**Análisis Desafío 1.**

**Elaborado por:**

* Maria Alejandra Gutierrez Rengifo
* Juan Camilo Perez

**Asignatura:** informática 2

**Docente:** Augusto Enrique Salazar Giménez

**Universidad de Antioquia**

**Facultad de Ingeniería**

**Programa de Ingeniería de Telecomunicaciones Medellín, 2025-1**

**1. Análisis del Problema**

El desafío consiste en recuperar una imagen original que ha sido alterada mediante una serie de transformaciones aplicadas a nivel de bits. Estas transformaciones incluyen operaciones como XOR, rotaciones y desplazamientos, aplicadas en un orden que no se conoce. Además, luego de cada transformación, se realizó un proceso de enmascaramiento con una máscara de color y un desplazamiento, cuyos resultados se almacenaron en archivos .txt.

El objetivo es identificar el orden correcto en que se aplicaron estas transformaciones, validar dicha secuencia utilizando los archivos de rastreo y, finalmente, revertir cada paso para reconstruir la imagen original.

**2. Diseño de la Solución**

Para abordar este problema, se propone una solución dividida en cuatro etapas principales:

a. Lectura de datos

Se cargan todos los archivos necesarios:

* La imagen distorsionada (ID.bmp)
* La imagen de modificación (IM.bmp), que pudo haberse usado en operaciones XOR
* La máscara de color (M.bmp)
* Los archivos .txt con los resultados intermedios del enmascaramiento y sus respectivas semillas

Estos elementos son fundamentales para poder verificar si una transformación se aplicó correctamente en cada paso.

b. Implementación de transformaciones

* Se desarrollarán funciones para simular las operaciones que pudieron aplicarse sobre la imagen original:
* XOR entre dos imágenes
* Rotaciones de bits (a izquierda y derecha)
* Desplazamientos de bits (a izquierda y derecha)

Estas funciones se implementarán utilizando únicamente arreglos y punteros, como lo establece el enunciado del reto.

c. Prueba de combinaciones

Debido a que no se conoce el orden de las transformaciones, se deben probar diferentes secuencias. Por cada combinación:

* Se aplica la transformación propuesta
* Se realiza el enmascaramiento usando la máscara y el desplazamiento indicado
* Se comparan los resultados con los datos del archivo .txt
* Si coinciden, se considera válida esa transformación en esa etapa.

d. Reconstrucción de la imagen

Una vez se identifica el orden correcto de las transformaciones, se aplican en sentido inverso para revertir sus efectos. Al final del proceso, se espera haber recuperado la imagen original (IO.bmp).

**3. Diseño Inicial**

Antes de empezar a programar, se definieron algunos principios básicos para organizar el trabajo:

* Dividir el proceso en módulos claros: lectura, transformación, validación y reconstrucción
* Realizar pruebas iniciales con imágenes pequeñas para facilitar la depuración
* Registrar los pasos que funcionen correctamente para evitar repetir procesos
* Mantener la solución lo más simple posible, sin estructuras complejas
* Seguir una lógica ordenada: primero identificar, luego revertir
* Este diseño no es definitivo, pero sirvió como una hoja de ruta para comenzar el desarrollo de forma organizada.

**4. Esquema del Algoritmo**

* Cargar las imágenes y archivos de rastreo
* Implementar las funciones necesarias para las transformaciones
* Probar distintas combinaciones de transformaciones y validarlas con los archivos .txt
* Registrar el orden correcto cuando haya coincidencia
* Revertir las transformaciones aplicándolas en orden inverso
* Exportar la imagen reconstruida