**Informe del Desarrollo del Proyecto: Reconstrucción de Imagen BMP**

**1. Introducción**

El objetivo del proyecto fue reconstruir una imagen que había sido transformada mediante operaciones de manipulación de bits, enmascaramiento y operaciones lógicas como el XOR. Se proporcionaron archivos de imagen, código ayuda y archivos de texto que contenían datos de enmascaramiento y semillas para la verificación de cada paso.

**2. Esquema de Tareas Definidas**

* Carga de las imágenes BMP.
* Lectura de los archivos de texto que contienen los valores de enmascaramiento.
* Aplicación de operaciones de reconstrucción:

1. XOR entre imágenes.
2. Rotaciones de bits.
3. Verificaciones de enmascaramiento usando los valores y semillas.

* Exportación de la imagen reconstruida en formato BMP.

**3. Algoritmos Implementados**

* **Carga de Imágenes**: Uso de la función loadpixels para cargar imágenes y extraer sus píxeles en arreglos lineales.
* **Carga de Archivos de Enmascaramiento**: Uso de la función loadSeedMasking para la lectura de semillas y valores RGB.
* **Operaciones de Reconstrucción**:

1. **XOR**: Se aplica un XOR bit a bit entre dos arreglos de píxeles.
2. **Rotación de Bits**: Cada byte de los píxeles es rotado a la derecha por un número determinado de bits.
3. **Verificación**: Se suma el valor del píxel más el de la máscara y se compara contra el archivo de referencia.

* **Exportación de Imagen**: Uso de la función exportImage para la creación de una nueva imagen a partir del arreglo de píxeles y guardado como BMP.

**4. Problemas de Desarrollo Afrontados**

* **Errores de Verificación**: Inicialmente las operaciones de verificación no coincidían debido a malinterpretación de las rotaciones de bits.
* **Direcciones de Archivos**: Se solucionó el problema de carga de archivos usando rutas absolutas en vez de relativas.
* **Control de Memoria Dinámica**: Se tuvieron que asegurar las liberaciones de memoria de todos los recursos asignados dinámicamente.

**5. Evolución de la Solución y Consideraciones para la Implementación**

* Implementación de las funciones faltantes como para aplicar el XOR, las rotaciones y hacer las verificaciones.
* Se realizaron verificaciones detalladas paso por paso para validar cada transformación intermedia.
* Se implementó un sistema de depuración que mostraba las sumas y comparaciones para identificar errores más fácilmente.
* La elección de tipos de datos como unsigned char\* para los píxeles optimizó el uso de memoria, ya que cada canal RGB cabe en 1 byte.

**6. Conclusión**

El proyecto permitió reforzar el manejo de imágenes en bajo nivel, las operaciones bit a bit, la manipulación de archivos externos. La metodología de verificación paso a paso fue crucial para alcanzar la reconstrucción correcta de la imagen original.

.