# **Desafío 1 — Reconstrucción de mensaje comprimido y encriptado**

**Curso:** Informática II  
 **Autores:** Carlos Mario Rendón Martínez y Michell  
 **Profesor:** Aníbal Guerra / Augusto Salazar  
 **Fecha:** 19 de septiembre de 2025

1. Contextualización

El desafío consiste en reconstruir un mensaje original que fue comprimido (usando RLE o LZ78) y luego encriptado (con rotación de bits y XOR). El único dato que tenemos es el mensaje final encriptado y un fragmento conocido del texto original.

La tarea es identificar qué método de compresión se usó, descubrir los parámetros de encriptación (rotación n y clave K), desencriptar el mensaje y finalmente descomprimirlo para recuperar el texto original.

2. Análisis del problema

Lo más difícil del reto es que no nos dicen:

* Qué compresión se usó (RLE o LZ78).
* Cuál es la cantidad de bits rotados.
* Cuál es la clave de la operación XOR.

Esto significa que primero debemos probar combinaciones hasta encontrar la correcta que revele el fragmento conocido.

Además, el profesor pone restricciones:

* Usar C++ con Qt.
* No se permite usar string ni STL.
* Usar punteros, arreglos y memoria dinámica.

3. Diseño inicial de la solución

De manera general, pensamos organizar la solución en tres bloques de funciones:

-Encriptación/Desencriptación

Archivos: encriptacion.cpp y encriptacion.h  
 Funciones principales:

* Aplicar rotación de bits (izquierda/derecha).
* Aplicar y revertir la operación XOR.

-Compresión/Descompresión

Archivos: compresion.cpp y compresion.h  
 Funciones principales:

* Implementación de RLE (básica).
* Implementación de LZ78 (usando un diccionario dinámico con punteros y arreglos).

-Programa principal

Archivo: main.cpp  
 Responsabilidades:

* Recibir el mensaje encriptado y el fragmento conocido.
* Probar combinaciones de n (1..7) y K (0..255).
* Llamar a las funciones de encriptación/desencriptación.
* Llamar a las funciones de compresión/descompresión.
* Verificar si el fragmento conocido aparece en el resultado.