Desafío 1 - Decodificación Imágenes BMP

Estudiantes:

Esteban Rodriguez Monsalve, [esteban.rodriguez3@udea.edu.co](mailto:esteban.rodriguez3@udea.edu.co)

Alejandro Bedoya Zuluaga, [alejandro.bedoyaz@udea.edu.co](mailto:alejandro.bedoyaz@udea.edu.co)

Profesor:

Aníbal Jose Guerra Soler

Informática II

12 de abril de 2025

**Diseño y análisis desafío #1-ingeniería inversa con mapas de bit**

**Introducción**

El desafío 1, cuyo objetivo principal es aplicar nuestros conocimientos en programación en C++ para resolver un problema complejo de ingeniería inversa relacionado con imágenes digitales. El desafío consiste en reconstruir una imagen original en formato BMP que ha sido alterada mediante una serie de transformaciones a nivel de bits —incluyendo operaciones como XOR, rotaciones y desplazamientos— combinadas con enmascaramientos basados en una máscara de color y desplazamientos aleatorios.

El proceso de transformación de la imagen se realizó en un orden desconocido, lo que representa un reto adicional, pues además de revertir cada operación correctamente, se debe deducir el orden exacto en que ocurrieron. Como única pista, se nos proporcionan los resultados de los enmascaramientos intermedios en archivos de texto, junto con la imagen final transformada, una imagen auxiliar (potencialmente usada para los XOR) y la máscara de color.

En este informe se presenta el análisis del problema, la estrategia de solución propuesta, la implementación de los algoritmos correspondientes y los resultados obtenidos. La solución propuesta busca no solo restaurar la imagen original, sino también servir como ejercicio práctico de razonamiento lógico, manipulación de estructuras de datos dinámicas, y aplicación avanzada de operaciones bit a bit en lenguaje C++, todo dentro del entorno de desarrollo Qt.

# **Planteamiento de la solución**

la estrategia principal consiste en revertir el proceso de transformación, identificando las operaciones que se aplicaron (XOR , rotaciones y desplazamiento), su orden y el lugar en el que se utilizó el enmascaramiento, para ello utilizaremos los archivos .txt proporcionados, los cuales contienen los resultados intermedios entre cada enmascaramiento, con esto validamos que las operaciones son las correctas.

1 - Implementación de operaciones bit a bit: vamos a implementar funciones propias en C++ para realizar las operaciones destacadas sobre los datos de la imagen BMP, utilizaremos arreglos dinámicos y punteros.

2 - Algoritmo de verificación de desenmascaramiento: A partir de la imagen modificada y la máscara proporcionada, se diseñará un mecanismo con la intención de simular un desenmascaramiento y comparar los resultados con los dados en los mapeos de validación, esto con la intención de validar la transformación y que sí coincida con la etapa de desencriptación respectiva.

3 - Deducción del orden de operaciones:Se planea utilizar un enfoque de búsqueda sistemática, por ejemplo el método búsqueda por prueba y error haciendo uso del feedback, donde se empieza desde un puntos intermedio, o aleatorio para verificar los cambios aplicando una operación sobre un conjunto pequeño de datos (sobre los primeros 100 arreglos), y verificamos si la coincidencia con el resultado es alta, haciendo uso el dataset proporcionado; según el nivel de coincidencia se ajusta la dirección del intento (por ejemplo, probar más adelante o más atrás en la imagen) una vez se tenga una coincidencia total en los primeros 100 arreglos procedemos a comparar los siguientes 100 arreglos, esto hasta completar todo el mapeo, para asegurarnos de que la coincidencia sea total, en caso contrario de que la coincidencia no sea la esperada en los primero 100 arreglos, podemos asegurar que la combinación no es la que buscamos por lo que proseguimos a buscar una combinación diferente.

4 - Reconstrucción de las imágenes en cada paso: Una vez identificadas el orden de las operaciones, aplicaremos estas sobre las imágenes transformadas, lo cual nos permitirá gradualmente ir recuperando la imagen original, esto lo haremos con el código proporcionado en el dataset.

# 5 - Exportación y validación visual : Finalmente, se generará una imagen de salida a partir de los datos reconstruidos y se validará visualmente y numéricamente su correspondencia con la imagen original esperada (cuando esté disponible).

Asi que estos son los pasos que nosotros tenemos en consideración .