = (macAddress[:8]).upper()
    else: macAddressNetId = (macAddress).upper()

    try:#se hace un request para poder obtener el contenido mas actualizado de las direcciones mac
        OuiLookUpResponse = requests.get('https://gitlab.com/wireshark/wireshark/-/raw/master/manuf')
        contenedor = OuiLookUpResponse.text.split("\n")
    except:#en caso de fallar el request se utiliza el archivo OUILookup.txt para realizar la busqueda del fabricante
        OuiLookUpFile = open(fileName , "r", encoding="utf8")
        contenedor = OuiLookUpFile.readlines()
        OuiLookUpFile.close()

    for i in contenedor:
        i = i.strip("\n")
        aux_1 = i.split('\t')
        if(str(aux_1[0]) == str(macAddressNetId)):
            return aux_1[2]

    return "Not found"
```

Ilustración 7.Función para buscar el fabricante mediante una dirección mac de entrada.

3. Diagrama de flujo del software:

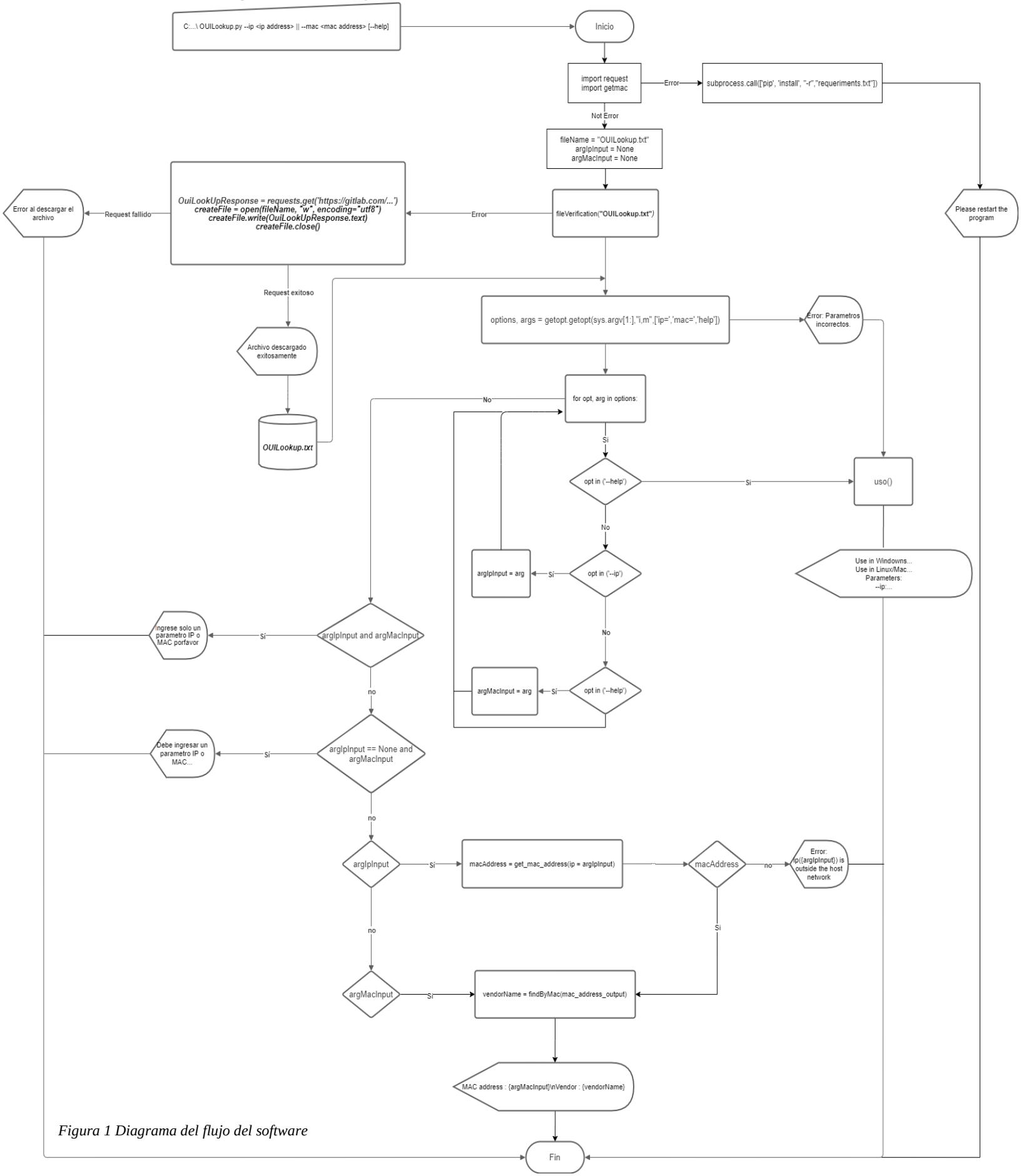


Figura 1 Diagrama del flujo del software