

MANUAL TECNICO

Diego Facundo Pérez Nicolau

202106538

IPC1

Con este comando en el programa llamamos a la primera ventana:

```

]      /**
-      * @param args the command line arguments
-      */
]      public static void main(String[] args) {
-          new inicio().setVisible(true); // TODO code application logic here
-      }

```

Con este sistema traemos y usamos datos:

```

-      */
-      public class datos {
-          private String categoria;
-          private String cantidad;
-
-          public datos(String categoria, String cantidad) {
-              this.categoria = categoria;
-              this.cantidad = cantidad;
-          }
-
-          public String getcategoria() {
-              return categoria;
-          }
-
-          public void setcategoria(String categoria) {
-              this.categoria = categoria;
-          }
-
-          public String getcantidad() {
-              return cantidad;
-          }
-
-          public void setcantidad(String cantidad) {
-              this.cantidad = cantidad;
-          }
-      }

```

Importamos todas las librerías a usar

```
package javaapplication10;
import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.image.BufferedImage;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.PrintWriter;
import javax.imageio.ImageIO;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.data.category.DefaultCategoryDataset;
/**...4 lines */
public class inicio extends javax.swing.JFrame {
    static int Cont1=0,CONTdatos=0;
    static datos datosGRA[]=new datos[1];
    static datos AUX[]=new datos[1];
    static String title;
```

Con esta parte del código es donde traemos a la grafica de los datos de la tabla:

```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
String leerruta=jTextField1.getText().trim();

    if (jTextField2.getText().equals("")) {
        JOptionPane.showMessageDialog(null, "Debe llenar todos los Campos");
    } else {{
        try {

            FileReader fr=new FileReader(leerruta);
            BufferedReader br=new BufferedReader(fr);

            String linea;
            while ((linea=br.readLine())!= null) {
                Conti++;
            }

            System.out.println("el contador es :"+Conti);
            datosGRA=new datos[Conti];
            AUX=new datos[Conti];
            System.out.println("se seteo el tamaño "+datosGRA.length);

            FileReader LR=new FileReader(leerruta);
            BufferedReader bF=new BufferedReader(LR);

            String linea2;
            while ((linea2=bF.readLine())!= null) {
                String [] datos2=linea2.split(",");
                String CATEGO=datos2[0];
                String CANTII=datos2[1];
                datosGRA[CONTdatos]=new datos(CATEGO, CANTII);
                AUX[CONTdatos]=new datos(CATEGO, CANTII);
                CONTdatos++;
            }

        } catch (Exception e2) {
            e2.printStackTrace();
        }

        for (int i = 0; i <datosGRA.length-1; i++) {
            System.out.println("categoria:"+ datosGRA[i].getcategoria());
            System.out.println("cantidad:"+ datosGRA[i].getcantidad());
            System.out.println("*****");
        }
        System.out.println("el segundo contador es: "+CONTdatos);
    }

}
//BUB();

```

Parte del código del método de ordenamiento de burbuja:

```

String auxCANT;
//APLICANDO EL ORDENAMIENTO BURBUJA
for (int i = 1; i < (CONTdatos - 1); i++) {
    for (int j = 1; j < (CONTdatos - 1); j++) {
        float num1=Float.parseFloat(datosGRA[j].getcantidad());
        float num2=Float.parseFloat(datosGRA[j + 1].getcantidad());
        if (num1 > num2) {
            //OBTENIENDO VALORES Y GUARDANDOSLOS
            auxCANT = datosGRA[j].getcantidad();
            auxCAT = datosGRA[j].getcategoría();

            //SETEANDO LA POSICION ACTUAL EL VALOR DE LA POSICION SIGUIENTE

            datosGRA[j].setcantidad(datosGRA[j + 1].getcantidad());
            datosGRA[j].setcategoria(datosGRA[j + 1].getcategoría());

            //SETEANDO LA POSICION SIGUIENTE CON LA POSICION ACTUAL
            datosGRA[j + 1].setcantidad(auxCANT);
            datosGRA[j + 1].setcategoria(auxCAT);
        }
    }
}

System.out.println("datos ordenados por su cantidad: ");
System.out.println(" ");
for (int i =0; i<CONTdatos; i++) {
    System.out.println(" su cantidad: " + datosGRA[i].getcantidad());
    System.out.println(" su categoría: " + datosGRA[i].getcategoría());

    System.out.println(" ");
}
//BUB ();
JFreeChart barras=null;
DefaultCategoryDataset datos=new DefaultCategoryDataset();
String Y =datosGRA[0].getcantidad();
String X =datosGRA[0].getcategoría();
title=jTextField2.getText();
for (int i = 1; i < datosGRA.length-1; i++) {
    float cantidad=Float.parseFloat(datosGRA[i].getcantidad());
    String categoria=datosGRA[i].getcategoría();
    datos.addValue(cantidad, "", categoria);
}

barras=ChartFactory.createBarChart(title, X, Y, datos, PlotOrientation.VERTICAL, true,true,false);

ChartPanel panel = new ChartPanel(barras);

```

Parte del código para generar reportes:

```

writer.println("<!DOCTYPE HTML>");
writer.println("<html>");
writer.println("<head>");
writer.println("<meta charset= " + " utf-8 " + " > ");
writer.println("<title>REPORTE DE datos </title>");

writer.println("</head>");

writer.println("<body bgcolor= " + " #FF5733 " + "text=" + "white" + ">");

writer.println("<H1 " + " align= " + "center" + ">" + "Juan Pedro Valle Lema</H1>");

writer.println("<H2 " + " align= " + "center" + ">" + "202101648</H2>");
writer.println("<HR " + "align= " + " center" + " size = " + "4" + " width =" + " 100% " + "color= " + " White" + " noshade>");
writer.println("<br>");

writer.println("<pre>" + " ALGORITMO : " + jTextField2.getText() + " TIEMPO :"+jTextField2.getText() + "</pre>");
writer.println("<pre>" + " VELOCIDAD : " + jTextField2.getText() + " PASOS :"+jTextField2.getText() + "</pre>");
writer.println("<pre>" + " ORDEN : " + jTextField2.getText());
writer.println("<br>");
writer.println("<HR " + "align= " + " center" + " size = " + "4" + " width =" + " 100% " + "color= " + " White" + " noshade>");

if (jTextField2.getText().equalsIgnoreCase("ascendente")) {
    String CatminA = datosGRA[1].getcantidad();
    String CantminA = datosGRA[1].getcategoria();
    String CatmaxA = datosGRA[CONTdatos - 1].getcantidad();
    String CantmaxA = datosGRA[CONTdatos - 1].getcategoria();

    writer.println("<div>");
    writer.println("<div style='float: left; margin: 0px 200px 0px 300px; padding: 10px; text-align: left; '>");
    writer.println("<table BORDER WIDTH=200% >");
    writer.println("<tr>");
    writer.println("<td>DATO MINIMO</td>");
    writer.println("</tr>");
    writer.println("<tr>");
    writer.println("<td>");
    writer.println("<table BORDER WIDTH=100%>");
    writer.println("<tr>");
    writer.println("<td>"+CantminA+"</td>");
    writer.println("<td>"+CatminA+"</td>");
    writer.println("</tr>");
    writer.println("</table>");
    writer.println("</td>");

    writer.println("</tr>");
    writer.println("</table>");
    writer.println("</div>");
    writer.println("<div style='float: right; margin: -20px 400px 0px 10px; padding: 10px; text-align: left; '>");

```