

Informe - Análisis y Diseño de la Solución

Estudiante: NEVIS DANIELA LARGACHA MANCO

Curso: Informática II

Semestre: 2025-1

Fecha: 09 de abril de 2025

Docentes: Aníbal Guerra, Augusto Salazar

1. Análisis del problema

El desafío se trata de recuperar una imagen original en formato BMP a la que llamamos IO a partir de una versión muy alterada ID, esta imagen fue modificada con varias operaciones a nivel de bits, como XOR, rotaciones y desplazamientos, además, después de cada una de esas transformaciones, se aplicó una especie de filtro o máscara (M) para enmascarar el resultado.

Lo complicado es que no sé en qué orden se aplicaron esas transformaciones, lo único que tengo como apoyo son algunos archivos .txt que guardan el resultado del enmascaramiento y una semilla que se usó en ese proceso, entonces, el objetivo sería descubrir el orden correcto de las transformaciones, deshacerlas una a una y así recuperar la imagen original.

2. ¿Cómo vamos a resolverlo?

La idea es construir una solución usando C++ con Qt, dividiendo el problema en partes pequeñas que pueda manejar mejor, cada operación como el XOR o las rotaciones, se podrá aplicar y también deshacer, lo cual es clave para llegar al resultado final.

El flujo general que estoy pensando es este:

- ❖ Leer las imágenes y los archivos relacionados.
- ❖ Aplicar y deshacer las transformaciones a nivel de bits.
- ❖ Comparar los resultados intermedios con los archivos .txt para ver si voy por buen camino.
- ❖ Encontrar el orden correcto en que se aplicaron las transformaciones.

- ❖ reconstruir la imagen original.

3. ¿Qué tareas hay que hacer?

- ❖ Leer archivos importantes:

- ID.bmp: la imagen distorsionada.
- IM.bmp: una imagen que se usó durante el proceso de alteración por ejemplo, en el XOR.
- M.bmp: la máscara que se aplicó.
- Archivos .txt: que guardan resultados intermedios y las semillas usadas.

- ❖ Programar las transformaciones bit a bit:

- XOR entre imágenes.
- Rotaciones máximo 8 bits hacia la izquierda y la derecha.
- Desplazamientos también hasta 8 bits en ambas direcciones.

- ❖ Aplicar el enmascaramiento:

- Comparar las partes modificadas de la imagen con la máscara.
- Usar la semilla y verificar si el resultado coincide con lo que dice el archivo .txt.

- ❖ Descubrir el orden correcto:

- Probar distintas combinaciones posibles de transformaciones.
- Ver cuáles coinciden con los archivos .txt.
- Una vez de con la secuencia correcta, usarla para invertir todo el proceso.

- ❖ Recuperar la imagen original:

- Revertir las transformaciones en el orden correcto.
- Exportar la imagen ya reconstruida.

4. ¿Qué se me podría complicar?

Hay varios puntos delicados en este desafío y son:

- ❖ Las rotaciones y desplazamientos pueden generar errores si no se manejan bien los límites de los bits.
- ❖ Comparar los resultados intermedios con los txt puede ser difícil, porque no siempre se trabaja con la imagen completa, sino con pedazos.
- ❖ No saber el orden en que se aplicaron las transformaciones hace que tengamos que probar muchas combinaciones.

5. ¿Cómo lo voy a implementar?

Voy a organizar el código en módulos para mantener todo bien ordenado: uno contendrá el punto de arranque del programa, otro se encargará de leer y guardar las imágenes BMP, otro incluirá las funciones para hacer y deshacer las operaciones bit a bit, uno aplicará la máscara y verificará los resultados, y finalmente uno intentará deducir el orden correcto de transformaciones para reconstruir la imagen.

La idea es ir avanzando por etapas, subiendo los progresos a un repositorio de GitHub, y hacer pruebas con los archivos del “Caso 1” que se nos dieron, para asegurarnos de que todo funciona como debería.