

Consultar Fabricantes a una API

Francisca Meyer Rivero, francisca.meyer@alumnos.uv.cl

Sebastian Rodríguez Rodríguez, sebastian.rodriguezro@alumnos.uv.cl

Lukas Scheel Belmar, lukas.scheel@alumnos.uv.cl

1. Introducción

En el siguiente informe se documenta el desarrollo de la herramienta de línea de comandos "OUILookup" implementada en Python, con el propósito de permitir la consulta del fabricante de una tarjeta de red a través de su dirección MAC o dirección IP. Este proyecto se enmarca en la utilización de una API REST de consulta de direcciones MAC, específicamente la disponible en <https://maclookup.app>, que posibilita acceder a una extensa base de datos de fabricantes de tarjetas de red.

A lo largo de este informe, se presentarán los detalles técnicos del programa "OUILookup," se describirá el proceso de implementación, se analizarán los resultados obtenidos a través de ejemplos prácticos, y se discutirán posibles áreas de mejora. Este informe servirá como una guía exhaustiva para comprender la funcionalidad, la eficacia y las implicaciones prácticas de la herramienta desarrollada.

2. Materiales y Métodos

2.1. Materiales:

Infraestructura Computacional:

Sistema Operativo: Se empleó un entorno Linux para el desarrollo del programa, con posibilidad de ejecución en sistemas Windows.

Herramientas de Desarrollo:

Python: Se utilizó el lenguaje de programación Python para la implementación del programa "OUILookup."

Bash (Linux): Se emplearon scripts de Bash para la ejecución de la herramienta en sistemas Linux

API REST:

Se hizo uso de la API REST pública de consulta de direcciones MAC disponible en <https://maclookup.app> para obtener información sobre fabricantes de tarjetas de red.

Comandos de Medición de Latencia:

Comandos "ping" y "PowerShell" (en sistemas Windows): Se utilizaron para medir la latencia entre la aplicación y la API REST.

2.2. Métodos:

Implementación de "OUILookup":

Desarrollo en Python: El programa "OUILookup" fue implementado en Python, aprovechando las capacidades del lenguaje para realizar consultas a la API REST y procesar los resultados.

Consulta a la API REST:

Algoritmo de Consulta: Se implementó un algoritmo que mide el tiempo transcurrido entre el envío de la solicitud y la recepción de la respuesta, utilizando timestamps de inicio y fin.

Medición de Latencia:

Uso de Comandos "ping" y "PowerShell": Se implementó un algoritmo para medir la latencia entre la aplicación y la API REST utilizando los comandos "ping" y "PowerShell."

3. Resultados

3.1. Diagrama de flujo

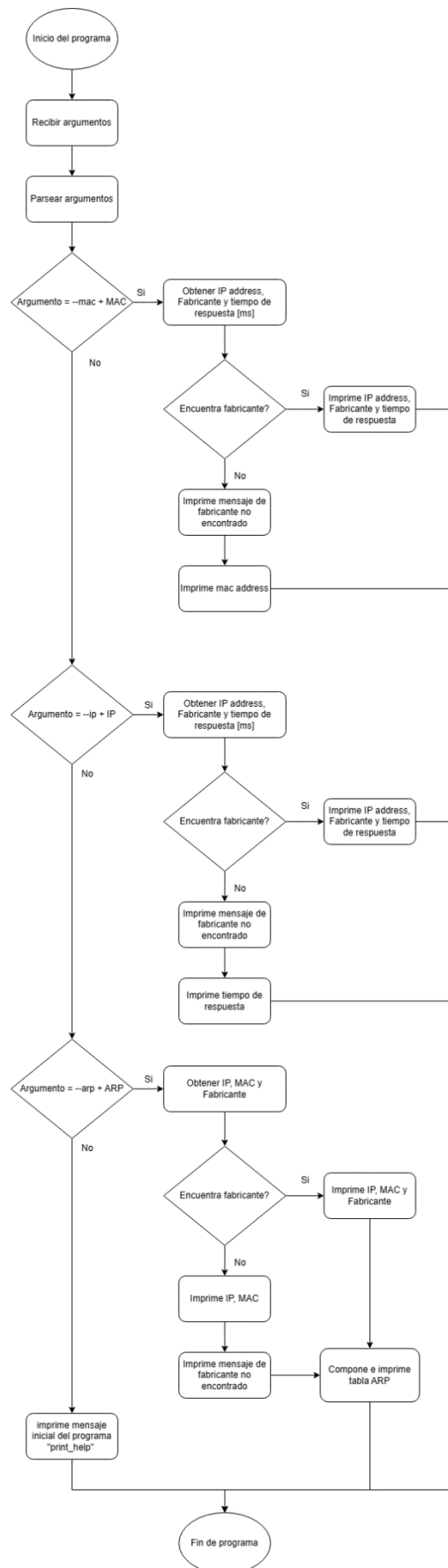


Figura 1: Diagrama de flujo de código

3.2. Preguntas

a) ¿Qué es REST? ¿Qué es una API?

REST, que significa Transferencia de Estado Representacional, es un estilo arquitectónico para diseñar sistemas distribuidos y servicios web. Es un conjunto de principios que se centran en la creación de servicios web eficientes, escalables y orientados a los recursos. Por otro lado, Una API, o Interfaz de Programación de Aplicaciones, es un conjunto de reglas y definiciones que permite que diferentes softwares se comuniquen entre sí. Proporciona una forma estructurada y documentada para que otras aplicaciones consuman las funciones o servicios de una aplicación o plataforma.

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

HTTP está estrechamente vinculado a las API REST al servir como el protocolo fundamental para la comunicación entre clientes y servidores en la web. Actúa como el canal en el cual se transmiten mensajes entre un navegador (cliente) y un servidor web. A través de URIs, HTTP identifica y localiza recursos en el servidor, y las representaciones de estos recursos se transfieren entre clientes y servidores en formatos como JSON o XML. En conjunto, HTTP y API REST establecen convenciones que facilitan la construcción eficiente y escalable de servicios web.

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

La dirección IP desempeña un papel fundamental en el acceso a recursos a través de una API REST al identificar de manera única la ubicación del cliente o del servidor que realiza la solicitud. La dirección IP del cliente que inicia la petición y la del servidor que aloja los recursos son esenciales para establecer la conexión. La dirección IP del cliente se utiliza para identificar la fuente de la solicitud, permitiendo al servidor enviar la respuesta correspondiente.

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 crucial considerar la latencia de red y el ancho de banda al evaluar el rendimiento de una API, ya que estos factores influyen directamente en la velocidad y eficiencia de la comunicación entre clientes y servidores. Altas latencias pueden resultar en tiempos de respuesta más lentos, afectando la velocidad de acceso a los recursos de la API. Asimismo, un ancho de banda limitado puede conducir a retrasos en la transferencia de datos, afectando negativamente el rendimiento general de la API.

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

La ejecución más lenta del programa utilizando API REST puede atribuirse a la transferencia de datos a través de la red en cada interacción cliente-servidor. Las múltiples solicitudes HTTP necesarias para completar tareas pueden aumentar el tiempo de ejecución, siendo la latencia de red, el tamaño de las respuestas y la complejidad de las operaciones factores influyentes. Es esencial optimizar la implementación y considerar estrategias como la caché para mejorar la eficiencia en el uso de API REST.

- f) **¿Cuáles 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 dirección MAC identifica dispositivos en una red local a nivel de la capa de enlace (capa 2), mientras que la dirección IP proporciona una identificación lógica para la conectividad global entre redes en la capa de red (capa 3). La dirección MAC es específica del hardware de la tarjeta de red, mientras que la dirección IP facilita la comunicación más allá de fronteras geográficas.

- 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?**

La arquitectura de redes, ya sea LAN o WAN, puede tener un impacto significativo en la accesibilidad y velocidad de respuesta de una API REST. En una LAN, donde los dispositivos están en proximidad física, la comunicación tiende a ser más rápida y eficiente, mejorando la accesibilidad y reduciendo la latencia. Por otro lado, en una WAN, que abarca distancias geográficas más amplias, la latencia puede aumentar debido a la mayor distancia y la intervención de dispositivos de red intermedios. La velocidad de respuesta de la API REST puede disminuir en una WAN debido a la necesidad de transferir datos a través de conexiones más extensas.

- 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 de red que dirige el tráfico de datos entre redes. Funciona tomando decisiones basadas en la información de las direcciones IP de origen y destino de los paquetes de datos. El enrutador determina la mejor ruta para enviar los datos desde el origen hasta el destino, optimizando la eficiencia de la red. En el contexto de una API REST, el enrutamiento de solicitudes se refiere a la asignación de solicitudes HTTP específicas a funciones o recursos correspondientes en el servidor.

- 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 sistema de numeración estándar conocido como el registro de puertos de la IANA (Internet Assigned Numbers Authority). Cada servicio o aplicación utiliza un puerto específico para la comunicación. Cuando un dispositivo envía datos a través de la red, la información del puerto de destino guía a la red para dirigir esos datos al servicio o aplicación correcto en el dispositivo de destino. Este enfoque facilita la comunicación eficiente y organizada entre aplicaciones y servicios en el entorno de red.

4. Discusión y conclusiones

En resumen, la creación de la herramienta de línea de comandos "OUILookup" en Python para la consulta de fabricantes de tarjetas de red a través de direcciones MAC o IP ha sido un proyecto exitoso. Aprovechando una API REST pública en <https://maclookup.app>, se logró acceder a una extensa base de datos de fabricantes. La infraestructura, basada en un entorno Python y comandos como "ping" y "PowerShell" para medir la latencia, ha demostrado ser sólida. Los resultados y respuestas a preguntas clave sobre REST, API, HTTP, direcciones IP y MAC, LAN, WAN, enrutadores y puertos proporcionan un entendimiento completo, destacando la importancia de considerar la latencia de red y el ancho de banda en el rendimiento de la API REST.

En términos de reflexiones finales, este informe sirve como una guía detallada para entender la funcionalidad y aplicabilidad práctica de la herramienta desarrollada. Identifica oportunidades de mejora, como la optimización de operaciones, y sienta las bases para investigaciones futuras en el diseño eficiente de herramientas basadas en API REST, con énfasis en la eficiencia y la escalabilidad en diversos entornos de red.