

Tarea 3: Consultar Fabricantes a una API

Alexander Farias Cifuentes, alexander.farias@alumnos.uv.cl

Nicolas Flores Calderon, nicolas.floresca@alumnos.uv.cl

Franco Contreras Caques, franco.contreras@alumnos.uv.cl

1. Introducción

A modo de introducción, en este informe se detallará el desarrollo de un código de programación desarrollado en Python, con el cual se puede consultar si una IP pertenece a una red, consultar el fabricante de una dirección MAC ingresada y también se puede visualizar la tabla ARP del dispositivo en cuestión.

Para el apartado de la consulta del fabricante de una dirección MAC se utilizará una API Rest de dirección “<https://maclookup.app/>”, de la cual estudiaremos su funcionamiento.

2. Materiales y Métodos

Como material primario e indispensable, utilizaremos un computador con el sistema operativo Windows 11, el cual posee integrado a las variables de entorno el lenguaje de programación Python. También usaremos una API Rest la cual está dada por la dirección <https://maclookup.app/>, la cual nos permitirá adquirir información importante al enviarle una dirección MAC.

A su vez, se utilizará el comando “ARP -A” dentro del código en Python para obtener la tabla ARP del dispositivo que esté ejecutando el código.

La forma en la que se accederá en el código a cada funcionalidad será a través de argumentos, es decir, al ejecutar el código en el “terminal” de debe especificar que función se quiere utilizar escribiéndola al lado del nombre del archivo. Si es para la función MAC o para IP se debe escribir el tipo de función más la dirección que se quiere analizar.

3. Resultados

3.1. Análisis del código.

A modo de análisis del código, hablaremos de las principales funciones que este debe cumplir.

Como primer apartado, el código al no recibir argumentos o al recibir el argumento “--help” debe mostrar en pantalla un cuadro de texto con ayuda para ejecutar bien el código. Esto se verá mejor ejemplificado en la figura 1.

```
PS C:\Users\Alexander\Documents\U\6to_Semestre\Redes 1\Tarea3> python OUILookup.py

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.
PS C:\Users\Alexander\Documents\U\6to_Semestre\Redes 1\Tarea3> python OUILookup.py --help

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.
PS C:\Users\Alexander\Documents\U\6to_Semestre\Redes 1\Tarea3> |
```

Figura 1

Como segunda función, está la funcionalidad “--ip + {inserte ip}”, la cual nos permite analizar si una ip pertenece a la red que se está utilizando, esto nos deja dos respuestas posibles. La primera, en la cual existe la ip dentro de la red, nos entrega la dirección MAC asociada a esta IP, el fabricante de esta MAC y el tiempo que se demoró en consultar el fabricante a la API. La segunda es en donde la ip no existe en la red, ahí nos muestra el siguiente mensaje: “ip is outside the host network”

Un ejemplo de esto se verá en la siguiente imagen (figura 2).

```
PS C:\Users\Alexander\Documents\U\6to_Semestre\Redes 1\Tarea3> python OUILookup.py --ip 192.168.1.15
Error: ip is outside the host network
PS C:\Users\Alexander\Documents\U\6to_Semestre\Redes 1\Tarea3> python OUILookup.py --ip 192.168.1.13
Direccion MAC: 90:55:de:5f:26:d8
Fabricante: Fiberhome Telecommunication Technologies Co.,LTD
el tiempo de la consulta fue de: 1.737353801727295 Segundos
PS C:\Users\Alexander\Documents\U\6to_Semestre\Redes 1\Tarea3> |
```

Figura 2

Como tercera funcionalidad, se puede corroborar el fabricante de una dirección MAC ingresada, cuya sintaxis debe ser así: “--MAC + {inserte MAC}”. Para esta funcionalidad puede tener dos respuestas, en donde encuentra el fabricante y en donde no, siendo la primera una salida con el nombre del fabricante y el tiempo que se demoró la consulta, y en el segundo caso no encontró fabricante. El ejemplo de como funciona esto se puede ver en la Figura 3.

```
PS C:\Users\Alexander\Documents\U\6to_Semestre\Redes 1\Tarea3> python OUILookup.py --mac 98:06:3c:92:ff:c5
Direccion MAC: 98:06:3c:92:ff:c5
Fabricante: Samsung Electronics Co.,Ltd
el tiempo de la consulta fue de: 1.6352174282073975 Segundos
PS C:\Users\Alexander\Documents\U\6to_Semestre\Redes 1\Tarea3> python OUILookup.py --mac 98:06:3f:92:ff:c5
Direccion MAC: 98:06:3f:92:ff:c5
Fabricante: not found
el tiempo de la consulta fue de: 2.074643611907959 Segundos
```

Figura 3

Como cuarta y última funcionalidad, se tiene el listado de las direcciones MAC e IP asociadas a la red del computador, esta funcionalidad se ejecuta como “--arp”. En este caso en particular, nos entrega la dirección IP, la dirección MAC y el fabricante de cada uno de los asociados. Esto se ve en la siguiente imagen (Figura 4)

```
PS C:\Users\Alexander\Documents\U\6to_Semestre\Redes 1\Tarea3> python OUILookup.py --arp
192.168.1.13 / 90:55:de:5f:26:d8 / Fiberhome Telecommunication Technologies Co.,LTD
192.168.1.1 / c4:9f:4c:bb:16:a0 / HUAWEI TECHNOLOGIES CO.,LTD
192.168.1.7 / ff:ff:ff:ff:ff:ff /
192.168.1.255 / 01:00:5e:00:00:16 /
224.0.0.22 / 01:00:5e:00:00:fb /
224.0.0.251 / 01:00:5e:00:00:fc /
224.0.0.252 / 01:00:5e:7f:ff:fa /
239.255.255.250 / ff:ff:ff:ff:ff:ff /
PS C:\Users\Alexander\Documents\U\6to_Semestre\Redes 1\Tarea3>
```

Figura 4

Ya que hemos ahondado en cada funcionalidad, para cerrar este apartado dejaremos el diagrama de flujo del programa creado.

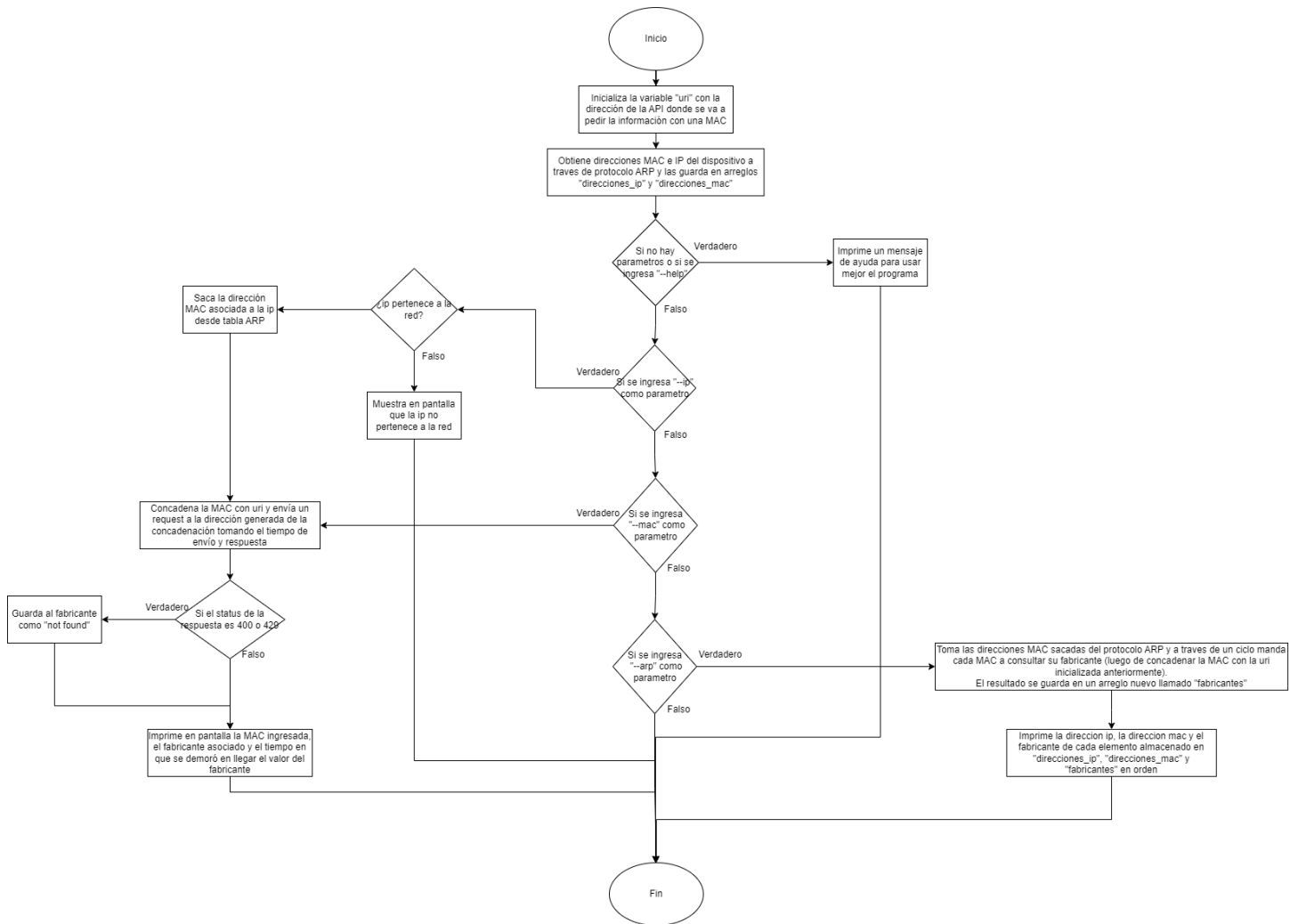


Figura 5

3.2. Preguntas

a) ¿Qué es REST? ¿Qué es una API?

Rest es una interfaz para conectar diversos sistemas basados en HTTP, cada recurso está identificado mediante URLS y se accede a través del protocolo señalado anteriormente "HTTP". Por otro lado una API es un mecanismo que permite a 2 componentes comunicarse entre sí mediante un conjunto de reglas y especificaciones, además sirve como una interfaz entre distintos programas de software.

b) ¿Cómo se relaciona el protocolo HTTP con las API REST y cuál es su función en la comunicación entre clientes y servidores?

Como se señaló anteriormente la API REST generalmente utiliza el protocolo HTTP para la comunicación entre el cliente y servidor. Su función principal es definir cómo se envían y reciben los mensajes a través de la web. En el contexto de las API REST, HTTP facilita la realización de operaciones esenciales como leer, crear, actualizar y eliminar recursos.

Al utilizar métodos HTTP estandarizados como GET, POST, PUT y DELETE, las API REST permiten una interacción eficiente y comprensible entre el cliente y el servidor. Cada uno de estos métodos se asigna a operaciones específicas que se necesitan en la comunicación web.

c) ¿Qué papel juega la dirección IP en el acceso a recursos a través de una API REST

La dirección IP identifica un dispositivo en la red, permitiendo que se envíen y reciban datos desde ese dispositivo. El papel que juega la dirección IP en el acceso a recursos en una API REST es cuando un cliente realiza una solicitud a un servidor, utiliza la dirección IP del servidor para establecer una conexión y acceder a los recursos proporcionados por la API.

d) ¿Por qué es importante considerar la latencia de red y el ancho de banda? ¿Cómo afectan estos factores al rendimiento de la API?

Es importante considerar la latencia de red y el ancho banda por su impacto directo en la eficiencia y la experiencia del usuario, ya que la latencia se sabe que es el tiempo en el que tardan los datos en viajar a través de la red, debido a esto afecta la rapidez en la cual la API responde a las solicitudes lo que sería decisivo para las aplicaciones que recurren respuestas en tiempo real. Por otro lado, el ancho de banda se sabe que determina la cantidad de datos que se transfiere en un tiempo dado y esto influye en el factor de rendimiento de la API debido a que se manejan grandes volúmenes de datos o archivos grandes.

Debido a todos los factores señalados anteriormente que determinan la confiabilidad, consistencia y escalabilidad de la API, donde afectan los factores como la eficiencia operativa como también la experiencia del usuario y el rendimiento de la API.

e) ¿Por qué el programa desarrollado utilizando API REST es más lenta su ejecución?

Es más lenta su ejecución debido a que debe buscar información que es parte de una red externa a la que nuestro dispositivo maneja, por ende, el tiempo de latencia que tendrá la petición de información será mucho mayor que pedir la información dentro de una red misma. Esto se ve reflejado en nuestro informe al momento de mostrar la tabla ARP, ya que contamos con las direcciones MAC e IP pero no con los fabricantes, por ende, se mandan las direcciones MAC mediante un request a la API Rest, y eso por cada MAC que existe hace que el tiempo de ejecución del programa se vea afectado de forma grave.

f) ¿Cuál es la diferencia entre la dirección MAC (Media Access Control) y la dirección IP, y en qué capa de la red se utilizan cada una de ellas?

La diferencia es que la dirección MAC es un identificador único asignado a la interfaz de red del dispositivo y por otro lado la dirección IP es un identificador que se asigna a un dispositivo en la red para su identificación y localización a nivel global.

Dirección mac se utiliza en la capa de enlace de datos en este caso “Capa 2”.

Dirección IP se utiliza en la capa de red en este caso “Capa 3”

g) ¿Cómo pueden las redes LAN (Local Area Networks) y WAN (Wide Area Networks) afectar la accesibilidad y la velocidad de respuesta de una API REST?

En el caso de las LAN, estas redes están diseñadas para operar en áreas geográficamente limitadas, como oficinas o edificios. Esto permite una comunicación más rápida y directa entre los dispositivos conectados. La infraestructura controlada de una LAN también ayuda a minimizar las interrupciones y los retrasos en la red, lo que se traduce en una mejor accesibilidad y una respuesta más rápida de la API REST.

Por otro lado, las WAN se extienden a través de grandes áreas geográficas, como ciudades, países o incluso continentes. Aunque las WAN permiten la conectividad a larga distancia, también introducen desafíos como una mayor latencia y una menor fiabilidad en comparación con las LAN. Estos problemas surgen debido a la mayor distancia que los datos deben recorrer y a la complejidad de mantener una red extensa y diversa. En una WAN, los datos a menudo pasan por múltiples routers y switches, lo que puede causar retrasos y pérdida de paquetes. Esto afecta negativamente la

velocidad de respuesta y la estabilidad de una API REST, especialmente cuando se accede a ella desde ubicaciones geográficamente distintas.

h) ¿Qué es un enrutador y cómo se utiliza para dirigir el tráfico de datos? ¿Qué relación tiene esto con el enrutamiento de solicitudes en una API REST?

Un enrutador es un dispositivo que dirige el tráfico de datos entre diferentes redes, direccionando paquetes de datos hacia sus destinos óptimos a través de la selección de rutas basada en una tabla de enrutamiento. Esta función se asemeja al enrutamiento en una API REST, donde el enrutador desempeña un papel análogo al dirigir las solicitudes de los clientes a los servidores correspondientes. En una API REST, el enrutamiento implica la interpretación de las solicitudes HTTP y su correspondencia con las operaciones específicas en el servidor, asegurando así que las solicitudes (como obtener, actualizar, borrar datos) lleguen al componente adecuado del servidor para su procesamiento.

i) ¿Cómo se asocian los puertos de red con servicios y aplicaciones específicas?

Los puertos de red funcionan como puertas de acceso distintivas en la comunicación de datos a través de internet y redes internas. Cada puerto, identificado por un número, está asociado con un servicio o aplicación específicos, permitiendo así una organización y distribución eficiente del tráfico de red. Esta asociación es crucial en entornos donde múltiples aplicaciones o servicios operan simultáneamente en un solo dispositivo o servidor.

Por ejemplo, los servidores web suelen utilizar el puerto 80 para tráfico HTTP (Hypertext Transfer Protocol) y el puerto 443 para HTTPS (HTTP Secure), que es una versión cifrada y segura de HTTP. Esta asignación de puertos asegura que las solicitudes enviadas a un servidor sean dirigidas correctamente al servicio correspondiente. En el caso de una API permite que las solicitudes HTTP, como GET o POST, sean enrutadas al servicio o aplicación adecuados dentro del servidor. Esto facilita la interacción entre diferentes aplicaciones y servicios en la web, permitiendo una comunicación efectiva y organizada.

4. Discusión y conclusiones

A modo de conclusión, este informe ha abordado el desarrollo de un código de programación en Python diseñado para realizar diversas consultas relacionadas con redes y dispositivos.

La capacidad del código para verificar la pertenencia de una IP a una red, identificar el fabricante de una dirección MAC y visualizar la tabla ARP del dispositivo proporciona herramientas valiosas para el análisis y gestión de redes.

La integración de la API Rest de dirección "https://maclookup.app/" para la consulta del fabricante de una dirección MAC ha enriquecido aún más la funcionalidad del código, demostrando la versatilidad y adaptabilidad del mismo en entornos de redes.

Este proyecto no solo destaca la eficiencia del desarrollo en Python, sino también la importancia de la colaboración con servicios externos para optimizar y ampliar las capacidades de la aplicación.

5. Referencias

<https://aws.amazon.com/es/what-is/api/>

<https://openwebinars.net/blog/que-es-rest-conoce-su-potencia/>

<https://www.adslzone.net/noticias/internet/direccion-ip-vs-direccion-mac-principales-diferencias/>

https://es.wikipedia.org/wiki/Protocolo_seguro_de_transferencia_de_hipertexto

<https://www.redeszone.net/tutoriales/configuracion-puertos/puertos-tcp-udp/>