

Informe Técnico:

A través del siguiente informe se detallará el diseño e implementación del código creado para la 'Tarea 1', el cual se encarga de la búsqueda de fabricante de tarjeta de red, dado un archivo de texto (.txt) y una dirección MAC o, en su defecto, una dirección IP como entrada por parámetro.

Diseño de la solución:

Como antecedente hemos de decir que el programa está hecho para Windows 10 y probado con la consola de este [cmd].

Iniciando el programa, se hace utilización del módulo *getopt* [para el cual se necesitará importar las librerías '*getopt*' y '*sys*'] el cual se encarga de analizar las opciones de línea de comandos en forma de secuencia de argumentos, dado los parámetros de la función *getopt()* se crean tres argumentos largos[con prefijo '--': '*ip=*', '*mac=*' y '*help*', de forma que los primeros dos son argumentos que reciben información [índice '=']. Su creación se realiza en un bloque '*try*', de forma que, de no cumplirse este bloque, se inicialice un bloque '*except*', para así validar la entrada de los parámetros establecidos, a su vez, en caso de no ingresar ningún parámetro, al final del código, se crea una excepción más. Cada validación de parámetros señala el error ocurrido e inicializa la función '*ayuda()*', la cual está encargada de señalar al usuario las instrucciones de uso del código. Todo lo anterior se puede ver claramente ilustrado en la Figura 1:

```
#Código Principal:
#Intenta conseguir valor de los parámetros:
try:
    opts, args = getopt.getopt(sys.argv[1:], "", ['ip=', 'mac=', 'help'])
#En caso de no conseguir ningún valor por parámetro, error:
except:
    print("\n **** ERROR: Parámetros incorrectos. ****\n")
    ayuda()
#Uso de los parámetros:
for opt, arg in opts:
    if opt in ('--help'):
        ayuda()
    elif opt in ('--mac'):
        dirMAC(arg)
    elif opt in ('--ip'):
        dirIP(arg)
#Obligatoriedad de parámetros:
print("\n **** ERROR: Se requiere al menos un parámetro para funcionar. ****\n")
ayuda()
```

Figura 1

Luego de la creación y validación de parámetros, se procede a recorrer la secuencia de argumentos creada por *getopt* y crear la consecuencia del ingreso de cada uno de estos. Dado los requerimientos y las características de la tarea se tomó la decisión de realizar cada acción de los parámetros mediante funciones, por tanto, *--help* ejecuta la función *ayuda()*, la cual está encargada, como se mencionó anteriormente, de imprimir en pantalla instrucciones de uso del código, tal y como se vé en la Figura 2.

```
#Instrucciones de uso [--help]
def ayuda():
    print(" *Utilización de parámetros:\n"
          "  ./" + sys.argv[0] + " --ip <IP> || --mac <MAC> || [--help]\n"
          "  \n *Lista de Parámetros:\n"
          "    --ip <dirección IP>: Señalará, a partir de la dirección IP local, el fabricante de la tarjeta de red.\n"
          "    --mac <dirección MAC>: Señalará, a partir de la dirección MAC, el fabricante de la tarjeta de red.\n"
          "    --help: Muestra información sobre el uso del programa y parámetros. [Opcional] \n"
          "    ****Se requiere el archivo de texto 'dir.txt' en el mismo directorio de este programa.**\n")
    exit(1)
```

Figura 2

Ahora, *--mac* realiza la ejecución de la función *dirMAC()*, la que está encargada de recorrer el archivo de texto requerido *dir.txt*, y encontrar la dirección MAC igual a la de entrada por parámetros, en caso de encontrarla, señalará el fabricante de la tarjeta de red, y en caso contrario se señalará que no se encuentra un fabricante para la dirección MAC dada. Todo esto se puede apreciar en la Figura 3:

```
#Busqueda de fabricante a partir de la dirección MAC [--mac]
def dirMAC(a):
    mac = a.upper()
    #Se abre el archivo 'dir.txt' en modo lectura, con formato de codificación utf8, para sobrepasar
    #errores causados por caracteres más allá del idioma ingles predefinido por el lenguaje.
    with open('dir.txt', 'r', encoding="utf8") as texto:
        lineas = texto.readlines()
        for linea in lineas:
            linea = linea.rstrip("\n")
            #Buscar direcciones MAC con extensión
            if (linea[8] == ':'):
                if (linea[0:17] == mac):
                    datos = linea.split("\t")
                    print("\nMAC: " + datos[0])
                    print("Fabricante: " + datos[-1])
                    break
            #Buscar direcciones MAC simples
            elif (linea[0:8] == mac):
                datos = linea.split("\t")
                print("\nMAC: " + datos[0])
                print("Fabricante: " + datos[-1])
                break
        #En caso de no encontrarse la MAC
    else:
        print("\nMAC: " + mac)
        print("Fabricante: Direccion MAC no encontrada")
    exit(1)
```

Figura 3

Por último el parámetro *-ip* se encarga de ejecutar la función '*dirIP*' ésta es responsable de verificar, según la dirección IP entregada, si la dirección IP está dentro de la red de área local, en caso de no estarlo señalarlo, y caso de encontrarla, buscar la dirección MAC correspondiente a la misma, para posteriormente entregarla a la función '*dirMAC*', para que esta última pueda encontrar el fabricante de la tarjeta de red de dicha dirección MAC, o no, e informarlo. El proceso de la función '*dirIP*' se presenta en la siguiente figura:

```
def dirIP(b):
    IP = b
    ip=" "+IP
    os.system('arp -a >Arp.txt')
    with open('Arp.txt', 'r') as texto:
        lineas = texto.readlines()
        for linea in lineas:
            linea = linea.rstrip("\n")
            linea=linea.split(" ")
            if(linea[0]==ip):
                mac=linea[1]
                mac=mac[3:11]
                mac=mac.replace("-",":")
                print(mac)
                dirMAC(mac)
                break
    exit(1)
```

Figura 4

Implementación:

En la siguiente figura se muestra el diagrama de flujo de la solución:

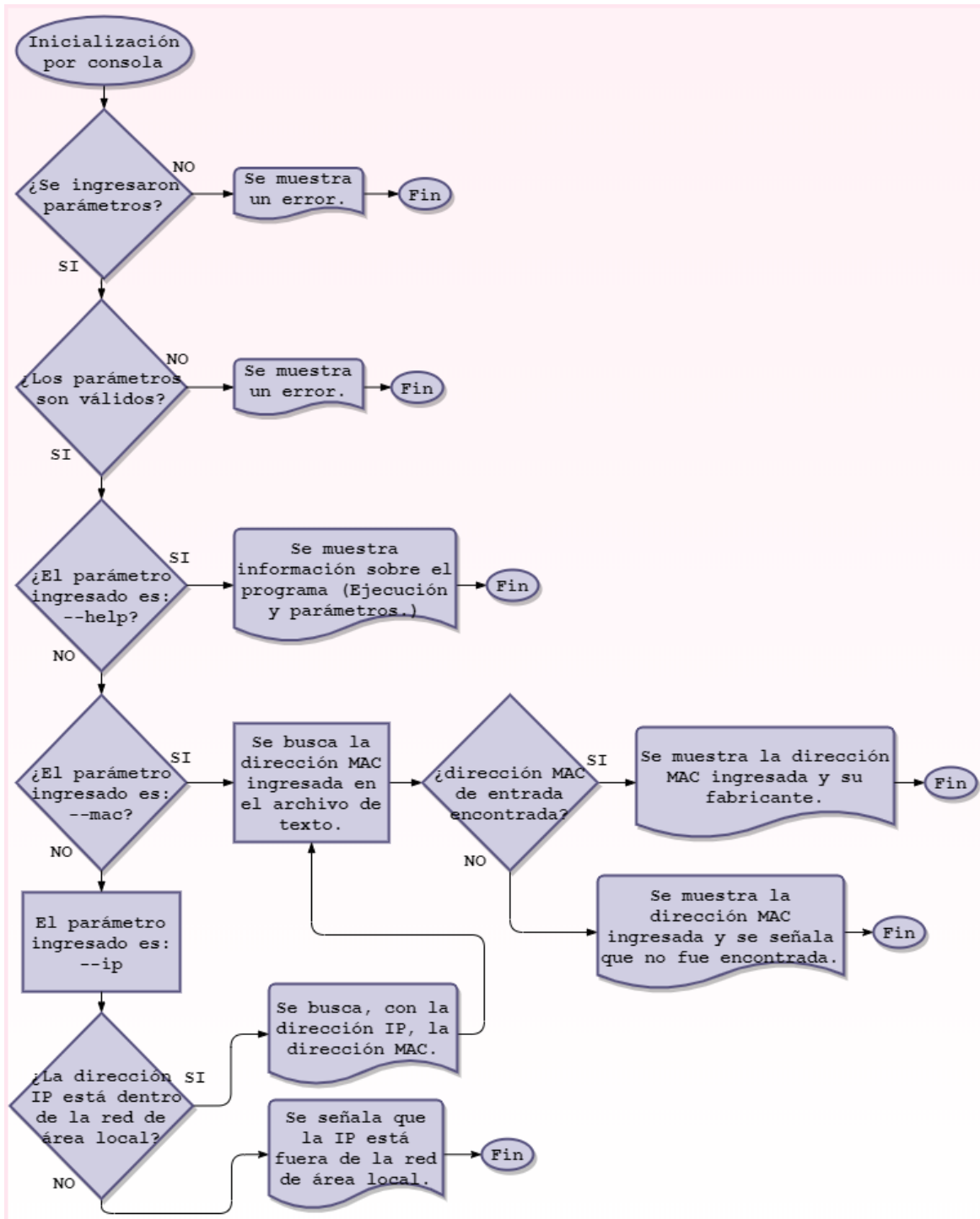


Figura 5