

## Tarea 3: Consultar Fabricantes a una API

Alexandra Aichele Movillo, [alexandra.aichele@alumnos.uv.cl](mailto:alexandra.aichele@alumnos.uv.cl)

Felipe Caru Prado, [felipe.caru@alumnos.uv.cl](mailto:felipe.caru@alumnos.uv.cl)

Andrés González Vilches, [andres.gonzalezvi@alumnos.uv.cl](mailto:andres.gonzalezvi@alumnos.uv.cl)

### 1. Introducción

El presente informe se centra en el programa que ha sido desarrollado con el propósito de identificar el fabricante de una tarjeta de red de tres maneras diferentes: a través de la dirección IP, la dirección MAC o la tabla ARP. Para poder conseguir esto, usamos la API REST pública “*maclookup.app*”, la cual tiene la información sobre los fabricantes de tarjetas de red en todo el mundo. Además, se incorpora la funcionalidad de medir la latencia entre el usuario y la API, ofreciendo una evaluación detallada de la comunicación en tiempo real.

En un contexto más amplio, esta herramienta tiene aplicaciones significativas en la administración y el mantenimiento de redes, brindando a los usuarios una visión más profunda de la infraestructura y la facilitando la identificación de dispositivos conectados.

A lo largo del informe, se van a presentar los detalles de los métodos de desarrollo, los resultados obtenidos y se discutirán las implicaciones de esta herramienta en el ámbito de las redes.

### 2. Materiales y Métodos

Materiales:

2.1 Para la ejecución del código se utilizó el sistema operativo Windows 10 22H2 de 64 bits y el intérprete de Python en su versión 3.11.4

2.2 Bibliotecas Utilizadas

Para la resolución del código se utilizaron las siguientes bibliotecas:

- **subprocess**: Se utilizó para ejecutar los procesos desde Python. por ejemplo, la tabla arp con el comando `–a`
- **getopt**: Ayuda a procesar opciones y argumentos mediante la línea de comandos.
- **sys**: esta biblioteca proporciona el acceso a variables dentro del código en este caso se usan para acceder a la línea de comandos
- **socket**: proporciona funciones para el manejo de los sockets. Se utiliza únicamente para la obtención de la dirección IP propia.

- **uuid:** proporciona funciones propias de UUID. En el código se utilizan para obtener la dirección MAC propia.
- **requests:** Se utiliza para las solicitudes con protocolo HTTP con la API “*maclookup.app*” y obtener información sobre las direcciones MAC.
- **re:** Proporciona operaciones para la búsqueda de texto, en el código se utiliza para analizar las líneas de la tabla arp y así extraer direcciones IP y MAC de esta.
- **time:** Esta Proporciona funciones para ayudarnos con el tiempo. En el código se implementó para las mediciones del tiempo de respuesta de las solicitudes a la API y mostrar por consola.

## 2.3 API

Se utilizó la API “*maclookup.app*” para lograr una comunicación Request-Response dentro del código.

Métodos:

## 2.4 Definición de URI de la API REST:

- Se seleccionó una API REST pública para consultar la información del fabricante de tarjetas de red basadas en direcciones MAC. Se utilizó la api disponible en “*maclookup.app*”

## 2.5 Desarrollo de OUILookup:

- Se desarrolló este código en Python para facilitar la consulta del fabricante a través de direcciones IP y MAC.
- El programa fue diseñado para aceptar parámetros de línea de comandos, como `--ip`, `--mac` y `--arp`, permitiendo consultas específicas.

## 2.6 Consulta del Fabricante por IP:

- Se desarrolló una función para obtener los datos de fabricación de una tarjeta de red basada en una dirección IP. Primero verifica si la IP proporcionada es la IP del host local. Si la es local, con la librería uuid obtiene la dirección MAC y luego consulta el fabricante. Si es que no es local, obtiene la tabla arp y busca la dirección MAC perteneciente a la dirección IP ingresada. Finalmente, imprime la dirección MAC, el fabricante y el tiempo de respuesta.

## 2.7 Consulta del Fabricante por MAC:

- Dada un dirección MAC ingresada en la línea de comandos, está se pasa directamente a la una función implementada donde se usa una API REST pública y se devuelve el fabricante y el tiempo que se demoró en realizar esa acción.

---

## 2.8 Obtención de la tabla ARP:

- Esta función obtiene la tabla arp del sistema, muestra la dirección IP, MAC, fabricante correspondiente y el tiempo de respuesta.

## 2.9 Obtener el fabricante:

- Se hace una consulta a la API REST pública `https://api.maclookup.app/v2/mac/{mac}`, esta devuelve el fabricante basado en la dirección MAC que le proporciona en la url. Finalmente se muestra el nombre del fabricante y el tiempo de respuesta.

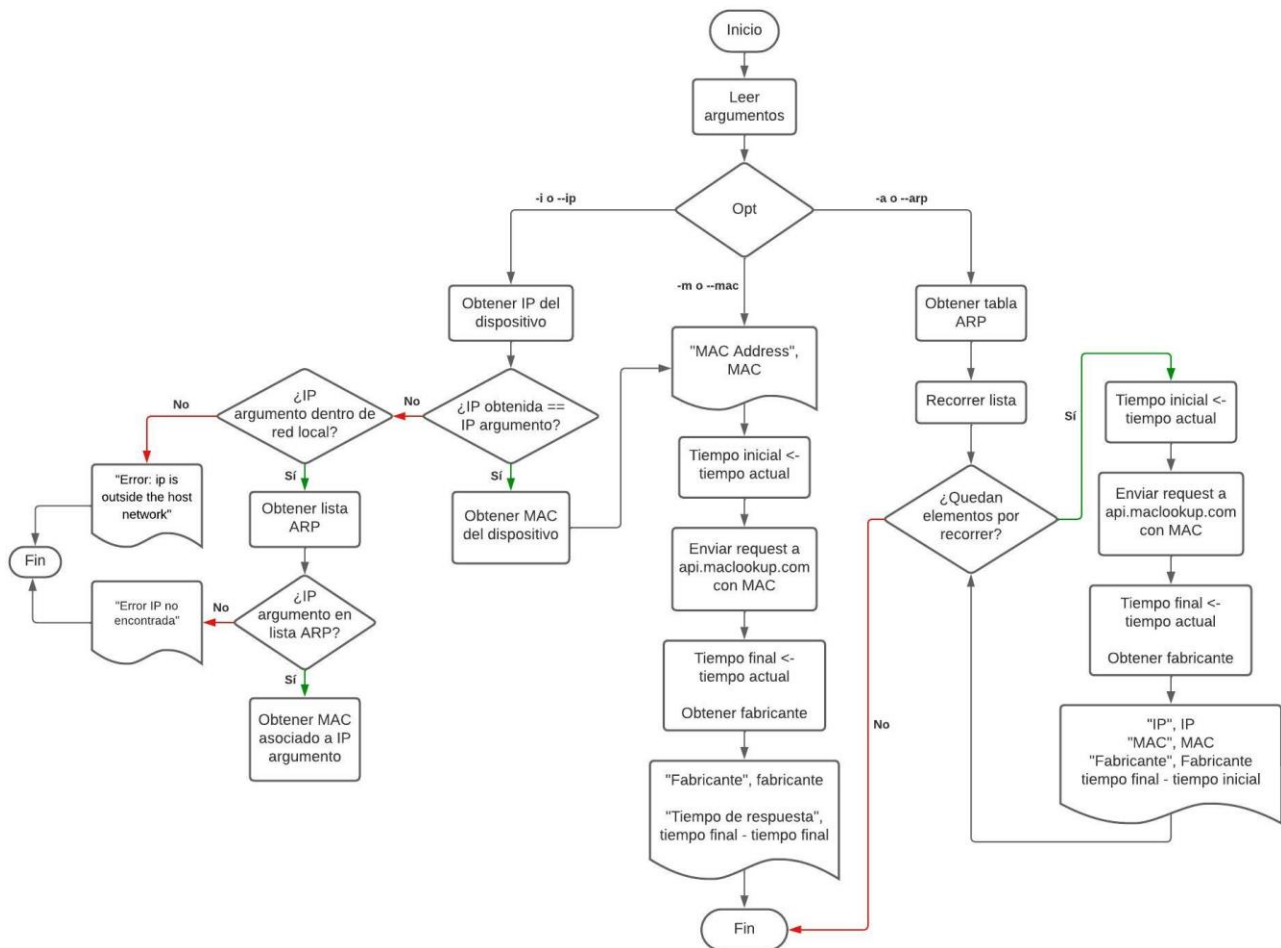
## 3. Main:

- Se encarga de procesar los argumentos de línea de comandos y llamar a las funciones apropiadas según los parámetros proporcionados. Permite consultar por IP, MAC o mostrar la tabla ARP.

### 3.1 Medición de la latencia:

- Usando una librería propia de Python llamada time, se puede invocar antes de hacer una petición a la API REST pública y luego se vuelve a llamar, teniendo esos dos tiempos se resta y se multiplica por 1000 para poder tener el tiempo en milisegundos.

### 3.2 Diagrama de Flujo:



## 3. Resultados

### Resultados

#### 3.1. Obtención de Direcciones MAC

Descripción del Proceso: Explica brevemente cómo se obtuvieron las direcciones MAC (a través de la dirección IP, directamente, o de la tabla ARP).

Las direcciones MAC se obtienen de diferentes formas dependiendo del comando ingresado por el usuario

Al obtener el fabricante por IP primero se verifica que la IP ingresada sea la IP del mismo dispositivo en que se ejecuta el programa, de ser así se obtiene la dirección MAC del dispositivo y su fabricante, en

caso de que no lo sea se verifica que la dirección IP esté dentro de la red local, si pertenece entonces busca la MAC correspondiente a esa IP dentro de la tabla ARP y con ella, su fabricante.

En caso de que ninguna de las condiciones anteriores se cumpla o que no se encuentre dentro de la tabla ARP saltará un error.

```
python OUILookup.py -i 192.168.1.1
```

```
MAC Address: 4c-ab-f8-d7-f6-50
```

Al obtener el fabricante por ARP se obtiene la tabla ARP y se obtiene la dirección MAC de cada uno de los dispositivos encontrados para obtener y desplegar su fabricante.

```
IP/MAC/Vendor
192.168.1.1      / 4c-ab-f8-d7-f6-50 /
192.168.1.220   / d0-fc-d0-a2-7b-7a /
192.168.1.221   / d0-fc-d0-a2-6a-eb /
192.168.1.222   / d0-fc-d0-a2-7a-f6 /
```

### 3.2. Consultas a la API maclookup.app

Método de Consulta: Las consultas fueron realizadas con peticiones request.get() a la siguiente URL: <https://api.maclookup.app/v2/mac/{mac}>, en donde {mac} corresponde a la dirección MAC de la que se quiere averiguar. La respuesta fue almacenada y procesada para mostrar solamente el dato del fabricante.

Respuestas de la API:

```
{"success":true,"found":true,"macPrefix":"D0FCD0","company":"HUMAX Co., Ltd.,"address":"HUMAX Village, 216, Hwan  
gsaoul-ro, Bu, Seongnam-si Gyeonggi-do 463-875, KR","country":"KR","blockStart":"D0FCD0000000","blockEnd":"D0FCD0  
FFFFFF","blockSize":16777215,"blockType":"MA-L","updated":"2022-01-20","isRand":false,"isPrivate":false}
```

### 3.3. Medición del Tiempo de Respuesta

Técnica de Medición: Para tomar el tiempo que demoraba en llegar la respuesta de la API desde que se enviaba el request correspondiente se usó la función time.time() que da como resultado el tiempo exacto en que es llamada la función, así, tomando el tiempo antes y después de la respuesta se obtuvieron los tiempos inicial y final respectivamente, los cuales al ser restados daban como resultado el tiempo de respuesta.

Resultados de Tiempos de Respuesta (10 pruebas por integrante):

Integrante	T. Mínimo	T. Máximo	T. Promedio
Andrés	741 ms	1052 ms	900 ms
Alexandra	312 ms	553 ms	422 ms
Felipe	574 ms	1799 ms	940 ms

---

### 3.4. Análisis de Resultados

Como era de esperarse el resultado del fabricante de cada una de las direcciones MAC buscadas fueron el mismo, sin embargo, el resultado del tiempo de obtención de los datos es muy distinto para cada uno de los dispositivos utilizados viéndose variaciones significativas tanto en el tiempo de respuesta de forma individual como el intervalo de tiempo de las 7 mediciones efectuadas aun cuando dos de los miembros tienen el mismo proveedor de internet, Movistar (Andrés y Alejandra).

### 3.5. Conclusiones Experimentales

En conclusión, el tiempo de obtención de respuesta de la API utilizada puede variar más o menos dependiendo de varios factores ligados a la conexión de cada dispositivo desde la que se esté efectuando, sin embargo, este tiempo es bastante corto y casi imperceptible con resultados de menos de 1 segundo en la mayoría de los casos. Además, la API da respuestas precisas, con bastantes datos relacionados a la MAC y su fabricante.

## 4. Discusión y conclusiones

La implementación exitosa de esta herramienta demuestra su utilidad como una solución a la identificación de los fabricantes de tarjetas de red. En esta herramienta la integración de la API REST garantiza datos actualizados, fiables y precisos, mientras que la medición de latencia en la conexión proporciona un rendimiento y seguridad de la consulta.

La herramienta es destacable ya que es completa y práctica no solo para la tarea sino también aborda la necesidad de encontrar los fabricantes de las tarjetas en una red mediante una IP específica de esta.

En conclusión OUILookup.py no solo se enfoca en el objetivo principal que es la consulta de fabricantes a través de direcciones MAC e IP, sino que va más allá al incorporar características que mejoran la precisión, actualización y efectividad de la gestión de la red.

## 5. Preguntas a responder

### a) ¿Qué es REST? ¿Qué es una API?

- REST: Representational State Transfer, o en español Transferencia de Estado Representacional, es un paradigma arquitectónico que establece principios para definir la comunicación entre cliente y servidor. Entre ellos está:
  - Arquitectura servidor/cliente.
  - No puede guardar estados, es stateless.
  - Exista la posibilidad de guardar información en el cache, a esto se le cacheability.
  - El sistema debe estar basado en capas.

- Interfaz uniforme.
- Puede extender la funcionalidad del cliente mediante el envío de código on demand. Esto es opcional.
- API: Interfaz de Programación de Aplicaciones, son mecanismos que permiten que dos componentes de software puedan comunicarse entre sí mediante definiciones y protocolos.

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

- El protocolo HTTP, desempeña un papel fundamental en la facilitación de la comunicación entre cliente y servidores en las API REST.
- El protocolo HTTP proporciona una infraestructura estándar para la transmisión de datos, utilizando métodos como GET, PUT, POST y DELETE, alineándose con las operaciones CRUD de las API REST.
- En la comunicación, el cliente envía solicitudes al servidor, este procesa la acción solicitada y devuelve respuestas HTTP que incluyen el estado de la operación y datos relevantes. Por lo tanto, la interacción coherente entre HTTP y las API REST sustenta la eficiencia y uniformidad en la comunicación entre sistemas.

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

- La dirección IP desempeña un papel importante en el acceso a recursos a través de una API REST. Este identificador único permite la identificación y ubicación de los clientes y servidores. Además, de implementar medidas de seguridad, redirigir tráfico, monitorear y analizar, integrar sistemas de seguridad y detectar ataques.

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

- La latencia de la red se refiere al tiempo que tarda un paquete en enviar datos. En este contexto, es de suma importancia considerar la latencia de red, ya que esta variable puede incidir directamente en la velocidad de respuesta de una API. En las situaciones donde la latencia de red es considerablemente alta, se traduce en un aumento del tiempo de respuesta, afectando de manera adversa a la experiencia de usuario al interactuar con la API y generando demoras perceptibles.
- En cuanto al ancho de banda, este concepto se refiere a la cantidad de datos que se puede transmitir en un periodo de tiempo determinado a través de una red. Por consiguiente, es esencial tener en cuenta el ancho de banda disponible, ya que si este limitado la api podría tener dificultades para manejar un alto volumen de solicitudes. Este escenario podría traducirse en tiempos de respuesta más lentos e, incluso, en la capacidad de procesar algunas solicitudes.

- 
- En consecuencia, la baja latencia de red y un ancho de banda adecuado es un factor determinante para mejorar el tiempo de respuesta y la capacidad de procesamiento de una API. Esto culmina como una experiencia mejorada para el usuario, una interacción más eficiente y una menor percepción de demoras.
- e) **¿Por qué el programa desarrollado utilizando API REST es más lenta su ejecución?**
- Visto en unidades anteriores del Curso en un modelo fin a fin pueden existir varios factores que hagan que la ejecución sea lenta.
  - Algunos de estos factores son la latencia de red, una gran cantidad de datos transferidos, que el código del servidor no sea muy eficiente, problemas con la conexión.
- 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?**
- Según lo visto en clases, se observa una gran diferencia entre las direcciones MAC y las direcciones IP. La dirección MAC se ubica en el nivel de capa 2 y se compone de 48 bits, expresados en dígitos hexadecimales, distribuidos en 12 dígitos organizados en 6 pares. Los tres primeros pares identifican al fabricante, mientras que los últimos 3 pares corresponden al modelo del dispositivo.
  - Por el contrario, las direcciones IP operan en el nivel de capa 3 y consta de 32 bits separados en grupos de 8 bits por puntos.
  - La distinción entre ambas direcciones es que las direcciones MAC dirigen el control de acceso a un medio en la capa 2 y las direcciones IP gestionan la conectividad a internet en el nivel de capa 3.
- g) **¿Cómo pueden las redes LAN (Local Área Networks) y WAN (Wide Área Networks) afectar la accesibilidad y la velocidad de respuesta de una API REST?**
- Una red LAN, o red de área local, se define así por su alcance limitado a un espacio físico reducido, como una casa, edificio, departamento. En el contexto de una API REST, tanto la red LAN como la API REST están conectadas a la misma red local lo que resulta fundamental para mejorar la accesibilidad y velocidad de respuesta de la API. Esto se traduce en una comunicación directa entre dispositivos, prescindiendo de la necesidad de pasar por una conexión a internet.
  - Una red WAN, o red de área amplia, abarcan dimensiones geográficas extensas que pueden incluir países o continentes. En relación con la API REST, la utilización de esta red puede incidir en la accesibilidad y velocidad de una respuesta. Dado que una red WAN abarca distancias considerables, los datos deben recorrer trayectos más extensos y atravesar múltiples puntos de enrutamiento antes de alcanzar su destino final. Esto puede impactar negativamente la velocidad de respuesta de la API.
- 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 facilita la conexión y el intercambio de datos entre dispositivos a través de internet, actuando como una puerta de enlace para dicha conexión. Operando a nivel



de la capa 3 del modelo OSI, estos dispositivos utilizan el protocolo de internet (IP) para dirigir paquetes de datos, cuyos encabezados incluyen información sobre las direcciones IP de origen y destino cuando estos pertenecen a redes distintas. En este sentido, el enrutador desempeña un papel esencial al posibilitar la comunicación efectiva entre dispositivos ubicados en redes separadas mediante la interpretación y dirección precisa de los paquetes de datos.

- La relación que existe entre ambos conceptos es la idea de dirigir el flujo de información de manera efectiva. Mientras que un enrutador de red dirige paquetes de datos entre redes, el enrutador en una API REST dirige solicitudes HTTP hacia los recursos correspondientes en el servidor. Ambos desempeñan un papel crucial en la gestión y optimización de la comunicación, ya sea a nivel de red o a nivel de aplicación.

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

- Los puertos de red se asocian con servicios y aplicaciones específicas mediante un proceso llamado “asignación de puertos”. Es un proceso mediante el cual un sistema operativo asigna un puerto de red a un servicio o aplicación específica. Hay muchos métodos para asignar los puertos, ej.: asignación de puertos dinámicos, asignación de puertos estáticos, asignación de puertos por nombre, asignación de puertos por dirección IP.
- Una vez que ya están asignados los puertos de red, el sistema operativo utiliza un protocolo de comunicación para comunicarse con la aplicación o servicio a través de ese puerto.
- Algunos de los protocolos son: TCP (transmisión control protocol), UDP (user datagram protocol), HTTP (hypertext transfer protocol), FTP (file transfer protocol).

## 6. Referencias

- [1] Python Software Foundation. “Python Documentation.” Python.org. Available: [<https://www.python.org/>]
- [2] Computer Weekly. “Red de Área Local (LAN).” Computer Weekly. [Enlace: <https://www.computer-weekly.com/es/definicion/Red-de-area-local-o-LAN>]
- [3] Amazon Web Services(AWS). “¿Qué es una API?” AWS. [ Enlace: <https://aws.amazon.com/es/what-is/api/>]
- [4] Cloudflare. “¿Qué es un enrutador?” Cloudflare. [Enlace: <https://www.cloudflare.com/es-es/learning/network-layer/what-is-a-router/>]
- [5] Cloudflare. “Cómo funcionan las APIs.” Cloudflare. [Enlace: <https://www.cloudflare.com/es-es/learning/security/api/how-do-apis-work/>]
- [6] Xataka, “¿Qué es una dirección IP y en qué se diferencia de la dirección MAC?” [Enlace: <https://www.xataka.com/basics/que-direccion-ip-que-se-diferencia-direccion-mac#:~:text=La%20direcci%C3%B3n%20MAC%20identifica%20la,Internet%20la%20direcci%C3%B3n%20IP%20p%C3%BAblica.>]
- [7] Amazon Web Services (AWS). “¿Qué es el enrutamiento? [Enlace: <https://aws.amazon.com/es/what-is/routing/#:~:text=Un%20enrutador%20es%20un%20dispositivo,cumplen%20principalmente%20tres%20funciones%20principales.&text=Un%20enrutador%20determina%20la%20ruta,un%20origen%20a%20un%20destino.>]
- [8] IONOS. “Puertos TCP Y UDP: Conoce sus funciones y diferencias.” [Enlace: <https://www.ionos.es/digitalguide/servidores/know-how/puertos-tcp-puertos-udp/>]