

Redes de Computadores, 2023 Tarea 3: Consultar fabricantes a una API.

1. Introducción

En el presente trabajo se desarrollará una aplicación llamada OUILookup, la cual será desarrollada en el lenguaje de programación Python, donde se implementará una herramienta la cual tendrá la finalidad de consultar el fabricante de una tarjeta de red dada su dirección MAC o su IP, para esto utilizaremos una API REST de consulta de direcciones MAC para acceder a la base de datos del fabricante.

El objetivo de este informe es detallar de manera clara como fue desarrollada la herramienta basada en líneas de comandos, la cual fue desarrollada en Python, además de la funcionalidad de esta y su respectivo diagrama de flujo. La herramienta será desarrollada en Windows por ende no se asegura su correcta funcionalidad en Mac Os o en Linux.

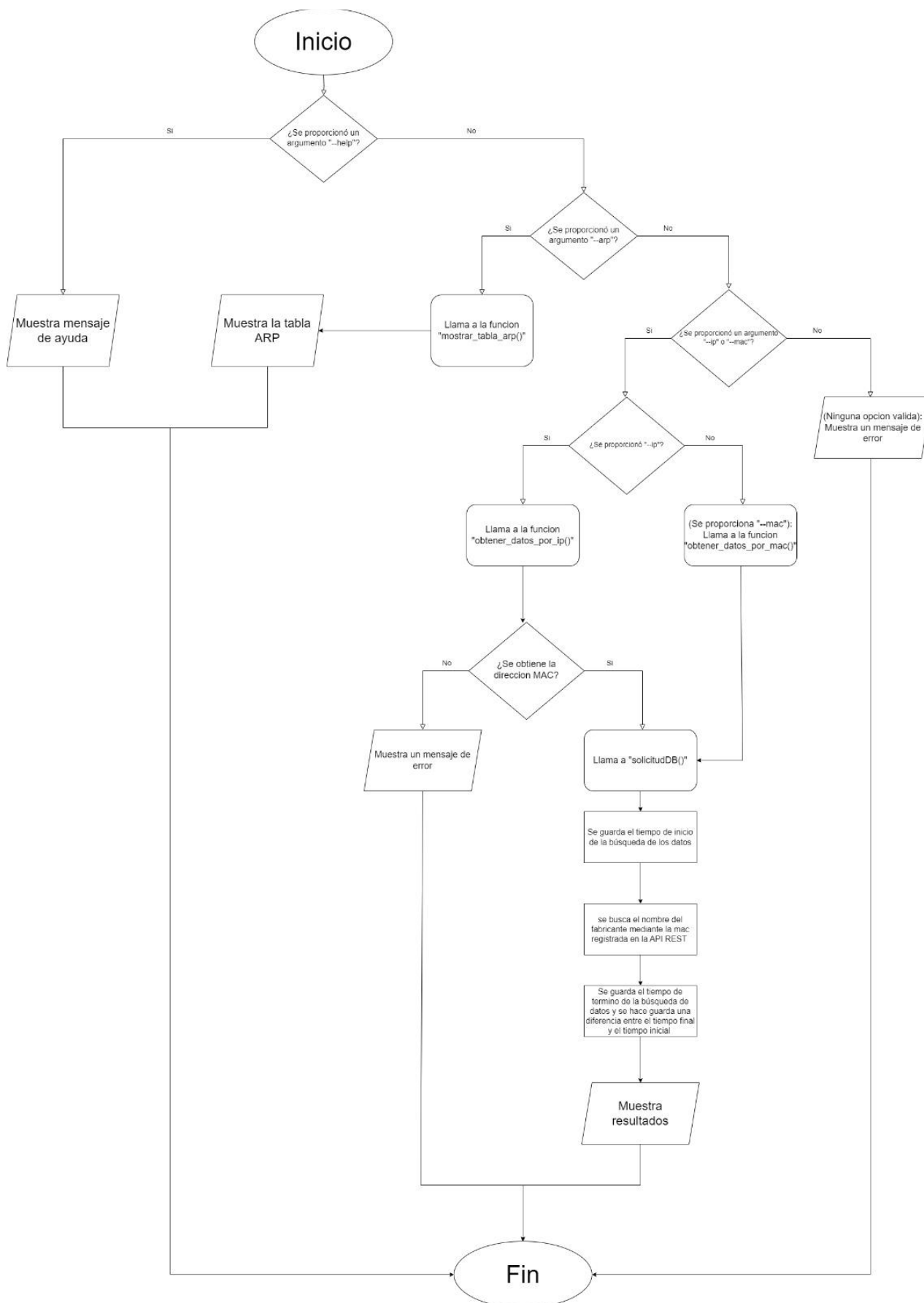
2. Materiales y Métodos

2.1. Materiales.

Para medir la latencia entre la API REST y nuestros computadores utilizaremos los comandos PING y PowerShell y el algoritmo que se nos fue entregado en la tarea.

2.2. Trabajo a realizar.

2.2.1. Diagrama de flujo.



-
1. Inicio del programa: El programa comienza su ejecución.
 2. ¿Se proporciona --help?: En este paso, el programa verifica si se proporcionó el argumento --help al ejecutar el programa.
 - Sí: Si se proporcionó --help, el programa muestra una ayuda que describe cómo usar el programa y luego termina la ejecución.
 - No: Si no se proporcionó --help, el programa continúa a la siguiente verificación.
 3. ¿Se proporciona --arp?: El programa verifica si se proporcionó el argumento --arp al ejecutar el programa.
 - Sí: Si se proporcionó --arp, el programa llama a la función mostrar_tabla_arp() que muestra la tabla ARP (una lista de direcciones IP y MAC en la red) y luego termina.
 - No: Si no se proporcionó --arp, el programa continúa a la siguiente verificación.
 4. ¿Se proporciona --ip o --mac?: El programa verifica si se proporcionó alguno de los argumentos -ip o --mac.
 - Sí: Si se proporcionó al menos uno de los argumentos, el programa continúa a la siguiente verificación.
 - No: Si no se proporcionó ninguno de los argumentos válidos, el programa muestra un mensaje de error y termina.
 5. ¿Se proporciona --ip?: El programa verifica si se proporcionó el argumento --ip.
 - Sí: Si se proporcionó --ip, el programa llama a la función obtener_datos_por_ip() para buscar información basada en la dirección IP.
 - ¿Se obtiene la dirección MAC?: La función obtener_datos_por_ip() intenta obtener la dirección MAC relacionada con la dirección IP.
 - Sí: Si se obtiene la dirección MAC, el programa llama a solicitudDB() para buscar información adicional en la base de datos y muestra los resultados. Luego termina.
 - No: Si no se obtiene la dirección MAC, el programa muestra un mensaje de error y termina.
 - No (se proporciona --mac en su lugar): Si se proporcionó --mac en lugar de --ip, el programa llama a la función obtener_datos_por_mac() para buscar información basada en la dirección MAC.
 - El programa llama a solicitudDB() para buscar información adicional en la base de datos, se guarda el tiempo de inicio de la búsqueda de la base de datos, luego busca el fabricante mediante de la mac registrada en la API REST, después del paso anterior se guarda el tiempo de término de la búsqueda de datos y se guarda una diferencia entre el tiempo final y el tiempo inicial, y muestra los resultados. Luego termina.

6. Fin del programa: El programa alcanza este punto y termina su ejecución.

2.2.2. Programa.

El programa esta en el repositorio y la explicación de su funcionamiento esta comentada en este mismo.

2.2.3. Preguntas a responder.

a) ¿Qué es REST? ¿Qué es una API?

REST (Transferencia de estado representacional) Es un estilo arquitectónico utilizado para el desarrollo de servicios web. Basado en la representación de recursos y en la comunicación sin estado entre el cliente y el servidor a través de estándares de HTTP, o en otras palabras es una forma de organizar y diseñar servicios web para que las computadoras puedan hablar entre sí de manera fácil y eficiente. Se basa en identificar y manipular recursos a través de operaciones estándar, como obtener, crear, actualizar o eliminar. Este se ha convertido en un enfoque utilizado para diseñar sistemas distribuidos en la web, utilizando un enfoque simple y eficiente en la creación de APIs web. REST esta basado en los principios de recursos, representación, estado cliente servidor, operaciones sin estado, operaciones estándar HTTP e hypermedia as the engine of application state.

Una API (Interfaz de Programación de Aplicaciones) es un conjunto de reglas y definiciones que permite que diferente software interactúen y se comuniquen entre sí, en palabras simples es darle instrucciones a una computadora para que haga algo específico, como obtener información de un sitio web o realizar una acción en una aplicación, ya sea a nivel de servicios web en la web o a través de bibliotecas y frameworks en el desarrollo de software. Existen varios tipos de APIs pero hay dos de las más comunes son las APIs basadas en la web y las APIs de bibliotecas o frameworks.

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 (Protocolo de Transferencia de Hipertexto) es el protocolo de comunicación utilizado para la transferencia de información que permite la comunicación entre clientes y servidores, o en palabras más sencillas es como un mensajero que lleva mensajes entre tu computadora a otra en internet. En relación con las APIs (Interfaz de Programación de Aplicaciones basada en Transferencia de Estado Representacional), HTTP actúa como el medio de comunicación para exponer servicios web mediante operaciones estándar como GET, POST, PUT y DELETE, identificadas por URI (Identificador Uniforme de Recursos) únicos asociados a cada recurso. La comunicación en HTTP es sin estado, lo que significa que cada solicitud es independiente. La representación de recursos en solicitudes o respuestas HTTP en una API REST transmite el estado representacional del recurso. Los datos se intercambian en formatos estándar como JSON o XML, facilitando la comprensión. En palabras simples, HTTP y API REST son reglas estándar que ayudan a las computadoras a hablar entre sí en internet y a compartir información de manera eficiente.

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

La dirección IP cumple un papel sumamente importante en el acceso a través de una API REST, ya que, Al realizar una solicitud a una API REST, nuestra dirección IP actúa como la identificación de origen de las solicitudes que realizamos. En términos de seguridad, se utilizan listas blancas y negras para permitir o bloquear solicitudes según direcciones IP específicas. La lista blanca asegura que solo dispositivos autorizados accedan, mientras que la lista negra bloquea accesos no autorizados o comportamientos maliciosos. Además, algunas APIs usan la dirección IP en la autenticación y autorización, asignando privilegios según la ubicación del cliente. En síntesis, la dirección IP es una herramienta clave para controlar el acceso a una API REST, asegurando la adecuada seguridad y autorización de las solicitudes.

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 red y el ancho de banda son esenciales para el rendimiento de una API. La latencia, relacionada con el tiempo de viaje de datos entre cliente y servidor, afecta directamente el tiempo de respuesta de la API. El ancho de banda, que determina la cantidad de datos transferibles en un período, es vital para una transmisión eficaz. Estos factores son clave para la escalabilidad, siendo cruciales con cargas de usuarios más altas. Una baja latencia y un ancho de banda adecuado son esenciales para evitar cuellos de botella. Además, una baja latencia mejora la experiencia del usuario, especialmente en aplicaciones interactivas o de transmisión de video en tiempo real. En términos de costos, un mayor ancho de banda puede generar gastos adicionales, especialmente en entornos en la nube. Optimizar su uso ayuda a controlar costos, contribuyendo al éxito general de la aplicación o servicio. En resumen, considerar la latencia y el ancho de banda es crucial para mejorar la eficiencia, velocidad y experiencia del usuario en el diseño de una API.

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

Cuando usas un programa que utiliza una API REST, lo rápido que funciona no depende solo de la API en sí, sino de cómo se implementa y del entorno en el que se ejecuta. Cosas como la velocidad de la conexión a internet, ciertos procesos de comunicación adicionales, la capacidad del servidor, el diseño de la API, posibles errores en la programación y si la API puede manejar muchas solicitudes a la vez pueden hacer que el programa sea más lento. Si la conexión a internet es lenta o si hay demasiada información innecesaria en las respuestas de la API, por ejemplo, puede afectar el rendimiento. Para hacer que el programa sea más rápido, se sugiere resolver estos problemas optimizando cómo se hacen las preguntas y se reciben las respuestas, usando almacenamiento en caché cuando sea posible y configurando tanto el programa como el servidor de manera eficiente.

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 dirección MAC y la dirección IP son identificadores en redes de computadoras con funciones diferentes y ubicación en capas distintas del modelo OSI. La Dirección MAC tiene la función de identificar de manera única dispositivos en una red local (LAN) según su interfaz de red y opera en la capa de enlace de datos (capa 2), facilitando la comunicación entre dispositivos conectados directamente. Además, está compuesta por 12 caracteres hexadecimales, a menudo presentados en pares separados por dos puntos. Y la Dirección IP permite la comunicación en redes extensas, como Internet, identificando un dispositivo y su ubicación en una red global, esta opera en la capa de red (capa 3), encargándose del enrutamiento de datos y la entrega de paquetes entre diferentes redes locales y su formato IPv4 utiliza cuatro conjuntos de números separados por puntos; IPv6 usa una notación hexadecimal más extensa.

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 una LAN, la latencia es baja debido a la proximidad física, mientras que en una WAN puede ser mayor debido a la distancia y la posibilidad de pasar por múltiples nodos. Las LAN tienden a tener un alto ancho de banda, garantizando transferencias rápidas, a diferencia de las WAN, que pueden tener limitaciones. La seguridad es mayor en LAN debido al control directo, mientras que en WAN, la exposición a amenazas puede afectar la velocidad. Las LAN son más fiables al ser controladas localmente, mientras que las WAN pueden experimentar interrupciones debido a su naturaleza distribuida. El ruteo de red es más directo en LAN, mientras que en WAN puede ser más complejo. Escalar una LAN es más fácil debido a la proximidad, mientras que escalar una WAN implica desafíos logísticos y técnicos. En resumen, las LAN ofrecen mejor velocidad y accesibilidad local, mientras que las WAN pueden tener mayores tiempos de respuesta. Se sugieren estrategias como optimización de consultas, uso eficiente de ancho de banda y técnicas de almacenamiento en caché para mejorar el rendimiento de una API REST en diferentes redes.

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, determinando la mejor ruta de origen a destino. Opera en la capa 3 del modelo OSI, utilizando direcciones IP. En el ámbito del software, el término se aplica al enrutamiento de solicitudes en aplicaciones web, como API REST. Aquí, el enrutador asigna solicitudes HTTP a controladores que manejan funciones específicas. En una API REST, el proceso implica definir rutas, mapearlas a controladores, recibir solicitudes, despacharlas al controlador correspondiente y procesarlas. La analogía con el enrutamiento de red radica en la toma de decisiones sobre la mejor ruta para transferir datos o procesar solicitudes.

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

El texto explica cómo se conectan puertos de red, servicios y aplicaciones usando TCP y UDP. Hay 65,535 puertos disponibles para cada protocolo, divididos en Bien Conocidos (0-1023), Registrados (1024-49151) y Dinámicos o Privados (49152-65535).

Los puertos Bien Conocidos son para servicios comunes como HTTP y FTP, los Registrados son para aplicaciones menos comunes, y los Dinámicos se usan temporalmente.

En la comunicación, los puertos guían los datos al servicio adecuado en el dispositivo de destino. Por ejemplo, HTTPS usa el puerto 443. La asignación de puertos puede variar, y algunos protocolos usan puertos TCP y UDP para funciones diferentes. La IANA tiene un registro de puertos y servicios en su sitio web.

3. Resultados

Resultados Mycol Z:

```
PS C:\Users\zinck\Downloads\para python> python OUILookup.py --ip 192.168.1.100
IP address: 192.168.1.100
MAC address: cc:46:d6:bc:a1:c7
Fabricante: Cisco Systems, Inc
Tiempo de respuesta: 0.9660487174987793 segundos
```

```
PS C:\Users\zinck\Downloads\para python> python OUILookup.py --mac 74:40:be:06:20:1a
MAC address: 74:40:be:06:20:1a
Fabricante: LG Innotek
Tiempo de respuesta: 0.8850350379943848 segundos
```

```
PS C:\Users\zinck\Downloads\para python> python OUILookup.py --arp

Interfaz: 10.145.213.172 --- 0x8
Dirección de Internet      Dirección física      Tipo
10.145.213.153             18-47-3d-53-ca-f3    din mico
10.145.255.254             cc-46-d6-bc-a1-c7    din mico
10.145.255.255             ff-ff-ff-ff-ff-ff    est tico
192.168.1.100              cc-46-d6-bc-a1-c7    din mico
224.0.0.2                  01-00-5e-00-00-02    est tico
224.0.0.22                 01-00-5e-00-00-16    est tico
224.0.0.251                01-00-5e-00-00-fb    est tico
224.0.0.252                01-00-5e-00-00-fc    est tico
239.255.255.250            01-00-5e-7f-ff-fa    est tico
255.255.255.255            ff-ff-ff-ff-ff-ff    est tico
```



```
PS C:\Users\zinck\Downloads\para python> python OUILookup.py --help
Uso: python OUILookup.py --ip <IP> --mac <MAC> --arp [--help]
--ip: IP del host a consultar.
--mac: MAC address a consultar.
--arp: Muestra la tabla ARP.
```

Resultados Diego:

```
PS C:\Users\W10\Desktop\OUILookup2> python OUILookup2.py --ip 192.168.0.7
IP address: 192.168.0.7
MAC address: 38:6b:1c:d0:1c:12
Fabricante: SHENZHEN MERCURY COMMUNICATION TECHNOLOGIES CO.,LTD.
Tiempo de respuesta: 0.9700169563293457 segundos
PS C:\Users\W10\Desktop\OUILookup2>
```

```
PS C:\Users\W10\Desktop\OUILookup2> python OUILookup2.py --mac 74:40:be:06:20:1a
MAC address: 74:40:be:06:20:1a
Fabricante: LG Innotek
Tiempo de respuesta: 0.777216911315918 segundos
PS C:\Users\W10\Desktop\OUILookup2>
```

```
Tiempo de respuesta: 0.777216911315918 segundos
PS C:\Users\W10\Desktop\OUILookup2> python OUILookup2.py --arp

Interfaz: 26.186.45.127 --- 0x7
Dirección de Internet      Dirección física      Tipo
26.0.0.1                   02-00-00-00-51-00    din mico
26.255.255.255             ff-ff-ff-ff-ff-ff    est tico
224.0.0.22                 01-00-5e-00-00-16    est tico
224.0.0.250                01-00-5e-00-00-fa    est tico
224.0.0.251                01-00-5e-00-00-fb    est tico
224.0.0.252                01-00-5e-00-00-fc    est tico
239.255.255.250            01-00-5e-7f-ff-fa    est tico
239.255.255.251            01-00-5e-7f-ff-fb    est tico
```

Resultados Damian:

4. Discusión y conclusiones.

5. Referencias.

<https://docs.python.org>

<https://www.youtube.com/watch?v=5RQY5Co3kbQ>

<https://www.youtube.com/watch?v=DmHBBBrLealc>

