PRACTICA 2 – IPC1 “F”

Lusvin Alexander Sicajá Ramírez
201602630

APLICACIÓN DE ORDENAMIENTO

MANUAL TECNICO

Métodos Principales

- **DECLARACIÓN DE VARIABLES PRINCIPALES**

```
protected Thread hilo;
protected Thread reloj;
protected FileNameExtensionFilter f = new
protected static File archivo;

protected static String m[][];
protected static String mdesordenada[][];
protected static String valorx;
protected static String valory;

protected static Integer cantidad[];
protected static Integer movimientos = 0;
protected static String nombre[];
protected static ChartPanel chpanel;
protected static Boolean tiempoGrafica;
protected static String nombreAlgoritmo;
```

- **METODO DE CARGA DE ARCHIVO UTILIZANDO JFILECHOOSER**

```
private void btnexaminarMouseClicked(java.awt.event.MouseEvent evt) {

    //Creamos el objeto file chooser
    JFileChooser buscador = new JFileChooser();

    buscador.setFileFilter(f);

    //comportamiento al seleccionar
    int respuesta = buscador.showOpenDialog(this);

    if (respuesta == JFileChooser.APPROVE_OPTION) {
        archivo = buscador.getSelectedFile();
```

- **METODO CARGADATOS, RECIBE EN SU PARAMETRO UN OBJETO TIPO FILE**

```
//Metodo que carga los datos del CSV a la variable texto-------
    static String cargaDatos(File f) throws IOException {
```

Este método se encarga de pasar los datos del archivo seleccionado con el FileChooser a una cadena de texto de tipo String.

- **METODO CREAR MATRIZ, RECIBE UNA CADENA DE TEXTO**

```
//Metodo que agrega los valores a una matriz de String

    static String[][] CrearMatriz(String texto) {
        String filas[] = texto.split("\n");
        int num colum = filas[0].split(",").length;
```

Esté método se encarga de pasar los datos de la cadena string a varias posiciones en una matriz, según el valor de separación.

- **METODOS DE SEPARACION EN VECTORES DE ENTEROS Y STRINGS**

```
//Metodo que separa la matriz en vectores de string la primer columna
static String[] separarNombre(String m[][]) {
    String vnombres[] = new String[m.length];
    for (int i = 1; i < m.length; i++) {
        vnombres[i] = m[i][0];
    }
    return vnombres;
}

//Metodo que separa la matriz en vectores de enteros la segunada columna
static Integer[] separarCantidad(String m[][]) {
    Integer vcantidad[] = new Integer[m.length];
    for (int i = 1; i < m.length; i++) {
        vcantidad[i] = Integer.parseInt(m[i][1]);
    }

    return vcantidad;
}
```

- **METODO DE ORDENAMIENTO BURBUJA**

```
//METODO ORDENAR MATRIZ EN BURBBLE SORT DESCENDENTE----------------------
    public void ordenarMatrizBurbble() throws InterruptedException {
        hilo = new Thread() {
            @Override
            public void run() {
                int editar = 1;
                for (int i = 1; i < cantidad.length; i++) {
                    // System.out.print(cantidad[i] + nombre[i]);
```

Dentro de éste método se encuentra el hilo que lleva el control del ordenamiento del vector cantidad el cual tiene un retardo con el método sleep() de 200 milisegundos.

- **METODO DEL CONTROL DEL TIEMPO**

```
//Metodo de conteo de reloj-----------------------------------------
public void relojito() throws InterruptedException {
    tiempoGrafica = true;
    reloj = new Thread() {
        @Override
        public void run() {
            while (tiempoGrafica) {
```

Este método se encarga de iniciar el hilo que llevará el conteo del tiempo.

- **METODO DE GENRACIÓN DEL GRAFICO DE BARRAS**

```
//Metodo para generar la gráfica-----------------------------------
public void generarGrafica() {
    DefaultCategoryDataset datos = new DefaultCategoryDataset();
    for (int i = 1; i < cantidad.length; i++) {
        datos.setValue(cantidad[i], nombre[i], " ");
```

- **METODO QUE LIMPIA EL GRÁFICO**

```
//Metodo para limpiar el jpanel y mostrar la grafica nueva----------------------------------------

    void limpiarJpanel() {
        jpgrafica.removeAll();
        jpgrafica.revalidate();
    }
```

Este método se encarga de limpiar el gráfico con el valor de ordenamiento actual, para luego sobre el panel se inserte el gráfico actualizado.

- **METODO PARA GENERAR EL REPORTE HTML**

```
public static void reporte() throws IOException {

    File archivoHtml;
    FileWriter escritor = null;

    try {
        archivoHtml = new File("Reporte.html");
        escritor = new FileWriter(archivoHtml);
```

- **METODO MAIN**

```
public static void main(String[] args) {
    try {
                UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
            } catch (ClassNotFoundException ex) {
                Logger.getLogger(View.class.getName()).log(Level.SEVERE, null, ex);
            } catch (InstantiationException ex) {
                Logger.getLogger(View.class.getName()).log(Level.SEVERE, null, ex);
            } catch (IllegalAccessException ex) {
                Logger.getLogger(View.class.getName()).log(Level.SEVERE, null, ex);
            } catch (UnsupportedLookAndFeelException ex) {
                Logger.getLogger(View.class.getName()).log(Level.SEVERE, null, ex);
            }


    View v = new View();
    v.setVisible(true);
```

El método main es el encargado de instanciar a la ventana principal de la aplicación, así mismo también posee un método de LookandFeel para visualizar la ventana del FILECHOOSER con la apariencia del SO.