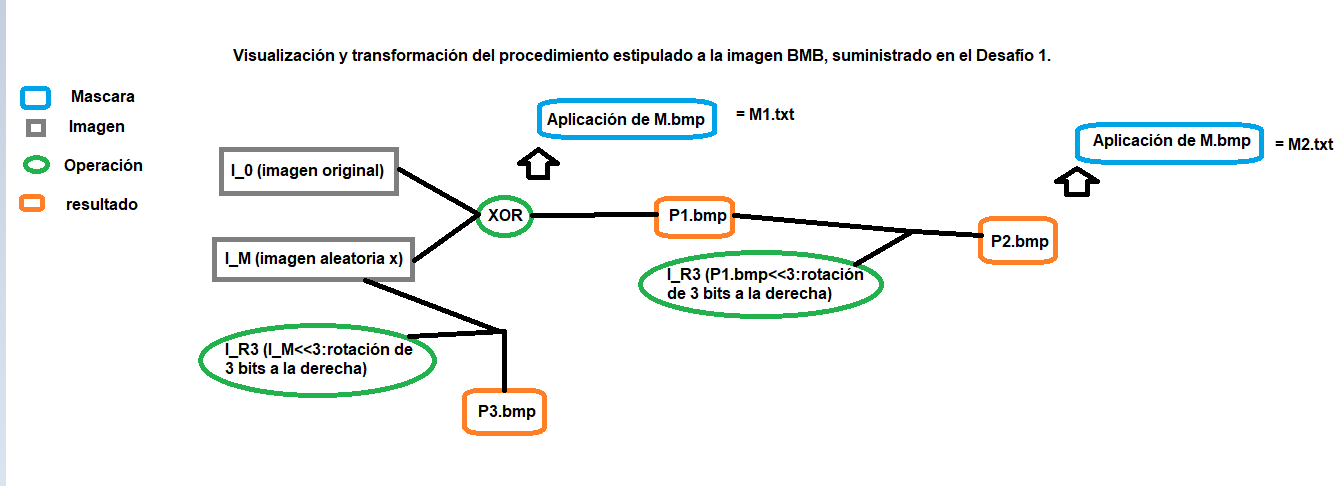
Informe del desarrollo del Desafío 1- Informática 2

Introducción

En este documento se menciona conceptos fundamentales para iniciar el desarrollo del desafío 1 en el curso de Informática 2, son necesarios para la comprensión y la solución del problema propuesto por los profesores del curso, básicamente se hace una consulta en fuentes de información y/o bibliotecas acerca de las operaciones a nivel de bits(bitwise) en C++, con el fin de comprender su sintaxis y que operaciones se pueden hacer con ellas, estas son: AND(&), OR, XOR(^), NOT(~), desplazar a la derecha (<<), desplazar a la izquierda (>>). Todas ellas me permiten hacer operaciones a nivel de bits y están relacionadas con el desafío propuestos por los profesores, ya que una imagen BMP está compuesta por pixeles, cada pixel equivale a 3 bytes y 1 byte equivale a 8 bits, esta información es importante ya que es posible manipular imágenes a nivel de bits en C++ y en este caso según el problema planteado es necesario conocer cómo hacerlo para aprender a enmascarar y desenmascarar una imagen BMP, entonces como estudiante me surgen varias preguntas:

¿Cómo puedo desenmascarar una imagen BMP teniendo como recursos los archivos txt que me indican que transformación tuvo la imagen a nivel de bits?¿Cómo es posible revertir esos procesos para llegar a la imagen original?¿es posible hacerlo sin usar estructuras ni STL?¿Cómo sería la sintaxis en C++ para lograr ese objetivo?¿Es suficiente con las explicaciones que me dieron en clases o debo ir más allá para solucionar el problema?¿Cuál es el mapa de aprendizaje para cumplir con los objetivos y desafíos propuestos en el curso?

Pasos para la solución al Desafío 1:

1. Nuevamente se realiza una lectura minuciosa al documento del desafío 1 que nos compartió el profesor, si bien se había realizado una representación gráfica de la posible interpretación del problema planteado, se logra complementar una parte importante que no había podido inferir del documento, la cual era en que pasos de la transformación de la imagen “I\_d” se generaban los archivos txt, entonces nuevamente comparto la imagen complementada con esa parte, que es la que se había compartido en el primer commit.  
     
     
   2. Teniendo en cuenta los requisitos para la entrega de la solución, cabe resaltar que el profesor indica que no se debe usar estructuras ni STL. Partiendo de esto, es importante recordar que se va a manipular archivos bmp(resultado de las transformaciones), archivos txt(contiene información de la transformación y la máscara aplicada) en los procesos. La lógica que se definió para recuperar la imagen original es la siguiente según el mapa de procesos construido a partir de la lectura del desafío 1:  
     
   A. **I\_R3**(rotación de bits a la derechas) aplicado **a I\_M**(imagen aleatoria) = **P3** entonces,   
   **P3 = XOR(I\_R3, I\_M)**  
     
     
   Nota: Como no se puede usar estructuras ni STL, entonces se toma como referencia las librerías usadas por el profesor en código ejemplo, también se empieza a desarrollar el código en el mismo main y al investigar sobre las librerías se obtiene la siguiente información:

* **#include <fstream> :** En este desafío me sirve para leer y escribir datos binarios o de texto en archivos, en este caso para usarla en los archivos bmp.
* **#include <QImage>:** Me sirve para cargar, manipular y guardar imágenes en diversos formatos, incluyendo BMP. Permite acceder fácilmente a los canales RGB de cada píxel. Puedo modificar píxeles individualmente con funciones como setPixel(), entre otras.
* **#include <QString>:** Me sirve para manejar cadenas de caracteres de forma segura, también para manejar rutas de archivo, me ayuda a interactuar con QImage.

En el mismo main del ejemplo del profesor se empieza trabajar, inicialmente se usan las librerías antes mencionadas, también las otras líneas de código básicas e indispensables para empezar a utilizar C++, las cuales son #include <iostream> y using namespace std; entonces primero que todo necesito cargar los insumos con los cuales voy a aplicar procesos inversos los cuales son:

* P3, I\_M, M, : creo funciones para extraer la información de los bits de las imágenes bmp y las guardo en variables apuntadas, con el fin de manipular y/o comparar su contenido.

