

Manual Técnico

Mauricio Antonio Castro Guerra

202100299

Se recurre a una nueva ventana en la cual se ubica el menú principal que utiliza la práctica.

```
package practica2;

/**
 *
 * @author marco
 */
public class Práctica2 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        new MENU().setVisible(true);
    }

}
```

Realizamos una ventana nueva en la cual ubicaremos las variables que posteriormente serán utilizadas a nivel global mediante el get and set.

```
public class DATA {
    private String CATO;
    private String CANTI;

    public DATA(String CATO, String CANTI) {
        this.CATO = CATO;
        this.CANTI = CANTI;
    }

    public String getCATO() {
        return CATO;
    }

    public void setCATO(String CATO) {
        this.CATO = CATO;
    }

    public String getCANTI() {
        return CANTI;
    }

    public void setCANTI(String CANTI) {
        this.CANTI = CANTI;
    }
}
```

Se utilizan todos los botones especificados en el sector de Design de una ventana.

```
private void initComponents() {

    buttonGroup1 = new javax.swing.ButtonGroup();
    buttonGroup2 = new javax.swing.ButtonGroup();
    jButton1 = new javax.swing.JButton();
    jButton2 = new javax.swing.JButton();
    Up = new javax.swing.JRadioButton();
    Down = new javax.swing.JRadioButton();
    Bub = new javax.swing.JRadioButton();
    Shell = new javax.swing.JRadioButton();
    jButton3 = new javax.swing.JButton();
    jTextField1 = new javax.swing.JTextField();
    jTextField2 = new javax.swing.JTextField();
    jPanel1 = new javax.swing.JPanel();
    jPanel2 = new javax.swing.JPanel();
    jButton4 = new javax.swing.JButton();
    jButton5 = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    jButton1.setFont(new java.awt.Font("Lucida Sans", 0, 12)); // NOI18N
    jButton1.setText("Examinar...");
    jButton1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
        }
    });
    getContentPane().add(jButton1, new org.netbeans.lib.awtextra.AbsoluteConstraints(369, 44, -1, -1));

    jButton2.setFont(new java.awt.Font("Lucida Sans", 0, 12)); // NOI18N
    jButton2.setText("Generar Gráfica");
    jButton2.addActionListener(new java.awt.event.ActionListener() {
```

Se utiliza una búsqueda en el ordenador para poder seleccionar la proveniencia de los datos a utilizar.

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    JFileChooser chooser = new JFileChooser();
    chooser.showOpenDialog(null);
    File F =chooser.getSelectedFile();
    String filename=F.getAbsolutePath();
    jTextField1.setText(filename);
}
```

En base a los datos obtenidos previamente, se realiza una gráfica en un JPanel, la cual no se posee ningún tipo de orden específico, mediante lectores de información.

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

    if (jTextField2.getText().equals("")) {
        JOptionPane.showMessageDialog(null, "Llene todos los Campos");
    } else {
        try {
            FileReader lec=new FileReader(LEER);
            BufferedReader tor=new BufferedReader(lec);

            String line;
            while ((line=tor.readLine())!= null) {
                CONTI++;
            }

            System.out.println("CANTIDAD DE DATOS :"+CONTI);
            DAG=new DATA[CONTI];
            AX=new DATA[CONTI];
            System.out.println("TAMAÑO DE SETEO: "+DAG.length);

            FileReader LEC=new FileReader(LEER);
            BufferedReader BUF=new BufferedReader(LEC);

            String line2;
            while ((line2=BUF.readLine())!= null) {
                String [] datos2=line2.split(",");
                String CATEG0=datos2[0];

                String CANTI=datos2[1];
                DAG[CONTI]=new DATA(CATEG0, CANTI);
                AX[CONTI]=new DATA(CATEG0, CANTI);
                CONTI++;
            }

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

            for (int i = 0; i <DAG.length-1; i++) {
                System.out.println("PAIS:"+ DAG[i].getCATO());
                System.out.println("CANTIDAD:"+ DAG[i].getCANTI());
                System.out.println("-----");
            }

            System.out.println("TAMAÑO: "+CONTI);

            JFreeChart filas=null;
            DefaultCategoryDataset datos=new DefaultCategoryDataset();
            String Y =DAG[0].getCANTI();
            String X =DAG[0].getCATO();
            titulo=jTextField2.getText();
            for (int i = 1; i < DAG.length-1; i++) {
                float can=Float.parseFloat(DAG[i].getCANTI());
                String cate=DAG[i].getCATO();
                datos.addValue(can, "", cate);
            }

            filas=ChartFactory.createBarChart3D(titulo, X, Y, datos, PlotOrientation.VERTICAL, true,true,false);

            ChartPanel panel = new ChartPanel(filas);
            panel.setPreferredSize(new Dimension(350,150));
            jPanel1.setLayout(new BorderLayout());
            jPanel1.add(panel,BorderLayout.NORTH);
            pack();
            repaint();

            try {
                BufferedImage imagen = new BufferedImage(jPanel1.getWidth(), jPanel1.getHeight(), BufferedImage.TYPE_INT_ARGB);
                jPanel1.paint(imagen.getGraphics());
                ImageIO.write(imagen, "png", new File("Desordenada.png"));
            } catch (Exception e) {
            }
        }
    }
}
```

Se realiza una gráfica abierta a 4 posibilidades mediante métodos de ordenamiento (Bubblesort y Shellsort) y enfoques de orden (Ascendente y Descendente), los cuales se ubicarán en un segundo JPanel.

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    if (Up.isSelected() && Bub.isSelected()){
        String xxx;
        String yyy;

        for (int i = 1; i < (CONTI - 1); i++) {
            for (int j = 1; j < (CONTI - 1); j++) {
                float num1=Float.parseFloat(DAG[j].getCANTI());
                float num2=Float.parseFloat(DAG[j + 1].getCANTI());
                if (num1 > num2) {
                    yyy = DAG[j].getCANTI();
                    xxx = DAG[j].getCATO();
                    DAG[j].setCANTI(DAG[j + 1].getCANTI());
                    DAG[j].setCATO(DAG[j + 1].getCATO());
                    DAG[j + 1].setCANTI(yyy);
                    DAG[j + 1].setCATO(xxx);
                }
            }
        }

        System.out.println("DATOS ORDENADOS POR CANTIDAD: ");
        System.out.println("-----");
        for (int i =0; i<CONTI; i++) {
            System.out.println(" CANTIDAD: " + DAG[i].getCANTI());
            System.out.println(" PAIS: " + DAG[i].getCATO());
            System.out.println("-----");
        }

        JFreeChart filas=null;
        DefaultCategoryDataset datos=new DefaultCategoryDataset();
        String Y =DAG[0].getCANTI();
        String X =DAG[0].getCATO();
        titulo=jTextField2.getText();

        for (int i = 1; i < DAG.length-1; i++) {
            float can=Float.parseFloat(DAG[i].getCANTI());
            String cate=DAG[i].getCATO();
            datos.addValue(can, "", cate);
        }

        filas=ChartFactory.createBarChart3D(titulo, X, Y, datos, PlotOrientation.VERTICAL, true,true,false);

        ChartPanel panel = new ChartPanel(filas);
        panel.setPreferredSize(new Dimension(350,150));
        jPanel2.setLayout(new BorderLayout());
        jPanel2.add(panel,BorderLayout.NORTH);
        pack();
        repaint(2);

        try {
            BufferedImage imagen = new BufferedImage(jPanel2.getWidth(), jPanel2.getHeight(), BufferedImage.TYPE_INT_ARGB);
            jPanel2.paint(imagen.getGraphics());
            ImageIO.write(imagen, "png", new File("Ordenada.png"));
        } catch (Exception e) {
        }
    }

    else if (Up.isSelected() && Shell.isSelected()){
        int salto, j, k;
        float aaa, eee;
        String yyy;
        String xxx;

        salto = ((CONTI - 1) / 2);
        while (salto > 0) {
            for (int i = salto + 1; i < CONTI; i++) {
                j = (i - salto);
                while (j >= 1) {
                    aaa = Float.parseFloat(DAG[j].getCANTI());
                    eee = Float.parseFloat(DAG[j + salto].getCANTI());
                }
            }
        }
    }
}
```

Se realiza un reporte en el cual se incluye la información brindada de una forma ordenada y desordenada, se incluye la información del estudiante que lo realizó, el tipo de ordenamiento manejado y las gráficas realizadas.

```
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String SFR="";
        if (Shell.isSelected()){
            SFR="Shellsort";
        } else if (Bubble.isSelected()){
            SFR="BubbleSort";
        }
        String Ruta = "Reporte.html";
        PrintWriter escritor = new PrintWriter(Ruta, "UTF-8");
        escritor.println("<DOCTYPE HTML>");
        escritor.println("<html>");
        escritor.println("<head>");
        escritor.println("<meta charset= " + " utf-8 " + " >");
        escritor.println("<title>Reporte: datos </title>");
        escritor.println("</head>");

        escritor.println("<h1 align= " + "center" + ">" + "Reporte de datos</h1>");
        escritor.println("<h2 align= " + "center" + ">" + "Nombre: Mauricio Antonio Castro Guerra </h2>");
        escritor.println("<h2 align= " + "center" + ">" + "Carnet: 202100299 </h2>");
        escritor.println("<h3>" + " Se está utilizando el método de ordenamiento:" + SFR + "</h3>");
        escritor.println("<br>");

        escritor.println("<div align= " + "center" + " size= " + "4" + " width= " + "100% " + "color= " + "White" + ">");
        escritor.println("<br>");
        escritor.println("<br>");
        escritor.println("<div align= " + "center" + ">");
        escritor.print("<img " + " src= " + "Desordenada.png" + ">");
        escritor.println("</div>");
        escritor.println("<br>");

        escritor.println("<h1 align= " + "center>" + "Información Desordenada :</h1>");
        escritor.println("<div style= " + " text-align:center; " + ">");
        escritor.println("<table border= " + " 2 " + " align= " + "center" + " width= " + "30% " + " >");
        escritor.println("<thead>");

        escritor.println("</thead>");
        escritor.println("<tbody>");
        for (int i = 0; i < CONTI; i++) {
            String cat = AM[i].getCATO();
            escritor.println("<td>" + cat + "</td>");

            String can = AM[i].getCANTI();
            escritor.println("<td>" + can + "</td>");

            escritor.println("</tr>");
        }

        escritor.println("</tbody>");
        escritor.println("</table>");
        escritor.println("</div>");

        escritor.println("<h1 align= " + "center" + " size= " + "4" + " width= " + "100% " + "color= " + "White" + ">");
        escritor.println("<br>");
        escritor.println("<div style= " + " text-align:center; " + ">");
        escritor.println("<table border= " + " 2" + " align= " + "center" + " width= " + "30% " + ">");
        escritor.println("<thead>");
        escritor.println("<tr>");
        escritor.println("<th>CANTIDAD</th>");
        escritor.println("</tr>");
        escritor.println("</thead>");
        escritor.println("<tbody>");
        for (int i = 0; i < CONTI; i++) {
            String cat = DAG[i].getCATO();
```