

Análisis y diseño de solución del Desafío I

Lady Vanessa Montenegro Alarcón

Steven Hernández Quintero

Facultad de Ingeniería, Universidad de Antioquia

259852 informática II

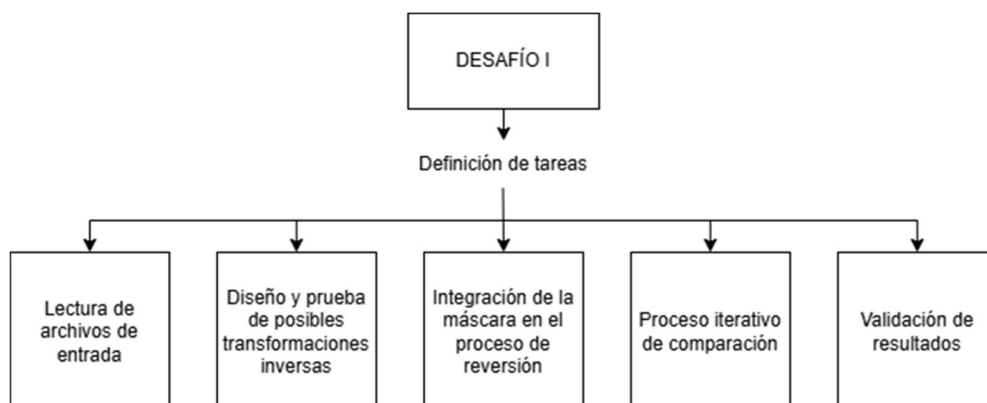
José Aníbal Guerra Soler

12 de abril de 2025

Análisis del problema

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. 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. Para abordar el problema, comenzaremos trabajando con la imagen codificada final, que representa el punto de partida de nuestro proceso de reversión.

Como paso inicial se almacenará la información de las imágenes proporcionadas junto con los archivos de texto, estas van a estar almacenadas en un tipo de dato que permite representar como mínimo valor numérico 0 y como máximo valor numérico 255, por esto mismo, como consideración inicial se debe tomar en cuenta que las operaciones efectuadas, como lo son transformación y suma de elementos, pueden generar valores que sobrepasen este límite. Para no trabajar con un volumen de datos tan grande se hará una copia de la imagen final desde donde indique la semilla de cada archivo de texto hasta el tamaño de la máscara. En ella se hará el proceso de pruebas de las posibles transformaciones inversas propuestas y se realizará una suma del valor de los bytes con la máscara, que deberá ser verificada contra los datos contenidos en los archivos de enmascaramiento, específicamente comprobando que la suma del primer elemento de cada arreglo coincidan con el valor esperado. De esta forma nos permitirá descartar las transformaciones que no se ajusten con el resultado esperado.



Algoritmos implementados

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 parcialmente dinámicos que contienen la información de los bytes de las imágenes, permitiendo la manipulación eficiente de los datos de la imagen. Dentro del programa principal se implementaron estructuras de control y condicionales que permitan el flujo de las operaciones.

Problemas de desarrollo

Durante el desarrollo de la solución se plantearon dos análisis erróneos. En ambos se consideró que la máscara tenía efectos permanentes sobre la imagen y que alteraban el orden de las operaciones que debían ser aplicadas.

Junto con esto se tuvo problemas con el IDE, ya que la manera en la que el depurador al encontrar valores nulos en arreglos de tipo unsigned char lo interpretaba como un fin de la cadena de caracteres y en consecuencia el visualizador de variables dejaba de mostrar el resto de la cadena, lo que generó confusiones en la implementación correcta de las funciones. Por otro lado, se daba un problema similar con los arreglos de tipo int, ya que no permite visualizar los elementos del arreglo.

Evolución de solución

Al principio se tenía planeado la implementación de funciones para las transformaciones que trabajaran sobre la porción del arreglo generado y otras para aplicar las operaciones sobre el arreglo original de la imagen. Sin embargo, se consideró innecesario pues podían ser modificadas y unificadas para manejar ambos arreglos. También se modificó el rango de datos a validar, de comparar todos los elementos de los arreglos a solo el elemento inicial de cada uno.

Dentro de las posibles transformaciones se hallan algunas de tipo irreversible puesto que generan una pérdida de información en los datos, debido a esto se omitió su invocación dentro del programa principal.

Por último, se implementó un casting de datos en una de las funciones para poder comparar sus valores sin afectar su contenido ni el tipo del arreglo.