Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Introducción a la Programación de Computadoras 1
Primer Semestre 2023

Enunciado proyecto 2

Ugallery

Sección	Catedrático	Auxiliar
A	Marlon Fransisco Orellana Lopez	José Daniel Alvarado Fajardo
B	William Estuardo Escobar Argueta	Diego Fernando Cortez Lopez
C	Moises Eduardo Velasquez Oliva	José Alejandro Santizo Cotto
D	Herman Igor Veliz Linares	Salvador de Jesus López Bautista José Eduardo Morales García
E	Neftali de Jesus Calderon Mendez	Erick Daniel Antillon Chinchilla
F	William Estuardo Escobar Argueta	Freddy Alejandro Monterroso
G	Edgar Francisco Rodas Robledo	Christofer William Borrayo López

APLICACIÓN:

A continuación, se definen y describen las vistas con las que debe contar la aplicación.

1. Ventana principal

La ventana principal es un menú y un acceso a la sección de biblioteca, editor y al convertidor.

La ventana principal debe ser capaz de trasladar el foco de la aplicación hacia la vista de la función a la que se acceda. La distribución de los menús y de los componentes propios de cada funcionalidad queda a discreción del estudiante.

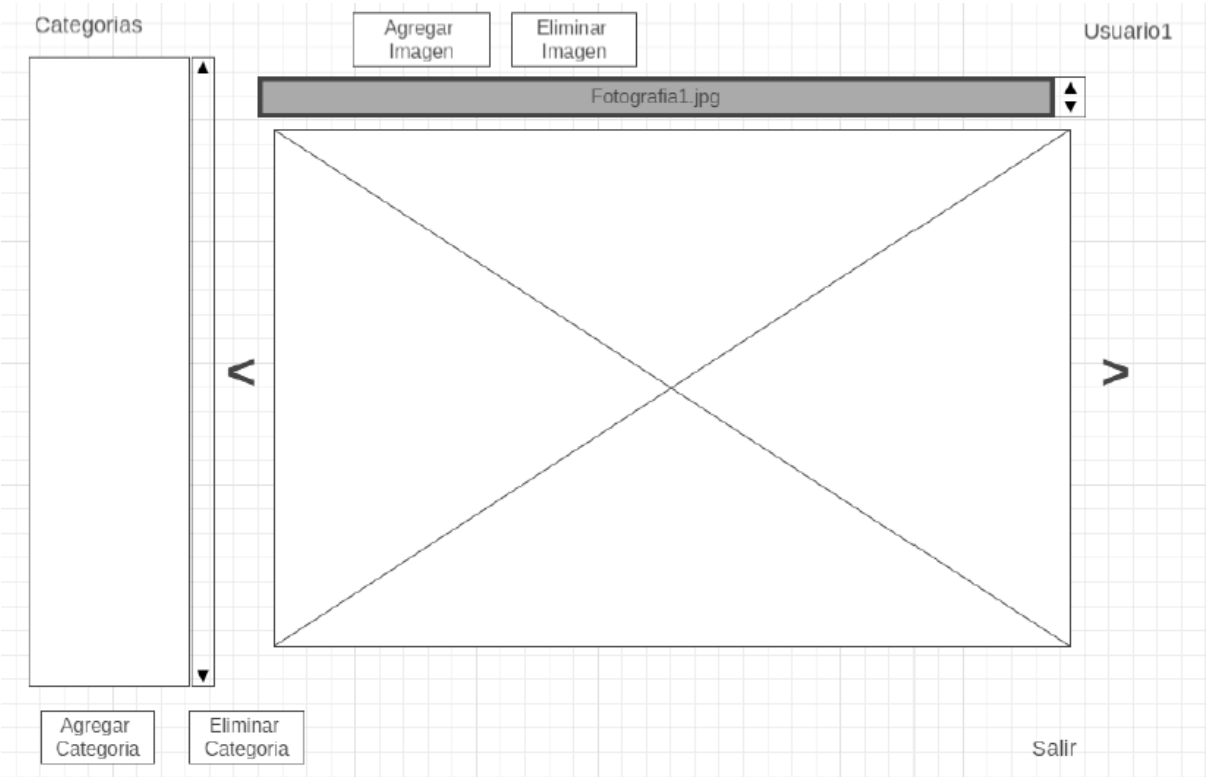
2. Biblioteca

El área de biblioteca es capaz de almacenar fotografías por categoría y por usuario, en la pantalla principal se pide un nombre de usuario y si el usuario no existe crea uno, al final del proyecto en la sección Almacenamiento se detalla la forma en la cual será implementado el almacenamiento en el proyecto.

Las categorías existen únicamente para el usuario que las crea, al inicio solo se tiene una categoría llamada “General”

Funcionalidades:

- Agregar imagen : puede agregar una o varias imágenes a la vez.
- Eliminar imagen
- Agregar categoría
- Eliminar categoría
- Siguiente imagen (Cambia de imagen a la siguiente)
- Imagen anterior (Regresa a la imagen anterior)



3. Editor

El módulo de edición puede cargar cualquier imagen en formato .jpg y ejecutar las operaciones que el usuario desee.

El módulo cuenta con un sistema de carpetas para que las imágenes resultantes sean guardadas y una carpeta temporal donde se almacenarán las conversiones de imágenes.

Importante: Para poder manipular las imágenes deben de ser convertidas a BMP y luego convertir la imagen manipulada resultante en una imagen JPEG como salida para el usuario. Se tiene que hacer uso de las clases provistas en el material.

Interfaz gráfica Editor

A. JPEG a BMP y viceversa

La primera tarea que debe poder hacer su proyecto es poder leer una imagen en formato JPEG y convertirla a BMP, y viceversa. Para esta parte del proyecto (y solamente en las partes que se le indique), se le permite utilizar las clases ImageIO y BufferedImage ya definidas en Java. Utilizando estas clases, usted debe:

1. Leer una imagen JPEG y generar una copia de la imagen en formato BMP.
2. Leer una imagen BMP y generar una copia de la imagen en formato JPEG.

Las imágenes convertidas a otro formato deben llevar el nombre de la imagen original con el prefijo converted, de tal manera que si su imagen se llama colores.jpeg, la imagen generada en formato bmp se llamaría converted-colores.bmp y debe ser almacenada en la carpeta temporal de su proyecto.

Mas adelante se le indicará como se ejecutará esta funcionalidad en su programa

B. Copia – JPEG

Para esta parte del módulo debe poder leer una imagen en formato JPEG y generar una copia de la misma. La copia no debe ser manipulada de ninguna manera. En este caso debe utilizar las clases de Java antes mencionadas para poder leer la imagen jpeg, sin embargo no las puede utilizar para realizar la copia. Para esto debe hacer lo siguiente:

1. Lea la imagen JPEG utilizando las clases de Java
2. Convierta la imagen a BMP
3. Cree una copia de esa imagen (BMP), utilizando la clase que se le provee
4. Convierta esa copia a JPEG

Dentro del material que se le provee para este proyecto, hay una clase llamada BMPHandlerCopy, la cual debe usar para hacer la copia.

La imagen copia debe llevar el nombre de la imagen original con el prefijo copia, de tal manera que si su imagen se llama colores.jpeg, la imagen copia se llamaría copia-colores.jpeg.

C. Rojo - Verde - Azul - Sepia

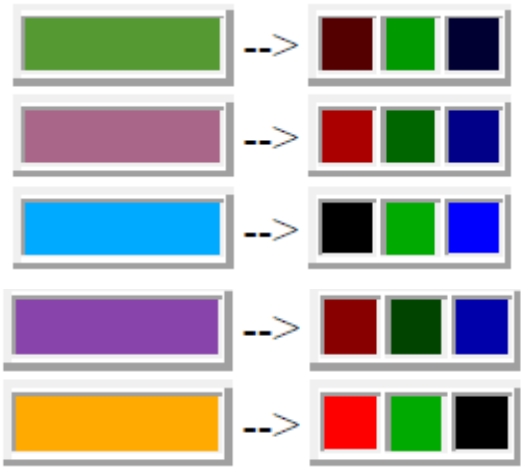
En la primera parte del proyecto, usted entenderá cómo están formados los colores en cada pixel. Su objetivo es separar su imagen original en tres tonos, rojo, verde y azul.

Imagen:

Usted va a trabajar inicialmente con una imagen en formato JPG, esta debe convertir a BMP, manipularla de esa manera y luego generar a partir de imágenes BMP, las imágenes JPEG correspondientes. A continuación, se le explica cómo está definida una imagen BMP.

Las imágenes están divididas en píxeles, en cada uno de los cuales se guarda un color específico dependiendo de su posición en la imagen. Este color, se forma por tonalidades de rojo, verde y azul. La tonalidad (cantidad de color) es un número del 0 al 255. Por ejemplo, si un píxel fuera a guardar el rojo puro, este se calificaría como R=255, V=0, A=0 que significa que el color se representa por 255 de rojo, 0 de verde y 0 de azul.

Supongamos los siguientes pixeles, cada uno tiene un color, y al lado mostramos las intensidades de rojo, verde y azul que se utilizan para poder formar ese color específico:

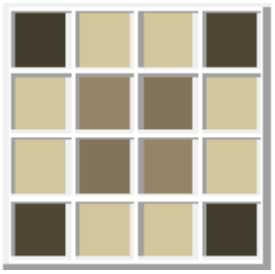


Para esta parte su tarea es leer del formato bmp de la imagen, la matriz de píxeles, guardando los datos de tal forma que separe la matriz de rojo, verde y azul. Tome en cuenta que el tamaño de dicha matriz será el número de píxeles en la imagen, y por lo tanto debe leer el tamaño de la imagen en el header. Para esta parte, debe utilizar lo anterior para generar las siguientes imágenes:

1. Tres imágenes sacadas de la imagen original, una que contenga solo las tonalidades de rojo, otra las tonalidades de verde, y otra las tonalidades de azul.
   Por ejemplo tenemos la siguiente imagen:

   Usted debería generar las siguientes imágenes:

2. Una imagen en tonalidad sepia, resultante de la imagen original. La cual se vería algo parecido a esto:



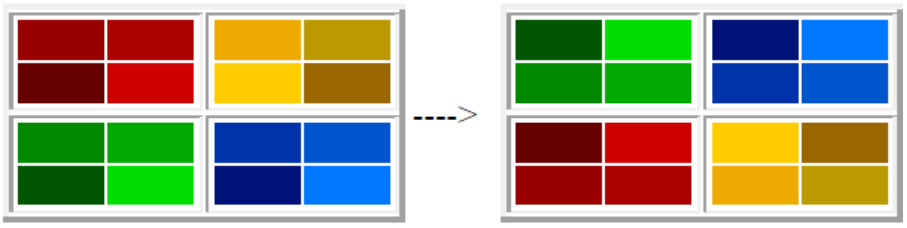
Entonces, para esta parte usted tiene que leer una imagen JPEG y generar solamente 4 imágenes a partir de esa (recuerde que tiene que convertir a BMP , generar 4 imágenes BMP y convertirlas a JPG). Si el nombre del archivo de la imagen fuera Image.jpeg, las imágenes para esta fase deben tener los siguientes nombres respectivamente:

1. Red-Image.jpeg, Green-Image.jpeg, Blue-Image.jpeg
2. Sepia-Image.jpeg

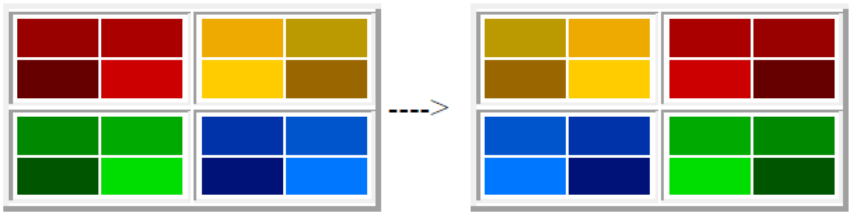
D. Modificar Imagen

Tiene que generar las siguientes imágenes a partir de la imagen original (JPEG). Recuerde que debe hacer lo mismo, leer en JPEG, convertir a BMP, manipular en BMP y luego convertir imágenes generadas a JPEG.

1. Imagen al revés de la original, rotando la imagen 180 grados sobre la línea horizontal de en medio. (Hrotation-Image.jpeg)



2. Al revés de la imagen original, pero rotando 180 grados en el eje vertical (Vrotation-Image.jpeg)



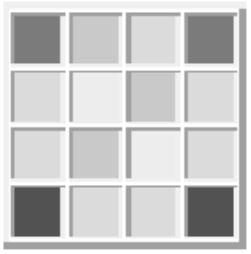
E. Blanco y negro

Tiene que generar las siguientes imágenes a partir de la imagen original (JPEG). Recuerde que debe hacer lo mismo, leer en JPEG, convertir a BMP, manipular en BMP y luego convertir imágenes generadas a JPEG.

El objetivo es convertir la imagen de colores a blanco y negro

Entrada:

Salida:



Para este módulo se le proveen las siguientes clases:

1. **ImageHandler:** Clase abstracta de la cual todos los handlers de este proyecto deben heredar, y ser forzados a implementar los métodos de lectura de imagen original y generación de imágenes.
2. **BmpHandlerCopy:** Handler que hereda de ImageHandler y crea una copia exacta de la imagen original en formato BMP. Esta es solo para darle una guía de cómo debería hacer con las demás.
3. **JPEGHandler:** Clase que ejecuta los handlers con un método estático.

Considerando lo anterior, su proyecto debe estar dividido en las siguientes clases:

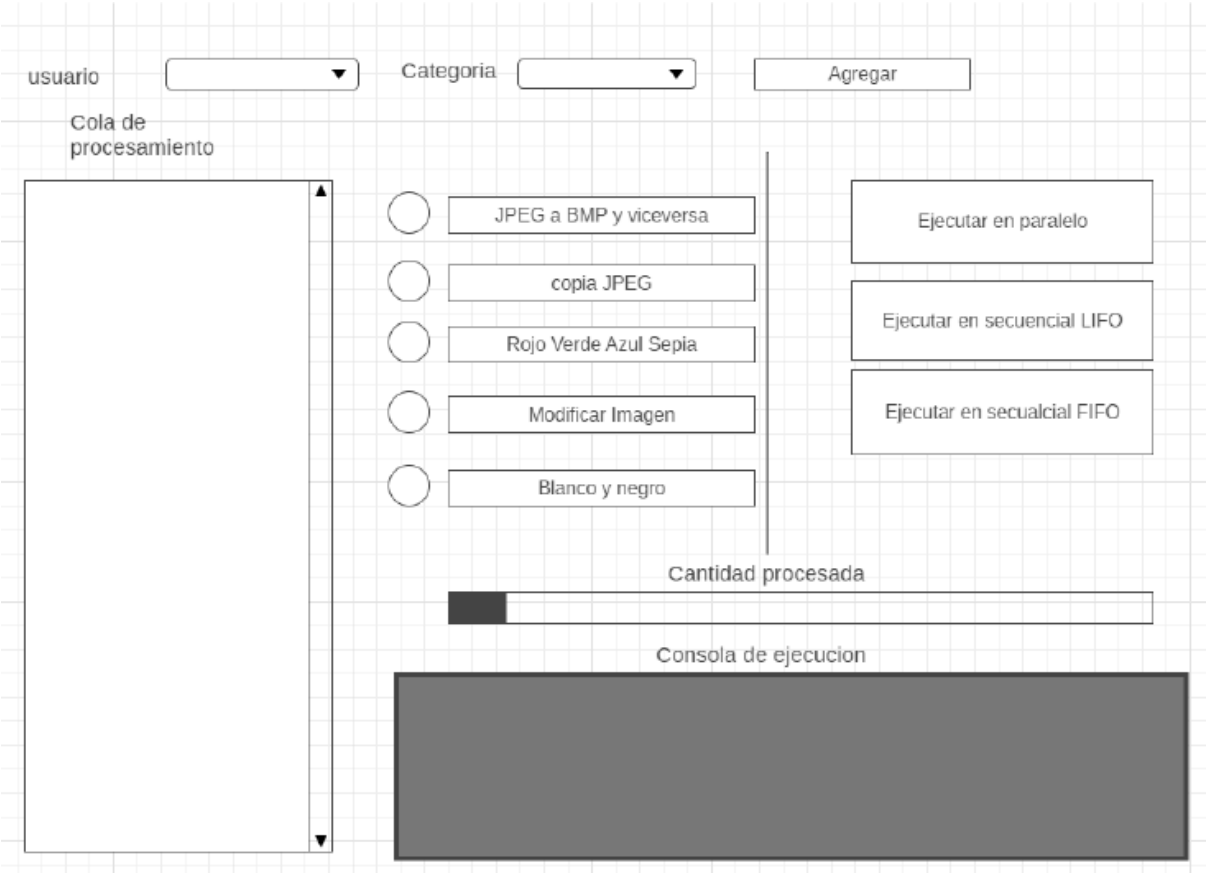
- Clase **JPEGtoBMPImage**, la cual se encarga de leer una imagen JPEG y genera una copia pero en formato BMP.
- Clase **BMPtoJPEGImage**, la cual se encarga de leer una imagen BMP y genera una copia pero en formato JPEG.
- Clase **JPEGImageCopy**, la cual se encarga de leer una imagen JPEG y crear una copia.
- Clase **JPEGImageHandlerColors**, la cual se encarga de generar las imágenes especificadas de colores y sepia.
- Clase **JPEGImageHandlerRotator**, la cual se encarga de generar las imágenes especificadas de transformación.
- Clase **JPEGImageHandlerBN**, la cual se encargue de generar las imágenes especificadas de blanco y negro.

- Clase llamada **JPEGHandler**, la cu ecutar las clases que se les especifique, la cual será invocada en el botón ejecutar.

Las clases **JPEGtoBMPImage**, **BMPtoJPEGImage**, **JPEGImageCopy**, **JPEGImageHandlerColors**, y **JPEGImageHandlerRotator**, **JPEGImageHandlerBN** deben heredar de la clase abstracta **ImageHandler**, e implementar los métodos abstractos de lectura y generación de archivos, los cuales se mandan a ejecutar por el método **runHandler** de la clase **JPEGHandler**.

4. Convertidor (Procesamiento por lotes)

En esta sección se implementará un procesamiento de las funciones anteriores, pero por lotes, para esto se seleccionara un usuario, luego se seleccionara alguna de sus categorías y se agregara las fotos de su categoría a la cola de procesamiento.



- **Agregar imágenes de categoría:** Agrega las imágenes que estén en la categoría y el usuario seleccionadas a la cola de procesamiento, se pueden agregar varias categorías de él mismo o distintos usuarios a la cola.
- **Seleccionar tipo de procesamientos:** Los lotes que estén en la cola de procesamiento pueden procesarse por “JPG a BMP” “Copia JPEG” “Rojo Verde Azul Sepia” “Modificar Imagen” “Blanco y negro”
- **Tipos de ejecución:**
    - **Ejecutar en paralelo:** Para esta opción se tienen que procesar todas las imágenes de manera simultánea (Se requiere el uso de hilos)

    - **Ejecutar en secuencia LIFO:** Procesa las imágenes una por una procesando primero la última que entró y terminado por la primera
    - **Ejecuta en secuencial FIFO:** Procesa las imágenes una por una, de la forma que fueron agregadas a la cola de procesamiento
- **Cantidad procesada:** Es una barra que indica el porcentaje de imágenes que han sido procesadas.
- **Consola de ejecución:** Imprime el nombre de la imagen procesada actualmente y que filtro le está aplicando.

Almacenamiento:
Esta sección aplica para todo el proyecto. Todas las estructuras heredarán de la clase abstracta **EstucturaDeDatos** e implementarán los métodos que requieran.

El programa es capaz de guardar y auto cargar la información al momento de abrir y cerrar, para esto se tiene que utilizar una implementación de la interfaz Serializable que provee java.

A continuación, se describe qué estructuras se deben de usar:

- **Biblioteca:**
  Almacenamiento usuarios: Lista Simple
  Almacenamiento categorías: Lista doblemente enlazada
  Almacenamiento imágenes dentro de las categorías: Lista circular doblemente enlazada
- **Convertidor**
    - **Cola de ejecución**: Cola simple

Requerimientos para el desarrollo del proyecto:

DOCUMENTACIÓN:

- Diagrama de Clases.

CONSIDERACIONES:

- Recuerde que en el formato BMP tiene dos bytes reservados (con valor cero) al final
- Recuérdese que el formato va al revés, es decir que BMP empieza a contar desde el píxel inferior izquierdo.
- Recuerde que en el formato BMP tiene dos bytes reservados (con valor cero) al final.
- Las clases y código entregado son de uso obligatorio.

RESTRICCIONES:

- La aplicación debe ser desarrollada en el lenguaje de programación Java.
- No se permite el uso de las funciones de librerías java.awt.Image o Javax,ImageIO o similares a excepción de los lugares donde se indica que puede ser utilizado. El estudiante debe implementar sus propias funciones de abrir, guardar y las modificaciones de las imágenes.
- La manipulación de imágenes en bmp tiene que ser creada por los estudiantes esta prohibido el uso de librerías externas.
- El código de las clases proporcionadas en el material NO puede ser modificado por los estudiantes
- Las estructuras de datos deben de ser creadas por los estudiantes
- No se permite el uso de estructuras que implemente Java (ArrayList, LinkedList, etc.).
- No se permite utilizar código copiado o bajado de Internet.
- El IDE por utilizar queda a discreción del estudiante (se recomienda el uso de NetBeans).
- Copias obtendrán una nota de 0 y reportará a la Escuela de Ciencias y Sistemas.

FECHA Y FORMATO DE ENTREGA:

Se debe de crear un repositorio privado en Github con el siguiente nombre:

- [IPC1]S12023-Proyecto2-<# de carnet>

Se debe de agregar al auxiliar a su repositorio como colaborador

- Sección A: **JoseDaniel12**
- Sección B: **Diego1399**
- Sección C: **josesantizo-dev**
- Sección D: **edushowy**
  lobje17
- Sección E: **eAntillon**
- Sección F: **Freddy-201212854**
- Sección G: **crisborr8**

Fecha de entrega:

- 05/05/2023

NO SE ACEPTAN ENTREGAS TARDÍAS