**Informe del Desafío 1**

Edwin Piedrahita Piedrahita

Facultad de ingeniería, Universidad de Antioquia, Informática 2

Prof. Augusto Enrique Salazar Jiménez

19 de septiembre de 2025

* **Análisis del desafío**

Como primero se tienen algunas entradas conocidas, como un mensaje comprimido y encriptado junto con un fragmento del texto original, ya por la parte que no se conoce está el método de compresión que puede ser por **RLE** o por **LZ78,** también se desconocen los parámetros de encriptación **(n)** que es el valor de la **rotación de bits**, y **(k)** que es el valor de la clave usada para la operación **XOR.**

Como segundo, se puede notar que para hallar los valores y las formas distintas en las que se puede recuperar el texto original son demasiadas, ya que hay dos formas distintas en las que se pudo usar la compresión, siete formas en las que se puede aplicar la rotación de bits, y 256 maneras de aplicar la clave correcta para realizar el XOR, lo cual nos daría una cantidad de 2 x 7 x 256 = 3584 combinaciones posibles para recuperar el texto original, lo cual dificulta encontrar y hallar los diferentes valores y combinaciones usadas para la compresión y encriptación.

También se tienen algunos desafíos como lo son las restricciones del lenguaje, como el no usar la STL, estructuras ni string, además de que se debe usar obligatoriamente la memoria dinámica, por otra parte, la complejidad de los algoritmos y operaciones a nivel de bits.

Base a lo dicho anteriormente lo que se tiene como objetivo es recuperar el mensaje original aplicando las operaciones en el orden correcto y con los valores correctos, en otras palabras, lo que se debe hacer es aplicar ingeniería inversa a los procesos realizados anteriormente para la compresión y encriptación del texto original.

* **Diseño de la solución**

La idea sería primero implementar una función que detecte cuales son los parámetros, como quien es **(k),** quien es **(n),** y cuál es el método que se uso para la descompresión, lo cual ayudara bastante para saber que combinaciones se usaron para la compresión y encriptación del texto original, para lo cual se usara él fragmento de texto original y así poder saber cuál combinación fue la que dio éxito.

Como segundo sería agregar una función que permita desencriptar el mensaje, en la cual se usara la operación **XOR** con su respectiva clave **(k),** además de la rotación de bits a la derecha una cantidad de **(n)** posiciones, ya que para la encriptación se uso la rotación de bits a la izquierda, las cuales se pueden realizar en funciones separadas.

Como tercero se realizará una función de descompresión, la cual va permitir obtener el mensaje original, acá también se pueden crear dos funciones, una para la descompresión por **RLE** y otra por **LZ78,** para que la función de descompresión sea más simple y fácil de entender**.**

Por último, se implementará una función, la cual manejará todo esto de forma más directa, la cual nos dará el mensaje encriptado, y el fragmento de texto original, generando mayor optimización para realizar estas operaciones, y así disminuir en gran cantidad el uso de la memoria.