

ALDO SAÚL VÁSQUEZ MOREIRA

CARNET: 202109754

FACULTAD DE INGENIERÍA

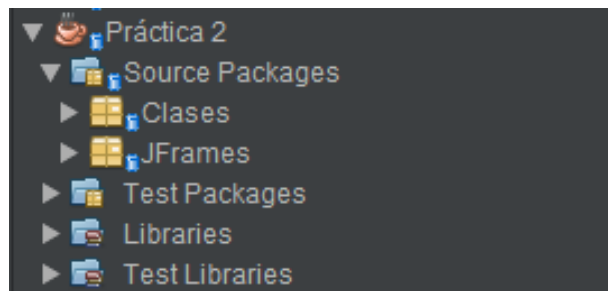
ESCUELA DE CIENCIAS Y SISTEMAS

LABORATORIO DE INTRODUCCIÓN A LA PROGRAMACIÓN 1



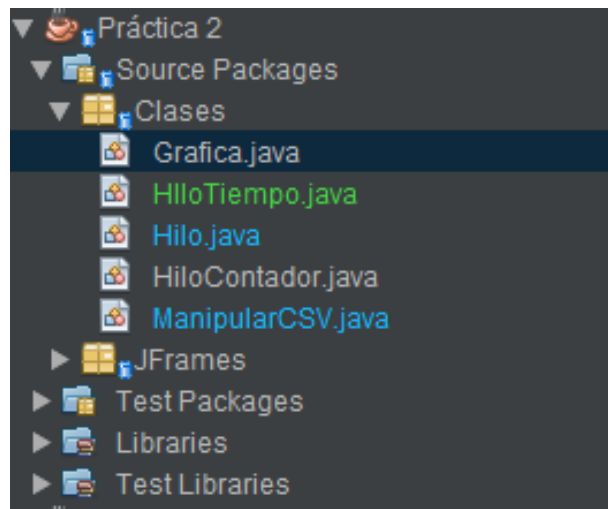
MANUAL TÉCNICO

El paquete del proyecto tiene la siguiente estructura:



En el cual se contienen los paquetes Clases y JFrames.

El paquete clases contiene cada una de las clases necesarias para el funcionamiento del programa:



La clase “Gráfica.java” es la encargada de recibir los datos del archivo .csv para luego representar los datos por medio de un gráfico de barras. Para cumplir con esta tarea esta clase utiliza la librería JFreeChart.

La clase “HiloTiempo.java” es una clase que herede de la clase Observable e implemente la interfaz Runnable. Todo lo anterior con la finalidad de crear un hilo que permita realizar el algoritmo de ordenamiento paralelamente a la toma del tiempo que este ordenamiento tarda.C

La clase “Hilo.java” cumple una función importante puesto que contiene todos los algoritmos de ordenamiento utilizados:

```
package Classes;

import Classes.Manipulacion;
import Classes.Grafica;
import Classes.HiloContador;
import javax.swing.JProgressBar;
import javax.swing.JOptionPane;

public class Hilo extends Thread {

    private void esperarXsegundos(int segundos) {
        try {
            Thread.sleep(segundos * 1000);
        } catch (Exception e) {
            Thread.currentThread().interrupt();
        }
    }

    public void quickSortAscendente(int[] datos, String datosK[], int a, int b) {
        int pivote = datos[a];
        String pivoteK = datosK[a];
        int i = a;
        int j = b;
        int aux;
        String auxK;
        while (i < j) {
            while (datos[i] <= pivote && i < j) {
                i++;
            }
            while (datos[j] > pivote) {
                j--;
            }
            if (i < j) {
                aux = datos[i];
                datos[i] = datos[j];
                datos[j] = aux;
                auxK = datosK[i];
                datosK[i] = datosK[j];
                datosK[j] = auxK;
            }
        }
        if (i < j) {
            quickSortAscendente(datos, datosK, i, j);
        }
    }
}
```

Esta hereda a la clase Thread puesto que el objetivo es obtener un hilo de esta clase ya que busca que el ordenamiento se realice de forma paralela a la graficación, conteo del tiempo, conteo de los pasos, etc.

También contiene un método interesante llamado “esperarXsegundos” el cual tiene un parámetro de tipo int el cual indicará la cantidad de segundos que queremos que el programa espere para realizar la siguiente tarea:

```
private void esperarXsegundos(int segundos) {
    try {
        Thread.sleep(segundos * 1000);
    } catch (Exception e) {
        Thread.currentThread().interrupt();
    }
}
```

La clase “HiloContador.java” es la encargada de llevar el conteo de los pasos de los algoritmos de ordenamiento. Esta es prácticamente una réplica de la clase “Hilo.java” por lo cual se hizo una copia de los arreglos de dicha clase

ya que el objetivo de este hilo no es ordenar los datos sino que llevar el conteo de los pasos de dicho ordenamiento.

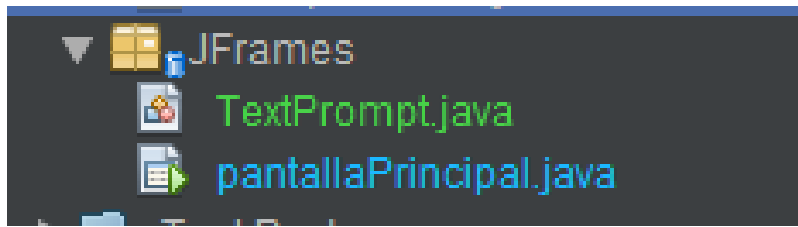
Y, por último, tenemos la clase “ManipularCSV.java” la cual es la encargada de convertir la información del archivo .csv y trasladarla a arreglos de tipo String e Integer.

Esta clase contiene dos métodos ya que como se indica, el usuario ingresa la información por medio de una caja de texto o por medio del explorador de archivos.

```
public void leerCSVString(String nombreArchivo) {
    try {
        lector = new BufferedReader(new FileReader(nombreArchivo));
        while ((lines = lector.readLine()) != null) {
            partesFila = lines.split(",");
            contador++;
            for (int i = 0; i < partesFila.length; i++) {
                if (i == 0) {
                    contadorFila = 1;
                    if (contadorFila == 0) {
                        valores[i] = partesFila[i];
                    }
                    if (contadorFila > 0) {
                        contadorFila = 1;
                        valores[contadorFila] = Integer.parseInt(partesFila[i]);
                        valores_contador[contadorFila] = Integer.parseInt(partesFila[i]);
                        valores_contador_nro_canchas[contadorFila] = Integer.parseInt(partesFila[i]);
                    }
                }
                if (i == 0) {
                    contadorFila = 1;
                    if (contadorFila == 0) {
                        valores[i] = partesFila[i];
                    }
                    if (contadorFila > 0) {
                        contadorFila = 1;
                        valores[contadorFila] = Integer.parseInt(partesFila[i]);
                        valores_contador[contadorFila] = Integer.parseInt(partesFila[i]);
                        valores_contador_nro_canchas[contadorFila] = Integer.parseInt(partesFila[i]);
                    }
                }
            }
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

```
public void leerCSVFile(File archivo) {
    try {
        lector = new BufferedReader(new FileReader(archivo));
        while ((lines = lector.readLine()) != null) {
            partesFila = lines.split(",");
            contador++;
            for (int i = 0; i < partesFila.length; i++) {
                if (i == 0) {
                    contadorFila = 1;
                    if (contadorFila == 0) {
                        valores[contadorFila] = Integer.parseInt(partesFila[i]);
                        valores_contador[contadorFila] = Integer.parseInt(partesFila[i]);
                        valores_contador_nro_canchas[contadorFila] = Integer.parseInt(partesFila[i]);
                    }
                }
                if (i == 0) {
                    contadorFila = 1;
                    if (contadorFila == 0) {
                        valores[contadorFila] = Integer.parseInt(partesFila[i]);
                        valores_contador[contadorFila] = Integer.parseInt(partesFila[i]);
                        valores_contador_nro_canchas[contadorFila] = Integer.parseInt(partesFila[i]);
                    }
                }
            }
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

En el paquete JFrames contamos con:



La clase “TextPrompt.java” se ubica aquí ya que su funcionamiento es propiamente para el JFrame pantallaPrincipal.

La clase “pantallaPrincipal.java” es la clase principal del proyecto y es la que contiene tanto las instancias de todos los hilos como las funciones de cada uno de los botones mostrados en el manual de usuario.

Importante es mencionar que se hizo uso de la clase File y BufferedWriter para la creación del reporte.

Además, esta clase, junto a las clases “ManipularCSV.java” y “Grafica.java” hacen uso del patrón de diseño Singleton para evitar crear más de una única instancia de esta clase ya que se busca la conservación de dicha información.