

Método Ventana Principal: en este método se configuraron los datos de la ventana, el tamaño , la ubicación , nombre de la ventana , los layouts .

```
public VentanaPrincipal() {
    super();

    Toolkit tamano=Toolkit.getDefaultToolkit();
    Dimension dimension =tamano.getScreenSize();
    int altura=dimension.height/2;
    int ancho=dimension.width/2;
    setBounds(ancho-415, altura-450, 830, 900);
    setResizable(false);
    setLayout(null);
    this.setTitle("Proceso de datos");
    panel.setBounds(40, 210, 740, 600);
    panel.setBackground(Color.gray);
    this.add(panel);
    agregar();

    this.addWindowListener(new WindowAdapter() {

        @Override
        public void windowClosing(WindowEvent we) {

        }

    });
}
```

Void agregar(): En este método se agregaron las etiqueta y botones al panel central del programa , se utilizo el setBounds para poder asignarle las posiciones y el tamaño de cada objeto .

```
void agregar() {

    JLabel etiquetaNombre = new JLabel("Ruta : ");
    etiquetaNombre.setBounds(50, 20, 50, 30);
    this.add(etiquetaNombre);

    JTextField txtRuta = new JTextField();
    txtRuta.setBounds(50, 50, 600, 30);
    this.add(txtRuta);

    JLabel etiquetaEdad = new JLabel("Nombre Grafica : ");
    etiquetaEdad.setBounds(50, 90, 200, 30);
    this.add(etiquetaEdad);

    JTextField txtNombre = new JTextField();
    txtNombre.setBounds(50, 120, 730, 30);
    this.add(txtNombre);

    JButton boton = new JButton("Buscar");
    boton.setBounds(680, 50, 100, 30);
    this.add(boton);

    JButton boton2 = new JButton("Aceptar");
    boton2.setBounds(680, 165, 100, 30);
    this.add(boton2);

    VentanaPrincipal referenciaPadre = this;

    boton.addActionListener(
        new ActionListener() {
```

boton2.addActionListener: en esta acción se procedió a leer el archivo seleccionado por el usuario, primero se creó un vector bidimensional que es el que recibe los datos ,luego de haber obtenido la ruta del archivo con la función File se leyó línea por línea el contenido de cada archivo , se utilizó una variable auxiliar que contenía el dato de cada línea , con charAt se fue separando carácter por carácter , dentro de un if se fue leyendo cada carácter y asignándolo al vector bidimensional que contendrá la información , se utilizo una variable “pos” para controlar la posición en x del vector , cada que el programa encontraba una “,” dicha variable aumentaba uno así el programa lograba detectar cada cambio de dato y accedía a la otra posición en x , como dicho archivo consta de 2 columnas cuando la variable llegaba a 1(por que se cuenta desde 0) se volvía a reiniciar y ahora trabajaba en una nueva posición en el eje Y, la variable cantidad fue utilizada para contar cuantas líneas había en el archivo que se estaba leyendo con eso se logra detectar la cantidad total de archivos.

```

boton2.addActionListener(
    new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {

            for(int i=0;i<100;i++){
                contenedor[i][0]="";
                contenedor[i][1]="";
            }
            cantidad=1;

            ruta=txtRuta.getText();
            nombre=txtNombre.getText();
            if((ruta.equals(""))||!(nombre.equals(""))){
                JOptionPane.showMessageDialog(null,"Campos vacios");
            }else{
                File archivo;
                FileReader fr;
                BufferedReader br;
                String nombre=ruta;
                try {
                    archivo=new File(nombre);
                    fr= new FileReader (archivo);
                    br= new BufferedReader(fr);
                    String a;
                    while((a=br.readLine())!=null){
                        cantidad=cantidad+1;
                        int pos=0;
                        for(int i=0;i<a.length();i++){
                            String aux2="" +a.charAt(i);
                            if(aux2.equals(",")) {

```

```

                                contenedor[cantidad][pos]=contenedor[cantidad][pos] + aux2;
                            }
                        }
                    }
                    br.close();
                    fr.close();

                    Grafica grafica=new Grafica(panel,cantidad,contenedor,nombre0);
                    grafica.start();

                } catch (IOException ex) {
                    JOptionPane.showMessageDialog(null,"Error al abrir el archivo: " + ex.getMessage());
                }
            }
        }
    }
}

```

Se instanciaron las variables que se encontraban en la ventana principal, con esto la clase Grafica podrá obtener los datos para seguir con el proceso

```

public class Grafica extends Thread {

    Panel panel;
    String[][]vector;
    String[][]ordenado;
    String[][]desordenado;
    int n;
    String nombre;
    public static int c=0,v=0,alg=0;
    public static int control=0,a=0;
    String algoritmo="",velocidad="",orden="";
    int pasos=0, tiempo=0;
    String dato[][]=new String[2][2];

    public Grafica(Panel panel,int n,String[][] vector,String nombre){
        this.vector=vector;
        this.panel=panel;
        this.n=n;
        this.nombre=nombre;
    }
}

```

Public void run (): Es el inicio de la ejecución del hilo Grafica se realizó mediante el método start() heredado de Thread. start() es un método que invoca a **run()** y devuelve inmediatamente el control a la tarea que lo ha llamado.

Se cargaron los datos enviados de la ventana principal , se utilizo un JOption pane para interactuar con el usuario , luego se guardaron los datos del vector en otro temporal antes de empezar a ordenarlos , luego ser reciben los datos seleccionados del usuario y se procede a verificar el tipo de ordenamiento .

```
public void run() {
    ordenado=new String[n+1][2];
    desordenado=new String[n][2];
    for(int i=0;i<n;i++){
        desordenado[i][0]=vector[i+1][0];
        desordenado[i][1]=vector[i+1][1];
    }
    for(int i=0;i<n;i++){
        dato[i][0]="";
        dato[i][1]="";
    }
    panel.setBackground(Color.white);
    VPrincipal vp=new VPrincipal();
    proceso(vector);
    int op=JOptionPane.showConfirmDialog(null,"Selecciona opciones ", "Ordenar dato",JOptionPane.YES_NO_OPTION,20);
    if(op==JOptionPane.YES_OPTION){
        vp.setVisible(true);
    }
    while(a!=1){
        System.out.print("");
        ovp.ordenar();
        vp.setVisible(true);
        a=vp.ordenar();
        if (vp.ordenar()==1){
            a=1;
            vp.dispose();
        }
    }
}
```

```
if(ordenar){
    algoritmo="burbuja";
    if(ordenar){
        ordena="ascendente";
        burbuja();
    }else if(ordenar){
        ordena="descendente";
        burbuja2();
    }
}
}else if(op==0){
    algoritmo="quicksort";
    if(ordenar){
        ordena="ascendente";
        quicksort(vector,1,n);
    }else if(ordenar){
        ordena="descendente";
        quicksort2(vector,1,n);
    }
    for(int i=0;i<n;i++){
        String [][]tem=new String [n][2];
        temp[i][0]=vector[i+1][0];
        temp[i][1]=vector[i+1][1];
    }
    quicksort2(temp,0,n-1);
}
}else if(op==1){
    algoritmo="ShellSort";
    if(ordenar){
        ordena="ascendente";
        ShellSort(vector, n);
    }else if(ordenar){
        ordena="descendente";
        ShellSort2(vector, n);
    }
}
```

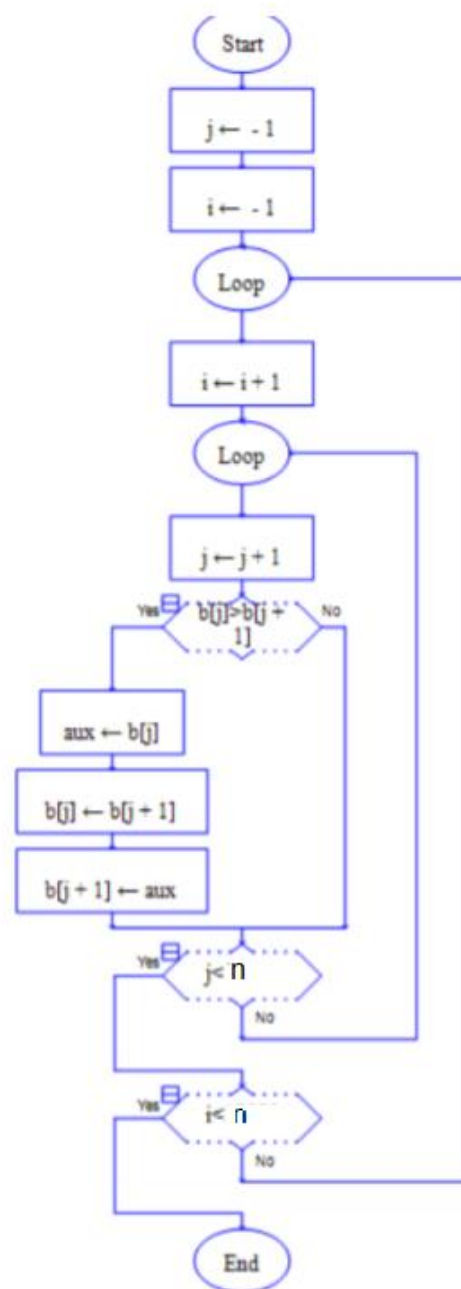
Bubble Sort():

```
public void burbuja() {
    double aux;
    String aux2;
    for(int i=1;i<n;i++){
        for(int j=1;j<n;j++){
            if( Double.parseDouble(vector[j][1])> Double.parseDouble(vector[j+1][1])){
                aux= Double.parseDouble(vector[j][1]);
                aux2=vector[j][0];
                //b[j]=b[j+1];
                vector[j][1]=vector[j+1][1];
                vector[j][0]=vector[j+1][0];
                //b[j+1]=aux;
                vector[j+1][1]=aux + "";
                vector[j+1][0]=aux2;
                proceso(vector);
            }
        }
    }
    dato[0][0]= vector[1][0];
    dato[0][1]=vector[1][1];

    dato[1][0]=vector[n][0];
    dato[1][1]=vector[n][1];
}

public void burbuja2() {
    double aux;
```

```
double aux1;
String aux2;
String aux3;
for(int i=1;i<n;i++){
    for(int j=1;j<n;j++){
        if( Double.parseDouble(vector[j][1])> Double.parseDouble(vector[j+1][1])){
            aux1= Double.parseDouble(vector[j][1]);
            aux2=vector[j][0];
            aux3=vector[j+1][1];
            //b[j]=b[j+1];
            vector[j][1]=vector[j+1][1];
            vector[j][0]=vector[j+1][0];
            //b[j+1]=aux1;
            vector[j+1][1]=aux1 + "";
            vector[j+1][0]=aux2;
            proceso(vector);
        }
    }
}
```



QuickSort

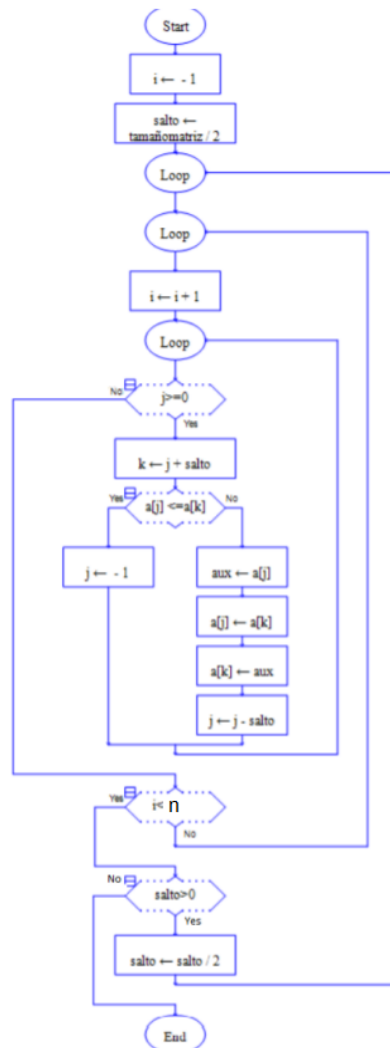
```

public void QuickSort(String a[][],int primero,int ultimo){
    int i=primero,j=ultimo;
    double pivote=Double.parseDouble(a[(primero+ultimo)/2][1]);
    double aux;
    String aux2;
    do{
        while(Double.parseDouble(a[i][1])<pivote){
            i++;
        }
        while(Double.parseDouble(a[j][1])>pivote){
            j--;
        }
        if(i<=j){
            aux=Double.parseDouble(a[i][1]) ;
            aux2=a[i][0];

            a[i][1]=a[j][1];
            a[i][0]=a[j][0];

            a[j][1]=aux+"";
            a[j][0]=aux2;
            i++;
            j--;
            proceso(a);
            ordenar(a);
            dato[0][0]= a[1][0];
            dato[0][1]=a[1][1];
            dato[1][0]=a[n][0];
            dato[1][1]=a[n][1];
        }
    }while(i<=j);
    if(primero<j){
        QuickSort(a,primero,j) ;
    }
    if(primero<i){
        QuickSort(a,i,ultimo) ;
    }
}

```



ShellSort

```

public void ShellSort(String vector[][],int n){
    int i,j,k,salto;
    double aux;
    String aux2;
    salto=n/2;
    while(salto>0){
        for( i=0;i<n;i++){
            j=i-salto;
            while(j>=1){
                k=j+salto;

                if(Double.parseDouble(vector[j][1])<=Double.parseDouble(vector[k][1])){
                    j=-1;
                }else{
                    aux=Double.parseDouble(vector[j][1]);
                    aux2=vector[j][0];

                    vector[j][1]=vector[k][1];
                    vector[j][0]=vector[k][0];

                    vector[k][1]=aux+"";
                    vector[k][0]=aux2;
                    j=j-salto;
                    proceso(vector);
                    ordenar(vector);
                    dato[0][0]= vector[1][0];
                    dato[0][1]=vector[1][1];

                    dato[1][0]=vector[n][0];
                    dato[1][1]=vector[n][1];
                }
            }
        }
    }
}

```

