

# MANUAL TÉCNICO

## Descripción general:

La aplicación fue realizada para simular los diferentes procesos que se llevan a cabo en una empresa de impresión de imágenes, desde que un cliente ingresa a la empresa hasta que recibe todas sus imágenes y abandona la empresa. La aplicación permite visualizar el estado de las estructuras de datos en memoria y también permite generar determinados reportes.

## Lenguaje de programación utilizado:

La aplicación fue desarrollada en el lenguaje de programación java.

## SO en los que puede ser utilizada:

Se puede utilizar en los sistemas operativos Windows, Mac OS, Linux.

Solo se necesita tener instalada la versión de Java SE 12 o superior.

## Descripción Técnica

Para el desarrollo se utilizaron listas enlazadas, haciendo uso de POO para tener una clase Nodo y una clase Lista en donde se podían ir guardando los datos necesarios.

Se hizo uso de la librería Gson para la lectura del archivo Json.

En este caso se trabajó con un proyecto Maven, por lo que para la instalación de las librerías sólo es necesario agregarlas como dependencias al archivo pom.xml.

Se usó la estructura map para recorrer el json y obtener los valores para crear los clientes

## Clientes

Para cada cliente se crearon los siguientes atributos:

```
public class Cliente {  
    private int id_cliente;  
    private String nombre_cliente;  
    private int img_color;  
    private int img_bw;  
    private int imgColorConstante;  
    private int imgBNConstante;  
    private int pasos;  
    private boolean tieneTodasSusImgs;
```

Se utilizaron dos atributos para la cantidad de imágenes ya que uno servía para mostrar siempre la cantidad de imágenes en el gráfico de las estructuras y otro para trabajar el algoritmo.

### Imágenes

Para cada imagen se crearon los siguientes atributos:

```
private String tipo;  
private int pasos;  
private Cliente cliente;  
private boolean sePuedeImprimir;
```

El atributo booleano se creó porque la imagen no se puede imprimir en el mismo paso que es enviada a la cola de impresoras, con esto se asegura que se imprima hasta el siguiente paso.

### Ventanillas

Para cada ventanilla se crearon los siguientes atributos:

```
private String numeroVentanilla;  
private ListaSimple listaImagenes;  
private boolean estaDisponible;  
private int idGrafo;  
private Cliente cliente;
```

## Nodo

Para cada nodo se crearon los siguientes atributos:

```
private Object dato;  
private Nodo siguiente;
```

## Lista Simple

Para cada lista simple se crearon los siguientes atributos.

```
private Nodo primero;  
private Nodo ultimo;  
private int size;
```

## Pila

Para el uso de la pila se utilizó una lista enlazada la cual con los métodos de insertar al final y de eliminar final, los cuales hacen la función de push y pop, de esta manera fue trabajada la pila.

## Cola

Para el uso de la cola se utilizó una lista enlazada la cual con los métodos de insertar al final y de eliminar inicio, los cuales hacen la función de primero en entrar primero en salir, de esta manera fue trabajada la cola.

Para el ordenamiento de las listas se utilizó ordenamiento burbuja, el cuál su algoritmo es el siguiente.

```
procedimiento DeLaBurbuja ( $a_0, a_1, a_2, \dots, a_{(n-1)}$ )  
  para  $i \leftarrow 1$  hasta  $n - 1$  hacer  
    para  $j \leftarrow 0$  hasta  $n - i - 1$  hacer  
      si  $a_{(j)} > a_{(j+1)}$  entonces  
         $aux \leftarrow a_{(j)}$   
         $a_{(j)} \leftarrow a_{(j+1)}$   
         $a_{(j+1)} \leftarrow aux$   
      fin si  
    fin para  
  fin para  
fin procedimiento
```