

Universidad de San Carlos de Guatemala
Escuela de ciencias y sistemas
Introducción a la computación y programación 1
Sección: F
Ing. William Estuardo Escobar Argueta
Aux. Héctor Josué Orozco Salazar



Práctica #2:

Algoritmos de ordenamiento (Java)

Manual técnico

Nombre: David Andrés Jimenez Paniagua

Carnet: 202004777

Fecha: 20 de Marzo de 2022

Este proyecto está desarrollado en Lenguaje Java, con el IDE Apache NetBeans. Para el desarrollo de este proyecto se utilizó la función drag & drop, para la realización de una interfaz gráfica.

Lectura de archivo:

La aplicación solo soporta archivos .csv. Con la ayuda de la librería nativa de java: java.io.*: se carga el archivo .csv, ya sea examinando los documentos o ingresando la ruta hacia el mismo, creando una tabla para verificar que los datos se cargaran correctamente.

```
//Modelo para la tabla
DefaultTableModel modeloTabla;
String direccion = txtRuta.getText();
File file = new File(direccion);
int m = 0;
try{
    String texto = "";
    Object cabeceras [] ={"X", "Y"};
    modeloTabla = new DefaultTableModel (cabeceras, 0);
    File abrir = file.getAbsolutePath();
    Object[] elemento = new Object[2];
    //Abre el archivo y si no está vacío realiza:
    if (abrir != null){
        FileReader carp = new FileReader(abrir);
        BufferedReader leer = new BufferedReader (carp) ;

        while ((texto = leer.readLine()) !=null){
            String registro[] = texto.split(",");
            elemento[0] = registro[0];
            ElementosX[m] = registro[0];
            OriginalesX[m] = registro[0];
            elemento[1] = registro[1];
            ElementosY[m] = Integer.parseInt(registro[1]);
            OriginalesY[m] = Integer.parseInt(registro[1]);
            m++;
            modeloTabla.addRow (elemento);
        }
        Tabla.setModel(modeloTabla);
    }
    //En caso de error, muestra el error
}catch (IOException e){
    JOptionPane.showMessageDialog(null,"Error: " +e);
}
}
```

Para realizar la carga, se necesitó un split, usando como separador “,”, el cual iba agregando los elementos del archivo en un par de arreglos (uno para X y otro para Y).

```

//Modelo para la tabla
DefaultTableModel modeloTabla;
//Creación del objeto FileChooser, para que el usuario elija el archivo
JFileChooser Doc = new JFileChooser ();
//Filtro para archivos CSV
FileNameExtensionFilter filtro = new FileNameExtensionFilter ("*.CSV", "csv");
//Le indicamos el fiiltro
Doc.setFileFilter(filtro);
//Ventana para seleccionar el archivo
int seleccion = Doc.showOpenDialog (this);
//Variable contador
int m = 0;
//Si se selecciona un archivo existente
if (seleccion == JFileChooser.APPROVE_OPTION) {
    //Selecciona el fichero
    File fichero = Doc.getSelectedFile();
    //Escribe la ruta del fichero seleccionado en el textField
    this.txtRuta.setText(fichero.getAbsolutePath ());
    //Si cumple con las características convertirá el archivo en una tabla
    try{
        String texto = "";
        Object cabeceras [] ={ "X", "Y"};
        modeloTabla = new DefaultTableModel (cabeceras, 0);
        File abrir = Doc.getSelectedFile();
        Object[] elemento = new Object[2];
        //Abre el archivo y si no está vacío realiza:
        if (abrir != null){
            FileReader carp = new FileReader(abrir);
            BufferedReader leer = new BufferedReader (carp);
            while ((texto = leer.readLine()) !=null){
                String registro[] = texto.split (",");
                elemento[0] = registro[0];
                ElementosX[m] = registro[0];
                OriginalesX[m]=registro[0];
                elemento[1] = registro[1];
                ElementosY[m] = Integer.parseInt(registro[1]);
                OriginalesY[m] = Integer.parseInt(registro[1]);
                m++;
                //Se agrega el elemento en la tabla
                modeloTabla.addRow (elemento);
            }
            Tabla.setModel(modeloTabla);
        }
        //En caso de error, muestra el error
    }catch (IOException e){
        JOptionPane.showMessageDialog(null,"Error: " +e);
    }
}

```

Graficar:

Para la graficación se utilizó la librería JFreeChart. Con la ayuda de los arreglos definidos anteriormente, se pasó esta información a un Dataset, con el cual se graficó.

```

//Declaración de variables auxiliares
int auxY;
String auxX;
String Nombre;
Nombre = txtTitulo.getText();
//Creación para almacenar datos
DefaultCategoryDataset datos = new DefaultCategoryDataset ();

if( Nombre.equals("") || Nombre.equals("TÍTULO DE LA GRÁFICA")){
JOptionPane.showMessageDialog(null, "Ingrese un nombre para la gráfica");
}else{

    //Asignación de datos X y Y en la grafica
    for(int i = 0;i<ElementosX.length ;i++){
        if(ElementosY[i] != 0){
            auxX = String.valueOf(ElementosX[i]);
            auxY = ElementosY[i];
            datos.setValue(auxY,"Países",auxX);
        }
    }

    //Creación de la grafica
    JFreeChart grafica = ChartFactory.createBarChart(Nombre,
        "Países",
        "Cantidades",
        datos,
        PlotOrientation.VERTICAL,
        true,
        true,
        false
    );

    //Instrucciones para hacer visible la grafica
    ChartPanel panel = new ChartPanel(grafica);
    panel.setMouseWheelEnabled (false);
    panel.setPreferredSize(new Dimension(1210, 450));

    PGráfica.setLayout (new BorderLayout ());
    PGráfica.add(panel, BorderLayout.NORTH);
    panel.removeAll();
    panel.revalidate();
    panel.repaint();

}

```

Algoritmos:

En este proyecto se utilizaron dos algoritmos de ordenamiento, los cuales son:

- **BubbleSort:**

Bubble Sort u Ordenamiento de burbuja, es el algoritmo de clasificación más simple que funciona intercambiando repetidamente los elementos adyacentes si están en el orden incorrecto. Acá fueron utilizados dos distintas maneras, la primera fue para ordenar el algoritmo de forma ascendente:

```
//ordenamiento BubbleSort ascendente
public void AlA() {
    int contador = 0;
    String Nombre;
    Nombre = txtTitulo.getText();
    DefaultCategoryDataset datos = new DefaultCategoryDataset();
    for (int i = 0; i < ElementosY.length - 1; i++) {
        for (int j = 0; j < ElementosY.length - i - 1; j++) {
            if ((int)ElementosY[j] != 0) {
                if ((int) ElementosY[j] > (int) ElementosY[j + 1]) {
                    // swap arr[j+1] and arr[j]
                    int temp = (int) ElementosY[j];
                    ElementosY[j] = ElementosY[j + 1];
                    ElementosY[j + 1] = temp;
                    contador++;
                }
            }
            Limpiar();
            contador++;
            String auxX = String.valueOf(ElementosX[j]);
            int auxY = ElementosY[j];
            txtContador.setText(String.valueOf(contador));
            datos.setValue(auxY, "Países", auxX);

            JFreeChart grafica = ChartFactory.createBarChart(Nombre,
                "Países",
                "Cantidades",
                datos,
                PlotOrientation.VERTICAL,
                true,
                true,
                false
            );

            //Instrucciones para hacer visible la grafica
            ChartPanel panel = new ChartPanel(grafica);
            panel.setMouseWheelEnabled(false);
            panel.setPreferredSize(new Dimension(1210, 450));

            PGrafica.setLayout(new BorderLayout());
            PGrafica.add(panel, BorderLayout.NORTH);

            panel.removeAll();
            panel.revalidate();
            panel.repaint();
        }
    }
}
```

Y la segunda forma, de manera descendente:

```
//ordenamiento BubbleSort Descendiente
public void A1D(){
    int contador = 0;
    String Nombre;
    Nombre = txtTitulo.getText();
    DefaultCategoryDataset datos = new DefaultCategoryDataset();
    for (int i = 0; i < ElementosY.length - 1; i++) {
        for (int j = 0; j < ElementosY.length - i - 1; j++) {
            if ((int)ElementosY[j] != 0) {
                if ((int) ElementosY[j] < (int) ElementosY[j + 1]) {
                    // swap arr[j+1] and arr[j]
                    int temp = (int) ElementosY[j];
                    ElementosY[j] = ElementosY[j + 1];
                    ElementosY[j + 1] = temp;
                    txtContador.setText(String.valueOf(contador));
                    Limpiar();
                    contador++;
                    String auxX = String.valueOf(ElementosX[j]);
                    int auxY = ElementosY[j+1];
                    txtContador.setText(String.valueOf(contador));
                    datos.setValue(auxY, "Países", auxX);
                }
            }
        }
    }
}
```

- **InsertionSort:**

La clasificación por inserción es un algoritmo de clasificación simple que funciona de manera similar a la forma en que clasifica las cartas en sus manos. La matriz se divide virtualmente en una parte ordenada y otra no ordenada. Los valores de la parte no ordenada se seleccionan y colocan en la posición correcta en la parte ordenada.

Al igual que en otro método, fueron utilizados dos distintas maneras, la primera fue para ordenar el algoritmo de forma ascendente:

```
//Ordenamiento InsertionSort Ascendente
public void A2A(){
    DefaultCategoryDataset datos = new DefaultCategoryDataset();
    int contador = 0;
    String Nombre;
    Nombre = txtTitulo.getText();
    int n = ElementosY.length;
    for (int i = 0; i < n; ++i) {
        int key = ElementosY[i];
        int j = i - 1;

        while (j >= 0 && ElementosY[j] > key) {
            ElementosY[j + 1] = ElementosY[j];
            j = j - 1;
            txtContador.setText(String.valueOf(contador));
            contador++;
            Limpiar();
            String auxX = String.valueOf(ElementosX[j+1]);
            int auxY = ElementosY[j+1];
            txtContador.setText(String.valueOf(contador));
            datos.setValue(auxY, "Países", auxX);
        }
        ElementosY[j + 1] = key;
    }
}
```

```

        JFreeChart grafica = ChartFactory.createBarChart(Nombre,
        "Países",
        "Cantidades",
        datos,
        PlotOrientation.VERTICAL,
        true,
        true,
        false
        );

        //Instrucciones para hacer visible la grafica
        ChartPanel panel = new ChartPanel(grafica);
        panel.setMouseWheelEnabled (false);
        panel.setPreferredSize(new Dimension(1210, 450));

        PGrafica.setLayout (new BorderLayout ());
        PGrafica.add(panel, BorderLayout.NORTH);

        panel.removeAll();
        panel.revalidate();
        panel.repaint();
    }
    ElementosY[j + 1] = key;
}

```

Y la segunda forma, de manera descendente:

```

//Ordenamiento InsertionSort Descendente
public void A2D(){
    DefaultCategoryDataset datos = new DefaultCategoryDataset();
    int contador = 0;
    String Nombre;
    Nombre = txtTitulo.getText();
    int n = ElementosY.length;
    for (int i = 0; i < n; ++i) {
        int key = ElementosY[i];
        int j = i - 1;

        while (j >= 0 && ElementosY[j] < key) {
            ElementosY[j + 1] = ElementosY[j];
            j = j - 1;
            txtContador.setText(String.valueOf(contador));
            contador++;
            Limpiar();
            String auxX = String.valueOf(ElementosX[j+1]);
            int auxY = ElementosY[j+1];
            txtContador.setText(String.valueOf(contador));
            datos.setValue(auxY, "Países", auxX);
        }
    }
}

```

```

        JFreeChart grafica = ChartFactory.createBarChart(Nombre,
        "Países",
        "Cantidades",
        datos,
        PlotOrientation.VERTICAL,
        true,
        true,
        false
        );

        //Instrucciones para hacer visible la grafica
        ChartPanel panel = new ChartPanel(grafica);
        panel.setMouseWheelEnabled (false);
        panel.setPreferredSize(new Dimension(1210, 450));

        PGrafica.setLayout (new BorderLayout ());
        PGrafica.add(panel, BorderLayout.NORTH);

        panel.removeAll();
        panel.revalidate();
        panel.repaint();
    }
    ElementosY[j + 1] = key;
}
}

```

Para cada método se añadió un contador de pasos, para conocer en cuántos movimientos realizaba el ordenamiento.

Reportes:

Para realizar los reportes se crea un archivo HTML, en el cual se presentan datos varios de la simulación, además del algoritmo original y ordenado en forma de tabla.


```

// para crear un archivo
File archivo = new File("Reportes\\"+Nombre+".html");
// aperturar un archivo
FileWriter escribir;
// crear nuevas lineas dentro un archivo
PrintWriter nuevaLinea;

// verifico si mi archivo existe o no existe
if(!archivo.exists()){
    try {
        //crear mi archivo
        archivo.createNewFile();
        // le pasao el archivo donde quiero escribir, true
        escribir = new FileWriter(archivo,true);
        // escribiendo dentro de mi archivo
        nuevaLinea = new PrintWriter(escribir);
        nuevaLinea.println("<!DOCTYPE html>\n" +
            "    <html lang=\"en\">\n" +
            "        <head>\n" +
            "            <meta charset=\"UTF-8\">\n" +
            "            <title>Document</title>\n" +
            "        </head>\n" +
            "        <body>");

        nuevaLinea.println("<h1>REPORTE DE ORDENAMIENTO</h1>");
        nuevaLinea.println("<h2>Nombre: David Andrés Jimenez Paniagua</h2>");
        nuevaLinea.println("<h2>Carnet: 202004777</h2>");
        nuevaLinea.println("<h1>Nombre de la gráfica: " +Nombre+ "</h1>");
        nuevaLinea.println("<h2>Algoritmo utilizado: " +Algoritmo+ "</h2>");
        nuevaLinea.println("<h2>LOs datos fueron ordenados de forma: " +Forma+ "</h2>");
        nuevaLinea.println("<h2>Número de pasos: " +pasos+ "</h2>");
        nuevaLinea.println("<h2>Datos Originales: </h2>");
        nuevaLinea.println("<table>");
    }
}

```

```

        nuevaLinea.println("<table>");
        for(int i=0; i<OriginalesY.length;i++){
            //solo cuando los datos existan
            if(OriginalesX[i] != null){
                nuevaLinea.println("<tr>");
                nuevaLinea.print("<td>");
                nuevaLinea.print(OriginalesX[i]);
                nuevaLinea.print("</td>");
                nuevaLinea.print("<td>");
                nuevaLinea.print(OriginalesY[i]);
                nuevaLinea.print("</td>");
                nuevaLinea.println("</tr>");
            }
        }
        nuevaLinea.println("</table>");

        nuevaLinea.println("<h2>Datos Ordenados: </h2>");
        nuevaLinea.println("<table>");
        for(int i=0; i<ElementosY.length;i++){
            //solo cuando los datos existan
            if(ElementosX[i] != null){
                nuevaLinea.println("<tr>");
                nuevaLinea.print("<td>");
                nuevaLinea.print(ElementosX[i]);
                nuevaLinea.print("</td>");
                nuevaLinea.print("<td>");
                nuevaLinea.print(ElementosY[i]);
                nuevaLinea.print("</td>");
                nuevaLinea.println("</tr>");
            }
        }
        nuevaLinea.println("</table>");
    }
}

```