DESAFIO I – INFORMATICA II – 2025/1  
ALEJANDRO NARANJO NARANJO  
CC 1041440121   
  
**INFORME**  
  
**I ANALISIS DEL PROBLEMA:**  
EL DESAFÍO CONSISTE EN RECONSTRUIR UNA IMAGEN BMP ORIGINAL QUE FUE SOMETIDA A UNA SERIE DE TRANSFORMACIONES A NIVEL DE BITS (ROTACIÓN, DESPLAZAMIENTO, XOR) Y LUEGO ENMASCARADA MEDIANTE LA SUMA CON UNA MÁSCARA RGB.

LA DIFICULTAD PRINCIPAL ESTA EN QUE EL ORDEN DE LAS TRANSFORMACIONESESDESCONOCIDO, Y SOLO SE CUENTA CON LA IMAGEN FINAL TRANSFORMADA, UNA IMAGEN ALEATORIA (IM), LA MÁSCARA UTILIZADA (M), Y ARCHIVOS CON RESULTADOS PARCIALES DEL ENMASCARAMIENTO.

CADA TRANSFORMACIÓN FUE SEGUIDA POR UNA OPERACIÓN DE ENMASCARAMIENTO QUE CONSISTE EN SUMAR UNA PORCIÓN DE LA IMAGEN RESULTANTE CON UNA MÁSCARA, DESDE UNA POSICIÓN ALEATORIA (SEMILLA). ESTOS RESULTADOS PERMITEN INFERIR EL ESTADO DE LA IMAGEN EN ESE PUNTO DEL PROCESO.

LA SOLUCIÓN AL PROBLEMA IMPLICA APLICAR **INGENIERÍA INVERSA** PARA DETERMINAR LAS OPERACIONES REALIZADAS Y SU ORDEN, A PARTIR DE LOS ARCHIVOS GENERADOS Y LAS IMÁGENES PROVISTAS.  
  
**II ALTERNATIVA DE SOLUCION PROPUESTA:**SE PLANTEA UNA SOLUCIÓN EN TRES ETAPAS:

1. IMPLEMENTAR FUNCIONES DE TRANSFORMACIÓN BIT A BIT: ROTACIONES, DESPLAZAMIENTOS Y XOR. ESTAS FUNCIONES PERMITIRÁN APLICAR Y DESHACER TRANSFORMACIONES SOBRE LOS DATOS DE IMAGEN.
2. VERIFICAR ENMASCARAMIENTOS: IMPLEMENTAR UN SISTEMA QUE, DADA UNA IMAGEN TRANSFORMADA Y UNA MÁSCARA M, SIMULE EL ENMASCARAMIENTO CON DIFERENTES SEMILLAS Y COMPARE EL RESULTADO CON LOS ARCHIVOS .TXT. ASÍ SE PODRÁ VERIFICAR SI UNA TRANSFORMACIÓN ES CORRECTA EN ESA ETAPA.
3. PROBAR COMBINACIONES DE TRANSFORMACIONES: USAR LÓGICA Y PRUEBAS SISTEMÁTICAS (FUERZA BRUTA LIMITADA) PARA APLICAR COMBINACIONES DE OPERACIONES Y VERIFICAR CUÁL DE ELLAS GENERA UNA COINCIDENCIA CON LOS ARCHIVOS DE ENMASCARAMIENTO. ESTO PERMITIRÁ DEDUCIR EL ORDEN CORRECTO DE OPERACIONES Y ASÍ RECONSTRUIR LA IMAGEN ORIGINAL.

**III ESQUEMA DE TAREAS:**FASE 1: LECTURA Y PREPARACIÓN DE DATOS

LEER IMAGEN TRANSFORMADA ID (IMAGEN FINAL DISTORSIONADA).

LEER IMAGEN ALEATORIA IM (USADA PARA XOR).

LEER MÁSCARA M (IMAGEN UTILIZADA PARA ENMASCARAMIENTO).

LEER ARCHIVOS .TXT DE RASTREO CON LOS RESULTADOS DEL ENMASCARAMIENTO.

FASE 2: IMPLEMENTACIÓN DE TRANSFORMACIONES BIT A BIT

FUNCIÓN DE XOR ENTRE DOS IMÁGENES RGB.

FUNCIÓN DE ROTACIÓN DE BITS POR CANAL RGB (A IZQUIERDA O DERECHA, HASTA 8 BITS).

FUNCIÓN DE DESPLAZAMIENTO DE BITS POR CANAL RGB.

FUNCIÓN PARA REVERTIR CADA OPERACIÓN (XOR ES REVERSIBLE, ROTACIÓN TAMBIÉN, DESPLAZAMIENTO NO SIEMPRE).

FASE 3: SIMULACIÓN DE ENMASCARAMIENTO

FUNCIÓN PARA APLICAR ENMASCARAMIENTO ENTRE UNA IMAGEN Y LA MÁSCARA M, DESDE UNA SEMILLA (DESPLAZAMIENTO S).

FUNCIÓN PARA COMPARAR EL RESULTADO DEL ENMASCARAMIENTO SIMULADO CON EL ARCHIVO .TXT CORRESPONDIENTE.

FASE 4: IDENTIFICACIÓN DEL ORDEN DE OPERACIONES

PROBAR COMBINACIONES POSIBLES DE TRANSFORMACIONES APLICADAS SOBRE ID.

DESPUÉS DE CADA TRANSFORMACIÓN, SIMULAR ENMASCARAMIENTO Y VERIFICAR CON ARCHIVOS .TXT.

SI HAY COINCIDENCIA CON EL ARCHIVO DE RASTREO → SE IDENTIFICA LA TRANSFORMACIÓN USADA EN ESA ETAPA.

REPETIR HASTA ENCONTRAR EL ORDEN COMPLETO.

FASE 5: RECONSTRUCCIÓN DE LA IMAGEN ORIGINAL

UNA VEZ IDENTIFICADO EL ORDEN, REVERTIR TODAS LAS TRANSFORMACIONES, EN ORDEN INVERSO.

GUARDAR LA IMAGEN ORIGINAL RECONSTRUIDA EN UN NUEVO ARCHIVO BMP.  
  
FASE 6: VALIDACIÓN

COMPARAR VISUALMENTE LA IMAGEN ORIGINAL CON IMÁGENES DE PRUEBA (SI ESTÁN DISPONIBLES).  
  
VERIFICAR SI LAS SUMAS RGB DE LA IMAGEN RECONSTRUIDA + M COINCIDEN CON LOS ARCHIVOS .TXT.

**IV ALGORITMOS A IMPLEMENTAR (PLAN DE DESARROLLO)**XORIMAGES(ID, IM): APLICA XOR ENTRE DOS IMÁGENES.

ROTATEBITSRIGHT(IMAGE, BITS): ROTA A LA DERECHA LOS BITS DE CADA CANAL RGB.

SHIFTBITS(IMAGE, BITS): DESPLAZA LOS BITS (A IZQUIERDA O DERECHA).

APPLYMASK(ID, M, S): SIMULA EL ENMASCARAMIENTO CON LA MÁSCARA M DESDE LA POSICIÓN S.

COMPAREWITHTRACEFILE(RESULTADO, ARCHIVO.TXT): COMPARA LOS VALORES OBTENIDOS CON LOS DEL ARCHIVO DE RASTREO.

RECONSTRUCTIMAGE(ID, IM, M, ARCHIVOS): FUNCIÓN PRINCIPAL PARA ENCONTRAR EL ORDEN DE TRANSFORMACIONES Y REVERTIRLAS.

**V PROBLEMAS DE DESARROLLO IDENTIFICADOS HASTA AHORA**LA CANTIDAD DE COMBINACIONES POSIBLES DE TRANSFORMACIONES PUEDE SER ALTA, ESPECIALMENTE SI HAY MUCHAS ETAPAS.

LA COMPARACIÓN ENTRE IMÁGENES Y LOS ARCHIVOS DE RASTREO REQUIERE PRECISIÓN BYTE A BYTE.

TRABAJAR CON PUNTEROS Y MEMORIA DINÁMICA SIN STL REQUIERE MUCHO CUIDADO PARA EVITAR ERRORES DE ACCESO O PÉRDIDA DE MEMORIA.

ES IMPORTANTE IMPLEMENTAR ESTRUCTURAS SIMPLES PERO EFICIENTES PARA ALMACENAR LOS DATOS INTERMEDIOS.

**VI EVOLUCION DE LA SOLUCION:**INICIALMENTE SE PENSÓ EN APLICAR LAS TRANSFORMACIONES EN ORDEN ALEATORIO Y VERIFICAR LA IMAGEN RESULTANTE. SIN EMBARGO, ESTO NO ERA EFICIENTE NI ESCALABLE. A PARTIR DEL ANÁLISIS DE LOS ARCHIVOS .TXT Y LA LÓGICA DEL ENMASCARAMIENTO, SE AJUSTÓ EL ENFOQUE HACIA COMPARAR PORCIONES ESPECÍFICAS DE LAS IMÁGENES Y EVALUAR TRANSFORMACIONES PARCIALES, LO CUAL PERMITE REDUCIR SIGNIFICATIVAMENTE EL NÚMERO DE COMBINACIONES A PROBAR.

LA IMPLEMENTACIÓN DE PEQUEÑAS FUNCIONES MODULARES FACILITARÁ LAS PRUEBAS INDEPENDIENTES Y MEJORARÁ LA LEGIBILIDAD DEL CÓDIGO.