Análisis y diseño de solución del Desafío I

Lady Vanessa Montenegro Alarcón
Steven Hernández Quintero

Facultad de Ingeniería, Universidad de Antioquia
259852 informática II

José Aníbal Guerra Soler
12 de abril de 2025

Este informe detalla el proceso de análisis y diseño de solución para el Desafío I, que consiste en reconstruir una imagen original a partir de una versión codificada mediante transformaciones aplicadas a nivel de bits. La única pista disponible para revertir este proceso son archivos de texto generados después de cada etapa de transformación. Además, cada etapa de transformación consiste en una única operación básica a nivel de bits, sin combinaciones o transformaciones compuestas, lo que simplifica en cierta medida el proceso de reversión. Esto significa que en ningún caso se aplicarán, por ejemplo, un desplazamiento y una operación XOR de manera simultánea como parte de una misma transformación, sino que cada una se aplicó de manera individual y secuencial.

Los archivos de texto generados después de cada transformación contienen dos piezas fundamentales de información. En la primera línea se encuentra la semilla, un valor numérico que indica la posición exacta donde comienza el enmascaramiento, a nivel de byte y no de píxel en la imagen transformada. El resto del archivo contiene el resultado de la suma entre los valores de los píxeles de la máscara y los valores originales de la imagen en esa etapa particular. Estos datos servirán como referencia para validar si hemos aplicado correctamente la transformación inversa durante el proceso de reconstrucción. El desarrollo de esta solución no solo pondrá a prueba nuestra capacidad para resolver problemas con información limitada, sino que también nos permitirá aplicar y fortalecer diversos conceptos del lenguaje C++, incluyendo el manejo de operaciones a nivel de bits, el uso eficiente de punteros, la gestión de memoria dinámica, y la implementación de funciones y estructuras de control adecuadas.

Para abordar el problema, comenzaremos trabajando con la imagen codificada final, que representa el punto de partida de nuestro proceso de reversión. Utilizando la función loadPixels, obtendremos acceso a los valores de los píxeles a través de un puntero que referencia el primer elemento del arreglo que los contiene. Con esta información, procederemos sistemáticamente a aplicar las posibles transformaciones inversas que podrían revertir las modificaciones originales. Después de cada transformación inversa propuesta se realizará una resta de pixeles entre la máscara y la posible imagen parcialmente revertida deberá ser verificada contra los datos contenidos en los archivos de enmascaramiento, específicamente comprobando que la semilla y los resultados de las sumas de píxeles coincidan con los valores esperados. De esta forma nos permitirá descartar las transformaciones que no produzcan los resultados esperados.

La implementación de la solución requerirá el desarrollo de varias funciones, cada una diseñada para manejar un tipo específico de transformación inversa. Estas funciones trabajarán principalmente con arreglos dinámicos que contienen la información de los píxeles, permitiendo la manipulación eficiente de los datos de la imagen. Sin embargo, es importante tener en cuenta ciertos riesgos durante este proceso, particularmente la posibilidad de desbordamientos de memoria. Las

operaciones a nivel de bits pueden, en algunos casos, generar valores que excedan la capacidad de almacenamiento definida para cada píxel, lo que podría provocar resultados inesperados o errores en el procesamiento. Para dar una posible solución a este problema, se implementarán controles que verifiquen los rangos de valores antes y después de cada operación, asegurando que los datos permanezcan dentro de los límites establecidos.

Este informe está sujeto a posibles cambios de análisis y el anexo de la implementación de la funciones diseñadas.