Informe de Análisis y Diseño
Desafío 1 - Informática II
Semestre 2025-1

Benjamin Ruiz Guarin
Kevin Jimenez Rincón

12 de abril de 2025

1. Análisis del problema

El desafío plantea un problema de ingeniería inversa donde se debe recuperar una imagen original que ha sido distorsionada mediante una serie de transformaciones bit a bit (como rotaciones, desplazamientos y operaciones XOR), seguidas por operaciones de enmascaramiento con una máscara de color.

No se conoce el orden de las transformaciones. Solo se dispone de:

- La imagen final distorsionada (ID).
- Una imagen aleatoria (IM) que puede haber sido usada en operaciones XOR.
- Una máscara M (más pequeña que la imagen original).
- Archivos `.txt` con desplazamientos y valores enmascarados.

El reto es descubrir el orden de las transformaciones, revertirlas, y recuperar la imagen original.

2. Diseño de la solución

El enfoque propuesto se basa en:

1. Leer la imagen distorsionada (ID), la IM y la máscara M.
2. Leer los archivos de enmascaramiento (M1.txt, M2.txt, ...), para extraer el desplazamiento y las sumas RGB.
3. Implementar funciones para las transformaciones bit a bit:
   - Operación XOR: `resultado[i] = img1[i] ^ img2[i]`
   - Rotación de bits (izquierda/derecha)
   - Desplazamiento de bits

4. Verificar cada paso aplicando el enmascaramiento y comparando los resultados con los archivos .txt.
5. Probar diferentes órdenes de transformaciones para encontrar la secuencia correcta.
6. Una vez hallado el orden, aplicar las transformaciones inversas para obtener la imagen original.

3. Esquema de tareas

- </> Leer imágenes BMP y convertir a arreglo RGB (punteros).
- </> Leer archivos `.txt` y extraer datos.
- </> Implementar operaciones bit a bit (XOR, rotación, desplazamiento).
- </> Simular el enmascaramiento y verificar resultados.
- </> Implementar lógica para deducir el orden de transformaciones.
- </> Invertir las transformaciones para reconstruir la imagen.
- </> Documentar el código y preparar presentación.

4. Desarrollo del trabajo

Dado el nivel de complejidad del problema, es fundamental adoptar una estrategia altamente organizada. Al tratarse de un trabajo colaborativo, se optó por un enfoque modular, en el que cada operación y tarea se implementa como una función independiente. Esta estructura permite trabajar de forma paralela y distribuir responsabilidades.

La idea es desarrollar primero todas las funciones necesarias —lectura de imágenes, operaciones bit a bit, enmascaramiento, verificación, etc.— de manera aislada y probada. Una vez que estas piezas estén completas y validadas, se integrarán en la lógica central que deduce el orden correcto de transformaciones aplicadas a la imagen final.

Además, cada módulo será acompañado de funciones de prueba unitarias para validar su funcionamiento con casos controlados antes de ser integrados al sistema principal.

Distribución de tareas

A continuación, se presenta la asignación de tareas según los módulos definidos:

Responsable	**Funcion**
Benjamin Ruiz	Lectura y escritura de archivos BMP, almacenamiento en arreglos dinámicos. Implementación del módulo de exportación de imágenes.
Kevin Jimenez	Implementación de operaciones bit a bit (XOR, rotaciones, desplazamientos) con punteros y validación de resultados.
Kevin Jimenez y Benjamin Ruiz	Lectura y análisis de archivos de enmascaramiento (`.txt`), y verificación con máscara M. Desarrollo del verificador de enmascaramiento.
Kevin Jimenez y Benjamin Ruiz	Diseño e implementación del algoritmo de búsqueda del orden correcto de transformaciones. Integración de módulos y pruebas finales.

Metodología de trabajo

Se utilizará un repositorio en Github para facilitar el control de versiones, y se realizarán commits regulares por cada avance funcional. Cada integrante probará sus funciones de manera independiente y dejará documentación mínima sobre su uso en el código.