

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela Ciencias y sistemas

MANUAL TÉCNICO

Introducción a la programación y computación 1

Rubén Alejandro Ralda Mejía

202111835

Guatemala 20 de marzo del 2022

Librerías

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.BorderFactory;
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.CategoryDataset;
import org.jfree.data.category.DefaultCategoryDataset;
```

Graficar

El botón examinar guarda la ruta en el textbox

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    JFileChooser archivo = new JFileChooser();
    int valor = archivo.showOpenDialog(this);
    if (valor == JFileChooser.APPROVE_OPTION) {
        txtruta.setText(archivo.getSelectedFile().getAbsolutePath());
    }
}
```

El botón generar gráfica, obtiene la ruta y el título (si no tiene título se utiliza un predeterminado).

```
// TODO add your handling code here:
String titulo = null;
if (!txtruta.getText().equalsIgnoreCase("")) {
    if (txtnombre.getText().equalsIgnoreCase("")) {
        titulo = "Grafica de ordenamiento";
        txtnombre.setText(titulo);
    } else {
        titulo = txtnombre.getText();
    }
}
File datos = new File(txtruta.getText());
BufferedReader br = null;
```

Con el archivo determino el número de líneas y creo los vectores con ese tamaño, luego lleno los datos en los vectores.

```
//tamaño del vector
Integer tamano = 0;
Scanner entrada = new Scanner(datos);
while (entrada.hasNext()) {
    entrada.nextLine();
    tamano++;
}
//llenar los datos en vectores separados
br = new BufferedReader(new FileReader(datos));
String[][] temporal = new String[tamano][2];
valoresy = new int[tamano - 1];
valoresx = new String[tamano - 1];
String line = br.readLine();
int i = 0;
String[] encabezadol = line.split(",");
line = br.readLine();
while (null != line) {
    temporal[i] = line.split(",");
    line = br.readLine();
    valoresy[i] = Integer.parseInt(temporal[i][1]);
    valoresx[i] = temporal[i][0];
    i++;
}
```

Luego grafico y lo muestro en el Jpanel. Invoco el método borrar para vaciar los datos si los hay.

```
//graficar
JFreeChart chart = mostrar(cat, titulo, encabezadol[0], encabezadol[1]);
ChartPanel panel = new ChartPanel(chart);
panel.setBorder(BorderFactory.createEmptyBorder(15, 15, 15, 15));
panel.setPreferredSize(new Dimension(400, 400));
panel.setBackground(Color.white);
jPanell.removeAll();
jPanell.setLayout(new BorderLayout());
jPanell.add(panel, BorderLayout.NORTH);
pack();
repaint();
//añadir los valores a la grafica
borrar();
for (int j = 0; j < tamano - 1; j++) {
    cat.setValue(valoresy[j], "", valoresx[j]);
}
```

Ordenar

Verifico si existen datos y se determina cual radiobutton esta seleccionado para llamar al método correspondiente a ordenar y luego como se deberá ordenar.

```

if (valoresy != null) {
    pasos = 0;
    int[] temporaly = new int[valoresx.length];
    String[] temporalx = new String[valoresx.length];
    for (int i = 0; i < valoresx.length; i++) {
        temporaly[i] = valoresy[i];
        temporalx[i] = valoresx[i];
    }
    if (insertion.isSelected()) {
        if (ascend.isSelected() == true) {
            metodoinsercion(temporaly, temporalx, 0);

        } else if (descend.isSelected() == true) {
            metodoinsercion(temporaly, temporalx, 1);
        }
        jButton4.setEnabled(true);
        algoritmo = "Ordenamiento por inserción";
    } else if (merge.isSelected() == true) {
        if (ascend.isSelected()) {
            metodoburbuja(temporaly, temporalx, 0);
        } else if (descend.isSelected() == true) {
            metodoburbuja(temporaly, temporalx, 1);
        }
        jButton4.setEnabled(true);
        algoritmo = "Ordenamiento por burbuja";
    }
} else {
    JOptionPane.showMessageDialog(this, "No se generado una grafica");
}

```

Este es el método de inserción cada vez que mueve un dato a otra posición se suma un paso y se actualiza la gráfica. Con el switch determina si es ascendente o descendente.

```

for (int siguiente = 1; siguiente < ordinary.length; siguiente++) {
    insercion = ordinary[siguiente];
    insercionx = ordenarx[siguiente];
    int moverElemento = siguiente;
    switch (tipo) {
        case 0: //ascendente
            while (moverElemento > 0 && ordinary[moverElemento - 1] > insercion) {
                ordinary[moverElemento] = ordinary[moverElemento - 1];
                ordenarx[moverElemento] = ordenarx[moverElemento - 1];
                moverElemento--;
                pasos++;
                labelpasos.setText(String.valueOf(pasos));
                borrar();
                for (int j = 0; j < ordinary.length; j++) {
                    cat.setValue(ordinary[j], "", ordenarx[j]);
                }
            }
            ordinary[moverElemento] = insercion;
            ordenarx[moverElemento] = insercionx;
            borrar();
            for (int j = 0; j < ordinary.length; j++) {
                cat.setValue(ordinary[j], "", ordenarx[j]);
            }
            break;
        case 1: //descendente
            while (moverElemento > 0 && ordinary[moverElemento - 1] < insercion) {
                ordinary[moverElemento] = ordinary[moverElemento - 1];
                ordenarx[moverElemento] = ordenarx[moverElemento - 1];
                moverElemento--;
                pasos++;
                labelpasos.setText(String.valueOf(pasos));
                borrar();
                labelpasos.setText(String.valueOf(pasos));
                borrar();
                for (int j = 0; j < ordinary.length; j++) {
                    cat.setValue(ordinary[j], "", ordenarx[j]);
                }
            }
            ordinary[moverElemento] = insercion;
            ordenarx[moverElemento] = insercionx;
            borrar();
            for (int j = 0; j < ordinary.length; j++) {
                cat.setValue(ordinary[j], "", ordenarx[j]);
            }
            break;
        default:
            throw new AssertionError();
    }
}
}

this.ordenadox = ordenarx;
this.ordenadoy = ordinary;

```

Para el método burbuja se hace lo mismo, pero con su respectivo algoritmo para ordenar los datos.

```

switch (tipo) {
    case 0://ascendente
        do {
            n = 0;
            for (i = 1; i < l; i++) {
                if (ordinary[i - 1] > ordinary[i]) {
                    temp = ordinary[i - 1];
                    ordinary[i - 1] = ordinary[i];
                    ordinary[i] = temp;
                    temp2 = ordenarx[i - 1];
                    ordenarx[i - 1] = ordenarx[i];
                    ordenarx[i] = temp2;
                    n = i;
                    pasos++;
                    labelpasos.setText(String.valueOf(pasos));
                    borrar();
                    for (int j = 0; j < valoresy.length; j++) {
                        cat.setValue(ordinary[j], "", ordenarx[j]);
                    }
                }
            }
            l = n;
            borrar();
            for (int j = 0; j < valoresy.length; j++) {
                cat.setValue(ordinary[j], "", ordenarx[j]);
            }
        } while (n != 0);
    break;
}

```

```

case 1://descendente
do {
    n = 0;
    for (i = 1; i < l; i++) {
        if (ordinary[i - 1] < ordinary[i]) {
            temp = ordinary[i - 1];
            ordinary[i - 1] = ordinary[i];
            ordinary[i] = temp;
            temp2 = ordenarx[i - 1];
            ordenarx[i - 1] = ordenarx[i];
            ordenarx[i] = temp2;
            n = i;
            pasos++;
            labelpasos.setText(String.valueOf(pasos));
            borrar();
            for (int j = 0; j < valoresy.length; j++) {
                cat.setValue(ordinary[j], "", ordenarx[j]);
            }
        }
    }
    l = n;
    borrar();
    for (int j = 0; j < valoresy.length; j++) {
        cat.setValue(ordinary[j], "", ordenarx[j]);
    }
} while (n != 0);
break;
default:
    throw new AssertionError();

```

Reportes

Luego de ordenar los datos se habilitará el botón para el reporte, creará un archivo y tendrá el nombre del título de la gráfica.

```

String cwd = System.getProperty("user.dir");
File dir = new File(cwd + "\\Reportes");
FileWriter escribir;
PrintWriter nuevaLinea;
if (!dir.exists() && !dir.isDirectory()) {
    try {
        dir.mkdir();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, e);
    }
}
File archivo = new File(cwd + "\\Reportes\\" + txtnombre.getText() + ".html");
archivo.delete();
try {
    archivo.createNewFile();
} catch (IOException ex) {
    Logger.getLogger(Inicio.class.getName()).log(Level.SEVERE, null, ex);
}

try {
    escribir = new FileWriter(archivo, true);
    nuevaLinea = new PrintWriter(escribir);
    nuevaLinea.println("<!DOCTYPE html>\n"
        + "<html lang=\"en\">\n"
        + "<head>\n"
        + "    <meta charset=\"UTF-8\">\n"
        + "    <title>Reporte</title>\n"
        + "</head>\n"
        + "<body>");

```

Luego con un ciclo para ir creando filas con la información de los datos ordenados.

```

nuevaLinea.println("<h1>Reporte </h1>");
nuevaLinea.println("<p><b>Nombre: </b>Rubén Ralda</p>");
nuevaLinea.println("<p><b>Carné: </b>202111835</p>");
nuevaLinea.println("<p><b>Algoritmo: </b>" + algoritmo + "</p>");
nuevaLinea.println("<p><b>Tiempo: </b>" + pasos + "</p>");
nuevaLinea.println("<p><b>Cantidad de pasos: </b>" + pasos + "</p>");
nuevaLinea.println("<table border=\"1\">");
for (int i = 0; i < ordenadox.length; i++) {
    if (ordenadox[i] != null) {
        nuevaLinea.println("<tr>");
        nuevaLinea.print("<td>");
        nuevaLinea.print("<b>" + ordenadox[i] + "</b>");
        nuevaLinea.print("</td>");
        nuevaLinea.print("<td>");
        nuevaLinea.print(ordenadoy[i]);
        nuevaLinea.print("</td>");
        nuevaLinea.println("</tr>");
    }
}
nuevaLinea.println("</table>");
nuevaLinea.println(" ");

```

Y lo mismo para los datos no ordenados.


```

nuevaLinea.println("<p><b>Datos no ordenados</b></p>");
nuevaLinea.println("<table border=\"1\">");
for (int i = 0; i < valoresx.length; i++) {
    if (valoresx[i] != null) {
        nuevaLinea.println("<tr>");
        nuevaLinea.print("<td>");
        nuevaLinea.print("<b> " + valoresx[i] + "</b>");
        nuevaLinea.print("</td>");
        nuevaLinea.print("<td>");
        nuevaLinea.print(valoresy[i]);
        nuevaLinea.print("</td>");
        nuevaLinea.println("</tr>");
    }
}
nuevaLinea.println("</table>");
nuevaLinea.println(" ");
nuevaLinea.println("</body>\n"
    + "</html>");
// me cierra mi archivo
escribir.close();
JOptionPane.showMessageDialog(this, "El reporte se ha creado con exito");

```