

ESCUELA CIENCIAS Y SISTEMAS

FASE

---

**1**

---

*Autor:*  
Rony Ormandy Ortíz  
Alvarez



# Índice general

1.	Introducción . . . . .	2
2.	Objetivo . . . . .	3
3.	Requerimientos . . . . .	4
3.1.	Requerimientos de hardware . . . . .	4
3.2.	Requerimientos de software . . . . .	4
4.	Clases del programa . . . . .	5
4.1.	Clase cola recepción . . . . .	5
4.2.	Clase Lista de ventanillas (simple) . . . . .	5
4.3.	Clase Lista de clientes atendidos (simplemente enlazada) . . . . .	6
4.4.	Clase Cola de impresión . . . . .	6
4.5.	Clase Lista de clientes en espera (Lista de listas) . . . . .	6
5.	Diagramas. . . . .	7
6.	Conclusiones . . . . .	8

## **1. Introducción**

El presente documento consta de diversas secciones en la cuales se pretende darle una idea sobre que recursos necesita su computador para poder ejecutarlo y como está estructurado el programa programa UDrawingPaper.

## 2. Objetivo

Que la persona que le dará el soporte o mantenimiento al programa comprenda las funcionalidades que debe tener el programa UDrawingPaper, así como su estructuración.

### **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

#### **3.2. Requerimientos de software**

El programa puede ser instalado en cualquier sistema operativo.

Tener instalado Java 8 o superior.

Como IDE se recomienda usar NetBeans.

## 4. Clases del programa

El programa consta de quince clases.

### 4.1. Clase cola recepción

En esta estructura es donde se ingresan los clientes al llegar a la empresa, en el menú deberá existir una opción que permita la carga masiva de los clientes que ingresarán a la empresa. La carga masiva se realizará por medio de un archivo con extensión (json) que contendrá la siguiente estructura:

```
1  {
2      "Cliente1": {
3          "id_cliente": "1",
4          "nombre_cliente": "1",
5          "img_color": "1",
6          "img_bw": "1"
7      },
8      "Cliente2": {
9          "id_cliente": "2",
10         "nombre_cliente": "2",
11         "img_color": "3",
12         "img_bw": "0"
13     },
14     "Cliente3": {
15         "id_cliente": "3",
16         "nombre_cliente": "3",
17         "img_color": "0",
18         "img_bw": "2"
19     }
20 }
```

Figura 1: Formato Json

### 4.2. Clase Lista de ventanillas (simple)

Se implementará una lista simplemente enlazada para el número de ventanillas que estarán en el proceso de simulación, cada ventanilla contará con una pila que se

utilizará para recibir las imágenes que un cliente desea imprimir.

### **Pila de imágenes**

El funcionamiento de la pila de cada ventanilla es el siguiente:

- En cada paso que el cliente está en ventanilla se inserta una imagen a la pila, con todas las especificaciones de la impresión.
- La pila se vacía cuando el cliente sale de la ventanilla.

### **4.3. Clase Lista de clientes atendidos (simplemente enlazada)**

Se guardará la información de todos los clientes que son atendidos por todas las ventanillas habilitadas durante el proceso de simulación. En cada nodo de la lista se almacenará la siguiente información:

- Nombre del cliente
- Ventanilla que lo atendió
- Número de imágenes impresas
- Cantidad total de pasos en el sistema

### **4.4. Clase Cola de impresión**

Cada impresora maneja una cola de impresión, al momento que un cliente termina de realizar su orden de impresión, las imágenes son enviadas desde la pila de la ventanilla hacia las distintas colas de acuerdo a las especificaciones solicitadas (imagen a color o en blanco y negro), se debe implementar un mecanismo de clasificación de imágenes que permitirá que en ningún momento una imagen sea enviada a la impresora incorrecta, de tal forma que se cumplan las especificaciones solicitadas por el cliente.

### **4.5. Clase Lista de clientes en espera (Lista de listas)**

Luego de terminar el proceso en ventanilla, cada uno de los clientes se almacenará en una única lista de espera, en la que uno de los clientes (nodos) contará con una lista de imágenes impresas, estas estarán mezcladas.. Después de que la imagen se genere, esta buscará a que cliente pertenece, y se almacenará en la lista de imágenes del cliente sin importar el tipo de impresión. Una vez que la lista de imágenes del cliente esté completa, se eliminará al cliente de la lista de espera.

## 5. Diagramas.

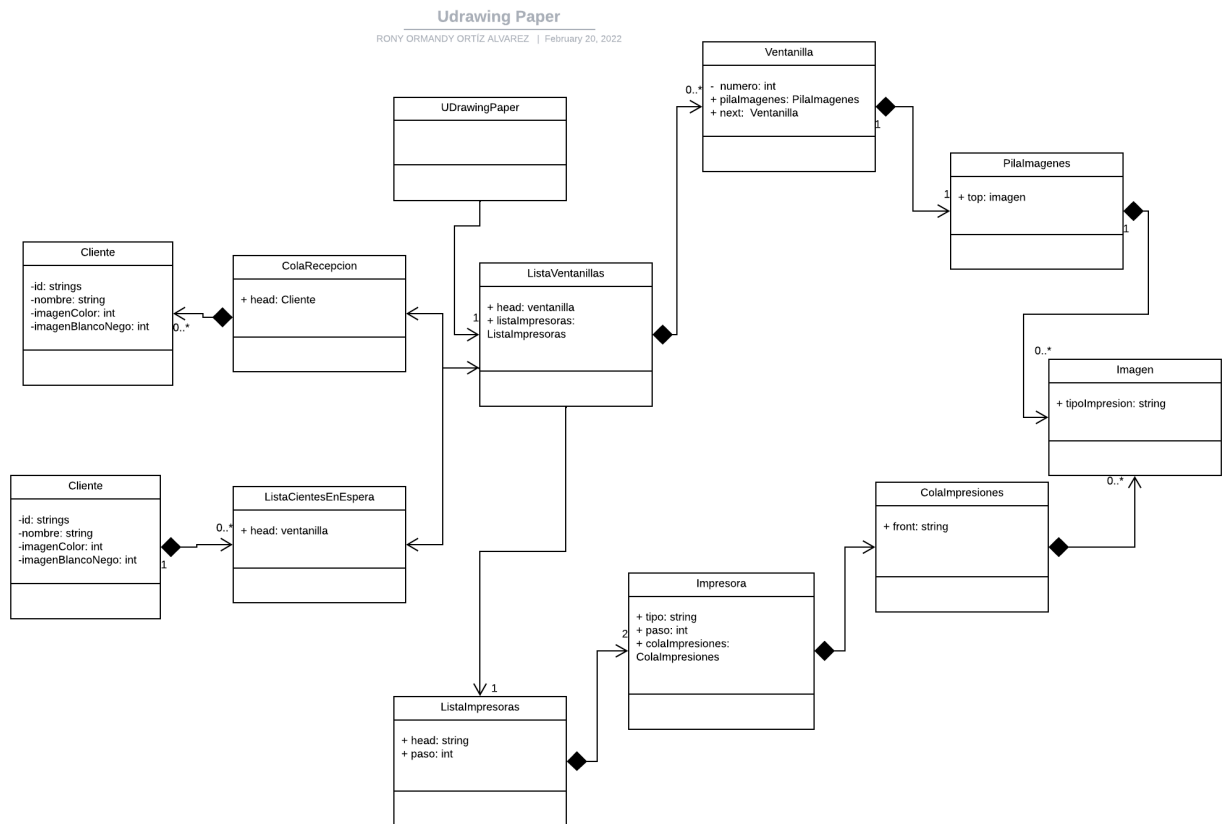


Figura 2: Diagrama de clases UML



## 6. Conclusiones

- Para el mantenimiento de este sistema se recomienda el uso del IDE de NetBeans.
- Se debe tomar en cuenta que este proyecto contiene librerías no desarrolladas por el autor de este documento.

# Bibliografía

- [1] PROYECTO *FASE1*