

Universidad de San Carlos de Guatemala
Facultad de ingeniería
Ingeniería en Ciencias y Sistemas
ESTRUCTURA DE DATOS



MANUAL TECNICO

Alberto Josue Hernández Armas 201903553

Guatemala 20 de febrero de
2022

INTRODUCCION

En el presente manual técnico podremos observar cómo es que está diseñado el código y explicando que forman la funcionalidad de una manera general, teniendo como objetivo principal que sea más entendible. Determinando los métodos utilizados y explicando algunas palabras claves refiriéndonos a sus propiedades demostrando cuál es su función dentro de los bloques de código que se nos presentan a continuación.

MANUAL TECNICO

Primero se muestran las librerías con las cuales se estarán trabajando durante todo el proyecto
El programa esta escrito en lenguaje Java:

```
4 //import jdk.nashorn.internal.parser.JSONParser;
5 import jdk.nashorn.internal.runtime.JSONListAdapter;
6
7 import jdk.nashorn.internal.parser.JSONParser;
8
9 import java.*;
10 import com.google.gson.*;
11 import java.io.PrintWriter;
12 import java.io.FileReader;
13 import java.util.Set;
14 import java.util.HashSet;
15
16 import java.io.BufferedReader;
17 import java.io.File;
18 import java.io.FileNotFoundException;
19 import java.io.FileReader;
20 import java.security.Key;
21 import java.util.ArrayList;
22 import java.util.Arrays;
23 import java.util.Iterator;
24 import java.util.Scanner;
25 import javax.print.DocFlavor;
26 import javax.swing.*;
27 import java.util.Random;
28 import javax.swing.plaf.synth.SynthTextAreaUI;
29
```

Luego se inicia directamente con la carga masiva de elementos al programa, para la construcción inicial de todas las listas enlazadas.

```

public static String getContentOfFile(String pathname) {
    File archivo = null;
    FileReader fr = null;
    BufferedReader br = null;

    try {
        // Apertura del fichero y creacion de BufferedReader para poder
        // hacer una lectura comoda (disponer del metodo readLine()).
        //para el commit final
        archivo = new File(pathname);
        fr = new FileReader(archivo);
        br = new BufferedReader(fr);
        // Lectura del fichero
        String content = "";
        String linea;
        while ((linea = br.readLine()) != null) {
            content += linea + "\n";
        }
        return content;
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        // En el finally cerramos el fichero, para asegurarnos
        //otro cambio solo para hacerlo bien
        // que se cierra tanto si todo va bien como si salta
        // no se cierra
    }
}

```

```

public static void main(String[] args) throws FileNotFoundException {
    ///////////CONVERSION DE JSON A JSONOBJECT Y GENERACION DE LA COLA DE RECEPCION(INSTANCIA)
    JFileChooser fileChooser = new JFileChooser();
    fileChooser.setCurrentDirectory(new File(System.getProperty("user.home")));
    int result = fileChooser.showOpenDialog(fileChooser.getParent());
    if (result == JFileChooser.APPROVE_OPTION) {...}
    Gson gsus = new Gson();
    JsonParser parsero = new JsonParser();
    JsonObject objetosito = (JsonObject)parsero.parse(getContentOfFile(dir));
    System.out.println(objetosito);
    Object keys = objetosito.keySet();
}

```

Para luego convertir el objeto Json en una estructura de lista enlazadas

```

Iterator<String> iter = objetosito.keySet().iterator();
while (iter.hasNext()) {
    String key = iter.next();
    try {
        cliente cliente = gsus.fromJson(objetosito.get(key), cliente.class);
        cliente.terminardeconstruir();
        cola_entrada.agrega(cliente);
        listatotaldeclientes.agrega(cliente);
        System.out.println("\nlista: ");
        nodo m = cola_entrada.First;
        while (m != null)
        {
            System.out.println(((cliente)m.value).vercliente());
            m = m.Next;
        } //IMPRESION DE LA COLA
    } catch (Exception e) {
        // Something went wrong!
    }
}

```

Para luego establecer la cantidad de ventanillas con las que contara el programa

```

System.out.println("Ingrese la cantidad de ventanillas");
int qv = Integer.parseInt(input.next());
for (int i = 0; i < qv; i++)
{
    vacio vc = new vacio();
    ventanillas.agrega(vc);
}
//nombre[0]="Alberto";nombre[1]="Sofia";nombre[5]="Roberto";nombre[6]="0"

```

Luego desplegar el menú y enseñar las diferentes acciones que puede realizar en programa, se queda en bucle para no cerrarse después de cumplir una acción

```
System.out.println("1.Realizar paso");
System.out.println("2.ver top 5 de color");
System.out.println("3. ver top 5 blanco y negro");
System.out.println("4.ver cliente que paso mas pasos(calculado al salir de la em
System.out.println("5. ver informacion de cliente especifico");
System.out.println("6. ver informacion del estudiante");
System.out.println("7. salir");
System.out.println("8. Reporte cola entrada");
System.out.println("9. Reporte Sala espera");
System.out.println("10. Reporte cola ventanillas");
System.out.println("8. Impresora Color");
String papanoel = input.next().toString();
if (papanoel.equals("1"))
{...}
else if(papanoel.equals("2"))
{...}
else if(papanoel.equals("3"))
{...}
else if(papanoel.equals("4"))
{...}
else if(papanoel.equals("5"))
{...}
else if(papanoel.equals("6"))
{...}
else if (papanoel.equals("8"))
{...}
else if (papanoel.equals("9"))
```

Impresión general de la memoria en tiempo real

```

// impresion cola entrada
System.out.println("\nver cola entrada");
nodo m2 = cola_entrada.First;
while (m2 != null)
{...} //IMPRESION DE LA COLA
System.out.println("ver ventanillas");
nodo ver = ventanillas.First;
while (ver != null)
{...}
//impresion sala espera
System.out.println("ver sala espera");
nodo espera = sala_espera.First;
while (espera != null) {...}
//impresion impresora color
System.out.println("ver impresora color");
nodo vercolor = imp_color.First;
while (vercolor != null) {
    System.out.println(((imagen) vercolor.value).getodo());
    vercolor = vercolor.Next;
}
//impresion impresora blanco y negro
System.out.println("ver impresora blanco y negro");
nodo verbn = imp_bn.First;
while (verbn != null) {...}
}

```

Funcionamiento diversos para el control de paso de información en el programa

```

///push para ventanillas y resta a cada cliente por cada paso(cuando iguala a cero push a sala de espera y a cada cola de impresion)
nodo jaja = ventanillas.First;
while (jaja != null)
{
    if(jaja.value.getClass()==cliente.class)
    {
        if(((cliente)jaja.value).img_color>0)
        {
            imagen img = new imagen( color: "color",((cliente)jaja.value).nombre_cliente,((cliente)jaja.value).id_cliente, pasos: 2);
            //imp_color.agrega(img);

            if(jaja.Next != null) {
                jaja = jaja.Next;
                ((cliente) jaja.Prev.value).color.agrega(img);

                ((cliente) jaja.Prev.value).img_color -= 1;
                jaja = jaja.Prev;
            }
            else
            {
                jaja = jaja.Prev;
                ((cliente) jaja.Next.value).color.agrega(img);

                ((cliente) jaja.Next.value).img_color -= 1;
                jaja = jaja.Next;
            }
        }
    }
}

```

Se realizan distintos tiros de información para que vaya fluyendo de manera constante sin interrupciones


```

nodo pepablanca = imp_bn.First;
if (pepablanca != null)
{
    if(((imagen)pepablanca.value).pasos>0)
    {
        if(pepablanca.Next != null) {
            pepablanca = pepablanca.Next;
            ((imagen) pepablanca.Prev.value).pasos -=1;
            pepablanca = pepablanca.Prev;
        }
        else
        {...}
    }
    else
    {
        nodo buscadueñoblanc = sala_espera.First;
        nodo fotitoblanc = imp_bn.popFirst();
        while (buscadueñoblanc != null)
        {
            if(((cliente)buscadueñoblanc.value).nombre_cliente.equals(((imagen)fotitoblanc.value).dueño))
            {
                if(buscadueñoblanc.Next != null)
                {
                    buscadueñoblanc = buscadueñoblanc.Next;
                    ((cliente)buscadueñoblanc.Prev.value).color.agrega(((imagen)fotitoblanc.value));
                    buscadueñoblanc = buscadueñoblanc.Prev;
                    break;
                }
            }
            else

```

Se finaliza con el pop de datos de la sala de espera, de los clientes que ya han recibido todas sus impresiones de vuelta

```
// AHORA LOS PUSH DE LOS CLIENTES EN LA SALA DE ESPERA CON SU PUSH ESPECIFICO JAJUY
nodo pafuera = sala_espera.First;
cliente pafueraaux = null;
while (pafuera != null)
{
    if(((cliente)pafuera.value).fotos==(((cliente)pafuera.value).color.tamaño+((cliente)pafuera.value).bn.tamaño))
    {
        pafueraaux =(cliente) sala_espera.popEspecifico(pafuera).value;

        //top5color.agregatop5color(pafueraaux);
        if(cientemaspasos == null)
            cientemaspasos = pafueraaux;
        else
            if (pafueraaux.contadordepasos > cientemaspasos.contadordepasos)
            {
                cientemaspasos = pafueraaux;
            }
    }
    pafuera = pafuera.Next;
}
```