

Informe Preliminar – Desafío 1

Curso: Informática II – Semestre 2025-2

Fecha: septiembre 19, 2025

Estudiante: Javier Ricardo Bernal Julio

1. Contextualización

El reto consiste en **reconstruir un mensaje original en texto plano** que ha pasado por dos procesos:

1. **Compresión**, utilizando alguno de los métodos: RLE o LZ78.
2. **Encriptación**, aplicando rotación de bits y una operación XOR con una clave.

Solo se dispone del mensaje final (comprimido y encriptado) y de un **fragmento conocido** del mensaje original.

El desafío busca poner a prueba la capacidad de análisis, deducción y programación en C++, aplicando estructuras de control, punteros, arreglos y memoria dinámica.

2. Análisis

- **Entradas del sistema:**
 - Mensaje comprimido y encriptado (cadena de caracteres).
 - Fragmento conocido del mensaje original.
- **Salidas del sistema:**
 - Mensaje original completo.
 - Método de compresión identificado (RLE o LZ78).
 - Parámetros de encriptación descubiertos (valor de rotación n y clave XOR K).
- **Restricciones técnicas:**
 - Implementación en C++ con framework Qt.
 - Sin uso de string ni librerías STL.
 - Obligatorio usar punteros, arreglos y memoria dinámica.
 - Eficiencia en el manejo de memoria.
- **Dificultades esperadas:**
 - Identificar correctamente el método de compresión.
 - Diseñar un proceso de búsqueda para encontrar n y K .

- Reconstruir el mensaje original asegurando que el fragmento conocido coincida.

3. Diseño

3.1 Arquitectura General de la Solución

1. Módulo de Descriptación

- Aplica XOR inverso con la clave K.
- Rota los bits en dirección contraria a n.

2. Módulo de Identificación de Compresión

- Prueba primero con RLE.
- Prueba luego con LZ78.
- Verifica si el resultado contiene el fragmento conocido → determina el método correcto.

3. Módulo de Descompresión

- Si es RLE: expandir pares (longitud, símbolo).
- Si es LZ78: reconstruir diccionario dinámico a partir de pares (índice, carácter).

4. Módulo de Validación

- Comprueba que el mensaje final contiene el fragmento conocido.
- Retorna el mensaje original completo.

3.2 Estructuras y Algoritmos

- Rotación de bits: operaciones con desplazamientos y máscaras.
- XOR: operación bit a bit reversible.
- RLE: uso de contadores de repeticiones y almacenamiento temporal en arreglos.
- LZ78: diccionario en memoria dinámica con punteros y arreglos.

4. Conclusión Preliminar

La solución se basa en separar el problema en tres etapas (desencriptar → detectar método → descomprimir). Esta modularidad facilita el análisis, la depuración y la implementación en C++ usando estructuras de bajo nivel. El diseño asegura que se pueda probar cada componente de manera independiente antes de integrarlos.