

Universidad de Costa Rica

Escuela de Ciencias de la Computación e Informática

Curso:

Lenguaje Ensamblador

CI-0118

Grupo 02

Proyecto del curso – Procesamiento de imágenes

Integrantes:

Kembly Paniagua Jiménez - B95822

Rodrigo Mendoza Quesada - C04813

Ciudad Universitaria Rodrigo Facio

I-2022

Resumen:

Para la realización de este proyecto se va a realizar un modelo SIMD, donde tomamos una instrucción y se le aplica a un grupo de datos simultáneamente, esto será utilizando las instrucciones vectoriales *avx2* las cuales son las encargadas de aplicar operaciones a un conjunto de datos de manera simultánea, esto mejora el rendimiento de grandes volúmenes de datos. Estas instrucciones son las ideales para este proyecto ya que se debe abrir una imagen en formato bmp, crear un bitmap en modo binario utilizando el lenguaje c++, en el cual a través del lenguaje ensamblador se realizará un programa con el cual se podrá modificar los valores para así poder modificar los bits que en este caso representan los pixeles de la imagen y así poder lograr que la imagen se vea como si se le hubiera aplicado un filtro.

Algunas de las consideraciones que se deben tomar en cuenta es que tiene que ser capaz de diferenciar los headers del BMP y los atributos de la imagen y hacer las transcripciones entre *Little endian* y *big endian* ya que de no ser tomadas en cuenta esto alterara el resultado obtenido, finalmente se tiene que poder generar un nuevo archivo ya transformado (con el filtro aplicado) de formato bmp de igual manera. Para mostrar los resultados se debe de crear una aplicación en Qt en la cual se debe de elegir la imagen desplegarla en pantalla y con botones poder ir aplicando los diferentes filtros y mostrando el resultado en pantalla.

Descripción Requisitos:

1. Abrir un archivo de tipo bmp y cargar los pixeles a un vector de memoria.

2. Creación de los filtros:

1. Ajustar el brillo:

Esto consiste en a cada píxel sumarle una constante para aumentarle el brillo y para bajarlo se le resta de igual manera una constante.

2. Contraste:

Para el filtro del contraste, se debe restar 128 a cada píxel, luego multiplicarlo por el factor de corrección el cual es dado por una formula y finalmente se vuelve a sumar 128.

F = factor de corrección.

$$F = 259 * (255 + C) / 255 * (259 - C)$$

3. Colores negativos:

Colores negativos, básicamente implicaría reemplazar cada pixel individual (R,G,B) por el pixel (255-R, 255- G, 255-B) respectivo.

4. Escalado:

Escalado de la imagen, se tendría que cuadruplicar el tamaño de la imagen, duplicar su dimensión en "X" y en "Y" para luego hacer una reubicación y respectiva copia de cada píxel adyacente en los nuevos espacios vacíos.

5. Espejado

Espejado de la imagen, se debe reubicar cada píxel en orden inverso en términos de su dimensión "X", esto implica mover los pixeles finales de la imagen por cada hilera hacia el frente de esta.

3. Creación de interfaz gráfica en qt.

1. Elegir una imagen.
2. Mostrar la imagen resultante en pantalla a través de un panel en Qt.
3. Generar un archivo nuevo con las modificaciones efectuadas a la imagen original y guardar el archivo en formato BMP.

Diseño de la interfaz:

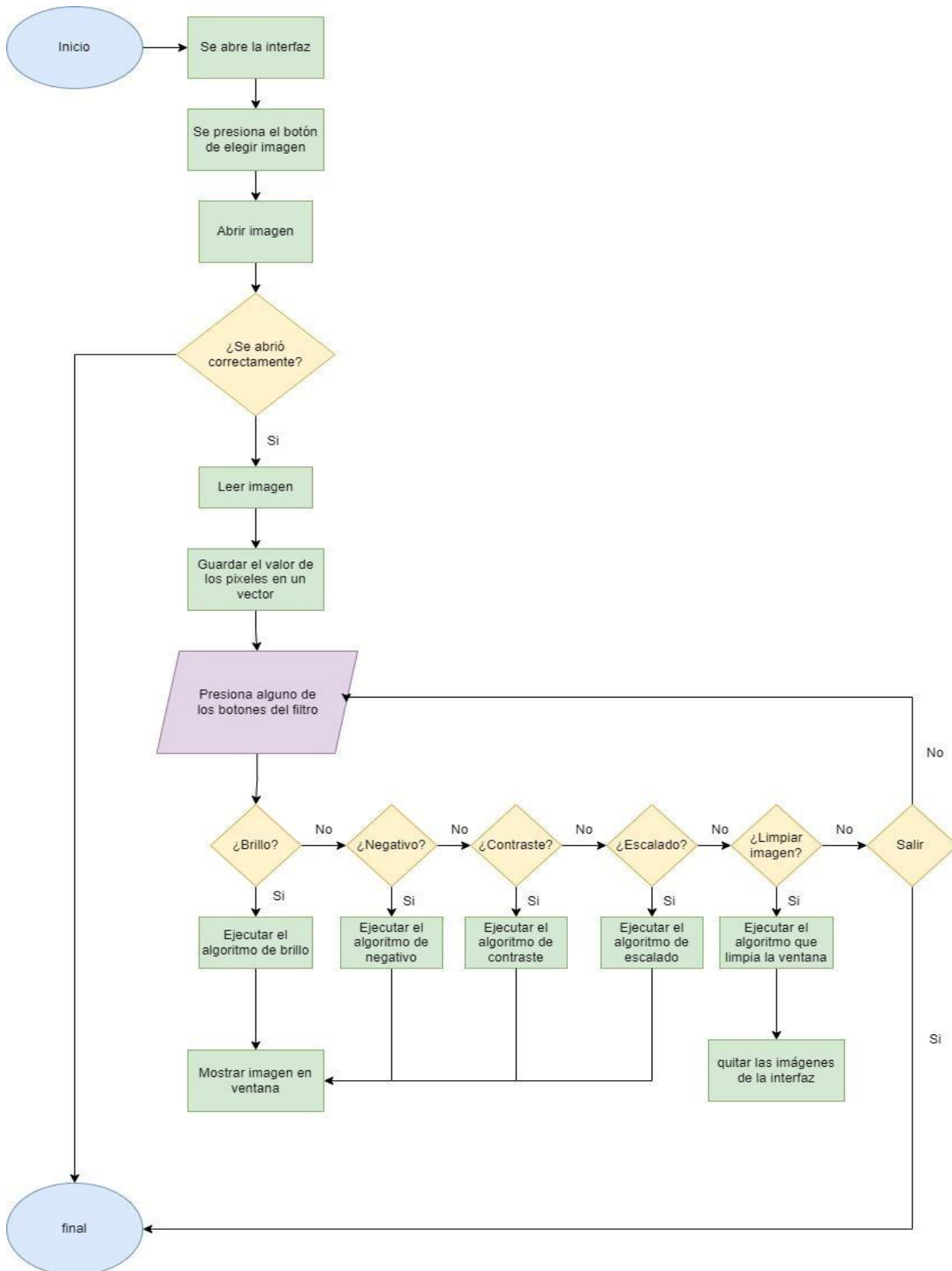
Se creará una ventana la cual tendrá dos *labels* una en la cual se mostrará la imagen original y la otra donde se mostrará la imagen editada. La ventana a su vez contara con botones los cuales serán los encargados de cambiar los filtros a la imagen, también tendrá un botón de elegir imagen el cual desplegara el explorador de archivos para que el usuario pueda elegir la imagen a su gusto.

Lista de tareas:

- Creación de la interfaz con QT.
- Creación de las clases encargadas de abrir la imagen bmp y crear su bitmap y el vector que lo contiene.
- Implementar el código en ensamblador que crea los filtros solicitados utilizando las instrucciones vectoriales.
- Enlazar la interfaz con el código en ensamblador.

- Implementar el código creado en NASM con la interfaz y el vector de imágenes para poder hacer los cambios respectivos y desplegar la imagen en la interfaz

Diagrama de Flujo:



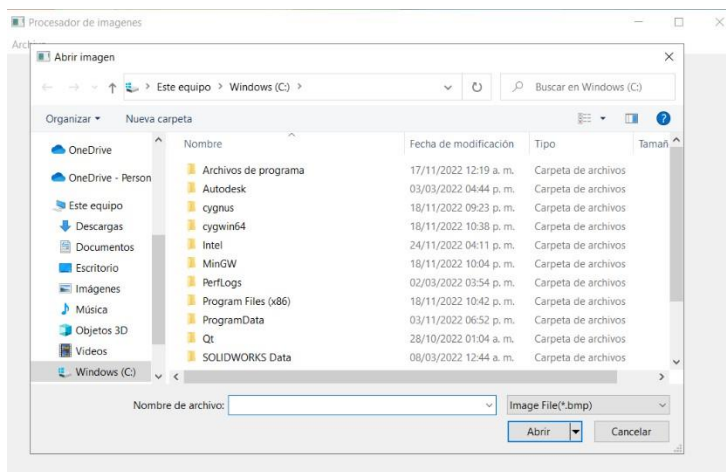
Bitácora

Fecha	Quien trabajó	Tarea	Descripción
17 noviembre	Rodrigo	Lectura del bmp y creación del vector de bits	Se implemento lectura y escritura del bmp
	Kembly	Creación de la interfaz	
	Kembly Rodrigo	Intento de creación de los filtros	Se agregaron algunos filtros en NASM
20 noviembre	Kembly	Filtro de negativo y brillo	Se agregó filtro negativo para las pruebas
	Rodrigo	Lectura del bmp y creación del vector de bits	Escritura del bmp
21 noviembre	Kembly	Se agrego el filtro del negativo a la interfaz	Se verifico el filtro negativo
	Rodrigo	Se agrego las imágenes en la interfaz	Prueba de generación de imagen alterada
22 noviembre	Kembly Rodrigo	Filtro contraste	filtro de contraste
23 noviembre	Kembly Rodrigo	Ensamblaje con QT	Se realizaron los cambios en el .pro para poder compilar el QT con el NASM
24 noviembre	Rodrigo	Algoritmo de escalado	Se intento la implementación del filtro de escalado, solo aumenta dos pixeles

Anexos (Descripción de la interfaz)



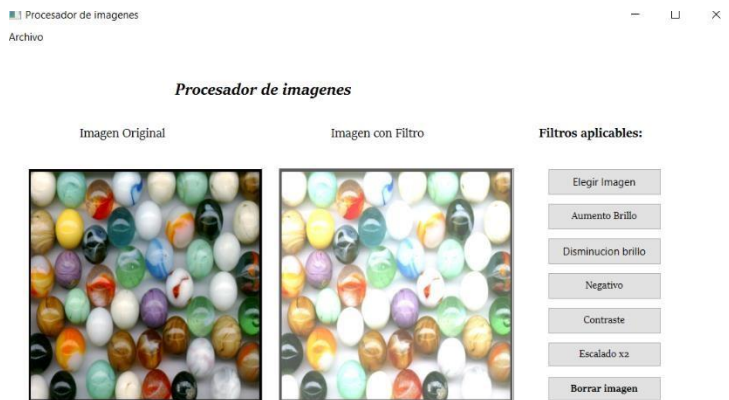
Al abrir la interfaz se despliega los botones y los *labels* donde se colocarán las imágenes.



Al presionar el botón de elegir imagen de despliega la ventana de explorador de archivos, para elegir la imagen



Luego muestra la imagen



Al darle al botón de brillo, lo aumenta y muestra la imagen.



Al presionar el botón de reducir brillo

Filtro de negativo.

Procesador de imágenes

Imagen Original



Imagen con Filtro



Filtros aplicables:

Elegir Imagen

Aumento Brillo

Disminucion brillo

Negativo

Contraste

Escalado x2

Borrar imagen

Filtro de contraste.

Procesador de imágenes

Imagen Original





Imagen con Filtro



Filtros aplicables:

Elegir Imagen

Aumento Brillo

Disminucion brillo

Negativo

Contraste

Escalado x2

Borrar imagen

Referencias

<https://www.youtube.com/watch?v=vqT5j38bWGg>

referencia: <https://poesiabinaria.net/2011/06/leyendo-archivos-de-imagen-en-formato-bmp-en-c/>

<https://solarianprogrammer.com/2018/11/19/cpp-reading-writing-bmp-images/>