ANÁLISIS DESAFIO 1

Sergio Andres Chaves Roa, Oscar Alexander Sepulveda

***Resumen*—En este trabajo se desarrolló una solución al desafío propuesto en la asignatura de Informática II, donde se planteó un problema de ingeniería inversa para recuperar un mensaje original a partir de su versión comprimida y encriptada. La práctica incluyó la implementación de algoritmos de desencriptación basados en rotación de bits y XOR, así como algoritmos de descompresión RLE y LZ78. Además, se utilizó un fragmento del mensaje original como pista para identificar los parámetros de cifrado y el método de compresión.**

***Palabras Claves*— Compresión, Encriptación, RLE, LZ78, XOR, Rotación de Bits, Ingeniería Inversa**

I. Objetivo General

Empleando los conocimientos adquiridos en el curso, desarrollar un programa en C++ que permita identificar el método de compresión, desencriptar y descomprimir un mensaje dado, validando los resultados mediante un fragmento conocido del mensaje original.

II. Objetivos específicos

Empleando la lógica de programación, implementar los algoritmos de desencriptación mediante rotación de bits y operación XOR con clave de un byte.
Empleando los conceptos estudiados, diseñar e implementar algoritmos de descompresión para los métodos RLE y LZ78 en C++.
Empleando un enfoque de validación, utilizar un fragmento conocido del mensaje original para determinar los parámetros de encriptación y el método de compresión aplicado.
Empleando buenas prácticas de programación, asegurar el uso eficiente de memoria mediante punteros, arreglos y memoria dinámica, evitando el uso de estructuras STL o tipo string.

III. marco teórico

La compresión de datos es un proceso mediante el cual se reduce el tamaño de un mensaje representado de manera más compacta. Entre los algoritmos más usados se encuentran RLE (Run Length Encoding), que codifica secuencias repetidas con pares (longitud, símbolo), y LZ78, que construye un diccionario dinámico de subcadenas para reducir redundancia [1].

La encriptación aplicada en el desafío combina dos operaciones a nivel de bits: la rotación de bits, que desplaza los bits de un byte circularmente, y la operación XOR, que modifica el valor de cada byte mediante una clave de un solo byte. Ambas operaciones son reversibles, lo que permite recuperar el mensaje original al aplicar las operaciones inversas [2].

El uso del código ASCII como base para el procesamiento facilita la manipulación de los datos, ya que cada carácter del archivo puede representarse como un byte entero (0–255), permitiendo aplicar directamente las operaciones de rotación y XOR antes de realizar la descompresión.

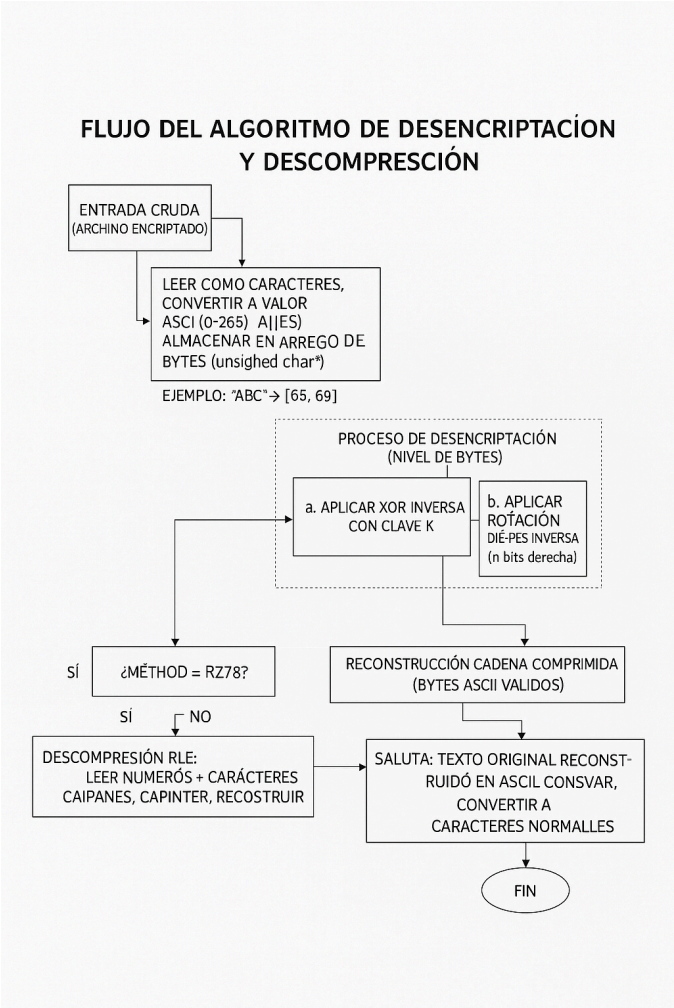

Fig. 1. Diagrama de Flujo análisis del desafío..

IV. Conclusión

El análisis de la solución propuesta para el problema de ingeniería inversa resalta la relevancia de un enfoque sistemático. La elección del código ASCII como fundamento para las operaciones se anticipa como un facilitador clave para la manipulación de datos a nivel de bits, prometiendo un proceso de desencriptación robusto. Se proyecta que la validación a través de un fragmento del mensaje original será una estrategia eficaz para la detección de métodos de compresión y la determinación de parámetros de encriptación. Esta aproximación subraya la importancia de considerar operaciones a nivel de bits, técnicas de compresión y métodos de encriptación como pilares fundamentales en el diseño de algoritmos de este tipo.