



Área de Ingeniería en Computadores

Algoritmos y Estructuras de Datos II

Proyecto II:

Paint++

Profesor:

Jose Isaac Ramírez Herrera

Estudiantes:

-Carlos Andrés Mata Calderón – 2019033834

-Joan Jesús Ugalde Zárate – 2020023722

-Li Hao Allan Chen Liang – 2019049482

Grupo:

1

Semestre I,2022

Tabla de contenido

<i>Introducción</i>	3
<i>Diagrama de Clases</i>	4
<i>Estructuras de Datos implementadas</i>	5
Vector	5
Lista Doblemente Enlazada:	5
Lista:	¡Error! Marcador no definido.
Matriz	5
Struct:	5
Puntero:	5
Hashtable	6
<i>Algoritmos Implementados</i>	6
<i>Problemas encontrados en forma de bugs</i>	9
<i>Jira</i>	9
<i>Referencias</i>	10

Introducción

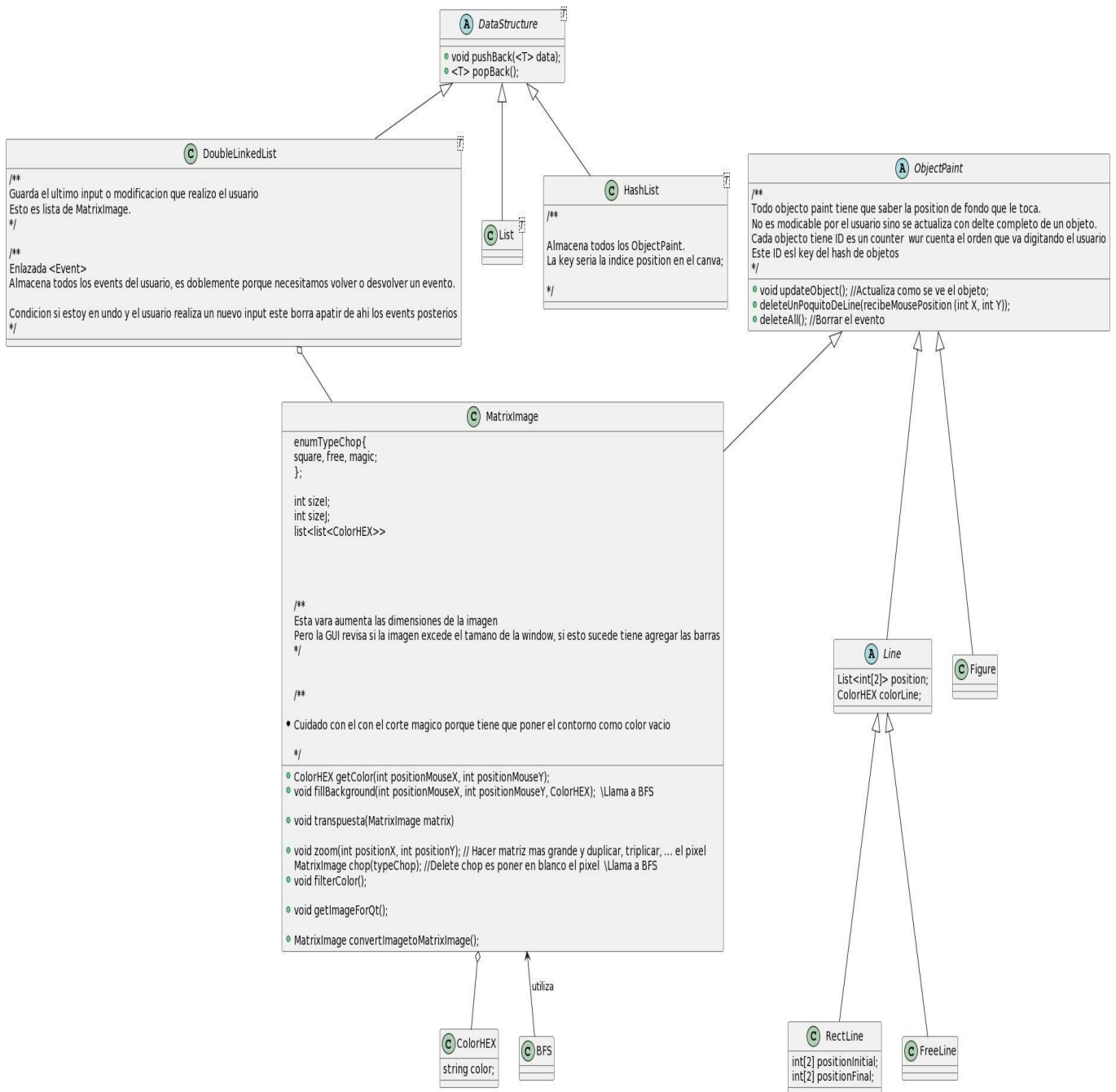
Para el presente trabajo, se desarrolla un programa similar a Microsoft Paint, utilizando el sistema operativo Linux y el lenguaje de programación C++. El programa se llama Paint++ y cuenta con varias herramientas en donde el usuario puede dibujar, insertar imágenes e interaccionar sobre un canvas blanco, el cual el usuario puede elegir el tamaño.

Al ser un trabajo grupal, se utiliza el sistema de control de versiones GitHub para un mejor orden y manejo del código. Se trabaja con varios branches, en donde cada Branch es utilizado para trabajar en una tarea en específico. Estas tareas están definidas en la herramienta para gestionar proyectos Jira, en donde se desglosa el proyecto en varias tareas y se le asigna a cada miembro del proyecto, esto para una desarrollarlo eficientemente.

El programa permite cargar y guardar imágenes en formato Bitmap (BMP), el cual es un formato de mapa de bits y se pueden comprimir sin pérdida de información. Para esto se utiliza una matriz en donde cada casilla va a simular el pixel de la imagen y va a contener los datos del color del pixel.

Para una mejor experiencia al utilizar el programa, se implementaron varias herramientas para interactuar con las imágenes. Estas herramientas son: agregar figuras geométricas y modificar sus dimensiones, borrador para poder borrar trazos o figuras creadas por el usuario, rotar el canvas o la imagen, color picker para obtener el color del pixel seleccionado, Paint-fill para pintar el área seleccionada en donde el área es definido por pixeles vecinos del mismo color, zoom y selección mágica. Además, tiene 2 estilos para dibujar: lápiz y lapicero. El lápiz permite al usuario elegir el tamaño y color del trazo y puede dibujar libremente. Al igual que el lápiz, el lapicero permite al usuario elegir el tamaño y color, pero solo permite dibujar líneas rectas.

Diagrama de Clases



<https://drive.google.com/file/d/1IHTNy77O4gxrYwdODVPuVkkmdRwKKcI9/view>

Estructuras de Datos implementadas

Vector:

Los vectores son lo mismo que las matrices dinámicas con la capacidad de cambiar su tamaño automáticamente cuando se inserta o elimina un elemento, y el contenedor maneja automáticamente su almacenamiento. Los elementos vectoriales se colocan en almacenamiento contiguo para que se pueda acceder a ellos y recorrerlos mediante iteradores. En los vectores, los datos se insertan al final. Insertar al final toma un tiempo diferencial, ya que a veces puede ser necesario extender la matriz. Eliminar el último elemento lleva solo un tiempo constante porque no se cambia el tamaño. Insertar y borrar al principio o en el medio es lineal en el tiempo.

Lista Doblemente Enlazada:

Es una lista lineal, en la cual cada nodo puede acceder al nodo anterior o al siguiente. Se utiliza esta estructura de datos para implementar la función de undo y redo. Para esto se necesita guardar los registros de actividad del usuario, y así si el usuario desea hacer un undo entonces se devuelve al nodo anterior y si desea un redo, avanza al nodo siguiente.

Array:

Un Array es una colección de elementos del mismo tipo ubicados en memoria, los cuales se manejan mediante índices para poder tener una referencia y manipular el array.

Matriz:

Una matriz es una estructura de datos conformada por filas y columnas, o en otras palabras una matriz es un vector conformado por más vectores. La matriz va a contener los datos de objetos Tarjeta y se accede a ella mediante punteros. Se utiliza esta estructura para guardar los datos de la imagen, en donde se va a guardar un pixel por casilla.

Struct:

Esta estructura de datos permite que el usuario pueda almacenar diferentes tipos de datos, excepto null y no permite usar como herencia de otra clase. Además “struct” por defecto va a ser público.

Puntero:

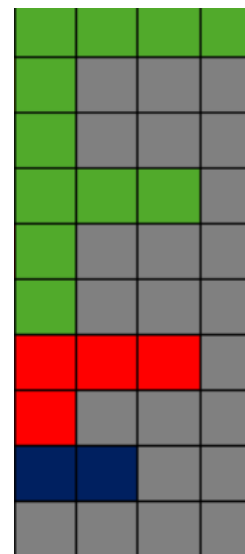
Los punteros son variables que se utilizan para almacenar la dirección de memoria de un objeto y que se deben declarar antes de ser utilizados. Estas variables funcionan para acceder a los datos que contiene la variable, cambiar su valor o para tener una referencia de la dirección de memoria en donde se ubica el objeto.

Algoritmos Implementados

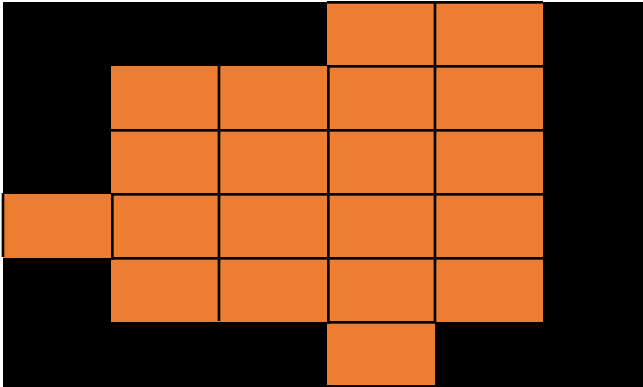
Algoritmo de búsqueda por anchura (BFS):

El algoritmo de búsqueda por anchura FIFO (First In, First Out), recorre los elementos por niveles, el nivel 1 es la raíz padre, el nivel 2 son los hijos, el nivel 3 son los hijos de los hijos y así sucesivamente. El algoritmo de búsqueda por anchura comienza a buscar el elemento del primer nivel y luego busca de izquierda a derecha los del siguiente nivel, así hasta encontrar el elemento solicitado.

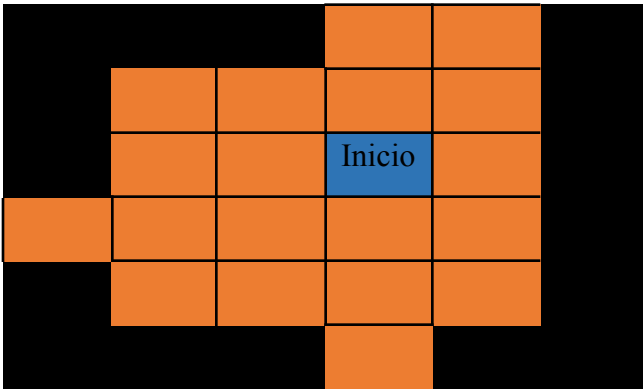
En el presente proyecto, se necesita implementar la función Paint-fill. Esta función permite al usuario rellenar de pintura, áreas conectadas del mismo color. Como se puede ver en la imagen, se selecciona el pixel (2,3) el cual es color negro. Entonces se busca los pixeles adyacentes con el mismo color, en este caso negro. Al terminar la búsqueda se reemplaza todos los pixeles negros adyacentes por pixeles de color gris.



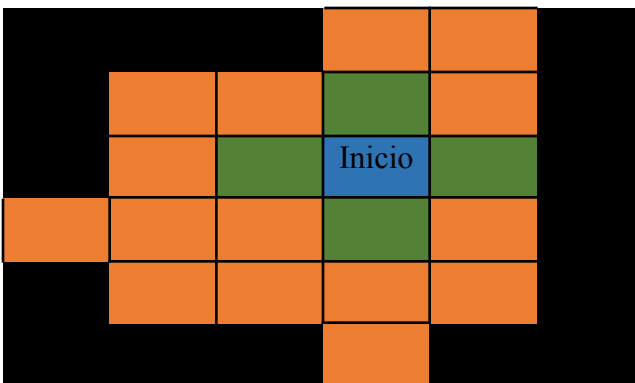
Se utiliza este algoritmo porque es más fácil y rápido encontrar los colores similares adyacentes al pixel seleccionado.



Se selecciona un pixel.



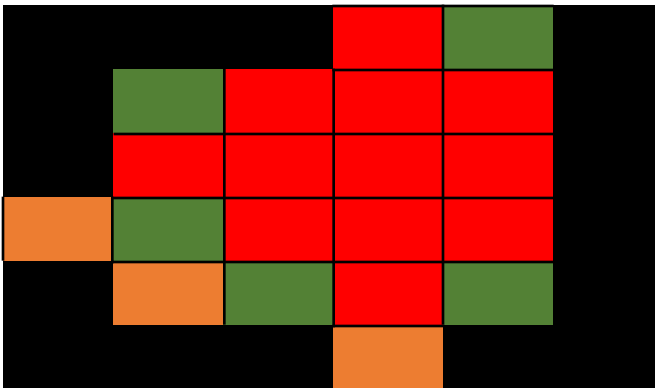
Se buscan los pixeles adyacentes, como son del mismo color entonces se guardan en una lista.



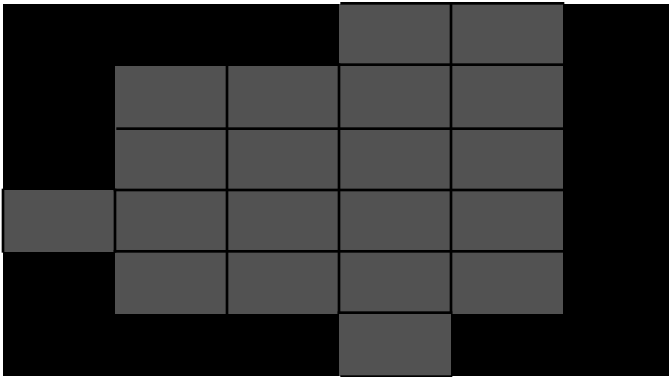
Se sigue repitiendo el paso anterior, si se encuentra un pixel de un color diferente, entonces se descarta.



Se repite este paso hasta verificar que el alrededor del área sea de diferente color.



Por último, se reemplaza el color.



Anexo

<https://drive.google.com/file/d/1WN5ylXQg2RybVWmUSg-BcLOYDW047wlJ/view?usp=sharing>

Problemas encontrados en forma de bugs

<https://github.com/Carlos12001/paintpp/issues>

The screenshot shows the GitHub interface for the repository **Carlos12001 / paintpp**. The **Issues** tab is selected, showing 2 open issues. The issues are:

- Undo y Redo dejan de funcionar** (#2), opened May 24, 2022 by Carlos12001.
- Rotacion a la mitad** (#1), opened May 24, 2022 by Carlos12001.

The interface includes a search bar, navigation links (Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, Settings), and a footer with GitHub's terms and privacy policy.






Jira



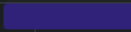
























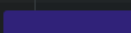













<https://achen031199.atlassian.net/jira/software/projects/PJALLAC/boards/1/backlog?selectedIssue=PJALLAC-53>

Projects / PAINT++ JALAC

Roadmap





Search roadmap 🔍

     Status category ▾























Epic	APR	MAY
	Tablero Sprint 1	
 PJALLAC-52 Destrasfo... DONE 		
 PJALLAC-16 Manejar u... DONE 		
 PJALLAC-18 Cargar las ima... DONE		
▼  PJALLAC-6 Implementar función Paint...		
 PJALLAC-12 Implementar A... DONE		
 PJALLAC-13 Manejar los co... DONE		
▶  PJALLAC-2 Implementar función Color...		
▶  PJALLAC-9 Herramienta para rotar el ...		
▶  PJALLAC-11 Funcionalidad de la Herr...		
▼  PJALLAC-25 Crear Canva		
 PJALLAC-35 Crear Col... DONE 		
 PJALLAC-32 Mostrar M... DONE 		
 PJALLAC-33 Crear en Blan... TO DO		
 PJALLAC-34 Utilizar List<Li... TO DO		
▼  PJALLAC-26 Interfaz QT +		
 PJALLAC-53 Crear pai... DONE 		
 PJALLAC-56 Crear sel... DONE 		
 PJALLAC-54 Crear Bot... DONE 		
 PJALLAC-57 Crear bot... DONE 		
 PJALLAC-55 Crear figu... DONE 		
▼  PJALLAC-28 Borrar trazos		

Projects / PAINT++ JALAC

Backlog

Search backlog Q     Epic ▾

▼ **Tablero Sprint 1** 28 Apr – 9 May (11 issues) 0 0 0 Complete sprint ...

	PJALLAG-51	Transformar archivo BMP a MatrixHEX	GUARDAR CANVAS/IMAGEN EN ...	DONE ▾	
	PJALLAG-35	Crear ColorHex	CREAR CANVA	DONE ▾	
	PJALLAG-19	Crear una lista doblemente enlazada	LLEVAR REGISTRO DE TODOS L...	DONE ▾	
	PJALLAG-52	Destrasformarlo a archivo BMP	GUARDAR CANVAS/IMAGEN EN ...	DONE ▾	
	PJALLAG-32	Mostrar MatrixImage	CREAR CANVA	DONE ▾	
	PJALLAG-53	Crear paint primitivo	INTERFAZ QT	DONE ▾	
	PJALLAG-56	Crear seleccion de colores	INTERFAZ QT	DONE ▾	
	PJALLAG-54	Crear Botones de las funciones	INTERFAZ QT	DONE ▾	
	PJALLAG-57	Crear boton de guardado	INTERFAZ QT	DONE ▾	
	PJALLAG-55	Crear figuritas para la funcion figura	INTERFAZ QT	DONE ▾	
	PJALLAG-16	Manejar una imagen por pixeles	GUARDAR CANVAS/IMAGEN EN ...	DONE ▾	

+ Create issue

Github

<https://github.com/Carlos12001/paintpp.git>

Referencias

- [[QGridLayout]](<https://chowdera.com/2021/04/20210403063919149m.html>)
- [[README.md Links]](<https://www.codegrepper.com/code-examples/whatever/how+to+create+link+in+readme.md>)
- [[unsigned char]](<https://stackoverflow.com/questions/75191/what-is-an-unsigned-char>)
- [[>> in numbers]](<http://www.cplusplus.com/forum/beginner/95670/>)
- [[unsigned char propieties]](<https://www.geeksforgeeks.org/maximum-value-of-unsigned-char-in-c/>)
- [[differences between #include "..." and include<...>]](<https://stackoverflow.com/questions/21593/what-is-the-difference-between-include-filename-and-include-filename>)
- [[implement our own Vector Class in C++]](<https://www.geeksforgeeks.org/how-to-implement-our-own-vector-class-in-c/>)
- [[create bitmap in c++]](<https://www.youtube.com/watch?v=vqT5j38bWGg>)
- [[iniatlize private static atribute]](<https://localcoder.org/error-iso-c-forbids-in-class-initialization-of-non-const-static-member>)
- [[What is "->" after function declaration?]](<https://stackoverflow.com/questions/12417197/what-is-after-function-declaration>)
- [[get current dir]](<https://www.delfistack.com/howto/cpp/get-current-directory-cpp/>)
- [[glog use 1]](<https://codeyarns.com/tech/2017-10-26-how-to-install-and-use-glog.html>)
- [[glog use 2]](<https://rpg.ifi.uzh.ch/docs/glog.html#:~:text=Google%20glog%20defines%20a%20series,verbose%20logging%20levels%2C%20and%20more.>)
- [[glog use 3]](<https://programming.vip/docs/basic-usage-of-glog.html>)
- [[under cursor widget qt]](<https://stackoverflow.com/questions/27363267/get-all-widgets-under-cursor>)
- [[number to string in c++]]([https://www.simplilearn.com/tutorials/cpp-tutorial/int-to-string-cpp#:~:text=The%20next%20method%20in%20this,data%20type%20into%20a%20string.&text=The%20to_string\(\)%20method%20is,string%3E%20or%20.](https://www.simplilearn.com/tutorials/cpp-tutorial/int-to-string-cpp#:~:text=The%20next%20method%20in%20this,data%20type%20into%20a%20string.&text=The%20to_string()%20method%20is,string%3E%20or%20.))
- [[create bin directory with cmake]](<https://stackoverflow.com/questions/6594796/how-do-i-make-cmake-output-into-a-bin-dir>)
- [[not resizeable window in qt]](<https://www.youtube.com/watch?v=--NkRG0UZTg>)
- [[color selector in qt]](https://www.youtube.com/watch?v=rtgOb03xl_U)
- [[qt input dialog text]](https://www.youtube.com/watch?v=wj1Ua_K1kY)
- [[not resizeable window]](<https://www.youtube.com/watch?v=--NkRG0UZTg>)
- [[Check for implement the BFS]](<https://www.geeksforgeeks.org/largest-connected-component-on-a-grid/?ref=gcse>)

