Tarea 02: Consultar Fabricantes de tarjetas de red

Ricardo Silva Pimentel, [ricardo.silvap@alumnos.uv.cl](mailto:ricardo.silvap@alumnos.uv.cl)

Joaquín Martínez, [joaquin.martinez@alumnos.uv.cl](mailto:%6a%6f%61qu%69%6e.%6d%61%72%74%69%6e%65z@a%6cu%6d%6e%6fs%2e%75%76%2ec%6c)

Alonso Vera, [alonso.vera@alumnos.uv.cl](mailto:a%6con%73%6f%2e%76%65%72%61@%61%6c%75m%6e%6f%73.%75%76%2ec%6c)

# Introducción

La identificación de fabricantes a partir de direcciones MAC es una práctica común en la administración de redes, ya que permite a los administradores conocer la procedencia de los dispositivos conectados. Esta tarea tiene como objetivo desarrollar una herramienta de línea de comandos en Python llamada **OUILookup**, que permite consultar el fabricante de una tarjeta de red a partir de su dirección MAC utilizando una API pública. La importancia de esta herramienta radica en su capacidad para simplificar la gestión de redes y proporcionar información valiosa para la identificación de dispositivos. En este informe, se presentará el problema que se busca resolver, el diseño y la implementación de la solución, así como los resultados de las pruebas realizadas y las conclusiones obtenidas.

# Descripción del problema y diseño de la solución

La tarea consiste en desarrollar una herramienta de línea de comandos llamada **OUILookup** que permita a los usuarios consultar el fabricante de una tarjeta de red a partir de su dirección MAC. Cada dispositivo de red está identificado por una dirección MAC única, pero esta dirección no proporciona información directamente accesible sobre el fabricante del dispositivo.

El desafío radica en implementar un codigo que permita a los usuarios ingresar una dirección MAC y recibir rápidamente información sobre el fabricante, utilizando para ello una API REST pública.

Los aspectos específicos del problema incluyen:

1. **Acceso a datos externos**: La herramienta debe interactuar con una API pública, lo que implica gestionar solicitudes HTTP y manejar posibles errores de conexión o respuestas inesperadas.
2. **Interfaz de línea de comandos**: La herramienta debe ser fácil de usar, permitiendo a los usuarios introducir direcciones MAC y recibir información sin complicaciones.
3. **Manejo de errores y respuestas**: Es fundamental que la herramienta maneje adecuadamente casos en los que la dirección MAC no se encuentra en la base de datos de la API, mostrando un mensaje claro al usuario.
4. **Eficiencia en la consulta**: La implementación debe ser lo suficientemente rápida para proporcionar respuestas en tiempo real, minimizando el tiempo de espera para el usuario.

**Diseño de la solución**

*Diagrama

Descripción generada automáticamente*La solución propuesta se basa en una arquitectura sencilla que integra una interfaz de línea de comandos con una API REST pública. El diagrama de clases a continuación ilustra la estructura básica del sistema:

**Estructura principal del código**

El programa se estructura principalmente en dos funciones clave: una encargada de realizar la consulta a la API y otra responsable de gestionar los argumentos proporcionados por el usuario en la línea de comandos. La combinación de ambas funciones permite que el software sea flexible y útil en la práctica, facilitando al usuario tanto la consulta de fabricantes como la navegación por las opciones del programa.

**Consultar\_fabricante\_mac():** Esta función es el núcleo del programa, ya que realiza la tarea principal: consultar a la API para obtener el fabricante de una dirección MAC. Sin esta función, el programa no cumpliría su propósito.

Texto

Descripción generada automáticamente

**Figura 1.** Estructura del “Consultar\_fabricante\_mac()”.

**Main():** Esta función organiza todo el flujo del programa. Maneja los argumentos de la línea de comandos y decide cuándo ejecutar la consulta o mostrar ayuda al usuario. Actúa como el controlador general del programa.

Texto

Descripción generada automáticamente

**Figura 2.** Estructura del “Main()”.

**Direcciones MAC Aleatorias**

Las **direcciones MAC aleatorias** son una característica implementada en dispositivos electrónicos para mejorar la privacidad y seguridad del usuario, especialmente en redes Wi-Fi. En lugar de utilizar la dirección MAC única y fija que se encuentra en el hardware del dispositivo, muchos sistemas operativos ahora permiten que los dispositivos generen direcciones MAC temporales al conectarse a redes [2].

* **Funcionamiento**: Esta tecnología permite a los dispositivos crear una nueva dirección MAC cada vez que se conectan a una red Wi-Fi, lo que dificulta el rastreo del dispositivo en diferentes redes. Según Apple, "el uso de direcciones MAC aleatorias ayuda a proteger la privacidad del usuario al ocultar su dirección MAC real" [1].
* **Beneficios de Privacidad**: La aleatorización de las direcciones MAC previene que terceros, como administradores de red o atacantes, rastreen el comportamiento del usuario a través de redes. Esto es especialmente útil en entornos públicos donde la vigilancia es más probable [2].
* **Implementación**: La funcionalidad está disponible en la mayoría de los sistemas operativos modernos, incluyendo **iOS** y **Android**. Los usuarios pueden habilitar o deshabilitar esta opción en la configuración de red de sus dispositivos, aunque en redes empresariales, esta funcionalidad puede interferir con la autenticación basada en MAC [1][2].
* **Consideraciones**: Aunque las direcciones MAC aleatorias mejoran la privacidad, pueden causar problemas en redes que dependen de la autenticación por MAC, lo que puede requerir ajustes adicionales en la configuración de red para permitir el acceso a dispositivos que utilizan direcciones aleatorias [2][3].

# Implementación

La implementación de OUILookup se realizó utilizando Python, debido a su simplicidad y la disponibilidad de bibliotecas útiles como requests para realizar solicitudes HTTP. A continuación, se describen las partes más importantes del código:

**Instalación de Dependencias**

Para ejecutar la herramienta, es necesario instalar la biblioteca requests. Esto se puede en la **Figura 1**:

Texto

Descripción generada automáticamente

**Figura 3.** Esta es una figura. El título debe estar centrado

**Uso de la Herramienta**

Una vez que se ha instalado la biblioteca, se puede utilizar la herramienta **OUILookup** a través de la línea de comandos. A continuación, se presentan algunos ejemplos de uso:

* Para consultar el fabricante de una dirección MAC específica:

Texto

Descripción generada automáticamente

**Figura 4.** La figura muestra la consulta de una MAC especifica.

* Para mostrar los fabricantes de los hosts disponibles en la tabla ARP:

Texto

Descripción generada automáticamente

**Figura 5.** Ya que se consulta a la API no es necesaria la tabla ARP.

* Para mostrar la ayuda y conocer las opciones disponibles:

Texto

Descripción generada automáticamente

**Figura 6.** Muestra comando para mostrar los parámetros del comando –help.

# Pruebas

Las pruebas del código se realizaron para garantizar que todas las funcionalidades trabajaran como se esperaba. Se definieron varios casos de prueba, incluyendo:

**Caso de éxito**: Consultar una dirección MAC válida.

**Resultado esperado**: Debería retornar el nombre del fabricante.

**Texto

Descripción generada automáticamente**

**Figura 7.** Muestra el resultado esperado para una MAC valida.

**Caso de no encontrado**: Consultar una dirección MAC que no exista en la BD.

**Resultado esperado**: Debería indicar que el fabricante no fue encontrado.

**Prueba 2**:

Texto

Descripción generada automáticamente

**Figura 8.** Muestra el resultado esperado para una MAC invalida.

También se realizaron pruebas para los macs que serán revisados en la tarea:

**Texto

Descripción generada automáticamente**

**Figura 9.** Respuesta de los 3 MAC a evaluar.

# Discusión y conclusiones

Los resultados obtenidos demostraron que **OUILookup** es capaz de consultar correctamente la información del fabricante a partir de direcciones MAC válidas y manejar adecuadamente los errores cuando las direcciones no se encuentran. Se aprendió la importancia de implementar un manejo de errores efectivo para mejorar la experiencia del usuario. Sin embargo, una limitación es que la herramienta depende de la disponibilidad y precisión de la API externa utilizada.

# Referencias

[1] https://support.apple.com/es-cl/102509

[2] <https://www.redeszone.net/noticias/wifi/direccion-mac-aleatoria-wifi/>

[3] https://telefonicatech.com/blog/mac-aleaotorias-y-privacidad-parte2

[4] https://maclookup.app/api-v2/documentation