

## *Informe Técnico*

### *Tarea 1: Implementación de un OUI Lookup*

*Redes de Computadores I -Universidad de Valparaíso*

*Marcelo Estay - Juan Larenas*

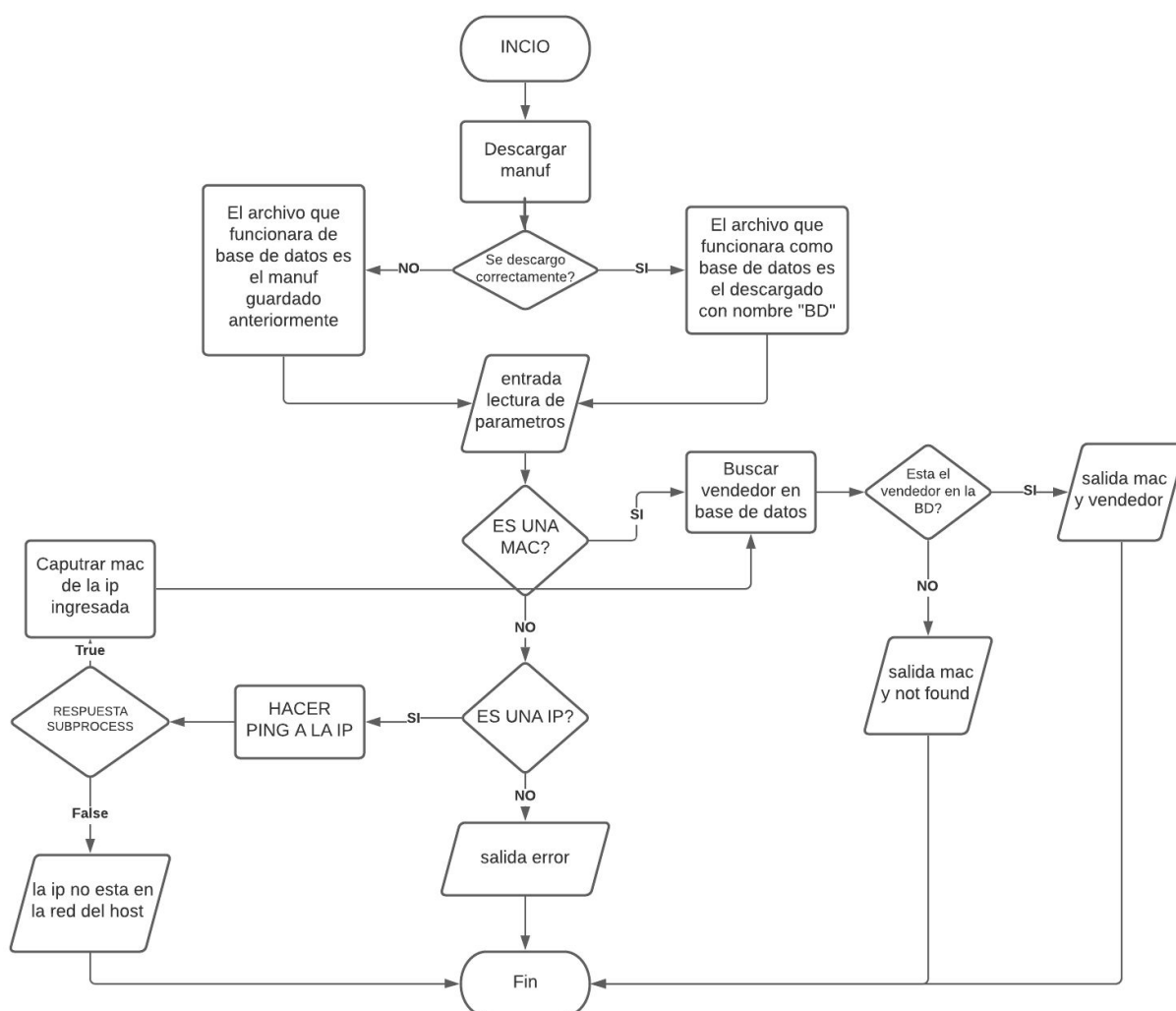
#### ***I. Introducción***

El objetivo de la **Tarea 1** es implementar una herramienta, basada en línea de comandos para consultar el fabricante de una tarjeta de red por medio de la dirección MAC o su IP. Este software se realizará utilizando el lenguaje de programación **Python**, dentro del Sistema Operativo **Windows**, siendo finalmente cargado a un repositorio **GitHub**.

## II. Descripción funcional.

### Narrativa del procesamiento.

El proyecto consiste en poder buscar en una base datos, los vendedores de la MAC a consultar, siendo esta ingresada de forma directa, o obtenida mediante la IP. Entendiendo que no todas las MAC existentes están registradas en dicha BD. Por medio del siguiente diagrama de flujo se expone la ejecución del programa.



El programa, en su inicio hace una conexión con la pagina web <https://gitlab.com/wireshark/wireshark/-/raw/master/manuf> para poder importar la base de datos actualizada donde se encuentra registrado los fabricante de las MACs. Si esta se rescata exitosamente, se almacena la información en un archivo llamado “**BD.txt**” en caso contrario se utilizará el archivo “**manuf.txt**” el cual es un respaldo offline de la BD.

Luego puede recibir como parámetros **--ip** o **--mac** realizará las tareas programada en cada uno de los casos que a continuación se detallaran.

## **Caso 1: Recibe --ip por parámetro y una IP por argumento**

En este caso, en los parámetros ingresados por consola, al momento de iniciar el programa, el usuario va a ingresar una IP. El software, hace un ping a esa **IP**, mediante el módulo subprocess, específicamente con la clase **run()**. Para fines del software, no es necesario mostrar todo el proceso de ejecución de dicho comando, por ende se le entrega **PIPE** a **stdout**, para capturar todo el proceso y no enviarlo a la consola. Se ejecuta el comando con **check = True** para poder detectar algún error en este ping, de esta manera, logramos determinar si la IP ingresada está (**check = True**) o no (**check = False**) en la red.

En el caso que la IP no se encuentre en la red, se enviará el error a la consola. Por lo contrario, si la IP se encuentra en la red, se ejecutará el módulo **getmac**, entregando como parámetro el **host (IP)**, con el fin de obtener la mac de este. Después de obtener la información con respecto a la mac, esta es analizada en una base de datos (manuf o BD), con el fin de encontrar su vendedor, en el caso, que el vendedor no se encuentre, se enviar la mac y el vendedor se entrega como **not found**, en caso contrario, se entrega la mac y vendedor.

## **Caso 2: Recibe --mac por parámetro y una Mac por argumento**

En este caso se recibirá una MAC como parámetro y una MAC por argumento, la cual será buscada en la base de datos (manuf o BD), que corresponde a la base de datos, donde están todas las mac y sus respectivos vendedores, si esta está dentro de esa base de datos, retornará su vendedor, pero en caso contrario, dirá que el vendedor no fue encontrado.

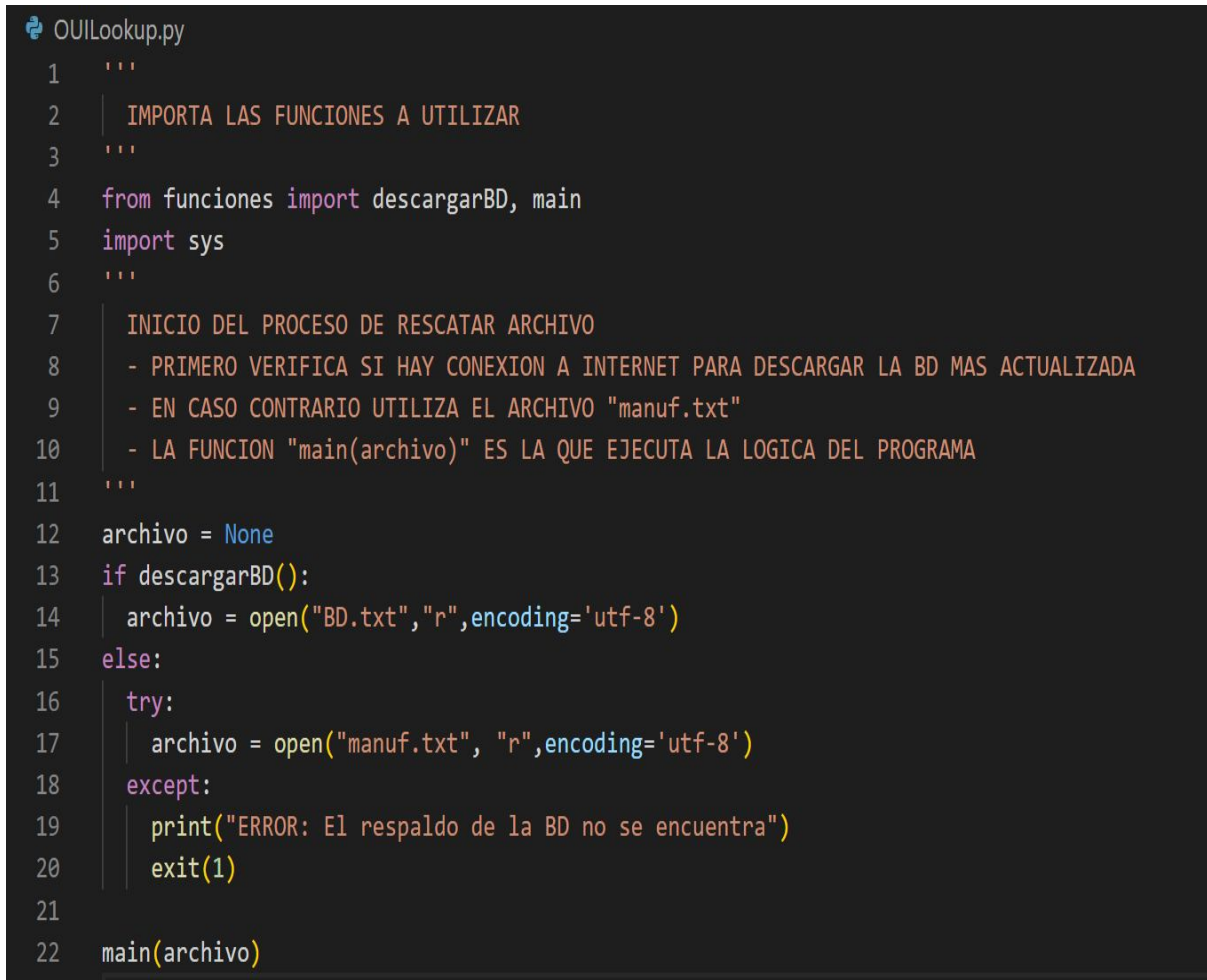
### ***III. Librerías a utilizar***

<b><i>Nombre</i></b>	<b><i>Función</i></b>	<b><i>Instalación</i></b>
requests	Facilita el trabajo con peticiones HTTP para extraer información de una página o enviar el contenido de un formulario de manera automatizada.	pip install requests
subprocess	Permite invocar procesos desde Python y comunicarse con ellos: enviar datos a la entrada (stdin) y recibir la información de salida (stdout).	pip install subprocess.run
subprocess import PIPE	Permite capturar el valor de entrada o de salida para ser utilizado como dato según lo especificado en <b>Caso 1</b>	al instalarse el anterior se instala este módulo
getmac import get_mac_address	Permite obtener las direcciones MAC de hosts remotos e interfaces locales	pip install getmac
getopt	El módulo getopt es el analizador de opciones de línea de comando original que soporta las convenciones establecidas por la función getopt de Unix. Analiza una secuencia de argumentos, como sys.argv y devuelve una secuencia de tuplas que contienen pares (opción, argumento) y una secuencia de argumentos no opcionales.	
os	Permite acceder a funcionalidades dependientes del Sistema Operativo.	
sys	Este módulo proporciona acceso a variables utilizadas o mantenidas por el intérprete y a funciones que interactúan estrechamente con el intérprete.	

**Nota:** solo los que tienen información en la columna de instalación son estrictamente necesarios que se instalen para la correcta ejecución del programa.

#### IV. Descripción del Programa

a continuación se procede a explicar el código.



```
OUILookup.py
1  '''
2  |  IMPORTA LAS FUNCIONES A UTILIZAR
3  |  '''
4  from funciones import descargarBD, main
5  import sys
6  |  '''
7  |  INICIO DEL PROCESO DE RESCATAR ARCHIVO
8  |  - PRIMERO VERIFICA SI HAY CONEXION A INTERNET PARA DESCARGAR LA BD MAS ACTUALIZADA
9  |  - EN CASO CONTRARIO UTILIZA EL ARCHIVO "manuf.txt"
10 |  - LA FUNCION "main(archivo)" ES LA QUE EJECUTA LA LOGICA DEL PROGRAMA
11 |  '''
12 archivo = None
13 if descargarBD():
14 |     archivo = open("BD.txt", "r", encoding='utf-8')
15 else:
16 |     try:
17 |         archivo = open("manuf.txt", "r", encoding='utf-8')
18 |     except:
19 |         print("ERROR: El respaldo de la BD no se encuentra")
20 |         exit(1)
21
22 main(archivo)
```

**Imagen 1.** Código de OUILookup.py

La **Imagen 1** hace referencia al código que une toda la lógica del programa, en donde se importan las funciones a utilizar y hace la apertura de la base de datos por medio de una condicional el cual verifica primeramente la conexión a internet para descargar la BD más actualizada y en caso contrario utiliza la de respaldo, si no encuentra este último lanza un mensaje de error y cierra el programa. Si encuentra la BD ejecuta la función main.

```

funciones.py > parametro
1  ''' SE IMPORTAN LAS LIBRERIAS NECESARIAS
2  PARA LA EJECUCION DE LAS FUNCIONES'''
3  import requests
4  import subprocess
5  import os
6  from subprocess import PIPE
7  from getmac import get_mac_address
8  import getopt
9  import sys
10
11  def obtener_MAC(host):
12      '''RETORNA MAC DEPENDIENDO DE LA IP QUE SE INGRESE'''
13      return get_mac_address(ip=host)
14
15  def descargarBD():
16      '''
17      DESCARGA LA BD DE LA PAGINA WEB: https://gitlab.com/wireshark/wireshark/-/raw/master/manuf
18      '''
19      try:
20          url = 'https://gitlab.com/wireshark/wireshark/-/raw/master/manuf'
21          myfile = requests.get(url)
22          archivo = open('BD.txt', 'wb')
23          archivo.write(myfile.content)
24          return True
25      except:
26          print("Sin conexión a internet para descargar BD")
27          print("Se usará la de Respaldo\n")
28          return False

```

**Imagen 2.** Código de funciones.py parte 1.

La **Imagen 2** hace referencia al inicio del código de funciones en donde se programan las acciones necesarias para lograr el objetivo del programa. Se importan las librerías necesarias, se define la función **obtener\_mac(host)** que recibe por parámetro la **IP** para luego retornar la MAC por medio de la metodo **get\_mac\_address(ip=host)**. Por último la función **descargarBD()** descarga la base de datos actualizada de la página web guardándolo en archivo de texto y retornando un True, en caso contrario imprime un mensaje de error y retorna un False.

```

30 def macBD(archivo, mac):
31     '''OBTIENE EL FABRICANTE DE LA MAC SEGUN
32     | BASE DE DATOS(MANUF.TXT O BD.TXT[DESCARGADA])
33     '''
34     companiaB = ""
35     companiaR = ""
36     macU = mac[:8]
37     linea = archivo.readline()
38     while linea:
39         nuevo = linea.split()
40         if "/" in nuevo:
41             macU = mac
42             if len(nuevo[2:]) <= 0:
43                 companiaB = "".join(nuevo[1:2])
44
45             if len(nuevo[2:]) > 0:
46                 companiaB = " ".join(nuevo[2:])
47
48             if(len(nuevo)) > 0:
49                 if macU.upper() == "".join(nuevo[0]):
50                     companiaR = companiaB
51                     break
52                 else:
53                     companiaR = "Not Found"
54
55             linea = archivo.readline()
56
57     print(f"MAC address : {mac}")
58     print(f"Vendedor      : {companiaR}\n")

```

Imagen 3. Código de funciones.py parte 2.

La **Imagen 3** sigue mostrando las demás funciones del archivo funciones.py en este caso la función **macBD** que recibe como parámetro la base de datos guardada en un archivo de extensión txt y la MAC a consultar. Por medio de un loop while se realiza la lectura de la BD por medio del método readline() el cual va guardando la información en la variable **línea** el cual en la variable **nuevo** se guarda la información en una lista separando la información para ser procesada y rescatar la mac y el vendedor.

```

61  def IpVerificadorMac(archivo, ip):
62      '''
63          DATO QUE EL MODULO SUBPROCESS NOS PERMITE OBTENER LOS DATOS ENVIADOS
64          A LA SALIDA EN CONSOLA LO OCUPAMOS PARA VERIFICAR LA SALIDA DEL COMANDO
65          PING HACIA UNA DIRECCION IP, EL CUAL ESTA INICIALIZADO EN TRUE, POR ENDE
66          SI NOS ARROJA UN VALOR DISTINTO DE ESTE SE IRA A LA SECCION EXCEPT, DE LO
67          CONTRARIO, LE PEDIMOS LA MAC QUE CORRESPONDE A ESA IP Y LA VERIFICAMOS
68          EN LA BASE DE DATOS
69      '''
70      try:
71          subprocess.run("ping " + ip, shell=True, check=True, stdout=PIPE)
72          macDef = obtener_MAC(ip)
73          macBD(archivo, macDef)
74      except:
75          print("Error: ip is outside the host network")
76

```

**Imagen 4.** Código de funciones.py parte 3.

La **Imagen 4** muestra la función **IpVerificadorMac** que recibe por parámetro la BD y la IP a verificar si este último se encuentra dentro de la red local, si se encuentra llama a la función **obtener\_mac(ip)** para recuperar la mac en una variable llama **macDef** y por último utiliza la función **macBD** para verificar si el vendor se encuentra dentro de la base de datos. En caso contrario manda un mensaje de error de que la ip no se encuentra dentro de la hist network.



```

77 v def main(archivo):
78 v     """
79     -UNION DE LAS DEMAS FUNCIONES
80     -GENERA LA OPCIONES POR INGRESO POR CONSOLA POR MEDIO DE GETOPT
81     """
82     ip = None
83     mac = None
84 v     try:
85         options, args = getopt.getopt(sys.argv[1:], "i,m", ['ip=', 'mac=', 'help'])
86 v     except:
87         print("Error: Parametros incorrectos.")
88         mensajeError()
89
90 v     if(options == []):
91         print("ERROR: No ingreso parametros")
92         mensajeError()
93
94 v     for opt, arg in options:
95 v         if opt in ('--help'):
96             mensajeHelp()
97 v         if opt in ('--ip'):
98             ip = arg
99 v         elif opt in ('--mac'):
100             mac = arg
101 v     try:
102 v         if(mac):
103             macBD(archivo, mac)
104 v         elif(ip):
105             IpVerificadorMac(archivo, ip)
106 v     except:
107         print("***ERROR EN LA EJECUCION DEL SCRIPT***")
108 v     finally:
109         archivo.close()
110
111 v def mensajeError():
112     print("Use: " + sys.argv[0] + " --ip <IP> | --mac <IP> [--help]")
113     exit(1)
114 v def mensajeHelp():
115     print("Use: " + sys.argv[0] + " --ip <IP> | --mac <IP> [--help]")
116     print("    --ip : specify the IP of the host to query.")
117     print("    --mac : specify the MAC address to query.")
118     print("    --help: show this message and quit.")
119     exit(1)
120

```

**Imagen 5.** Código de funciones.py parte 4.

La **Imagen 5** muestra las últimas funciones del archivo funciones.py. La función **main** recibe por parámetro la BD y además inicia los opciones para ingresar por consola, definiendo las variables que van a recibir la información a evaluar y por medio de un manejo de excepciones realiza la ejecución de las funciones anteriores para lograr el objetivo del programa. Por último las funciones de **mensajeError** y **mensajeHelp** envían información con respecto al uso indicado del programa.

## V. *Imágenes de pruebas de ejecución.*

```
PS C:\Users\Juan\Desktop\tRedes> py OUIlookup.py --ip 192.168.22.2
Error: ip is outside the host network
```

En el caso anterior, envía este error a la consola, habiendo obtenido una respuesta con error del comando ping, es decir la ip no está en la red.

```
PS C:\Users\Juan\Desktop\tRedes> py OUIlookup.py --mac 9e:09:47:19:b8:09
MAC address : 9e:09:47:19:b8:09
Vendedor    : Not Found
```

Al ingresar por parámetro una mac, esta es verificada en una base de datos, con el fin de obtener su vendedor, pero en el caso anterior la mac ingresada, en la base de datos no tiene vendedor, por ende se imprime por consola Not Found.

```
PS C:\Users\Juan\Desktop\tRedes> py OUIlookup.py --mac
Error: Parametros incorrectos.
Use: OUIlookup.py --ip <IP> | --mac <IP> [--help]
```

En el ingreso de parámetros se necesita, un indicador de lo que se va a buscar por ejemplo --ip, y además el valor del indicador. En el caso anterior el valor del indicador --mac, no está ingresado, solamente se sabe que se quiere buscar una mac, pero no se sabe cual es, por ende se envía un error al usuario, y además, la forma correcta de uso.

```
PS C:\Users\Juan\Desktop\tRedes> py OUIlookup.py --mac 9e:09:47:19:b8:09
Sin conexión a internet para descargar BD
Se usará la de Respaldo

MAC address : 9e:09:47:19:b8:09
Vendedor    : Not Found
```

La base de datos, llamada manuf, está en en algun lugar de la internet, por ende al momento de iniciar el software, esta debe ser descargada, en este caso no se cuenta con una conexión a internet, por ende se ocupa el manuf de respaldo, que más que nada es una advertencia al usuario de la forma en que se procederá.

```
C:\Users\maemg\Desktop\tRedes>py OUILookup.py --help
Use: OUILookup.py --ip <IP> | --mac <IP> [--help]
    --ip : specify the IP of the host to query.
    --mac : specify the MAC address to query.
    --help: show this message and quit.
```

En el caso de tener dudas de como poder ejecutar el programa, existe la opción de ingresar --help, de esta manera poder obtener indicaciones de como ejecutar el programa de manera exitosa.

```
C:\Users\maemg\Desktop\tRedes>py OUILookup.py --mac 20:0A:0D:40:00:00/28
MAC address : 20:0A:0D:40:00:00/28
Vendedor    : Virtium
```

En esta prueba se ingresa una dirección física que se encuentra registrada dentro de la base de datos por lo cual imprime su vendedor.