

Informe Desafío I

Por:

Isabela Aguirre Ceballos

Juan Felipe Higueta Pérez

Informática I

Universidad de Antioquia

2025

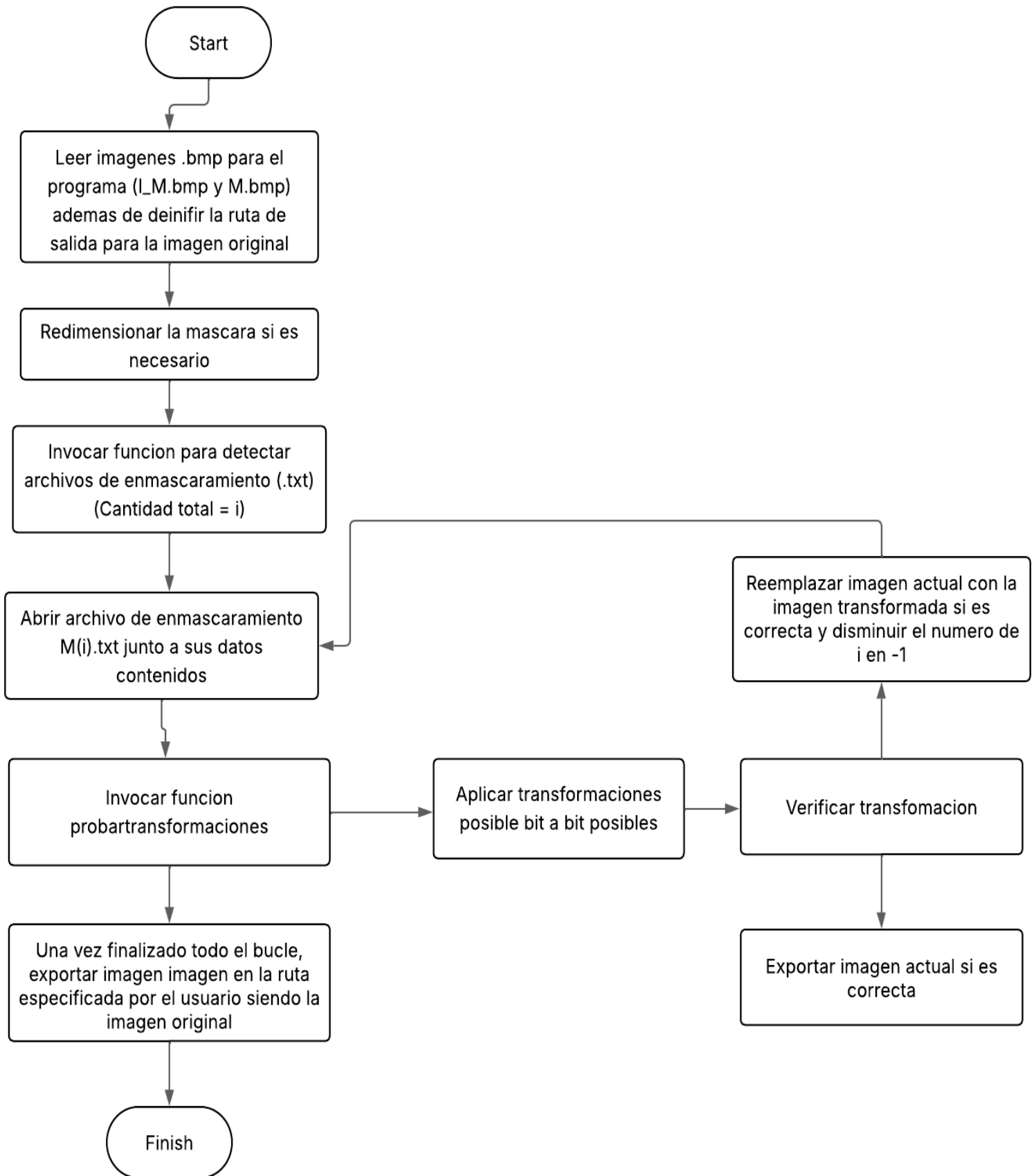
Análisis del problema

En este desafío, nos enfrentamos a la tarea de reconstruir una imagen original (IO) a partir de una imagen final (ID) que ha sido distorsionada mediante varias transformaciones a nivel de bits. Estas transformaciones incluyen operaciones XOR con otra imagen (IM), rotaciones y desplazamientos de bits. Sin embargo, el orden de las transformaciones y los parámetros exactos de cada una no se conocen de antemano.

Tras cada transformación, se realizó un proceso de enmascaramiento, donde se extrajo una porción de la imagen transformada y se sumó píxel por píxel con una máscara de color (M). El resultado de esta suma se guardó en varios archivos .txt, que contienen dos datos importantes: el número de desplazamiento (s) que indica desde qué punto de la imagen transformada se extrajo la porción, y los valores RGB resultantes de la suma con la máscara.

El objetivo del proyecto es reconstruir la imagen original utilizando solo la imagen final (ID), la imagen de referencia (IM), la máscara (M) y los archivos .txt que contienen los datos de enmascaramiento. Dado que no sabemos qué transformaciones se aplicaron ni en qué orden, la solución consiste en probar diferentes combinaciones posibles de transformaciones, verificando después de cada intento si el resultado coincide con los datos del archivo .txt correspondiente.

Esquema



Algoritmos implementados

Operación XOR - aplicarXOR

Esta función realiza la operación XOR entre dos imágenes, píxel por píxel, para deshacer transformaciones previas. Se aplica entre la imagen distorsionada (ID) y la imagen de referencia (IM). La función opera en los tres canales de color (Rojo, Verde y Azul) de cada píxel. El resultado es una imagen donde las transformaciones previas de XOR se revierten.

Rotación de Bits - rotarBits

El algoritmo de rotación de bits mueve los bits de cada byte de la imagen hacia la izquierda o hacia la derecha. Este proceso se aplica para revertir las transformaciones que involucraron una rotación de bits en la imagen. La función permite restaurar los valores de los píxeles a su estado original si se conoce el número de posiciones que se rotaron.

Desplazamiento de Bits - desplazarBits

El desplazamiento de bits mueve los bits de cada byte hacia la izquierda o hacia la derecha, y los bits que se salen se pierden. La función desplazarBits permite revertir transformaciones que consisten en desplazamientos de bits. Aunque es más difícil de revertir que la rotación, la función permite manejar las transformaciones de desplazamiento al conocer los valores de desplazamiento correctos.

valores almacenados en los archivos .txt para verificar si el enmascaramiento fue correcto.

Verificación contra archivo .txt - verificarEnmascaramiento

La función verificarEnmascaramiento compara los valores RGB de la porción de la imagen enmascarada con los datos almacenados en los archivos .txt. Si los valores coinciden, significa que la transformación aplicada fue correcta. Esta función es clave para validar cada paso de la reconstrucción de la imagen original.

Prueba de Transformaciones - probarTransformaciones

La función probarTransformaciones es la encargada de aplicar las diferentes transformaciones (XOR, rotación y desplazamiento) sobre una copia de la imagen. Al probar una transformación, se compara el resultado con los datos del archivo .txt correspondiente. Si se encuentra una coincidencia, la imagen original se reemplaza por la imagen transformada y se continúa con el siguiente paso. Esta función es esencial para probar las combinaciones correctas de transformaciones y restaurar la imagen original.

Obtención de cantidad de .txt a trabajar - detectarArchivosDeEnmascaramiento

La función detectarArchivosDeEnmascaramiento analiza la carpeta y cuenta el número total de archivos .txt disponibles. Esto permite llevar el control de cuántas pistas de enmascaramiento se tienen y de qué archivo se debe partir para realizar las pruebas.

Redimensionar Máscara - redimensionarMascara

La función redimensionarMascara ajusta el tamaño de la máscara (M) para que coincida con las dimensiones de la imagen que estamos procesando, ya sea la imagen original transformada o la imagen final encriptada. Esta función es útil para asegurar que el tamaño de la máscara sea siempre compatible con la imagen en cuestión.

Problemas Encontrados Durante el Desarrollo

A lo largo del desarrollo, surgieron varios problemas, que se resolvieron mediante pruebas y ajustes. A continuación, se detallan algunos de los principales problemas encontrados:

1. Problema con la Función de Enmascaramiento

Al trabajar en la función de enmascaramiento, el programa se cayó durante las pruebas debido a un error lógico. Este error ocurría cuando, al sumar los valores de la máscara con los datos del archivo .txt, se excedían los límites de la máscara, lo que causaba errores de memoria.

Solución: Para solucionarlo, establecimos un límite en la cantidad de elementos que se podían sumar, basándonos en el tamaño total de la máscara. Esto evitó los desbordamientos y corrigió el error.

2. Problema con la Identificación de Archivos .txt

En algún momento, se colaba un archivo .txt innecesario, lo que generaba un archivo adicional que no debía ser procesado. Para resolverlo, implementamos una función que identificaba correctamente los archivos de pistas, basándonos en el patrón de nomenclatura de los archivos Mx.txt , asegurando que solo se procesaran los archivos necesarios.

3. Problema con la Verificación de Transformaciones

A medida que avanzábamos en el proyecto, nos enfrentamos a un problema con la verificación de las transformaciones. El proceso de verificación fallaba en la primera transformación, lo que afectaba a todo el flujo de transformaciones posteriores y provocaba que la imagen final no se modificara.

Evolución de la solución y consideraciones para la implementación

Evolución de la solución:

Análisis inicial: Se entendió que el problema consistía en reconstruir una imagen original (IO) a partir de una imagen distorsionada (ID) usando transformaciones como XOR, rotaciones y desplazamientos, junto con una máscara (M) y archivos de enmascaramiento (.txt).

Desarrollo de funciones clave:

- **Transformaciones:** Implementamos funciones para XOR, rotación y desplazamiento de bits.
- **Prueba de transformaciones:** Creamos una función para probar todas las combinaciones posibles de transformaciones y verificar cuál genera los resultados correctos.
- **Iteración:** Ajustamos y refinamos las funciones para mejorar la precisión y eficiencia en el proceso de reconstrucción de la imagen.

Consideraciones para la implementación:

- **Gestión de memoria:** Es crucial liberar correctamente la memoria para evitar fugas, especialmente al trabajar con imágenes grandes.
- **Precisión en las transformaciones:** Las transformaciones deben aplicarse con precisión para garantizar que los resultados intermedios sean correctos.
- **Tamaño de la máscara:** La máscara debe coincidir en tamaño con la imagen en cada paso. Es importante redimensionarla correctamente.

- **Orden de transformaciones:** Las transformaciones deben aplicarse en orden inverso y probarse todas las combinaciones posibles.
- **Pruebas exhaustivas:** Se requieren pruebas exhaustivas para asegurar que la solución funcione correctamente en todos los casos posibles.

En resumen, la solución se ha refinado para aplicar correctamente las transformaciones, verificar los resultados con los archivos de enmascaramiento y reconstruir la imagen original. Las consideraciones clave son la gestión de memoria, precisión y eficiencia.