

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Área de Desarrollo de Software

Laboratorio Estructura de Datos

Proyecto 1

UDrawing Paper

Implementación de estructuras lineales

Nombre: Eddy Fernando Díaz Galindo

Registro académico: 201906558

Instructor: Alex Lopez

Fecha de Entrega: 20/02/2022

CLASES DE UDRAWING PAPER

Clase	Descripción
Cliente	Clase nodo la cola cliente que recibe todos los datos que trae el archivo con extension json
ColaCliente	Clase cola encargada de agregar clientes a la cola de recepción y genera el grafo de dicha cola
Comienzo	Clase sub-principal que se encarga de llenar la lista de ventanilla según los pasos que se den con clientes y genera los grafos de dicha lista y pila.
Imagenes	Clase nodo que contiene las imágenes a imprimirse que lleva el cliente a la ventanilla.
ListaVentanilla	Clase lista que contiene los metodos de agreagar ventanillas cuando el usuario ingrese la cantidad.
Main	Clase inicial y principal del proyecto que ejecuta el menu inicial, hace instancia a clases que son las creadoras de mas estructuras de datos.
Massive	Clase sub-principal que se encarga de analizar el archivo json de la carga masiva para luego agregarlos a cola o agregar aleatoriamente mas clientes.
PilaImagenes	Clase pila encargada de recibir las imágenes a imprimir con los datos del cliente y la cantidad de imágenes a color o blanco y negro.
Ventanilla	Clase nodo encargada de guardar la cantidad de ventanillas que se abran al inicio del programa.

Fuente: Elaboración Propia, 2022

MÉTODOS PRINCIPALES DE UDRAWING PAPER

MÉTODO MAIN: Encargado de iniciar el programa con el menu ejecutandose con ciclo while para la constante repetición del programa.

```
7  ▶ public static void main(String[] args) {  
8      Massive massive = new Massive();  
9      Comienzo comienzo = new Comienzo();  
10     int pasos = 1;  
11  
12     boolean preparativo = false;  
13     while (true){  
14         menu();  
15         try{  
16             System.out.println("Seleccione una opcion (número): ");  
17             int opcion=0;  
18             String Srutasou;
```

Fuente: Elaboración Propia, 2022

```
private static void menu(){  
    System.out.println("1. Carga masiva de clientes");  
    System.out.println("2. Cantidad de ventanillas");  
    System.out.println("3. Ejecutar paso");  
    System.out.println("4. Estado en memoria de las estructuras");  
    System.out.println("5. Reportes");  
    System.out.println("6. Acerca de");  
    System.out.println("7. Salir");  
}
```

Fuente: Elaboración Propia, 2022

MÉTODO ANALYZER: Encargado de parsear el archivo de entrada con extensión Json para luego mandarse a la cola de recepción.

```

20      public void Analyzer(String path){
21          JSONParser parser = new JSONParser();
22
23          try{
24              Object obj = parser.parse(new FileReader(path));
25              JSONObject jsonObject = (JSONObject) obj;
26
27              jsonObject.keySet().forEach(keyStr
28                  ->{
29                  if(((JSONObject) jsonObject.get(keyStr)).keySet() != null){
30                      ((JSONObject) jsonObject.get(keyStr)).keySet().forEach(keyStr2
31                          -> {
32                          Object keyvalue2 = ((JSONObject) jsonObject.get(keyStr)).get(keyStr2);
33                          if (keyStr2.equals("nombre_cliente")){
34                              nombre_cliente = keyvalue2.toString();
35                          }else if(keyStr2.equals("img_bw")){
36                              img_bw = Integer.parseInt(keyvalue2.toString());
37                          }else if(keyStr2.equals("img_color")){
38                              img_color = Integer.parseInt(keyvalue2.toString());
39                          }else if(keyStr2.equals("id_cliente")){
40                              id_cliente = keyvalue2.toString();
41                          }
42                      });
43                  }
44                  NuevoCliente(id_cliente,nombre_cliente,img_color,img_bw);
45              });
46          }catch (Exception e){
47

```

Fuente: Elaboración Propia, 2022

MÉTODO REALIZARPASO: Encargado de agregar aleatoriamente a clientes con datos utilizando la librería random.

```

public void realizarPaso(){
    int MinCliente = 0;
    int MaxCliente = 3;

    String ids;
    int MinNombre = 0;
    int MaxNombre = 18;
    int MinImagenes = 0;
    int MaxImagenes = 4;
    String[] nombres = {"Aguilar", "Alonso", "Álvarez", "Arias", "Benítez", "Blesa", "Bravo", "Caballero", "Cabrera", "Calvo", "Cambil", "Campos",

    Random random = new Random();

    int IngresaCliente = random.nextInt( bound: MaxCliente - MinCliente +1) + MinCliente;
    System.out.println("Ingresaron: " + IngresaCliente);
    for(int i = 1; i<=IngresaCliente; i++){
        id++;
        ids = "ran"+id;
        int NombreCliente = random.nextInt( bound: MaxNombre + MinNombre) + MinNombre;
        int ImageColor = random.nextInt( bound: MaxImagenes + MinImagenes) + MinImagenes;
        int ImageBW = random.nextInt( bound: MaxImagenes + MinImagenes) + MinImagenes;
        NuevoCliente(ids,nombres[NombreCliente],ImageColor,ImageBW);
    }
}

```

Fuente: Elaboración Propia, 2022

MÉTODO IMPRIMIRCLIENTE: Los métodos imprimir se caracterizan por ser cortos y con un solo ciclo while con el código imprimir de los datos de tenga la estructura de datos.

```
public void imprimirCliente(){
    Cliente tmp1 = inicio;
    System.out.println("-----Cola de Recepción-----");
    while ( tmp1 != null){
        if(tmp1.siguiente == null) {
            System.out.print("[id " + tmp1.id_cliente + " | nombre " + tmp1.nombre_cliente + " | color " + tmp1.img_color + " | bw " + tmp1.img_bw + " ] ->");
        }else{
            System.out.print("[id " + tmp1.id_cliente + " | nombre " + tmp1.nombre_cliente + " | color " + tmp1.img_color + " | bw " + tmp1.img_bw + " ] ->");
        }

        tmp1 = tmp1.siguiente;
    }
    if(tmp1 == null){
        System.out.println("NULL");
    }

    System.out.println("");
}
```

Fuente: Elaboración Propia, 2022

MÉTODO GRAFO: Es el encargado de generar una variable tipo String para luego analizarla y crear el archivo con extensión dot para la creación visual de las estructuras.

```
public void grafo(){
    graph += "digraph G { \n";
    graph += "label = \"Cola Recepción\" \n";
    Cliente tmp1 = inicio;
    int i = 0;
    while ( tmp1 != null){
        graph += "a"+i+"[label=\"CLIENTE: "+tmp1.id_cliente+" \n IMGC: "+tmp1.img_color+" \n IMGBW: "+tmp1.img_bw+" \"] \n";
        i++;
        tmp1 = tmp1.siguiente;
    }
    graph += "{rank = same; \n";

    for(int a = 0; a<i-1; a++){
        graph += "a"+a+" -> a" + (a+1) + " \n";
    }
    graph += "} \n";
    graph += "}";
    //System.out.println(graph);
    GenerarGrafo();
}
```

Fuente: Elaboración Propia, 2022

MÉTODO LLEGACLIENTE: Este método es el encargado de llenar la estructura de datos que tiene la cola de recepción de clientes, todas las estructuras tienen estos métodos y se caracterizan por tener la sentencia if con terminos null.

```

public void LlegaCliente(String id_cliente,String nombre_cliente,int img_color,int img_bw) {
    Cliente nuevo = new Cliente(id_cliente, nombre_cliente, img_color,img_bw);

    if (inicio == null) {
        inicio = nuevo;
    } else {
        Cliente tmp = inicio;
        while (tmp.siguiete != null){
            tmp = tmp.siguiete;
        }
        tmp.siguiete = nuevo;
    }
}

```

Fuente: Elaboración Propia, 2022

MÉTODO GENERAR GRAFO: Este método es el encargado de convertir el archivo con extension dot a png para ver de forma gráfica las estructuras de datos.

```

public void GenerarGrafo(){
    FileWriter fichero = null;
    PrintWriter pw = null;
    /*Crea un archivo con extensión .dot con el texto de la variable graph*/
    try{
        fichero = new FileWriter( fileName: "ColaRecepcion.dot");
        pw = new PrintWriter(fichero);
        pw.write(graph);
        pw.close();
        fichero.close();
    }catch(Exception e){
        System.out.println("Error en generar dot de la cola");
    }finally {
        if(pw!=null){
            pw.close();
        }
    }
    /*Convierte el archivo con extensión .dot a .png */
    try{
        ProcessBuilder proceso;
        proceso = new ProcessBuilder( ..command: "dot", "-Tpng", "-o", "GraphRecepcion.png", "ColaRecepcion.dot");
        proceso.redirectErrorStream(true);
        proceso.start();
    }catch (Exception e){
        System.out.println("Error en generar png de la cola");
    }
}

```

Fuente: Elaboración Propia, 2022