

Universidad De San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas

Estructura de Datos
Sección "A"



“MANUAL TÉCNICO”

Hugo Sebastian Martínez Hernández

Carné: 202002793

Objetivos

General:

Brindar al lector una guía que contenga la información del manejo de clases, atributos, métodos y del desarrollo de la aplicación desarrollada en Java para facilitar futuras actualizaciones y modificaciones realizadas por terceros.

Específicos:

- Dar más información al lector de las herramientas utilizadas para el desarrollo de la app como lo puede ser el editor, SO, entre otros.
- Proporcionar al lector una idea más precisa de los métodos y clases creadas para el desarrollo de la aplicación.

Introducción

Este manual técnico tiene como finalidad dar a conocer al lector que pueda requerir hacer modificaciones futuras al software el desarrollo de la aplicación indicando el IDE utilizado para su creación, su versión, requerimientos del sistema, etc...

La aplicación tiene como objetivo la generación de imágenes por medio del arte de pixel o también conocido como pixel art, que consiste en generar imágenes por medio de pixeles. Esta aplicación tendrá dos tipos de usuario: el administrador y el usuario. Los administradores podrán realizar carga de usuarios, crear y modificar, además también podrán ver ciertos reportes que le fueron asignados a su tipo de usuario. Los usuarios tendrán la capacidad de cargar sus capas con las crearán sus imágenes, podrán cargar sus imágenes y sus álbumes. A partir de la información que cargue el usuario se podrán generar las distintas imágenes y sus respectivos reportes.

Descripción de la Solución

Para poder desarrollar esta práctica se analizó lo que se solicitaba y se trató de ir un poco más allá realizando algunas validaciones para que el programa funcione de una manera mucho mejor y que sea más atractivo de usar para el usuario.

Entre las consideraciones encontramos con mayor prioridad están:

- Que se cumpliera con la carga de todos los archivos solicitado(usuarios, capas, imágenes y álbumes) por medio del archivo JSON que será proporcionado por la escuela de ciencias y sistemas.
- Se podrá cargar cualquier archivo JSON con n cantidad de datos que cumpla con los lineamientos dados en el proyecto.
- Cumplir con las especificaciones de las estructuras de datos para así cumplir con el objetivo principal de este proyecto.
- Realizar los reportes solicitados para así poder sacar estadísticas de los clientes que visitan la empresa.

Descripción de algunas funciones principales

- Carga Masiva: esta clase contendrá las 4 cargas masivas que se realizan dentro de la aplicación, la carga de capas carga de imágenes, carga de álbumes y la carga de clientes. Cada una de estas cargas se almacenará en su respectiva estructura para no perder los datos.

Listas solicitadas para el funcionamiento del programa:

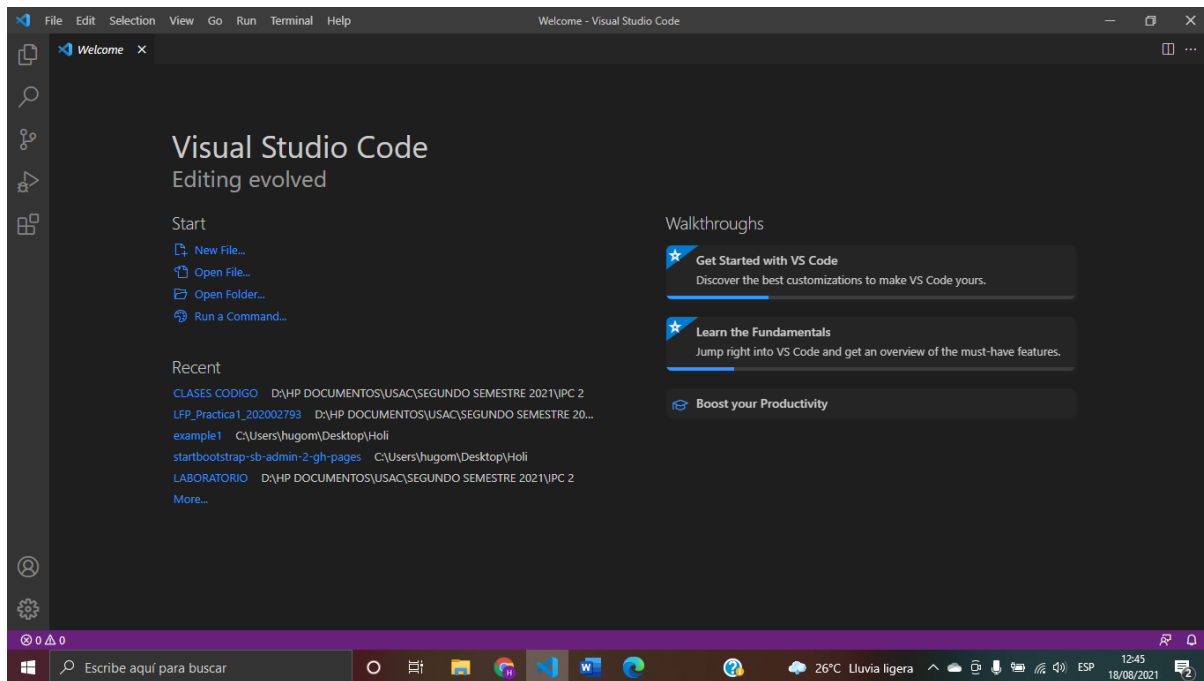
- Árbol B: esta estructura será la encargada de almacenar a todos los usuarios que se registren dentro de la aplicación. El árbol es de orden 5 es decir que puede almacenar un máximo de 4 clientes por cada rama o página y podrá tener un máximo de 4 hijos cada página. En esta estructura se implementaron el método de crear, modificar y buscar.
- Árbol Binario de Búsqueda (ABB): en esta estructura se guardan las capas de los usuarios, cada capa se guardará con su respectivo ID que debe ser único y además tendrá un atributo de tipo matriz dispersa que se encargará de guardar la matriz que conforma a esa capa. A continuación, se mencionan algunos métodos importantes de esta estructura:
 - Insertar: la función de insertar recibirá como parámetro un valor de tipo Capa que tendrá el id y la matriz dispersa.
 - AgregarAMatrizDispersa: este método es importante para la generación de la imagen. El método recibe como parámetro una matriz dispersa que es la de la imagen que se formara con dicha capa y agrega todos los pixeles a esa matriz de la imagen que se está formando.
 - Los demás métodos sirven para recorrer el árbol y retornaran diferentes valores, la mayoría sirve para el área de reportes y otros simplemente para verificar la estructura.
- Árbol auto balanceado (AVL): esta estructura almacenara las imágenes del usuario, estas imágenes se guardarán con un id y con el listado de capas que pertenecen a dicha imagen. A continuación, se mencionarán algunos métodos importantes para esta estructura:
 - Insertar: este método será el encargo de empezar el proceso de inserción en árbol AVL, recibirá como parámetro un valor de tipo imagen.
 - RotacionDerecha: este método se encargará de realizar una rotación a la derecha en caso el árbol la necesite para estar balanceado.
 - RotacionIzquierda: este método se encargará de realizar una rotación a la izquierda en el caso que el árbol la necesite para estar balanceado
 - dobleIzqDer: este método realizara una doble rotación, primero rota a la izquierda y luego a la derecha.
 - dobleDerIzq: este método realizara una doble rotación, primero rota a la derecha y luego a la izquierda.

- Balancear: este método será la medula espinal de las rotaciones, ya que se encargará de realizar las llamadas a las rotaciones según el árbol lo necesite, evaluará las alturas y con ello tomará la decisión de qué hacer con el árbol.
 - Eliminar: este método servirá para poder eliminar una imagen de la estructura AVL.
 - encontrarMenor: este método será de utilidad para encontrar el menor de los mayores que será de utilidad en caso el árbol la necesite en el momento de eliminar.
 - Todas las demás funciones son útiles para la generación de los reportes.
- Matriz Dispersa: esta matriz jugo en papel importante en la generación de las imágenes, en esta estructura se almacenaban los colores que tenía cada capa y a partir de ahí se pueden generar todas las imágenes que el usuario solicite. A continuación, algunas funciones útiles para esta estructura:
 - insertarEnFila: este método es el encargado de insertar un valor en una fila.
 - insertarEnColumna: este método es el encargado de insertar un valor en una columna.
 - encontrarColumna: este método servirá para poder verificar si la columna ya está creada.
 - encontrarFila: este método servirá para poder verificar si la fila ya está creada.
 - buscarRepetidos: método útil para saber si el valor que se quiere ingresar ya está con algún valor.
 - crearFila: este método será el encargado de crear una fila si es que no existe.
 - crearColumna: este método será el encargado de crear una columna si es que no existe.
 - insertarNodo: este método será el encargado de iniciar el proceso de inserción de un nuevo nodo en la matriz.
 - Los demás métodos son útiles para generar los reportes respectivos.
- Lista doblemente enlazada: esta estructura será la encargada de almacenar los álbumes cargados por el usuario, esta estructura tiene dos métodos principales que son: el insertar, que será el encargado de insertar un elemento al final de lista y el método de eliminar que será el encargado de eliminar un elemento de la lista doble.
 - Cola: esta estructura fue de utilidad únicamente para poder realizar algunos recorridos como por ejemplo el recorrido por niveles que se solicitaba en algunos reportes.
 - Lista simple: esta estructura fue de utilidad para guardar las imágenes que conformaban a un álbum, y además también fue útil en otras funcionalidades de la aplicación.
 - TablaHash: esta estructura fue de utilidad para guardar a todos los mensajeros, en esta tabla se almacenan por medio de la función de doble dispersión.
 - dobleDispersion: este método se encarga de resolver las colisiones en caso hubieran usando el método de doble dispersión con la ecuación $(llv \bmod 7 + 1) * i$
 - validarFactorCarga: este método verifica el factor de carga que es de 75%, cuando se alcance este 75% de llenado se aumentara el almacenamiento *2.

- ListaAdyacencia: esta estructura fue de utilidad para realizar los grafos, esta lista es una lista de listas.

IDE

El IDE con el que se desarrolló practica fue Visual Studio Code versión 1.59.0, esto es porque al desarrollador le resulta más fácil poder manejar los archivos, correr el programa y por comodidad del programador, sin embargo, se deja a criterio del programador el IDE de su preferencia.



Requerimientos:

- Instalar JAVA 1.8.0_111, o versiones similares no tan desactualizadas ya que podría dar algún tipo de problema en su ejecución
- Las especificaciones del equipo utilizado para el desarrollo de la práctica se verán reflejadas en la siguiente imagen:

Acerca de

El equipo está supervisado y protegido.

[Ver detalles en Seguridad de Windows](#)

Especificaciones del dispositivo

HP Pavilion Laptop 15-cw1xxx

Nombre del dispositivo	LAPTOP-53F8AST2
Procesador	AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz
RAM instalada	12.0 GB (9.92 GB utilizable)
Id. del dispositivo	8C08C343-367E-4478-92D3-C4A43A3A6CD4
Id. del producto	00327-30712-16056-AAOEM
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil	La entrada táctil o manuscrita no está disponible para esta pantalla

- Las especificaciones del sistema operativo utilizado para el desarrollo de la practica también se verán reflejadas en la siguiente imagen:

Especificaciones de Windows	
Edición	Windows 11 Home Single Language
Versión	21H2
Se instaló el	17/12/2021
Compilación del SO	22000.493
Experiencia	Paquete de experiencia de características de Windows 1000.22000.493.0
Contrato de servicios de Microsoft	
Términos de licencia del software de Microsoft	

Librerías Utilizadas

Algunas clases propias de Java que se utilizaron:

- **BufferedReader:** clase útil para leer un archivo en java.
- **Json Simple:** esta librería se utilizó para poder leer un archivo JSON.
- **Swing:** se usó para desarrollar la parte gráfica de la aplicación.