

ESCUELA CIENCIAS Y SISTEMAS

UDRAWING PAPER

---

**2**

---

*Autor:*  
Rony Ormandy Ortíz  
Alvarez



# Índice general

1.	Introducción . . . . .	2
2.	Objetivos . . . . .	3
3.	Requerimientos . . . . .	4
3.1.	Requerimientos de hardware . . . . .	4
3.2.	Requerimientos de software . . . . .	4
4.	Fundamentos Teóricos . . . . .	4
4.1.	Java . . . . .	4
4.2.	Tipos de datos abstractos . . . . .	4
5.	Paquetes . . . . .	5
5.1.	Clases . . . . .	5
5.2.	GUI . . . . .	5
6.	Métodos del árbol B . . . . .	6
6.1.	Eliminación . . . . .	6
7.	Flujo del programa . . . . .	7
7.1.	Diagrama de clases . . . . .	7
8.	Recomendación . . . . .	8

## 1. Introducción

UDrawing es una aplicación de escritorio que puede ejecutarse independientemente del sistema operativo de la computadora, por lo cual está se desarrollo en el lenguaje de programación Java, la aplicación permite a los clientes de la empresa UDrawing Paper registrar imágenes especiales construidas por capas.

La principal función de la aplicación consiste en un generador de imágenes por capas, la aplicación cuenta con un conjunto de conjunto de capas cargadas previamente y almacenadas en memoria para ser utilizadas, estas capas se utilizarán para genera imágenes hechas con pixeles, cada capa contiene la información de los distintos pixeles y al colocar una capa sobre otra estas irán formando una imagen más completa.

El sistema es capaz de generar imagen seleccionando las capas deseadas

## 2. Objetivos

- Simular la maquina de ensamblaje desarrollada por Digital Intelligence.
- Predecir el tiempo de ensamblaje de un producto.

## 3. Requerimientos

### 3.1. Requerimientos de hardware

Requerimientos mínimos de hardware de la computadora personal o laptop:

- \* Tipo de CPU: 266 MHz o superior, 2 o mas núcleos.
- \* Memoria: 4GB de RAM o superior.
- \* Se recomienda utilizar Netbeans como IDE

### 3.2. Requerimientos de software

- \* El programa puede ser instalado en cualquier sistema operativo.
- \* Tener instalado Java 8.

## 4. Fundamentos Teóricos

### 4.1. Java

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes.

### 4.2. Tipos de datos abstractos

Es un modelo matemático compuesto por una colección de operaciones definidas sobre un conjunto de datos para el modelo.

Las estructuras utilizadas y su utilidad en esta aplicación son:

1. **Árbol B:** Esta estructura se utilizará para guardar los clientes registrados en el sistema.
2. **Matriz Dispersa:** Esta estructura se utilizará para guardar la información de una capa, es decir las posiciones y colores de los píxeles utilizados para generar imágenes.
3. **Árbol binario de búsqueda (ABB):** Esta estructura se utilizará para guardar todas las capas con las cuales se pueden generar distintas imágenes.

4. **Árbol AVL:** Esta estructura se utilizará para guardar cada una de las imágenes de un cliente.
5. **Lista circular doblemente enlazada:** Esta estructura se utilizará para guardar el listado de álbumes por cada cliente, a su vez cada álbum posee una sublista de imágenes el tipo de sublista queda a discreción del estudiante.

## 5. Paquetes

El programa consta de 1 paquetes:

1. UDrawingPaperF2: contiene 3 clases.

### 5.1. Clases

#### Cliente

Tiene los datos para registrar el cliente

#### Admin

### 5.2. GUI

#### Login

Esta clase es la que inicia

## 6. Métodos del árbol B

### 6.1. Eliminación

Primero encontraremos el nodo perteneciente al árbolB, lo localizamos y lo eliminamos. Caso 1

En el caso 2 cuando eliminamos y solo queda un nodo rompiendo las especificaciones que deben de ser como mínimo dos valores en el árbolB.

Para realizar la eliminación la dividiremos en tres casos: //En la inserción es la división y en la eliminación el la unión

1. **Caso 1:** Eliminar una clave y que al eliminarla siga siendo un nodo valido.
2. **Caso 2:** Eliminar una clave y que al eliminar no siga siendo un nodo valido. Primero se presta al izquierdo y luego al derecho.
3. **Caso 3:** Cuando ningún nodo adyacente tiene la cantidad de claves suficientes para prestar.

## 7. Flujo del programa

### 7.1. Diagrama de clases

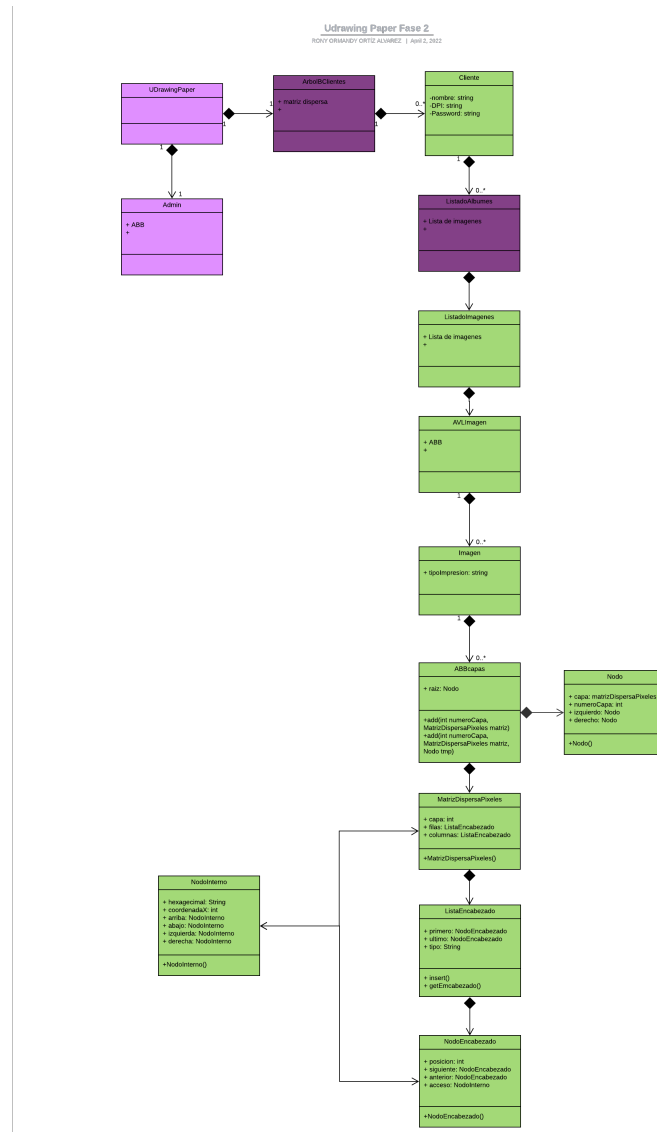


Figura 1: Diagrama de clases de la aplicación.



## 8. Recomendación

Al modificar este sistema se recomienda usar un sistema de control de versiones como Git.