# Evidencia del Proceso de Análisis y Diseño de la Solución

1. **Análisis del Problema**

**1.1. Descripción General**

El problema consiste en realizar ingeniería inversa sobre un mensaje que ha sido sometido a un proceso de compresión (RLE o LZ78) y posterior encriptación (rotación de bits + XOR). Se conoce un fragmento del texto original que servirá como referencia para determinar los parámetros y algoritmos utilizados.

* 1. **Restricciones Técnicas**
* Implementación en C++ con framework Qt
* Prohibido uso de STL (no string, vector, map, etc.)
* Uso obligatorio de punteros, arreglos y memoria dinámica
* Solo caracteres A-Z, a-z, 0-9 (sin tildes, ni Ñ, ni puntuación)
* Gestión manual de memoria

**1.3. Desafíos Principales**

* Implementar algoritmos de compresión sin bibliotecas estándar
* Realizar ingeniería inversa para descubrir parámetros de encriptación
* Gestionar memoria manualmente sin fugas
* Crear estructuras de datos personalizadas

**2. Alternativa de Solución Propuesta**

**2.1. Enfoque General**

Se implementará un sistema modular que:

* Realice fuerza bruta controlada para descubrir parámetros
* Implemente ambos algoritmos de compresión
* Proporcione utilidades para manipulación de cadenas
* Gestione memoria de forma segura

**2.1. Diagrama de flujo**

Entrada: Mensaje Encriptado

**2.3. Algoritmos a Implementar**

Si

No

Resultado valido

Si

No

Fuerza Bruta

Si

No

Error: No se pudo descomprimir

Mensaje Original  
Recuperado

Intentar Decompresion   
LZ78

Resultado Valido

Intentar descompresión   
RLE

Desencriptar el mensaje

Parámetros encontrados

Coincide con la entrada

**2.3.1. Encriptación/Desencriptación**

* Rotación de bits izquierda/derecha
* Operación XOR con clave de 8 bits
* Combinación: Rotación → XOR

**2.3.2. Compresión RLE**

* Codificación: Secuencias repetidas → (count)(char)
* Decodificación: (count)(char) → Secuencia repetida

**2.3.4. Ingeniería Inversa**

* Fuerza bruta sobre n (1-7) y K (0-255)
* Verificación contra fragmento conocido
* Validación de resultados