

# Informe Tarea 3

Maximiliano Casas Echeverría, [maximiliano.casas@alumnos.uv.cl](mailto:maximiliano.casas@alumnos.uv.cl)

Alex Rodríguez Celis, [alex.rodriguez@alumnos.uv.cl](mailto:alex.rodriguez@alumnos.uv.cl)

Fernando Zamora Fuentes, [fernando.zamora@alumnos.uv.cl](mailto:fernando.zamora@alumnos.uv.cl)

## 1. Introducción

En la actualidad es importante saber el funcionamiento interno de comunicación que presentan los computadores entre sí para poder configurar y proteger nuestras redes y dispositivos de una mejor manera. Mediante el siguiente informe se tiene como finalidad explicar la implementación de una herramienta desarrollada a base de comandos, para poder consultar e identificar el fabricante de una tarjeta de red con tan solo conocer su dirección IP o MAC, adicional a esto se encuentra la función de solicitud arp en el cual se conocerán todas las direcciones IP que se encuentran en la misma red que la nuestra. Y para reforzar el análisis de este informe ocuparemos una API REST en función de nuestra herramienta y su latencia respectiva en los resultados de cada uno de los integrantes. En las siguientes secciones se explicará detalladamente más el procedimiento de como hicimos estos procesos.

## 2. Materiales y Métodos

### 2.1. Materiales

En esta subsección explicaremos los materiales que usamos para el desarrollo de esta herramienta mediante la fabricación de comandos en el lenguaje de programación Python.

- Un computador con sistema operativo Windows.
- El editor de código Visual Studio Code.
- Una API REST

Se utilizó el editor de código Visual Studio Code, que presenta una amplia cantidad de librerías y funcionalidades muy cómodas para el programador tanto en tema de interfaz como codificación. Además utilizaremos una API REST de consulta de direcciones MAC para acceder a la base de datos de fabricantes. En este caso ocuparemos una API publica (<https://maclookup.app/>) para el desarrollo de esta tarea.

## **2.2. Métodos**

Para el desarrollo de los comandos ocuparemos los siguientes métodos (librerías):

**argparse:** Es una herramienta completa de análisis de argumentos de línea de comandos, y maneja argumentos opcionales y requeridos.

**ipaddress:** Este módulo incluye clases para trabajar con direcciones de red de tipo IPv4 e IPv6.

**Subprocess:** Da la posibilidad de ejecutar un comando y esperar a que este finalice.

**Requests:** Esta biblioteca facilita realizar solicitudes HTTP, como obtener datos de una página web o interactuar con servicios web y API.

**Getmac:** Se utiliza para cuando se desea escribir la dirección MAC en un analizador de red o cuando se necesite saber qué protocolos están actualmente en uso en cada adaptador de red de un equipo.

### 3. Resultados

#### 3.1. Menú de opciones

```
PS C:\Users\Fernando\Desktop\u\Redes2> python tarea2-OUILookup.py --help
usage: tarea2-OUILookup.py [-h] [--ip IP] [--mac MAC] [--arp]

Consulta el fabricante de una tarjeta de red dada su dirección MAC o IP.

options:
  -h, --help  show this help message and exit
  --ip IP      IP del host a consultar.
  --mac MAC    MAC a consultar.
  --arp        Muestra los fabricantes de los host disponibles en la tabla ARP.
```

#### 3.2. Búsqueda por dirección MAC

```
PS C:\Users\Fernando\Desktop\u\Redes2> python tarea2-OUILookup.py --mac 0C:54:15:50:F1:D6

Dirección MAC: 0C5415
Fabricante: Intel Corporate
Tiempo de ejecución: 0.8950672149658203 segundos
```

##### Integrante 1

```
PS C:\Users\plugs> python OUILookup.py --mac A8:A1:59:42:FC:F6

Dirección MAC: A8A159
Fabricante: ASRock Incorporation
Tiempo de ejecución: 0.8833329677581787 segundos
```

##### Integrante 2

```
PS C:\Users\max casas\Downloads> python OUILookup.py --mac A8:A1:59:42:FC:F6

Dirección MAC: A8A159
Fabricante: ASRock Incorporation
Tiempo de ejecución: 2.6235435009002686 segundos
```

##### Integrante 3

### 3.3. Solicitud ARP

```
PS C:\Users\Fernando\Desktop\u\Redes2> python tarea2-OUILookup.py --arp
```

Tabla ARP:

Interfaz: 192.168.1.87 --- 0xb

Dirección de Internet	Dirección física	Tipo
192.168.1.1	fc-12-63-1d-06-80	dinámico
192.168.1.255	ff-ff-ff-ff-ff-ff	estático
224.0.0.22	01-00-5e-00-00-16	estático
224.0.0.251	01-00-5e-00-00-fb	estático
224.0.0.252	01-00-5e-00-00-fc	estático
239.255.255.250	01-00-5e-7f-ff-fa	estático
255.255.255.255	ff-ff-ff-ff-ff-ff	estático

Interfaz: 192.168.1.87

192.168.1.1	fc-12-63-1d-06-80
192.168.1.255	ff-ff-ff-ff-ff-ff
224.0.0.22	01-00-5e-00-00-16
224.0.0.251	01-00-5e-00-00-fb
224.0.0.252	01-00-5e-00-00-fc
239.255.255.250	01-00-5e-7f-ff-fa
255.255.255.255	ff-ff-ff-ff-ff-ff

#### Integrante 1

```
PS C:\Users\plugs> python OUILookup.py --arp
```

Tabla ARP:

Interfaz: 192.168.0.51 --- 0x9

Dirección de Internet	Dirección física	Tipo
192.168.0.1	78-6a-1f-5b-40-6b	dinámico
192.168.0.255	ff-ff-ff-ff-ff-ff	estático
224.0.0.22	01-00-5e-00-00-16	estático
224.0.0.251	01-00-5e-00-00-fb	estático
224.0.0.252	01-00-5e-00-00-fc	estático
239.255.255.250	01-00-5e-7f-ff-fa	estático
255.255.255.255	ff-ff-ff-ff-ff-ff	estático

Interfaz: 192.168.0.51

192.168.0.1	78-6a-1f-5b-40-6b
192.168.0.255	ff-ff-ff-ff-ff-ff
224.0.0.22	01-00-5e-00-00-16
224.0.0.251	01-00-5e-00-00-fb
224.0.0.252	01-00-5e-00-00-fc
239.255.255.250	01-00-5e-7f-ff-fa
255.255.255.255	ff-ff-ff-ff-ff-ff

#### Integrante 2

```
PS C:\Users\max casas\Downloads> python OUILookup.py --arp

Tabla ARP:

Interfaz: 192.168.100.50 --- 0xe
Dirección de Internet      Dirección física      Tipo
192.168.100.1              e0-00-84-e7-52-0b    dinámico
192.168.100.2              84-a0-6e-17-01-1f    dinámico
192.168.100.11             d0-57-94-ee-26-10    dinámico
192.168.100.62             90-34-fc-c3-bf-fd    dinámico
192.168.100.102            fc-d5-d9-cc-0c-d3    dinámico
192.168.100.103            fc-d5-d9-cc-08-53    dinámico
192.168.100.163            b0-99-d7-cc-b0-2c    dinámico
192.168.100.255            ff-ff-ff-ff-ff-ff    estático
224.0.0.22                 01-00-5e-00-00-16    estático
224.0.0.251                01-00-5e-00-00-fb    estático
224.0.0.252                01-00-5e-00-00-fc    estático
239.255.255.250            01-00-5e-7f-ff-fa    estático
255.255.255.255            ff-ff-ff-ff-ff-ff    estático
```

## Integrante 3

## 3.4. Búsqueda por IP (casos posibles)

```
PS C:\Users\Fernando\Desktop\u\Redes2> python tarea2-OUILookup.py --ip 192.168.1.1

Dirección IP: 192.168.1.1
Estado: En la misma red
MAC: fc:12:63:1d:06:80

Dirección MAC: FC1263
Fabricante: ASKEY COMPUTER CORP
Tiempo de ejecución: 0.6740505695343018 segundos
```

## Integrante 1

```
PS C:\Users\plugs> python OUILookup.py --ip 192.168.0.255

Dirección IP: 192.168.0.255
Estado: Fuera de la red
MAC: 0c:9a:3c:89:f2:88

Dirección MAC: 0C9A3C
Fabricante: Intel Corporate
Tiempo de ejecución: 0.9557404518127441 segundos
```

## Integrante 2

```
PS C:\Users\max casas\Downloads> python OUILookup.py --ip 192.168.100.62

Dirección IP: 192.168.100.62
Estado: Fuera de la red
MAC: 90:34:fc:c3:bf:fd

Dirección MAC: 9034FC
Fabricante: Hon Hai Precision Ind. Co.,Ltd.
Tiempo de ejecución: 1.6956813335418701 segundos
```

### Integrante 3

#### 3.5 Preguntas a responder

a) ¿Qué es REST? ¿Qué es una API?

R: REST es un estilo arquitectónico para el diseño de sistemas distribuidos, especialmente aquellos que operan a través de la World Wide Web. REST se basa en un conjunto de principios y restricciones que, cuando se aplican correctamente, facilitan la creación de sistemas web escalables, eficientes y mantenibles.

Una API es un conjunto de reglas y herramientas que permite que diferentes aplicaciones se comuniquen entre sí. Las APIs pueden ser utilizadas para acceder a servicios web, bases de datos, bibliotecas de funciones y otros componentes 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?

R: HTTP actúa como el protocolo de transporte para las API REST, proporcionando un conjunto de reglas y estándares que permiten la comunicación eficiente entre clientes y servidores. La función del protocolo HTTP (Protocolo de Transferencia de Hipertexto) en la comunicación entre clientes y servidores es facilitar la transferencia de datos a través de la World Wide Web.

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

R: Cuando alguien quiere usar información o funciones proporcionadas por una API REST, le envía una solicitud a través de Internet al servidor que aloja esa API. La dirección IP del servidor es esencial para que la solicitud llegue al lugar correcto en la red. A menudo, además de la dirección IP, se utiliza un nombre de dominio, que es más fácil de recordar, para acceder al servidor de 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?

R: La velocidad de respuesta de red y la capacidad de transferencia son aspectos fundamentales que deben tenerse en cuenta al trabajar con APIs, ya que influyen de manera notable en cómo se desempeña y cuán eficiente es la comunicación entre los usuarios y los servidores.

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

R: La velocidad de ejecución de un programa desarrollado utilizando una API REST puede estar influenciada por varios factores, como la latencia de red, número de solicitudes, tamaño de los datos transferidos, carga del servidor, diseño de la API, autenticación y autorización, entre otros.

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?

R: La dirección MAC se utiliza para la comunicación en una red local, mientras que la dirección IP se utiliza para la comunicación en redes más amplias, incluyendo Internet. La dirección MAC opera en la capa de enlace de datos, mientras que la dirección IP opera en la capa de red.

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?

R: La elección entre LAN y WAN para una implementación de API REST dependerá de los requisitos de accesibilidad y rendimiento. Mientras que las LAN ofrecen baja latencia y alta velocidad para comunicación interna, las WAN permiten la conexión a través de distancias más grandes, pero pueden tener mayores desafíos en términos de latencia y velocidad de respuesta.

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?

R: Un enrutador es un dispositivo de red que se utiliza para dirigir el tráfico de datos entre diferentes redes. Su función principal es determinar la ruta más eficiente para enviar datos desde la fuente hasta el destino. aunque el enrutador de red y el enrutador de API REST operan en capas diferentes de la arquitectura, comparten el principio fundamental de dirigir el tráfico hacia su destino.

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

R: La asociación de puertos de red con servicios y aplicaciones específicas se basa en un sistema de números de puerto estándar. Este sistema facilita la identificación y el enrutamiento de datos al garantizar que diferentes servicios y aplicaciones utilicen puertos específicos y bien conocidos para la comunicación en una red.

---

#### 4. Discusión y conclusiones

En el transcurso de esta tarea, la creación de esta herramienta de comandos ha destacado la importancia de comprender y optimizar la comunicación interna entre computadoras. La capacidad de identificar el fabricante de una tarjeta de red mediante la dirección MAC o IP no solo es un avance técnico, sino también una contribución valiosa al ámbito de la seguridad y configuración de sistemas.

Al utilizar una herramienta de línea de comandos en Python y aprovechar una API REST publica, hemos logrado no solo obtener información precisa sobre los fabricantes de tarjetas de red, sino también incorporar mediciones de latencia que enriquecen nuestra comprensión del rendimiento de la aplicación.

En resumen, la implementación de esta línea de comandos no solo cumple con sus objetivos originales, sino que también destaca la necesidad continua de comprender y mejorar la interconexión de sistemas en un mundo digital en constante evolución.