Manual Técnico

Para el desarrollo de este programa fue necesario el uso de 5 librerías externas. JFreechart y JCommon para poder implementar las gráficas de barra, JSON para poder leer archivos y decodificarlos, iText y JTattoo para la creación de los reportes en PDF y HTML. Con la ayuda de estas librerías, el uso de Threads (Hilos) en Java y algoritmos de ordenamiento se pudo brindar un programa el cual lee archivos JSON, los decodifica, almacena sus variables y al iniciar el ordenamiento con la ayuda de los hilos estimar el tiempo que toma en ordenar una n cantidad de datos y contando los pasos que realizo durante todo el proceso, para así obtener una gráfica ordenada con sus reportes.

Variables globales

```
//Variables
    File archivo;
    FileReader fr;
    BufferedReader bfr;
    public static int[] datosArreglos;
    static int b;
    static int contador = 0;
    static String title;
    //Grafica 1
    public static DefaultCategoryDataset dataset;
    public static JFreeChart chart;
    public static ChartPanel panel;
    public static JPanel panelNuevo;
    //Grafica 2
    public static DefaultCategoryDataset dataset2;
    public static JFreeChart chart2;
    public static ChartPanel panel2;
    public static JPanel panelNuevo2;

    public static boolean iniciahilo = true;
    public static boolean iniciahilo2 = true;
```

Estas variables globales staticas servirán para todo el proceso y funcionamiento de nuestro programa, tenemos el arreglo principal el cual contará con los valores leídos en el JSON y variables especiales para las gráficas.

Pantalla Principal

Se usó drag and drop para la creación de la interfaz gráfica.

Botón examinar para la lectura del JSON

```
String contenido = "";
JFileChooser fc = new JFileChooser();
int valor = fc.showOpenDialog(pMain);
if (valor == JFileChooser.APPROVE_OPTION)
{
    System.out.println(" " + fc.getSelectedFile());

    lb_ruta.setText(String.valueOf(fc.getSelectedFile()));
    archivo = fc.getSelectedFile();
} else
{
    JOptionPane.showMessageDialog(this, "No se ha seleccionado un archivo");
}

fr = new FileReader(archivo);
bfr = new BufferedReader(fr);
String linea;

while ((linea = bfr.readLine()) != null)
{
    contenido += linea;
}
```

Mediante JFileChooser se escoge le archivo a decodificar, posteriormente se lee este archivo y su contenido se almacena en una variable de tipo String.

Decodificación de del archivo JSON

```
JSONParser parser = new JSONParser();
            try
            {
                Object obj = parser.parse(contenido);
                JSONObject jsonObj = (JSONObject) obj;
                System.out.println(jsonObj);
                //Datos
                title = (String) jsonObj.get("title");
                lb_titulograf.setText(" " + title);
                System.out.println("title = " + title);
                //Arreglo
                Object jsonArrayDatos = jsonObj.get("dataset");
                JSONArray datos = (JSONArray) jsonArrayDatos;
                arreglo_desordenado = String.valueOf(datos);
                System.out.println(arreglo_desordenado);
                System.out.println(datos.size());

                datosArreglos = new int[datos.size()];

                //Me imprimo los valores
                for (Object i : datos)
                {
                    String a = String.valueOf(i);
                    b = Integer.parseInt(a);
                    System.out.println(b);
                    //Asignarle el valor al arreglo

                    datosArreglos[contador] = b;
                    System.out.println("Contador " + contador);
                    contador++;

                }
            }
```

Al usar la librería JSON y sus métodos se obtiene le título de la gráfica y el arreglo del dataset el cual se almacenará como valores enteros en el arreglo declarado anteriormente.

Clase Cronometro

```
public static int hora = 0;
    public static int minuto = 0;
    public static int segundo = 0;
    public static String reloj;
    JLabel etiqueta;
    Cronometro(JLabel lb_cronometro) {
        this.etiqueta = lb_cronometro;
    }
    @Override
    public void run(){
        try
        {
            int x = 0;
            while(FrameMain.iniciahilo){
                Thread.sleep(1000);
                ejecutarHilo(x);
                x++;
            }

        } catch (Exception e)
        {
            System.out.println("EXCEPTION EN EL CRONOMETRO" + e.getMessage());
        }
    }

    private void ejecutarHilo(int x) {
        //System.out.println(x +" - "+ Thread.currentThread().getName());
        segundo++;
        if(segundo > 59){
            segundo = 0;
            minuto ++;
            if(minuto>59){
                minuto = 0;
                hora++;
            }
        }

        String textoSegundo = "";
        String textoMinuto = "";
        String textoHora = "";

        if(segundo < 10){
            textoSegundo = "0"+segundo;
        }else{
            textoSegundo = ""+segundo;
        }

        if(minuto < 10){
            textoMinuto = "0"+minuto;
        }else{
            textoMinuto = ""+minuto;
        }

        if(hora < 10){
            textoHora = "0"+hora;
        }else{
            textoHora = ""+hora;
        }

        reloj = textoHora+":"+textoMinuto+":"+textoSegundo;
        etiqueta.setText(reloj);
    }
```

Esta clase trabaja con hilos y simula un cronometro el cual cuenta en horas, minutos y segundos.

Método ordenamiento Quicksort

```java
public static void quicksort(int A[], int izq, int der) {
        try
        {
            int pivote = A[izq];
            int i = izq;
            int j = der;
            int aux;

            while (i < j)
            {

                while (A[i] <= pivote && i < j)
                {
                    i++;
                }

                while (A[j] > pivote)
                {
                    j--;
                }

                imprimirConsola();
                FrameMain.imprimirGrafica();
                Thread.sleep(750);

                if (i < j)
                {
                    aux = A[i];
                    A[i] = A[j];
                    A[j] = aux;
                    pasos++;
                    pasoslb.setText(String.valueOf(pasos));
                    FrameMain.imprimirGrafica();


                }
            }
            A[izq] = A[j];
            A[j] = pivote;
            FrameMain.imprimirGrafica();
            if (izq < j - 1)
            {
                quicksort(A, izq, j - 1);
            }
            if (j + 1 < der)
            {
                quicksort(A, j + 1, der);
            }

        } catch (InterruptedException e)
        {
        }
    }
```

Este método es un algoritmo de ordenamiento, es conocido por ser uno de los más rápidos, ya que funciona de la siguiente manera: existe un variable pivote, dos variables que funcionan como flechas las cuales se van moviendo una de derecha a izquierda y la otra de izquierda a derecha, al momento de encontrarse el pivote anteriormente seleccionado se coloca en esa posición y todo número mayor que él ira de lado derecha y los menores a la izquierda, realizando los debidos cambios de posición. Al termina eso, mediante la recursividad se ordena todos los números que se encuentren a la izquierda del pivote implementando lo anteriormente mencionado hasta logar el ordenamiento del lado izquierdo como el derecho. En este caso mandamos a dormir nuestro ordenamiento y sumamos al contador pasos cuando se efectué uno e imprimimos la gráfica para observar en tiempo real como ocurre este ordenamiento.

Método de la gráfica

```
public static void imprimirGrafica() {
        dataset = new DefaultCategoryDataset();
        for (int i = 0; i < datosArreglos.length; i++)
        {
            dataset.addValue(datosArreglos[i], String.valueOf(datosArreglos[i]), "");

        }
        chart = ChartFactory.createBarChart3D(title, "Valores", "", dataset, PlotOrientation.VERTICAL, true, true, false);
        panel = new ChartPanel(chart);
        panel.setLayout(null);
        panel.setBounds(0, 0, 779, 482);
        panelNuevo = new JPanel();
        panelNuevo.setVisible(true);
        panelNuevo.setBounds(30, 150, 779, 482);
        panelNuevo.add(panel);
        pMain.add(panelNuevo);
    }
```

Este método se manda a llamar al momento de realizar un cambio en el ordenamiento, y con la librería JFreechart se crea una gráfica de barra.

Método para crear HTML

```java
public static void html1(String fechaActual) {
        String nombre_reporte;
        File reporteHTML;
        FileWriter fw;
        BufferedWriter buff;
        String contenidoHTML;

        try
        {
            nombre_reporte = "E:\\Practica 2 IPC\\" + FrameMain.title + "_" + fechaActual + ".html";
            reporteHTML = new File(nombre_reporte);
            fw = new FileWriter(reporteHTML);
            buff = new BufferedWriter(fw);
            //Texto en HTML
            contenidoHTML = "<html>\n"
                    + "    <head>\n"
                    + "        <title>PRACTICA 2 </title>\n"
                    + "    </head>\n"
                    + "    <body>\n"
                    + "        <h2>Alvaro Norberto Garcia Meza</h2>\n"
                    + "        <h3>Carnet: 202109567</h3>\n"
                    + "        <h3>Ordenamiento Utilizado: QUICKSORT</h3>\n"
                    + "        <h3>Transcurrieron: " + Cronometro.reloj + "</h3>\n"
                    + "        <h3>Pasos efectuados: " + pasosEfectuados + "</h3>\n"
                    + "        <h3>Arreglo Desordenado: " + FrameMain.arreglo_desordenado + "</h3>\n"
                    + "        <h3>Arreglo Ordenado: [" + datos_ordenados + "]</h3>\n"
                    + "        <div style=\"text-align: center;\">\n"
                    + "            <img src=\"E:\\\\Practica 2 IPC\\\\Grafica1.png\">"
                    + "        </div>"
                    + "    </body>\n"
                    + "</html>";
            buff.write(contenidoHTML);
            buff.close();
            fw.close();
            //pdfOrden1(fechaActual, contenidoHTML);
            Document documentoHTML = new Document();
            documentoHTML.open();
            HTMLWorker htmlWorker = new HTMLWorker(documentoHTML);
            htmlWorker.parse(new StringReader(contenidoHTML));
            documentoHTML.close();

        } catch (Exception e)
        {
        }
    }
```

Con este método y la librería itext se puede crear páginas Web utilizando HTML el cual contiene los datos requeridos.

Método para la creación de PDFs

```java
public static void pdfOrden1(String fechaActual){
        if (!FrameMain.iniciahilo)
        {
            System.out.println("Se ha terminado el ordenamiento, precede a generar el pdf y limpiar todo");
            //Hacer png la gráfica
            try
            {
                ChartRenderingInfo info = new ChartRenderingInfo(new StandardEntityCollection());
                File fileGraf = new File("E:\\Practica 2 IPC\\Grafica1.png");
                ChartUtilities.saveChartAsPNG(fileGraf, FrameMain.chart, 500, 400, info);
            } catch (IOException e)
            {
            }
            //Creacion del pdf
            Document documento = new Document(PageSize.LETTER);
            Image grafica = Image.getInstance("E:\\Practica 2 IPC\\Grafica1.png");
            OutputStream archivo;
            archivo = new FileOutputStream("E:\\Practica 2 IPC\\" + FrameMain.title + "_" + fechaActual + ".pdf");
            PdfWriter.getInstance(documento, archivo);
            //Abrir
            documento.open();
            //Paragraphs
            Paragraph nombre = new Paragraph();
            nombre.add("Alvaro Norberto García Meza");
            nombre.setAlignment(Element.ALIGN_LEFT);

            Paragraph carnet = new Paragraph();
            carnet.add("Carnet: 202109567");
            carnet.setAlignment(Element.ALIGN_LEFT);

            Paragraph tipo = new Paragraph();
            tipo.add("Ordenamiento utilizado: QUICKSORT");
            tipo.setAlignment(Element.ALIGN_LEFT);

            Paragraph tiempo = new Paragraph();
            tiempo.add("Transcurrieron: " + Cronometro.reloj);
            tiempo.setAlignment(Element.ALIGN_LEFT);

            Paragraph pasos = new Paragraph();
            pasos.add("Pasos efectuados: " + pasosEfectuados);
            pasos.setAlignment(Element.ALIGN_LEFT);

            Paragraph datosDesor = new Paragraph();
            datosDesor.add("Arreglo Desordenado: " + FrameMain.arreglo_desordenado);
            datosDesor.setAlignment(Element.ALIGN_LEFT);

            Paragraph datosOrde = new Paragraph();
            datosOrde.add("Arreglo Ordenado: " + "[" + datos_ordenados + "]");
            datosOrde.setAlignment(Element.ALIGN_LEFT);

            //Agregar gráfica
            grafica.setAlignment(Element.ALIGN_CENTER);
```

Este método crea un pdf con los datos requeridos.