Informe del Trabajo Práctico N°1: Análisis de Tramas IEEE 802.15.4

# Introducción

Este trabajo práctico tiene como objetivo analizar tramas de comunicación según el estándar IEEE 802.15.4. Se implementó un programa en Python que verifica la validez de las tramas, considerando longitud correcta, checksum válido y la presencia de secuencias de escape.

# Descripción del Algoritmo

El programa sigue los siguientes pasos para procesar las tramas:

* **Leemos el archivo:** Leemos el archivo “tramas\_802-15-4.log” que contiene toda la información con las tramas y la guardamos en una variable para poder ir usando los datos
* **Identificación de tramas:** Se detectan las tramas mediante el delimitador de inicio 0x7E. Siempre que se encuentre un delimitador de inicio 0x7E se realizan las

verificaciones necesarias para saber si es parte del mensaje o no, buscando el correspondiente 0x7D.

* **Manejo de secuencias de escape:** Cuando se encuentra 0x7D, se analiza el byte anterior para corregir los datos escapados. Se eliminan las secuencias de escape y se registra la posición de estas en la trama.
* **Validación de longitud:** Se verifica que la longitud especificada en la trama coincida con el tamaño real de los datos útiles y se guarda en una lista. Para las tramas con longitud incorrecta se las guarda en una lista aparte.
* **Verificación del checksum:** Se calcula el checksum de la trama para asegurar la integridad de los datos y se compara con el valor proporcionado. Las tramas con checksum incorrecto se guardan en una lista.
* **Resultado:** Muestra el resultado de cantidad de tramas totales, tramas de longitud correcta, tramas con longitud incorrecta, tramas con longitud correcta y checksum correcto, tramas con longitud correcta y checksum incorrecto y tramas con secuencia de escape.

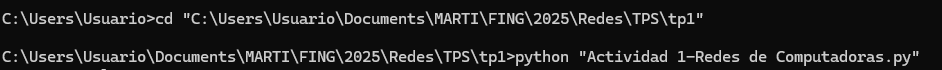
# Ejecución del Algoritmo:

A continuación, se mostrarán los pasos a seguir para la ejecución del archivo en dos sistemas operativos: Windows y Linux

*Windows:*

1. Abrir la consola de Windows (CMD)
2. Presionar *Win + R*, escribir *cmd* y presionar *Enter*.
3. Ubicar el archivo .py (Asegurarse de conocer la ruta completa del archivo Python.)
4. Elegir el directorio donde se encuentra el archivo.
5. Ejecutar cd “ruta de acceso”
6. Ejemplo: *cd "C:\Users\Usuario\Documents\MARTI\FING\2025\Redes\TPS\tp1*"
7. Ejecutar el archivo a través del comando Python
8. Ejemplo: *python "Actividad 1-Redes de Computadoras.py"*
9. Si el script necesita un archivo como *"Tramas\_802-15-4.log"* y no lo encuentra, verificar que se encuentre en la misma carpeta que mi archivo .py

Ejemplo de comandos en consola:



*Linux:*

Para ejecutar un archivo .py en Linux desde la consola, sigue estos pasos:

1. Abre la terminal.
2. Navega al directorio donde se encuentra tu archivo .py usando el comando cd:  
   cd /ruta/del/directorio
3. Ejecuta el archivo Python usando el comando python3 (o python, dependiendo de cómo esté configurado tu sistema): python3 nombre\_del\_archivo.py
4. Si estás usando una versión de Python 2, entonces usarías:

python nombre\_del\_archivo.

# Resultados del Algoritmo

Estos son los resultados obtenidos luego de ejecutar el programa:

