

MANUAL TECNICO:

Tabla de Contenido

Introducción	1
Objetivos	1
Información destacada	2
Conocimientos previos	2
1. Requerimientos	3
2. Instalación y configuración	4
3. Diagramas de flujo y código	5

Introducción

El presente documento describe los describe la serie de pasos que realiza algunos de los métodos o funciones que conforman el programa computacional de ordenamiento de datos por medio de diagramas de flujo, como también se detalla los conocimientos previos que debe tener el lector de este manual para comprender de una mejor manera el funcionamiento de cada uno partes del código que son parte del programa.

Objetivos

Instruir el uso adecuado del programa computacional, describiendo el diseño y la lógica del programa por medio de diagramas de flujo. Describir al usuario el funcionamiento del programa para el mejor uso de él y demostrar el proceso necesario para su ejecución.

Información destacada

El manual técnico hace referencia a la información necesaria con el fin de orientar al personal en la concepción, planteamiento análisis programación e instalación del sistema. Es de notar que la redacción propia del manual técnico está orientada a personal con conocimientos en sistemas y tecnologías de información, conocimientos de programación avanzada sobre Java y diagramas de flujo, responsables del mantenimiento e instalación del programa computacional en el computador.

Conocimientos Previos

Los conocimientos mínimos que deben tener las personas que operarán las páginas y deberán utilizar este manual son:

- Conocimientos y entendimientos en diagramas de flujos
- Conocimientos en Java
- Conocimiento básico de Windows
- Conocimiento en el funcionamiento de hilos en java
- Conocimiento en la librería JFreeChart
- Conocimiento en HTML

1. Requerimientos

El sistema puede ser instalado en cualquier sistema operativo que cumpla con los siguientes requerimientos:

- Mínimo 512 MB en RAM
- Espacio en disco: 124 MB para JRE; 2 MB para Java Update
- IDE NetBeans, o compatible con Java
- Navegador web

2. Instalación y configuración

Para el proceso de instalación de esta aplicación únicamente es necesario tener instalado un IDE que sea compatible con Java para ejecutar la aplicación desde la terminal de este.

No es necesario tener alguna configuración ya que la configuración que trae por determinado el IDE es la necesaria para que el funcionamiento del programa sea posible.

3. Diagramas de Flujo y Código

Los diagramas de flujo nos ayudan a organizar de una mejor manera nuestro código y ver a futuro como realmente funcionara y es por eso por lo que son de gran importancia al momento de realizar un programa. A continuación, se muestran algunos de los diagramas que representan algunos métodos o funciones que conforman el código del programa de ordenamiento de datos.

Código de Leer Archivo

```

43     int numLinea = 0;
44     String titulos = "";
45     String valores = "";
46     try {
47         FileReader lectura = new FileReader(vista.campoRuta.getText());
48         BufferedReader archivoLeer = new BufferedReader(lectura);
49         String linealeida;
50         while ((linealeida = archivoLeer.readLine()) != null) {
51             if (numLinea == 0) {
52                 titulos += linealeida;
53                 numLinea++;
54             } else {
55                 valores += linealeida + ",";
56                 numLinea++;
57             }
58         }
59         archivoLeer.close();
60         lectura.close();
61         String[] datos = valores.split(",");
62         encabezado = titulos.split(",");
63         nombres = new String[numLinea - 1];
64         cantidades = new double[numLinea - 1];
65         int posicion = 0;
66         //////////////////////////////////////
67         for (int i = 0; i < nombres.length; i++) {
68             nombres[i] = datos[posicion];
69             posicion += 2;
70         }
71         //////////////////////////////////////
72         posicion = 0;
73         for (int i = 0; i < cantidades.length; i++) {
74             cantidades[i] = Double.parseDouble(datos[posicion + 1]);
75             posicion += 2;
76         }

```

Esta parte de código representa el funcionamiento que el programa realiza para capturar los datos que contiene el archivo que el usuario ingreso para generar una gráfica de barras en el programa.

Como se puede observar en el código, el programa lee una por una las líneas del archivo para ir obteniendo los datos de este, al finalizar la obtención de los datos del archivo se procede a crear 3 arreglos en el que uno contendrá los valores, otro contendrá los nombres y el ultimo el encabezado de cada eje de la gráfica.

Código de Generar Grafica

```
grafica.graficar(vista.campoAlgoritmo.getText(), cantidades, nombres, encabezado);  
vista.panelGrafica.removeAll();  
vista.panelGrafica.add(grafica);  
vista.panelGrafica.revalidate();  
vista.panelGrafica.repaint();
```

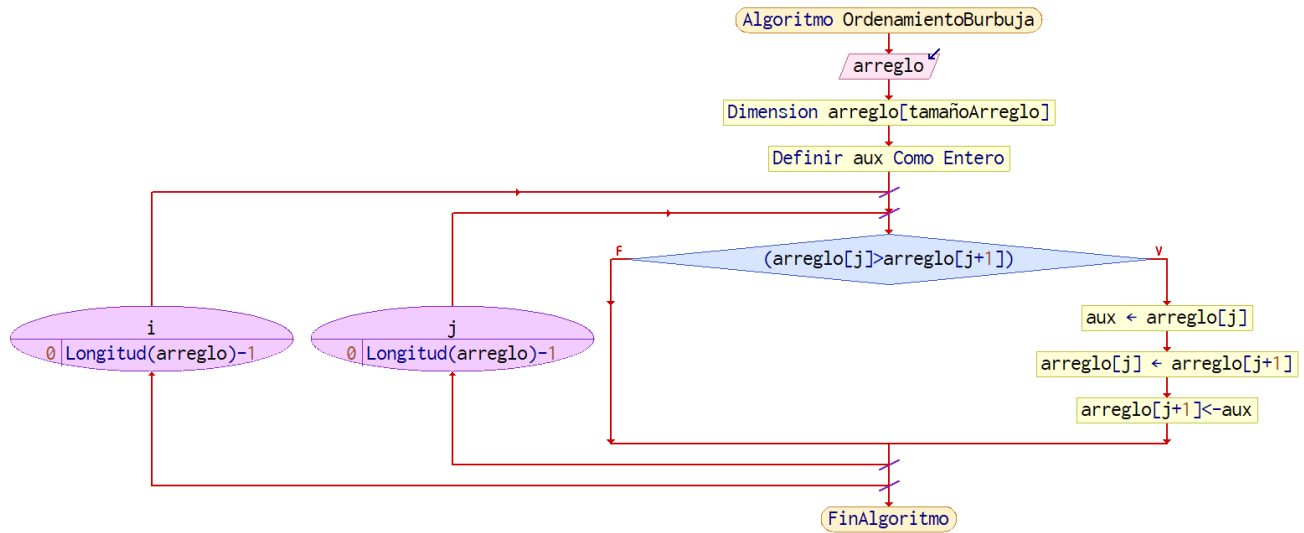
El funcionamiento de esta parte del código del programa es generar una gráfica de barras que represente los datos que el usuario ingreso por medio del archivo .csv.

Como bien se puede observar lo que realiza este código es llamar un método nombrado “grafica” por medio del objeto “grafica” el cual es una instancia de la clase “GraficaBarra”.

El método grafica recibe como parámetro el nombre del título de la gráfica, los valores, los nombres y los encabezados de cada eje.

El objeto “grafica” ahora tendrá almacenado los valores ingresados al momento de llamar la función “grafica” y el objeto ya que es un panel se agregará al panel que está contenido en la ventana principal, representando así los valores por medio de una gráfica de barras.

Diagrama de Flujo de Ordenamiento Burbuja

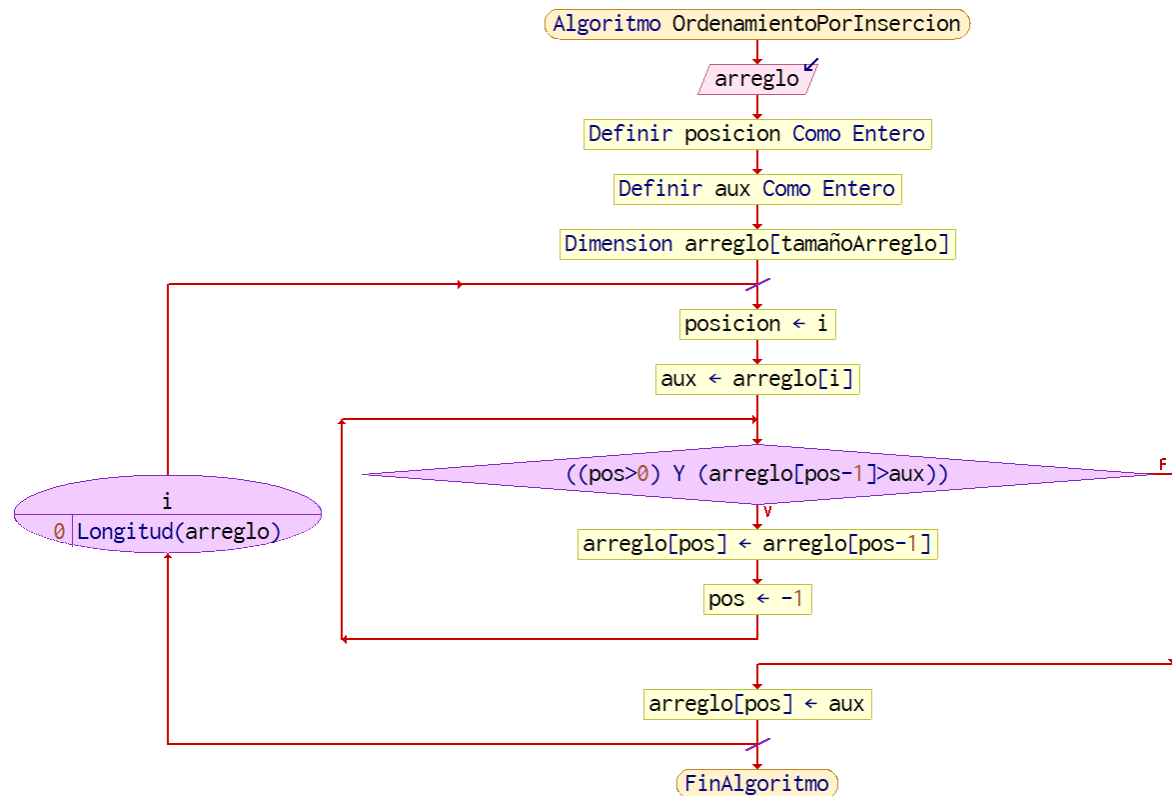


Este diagrama de flujo representa el comportamiento de la función que realiza para ir ordenando los datos de una manera llamada “ordenamiento burbuja”.

Como se puede observar la función recibe como parámetro un arreglo en el cual los datos se encuentran desordenados y el arreglo es recorrido posición por posición para ser ordenado.

En la función se crea una variable auxiliar de tipo entero, y esta sirve al momento en el que el arreglo es comparado para determinar si el primer valor es mayor que el segundo.

Diagrama de Flujo de Ordenamiento por Inserción

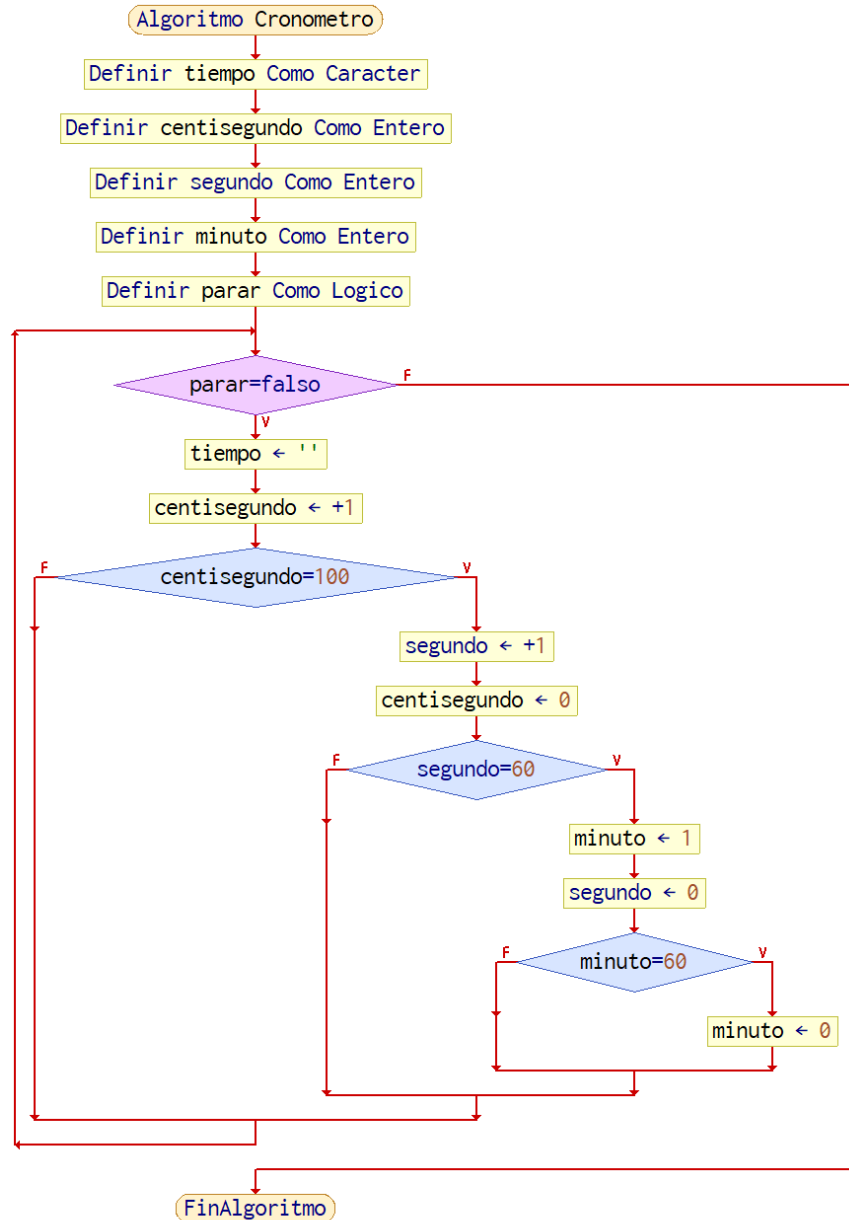


Este diagrama de flujo representa el comportamiento de la función que realiza para ir ordenando los datos de una manera llamada “ordenamiento por Inserción”.

Como se puede observar la función recibe como parámetro un arreglo en el cual los datos se encuentran desordenados y el arreglo es recorrido posición por posición para ser ordenado.

En la función se crea una variable auxiliar, y esta sirve al momento en el que el arreglo es comparado para determinar si el primer valor es mayor que el segundo.

Diagrama de Flujo de Tiempo



En este diagrama de flujo se puede observar la lógica del funcionamiento del método “tiempo” el cual representa el tiempo que se toma un algoritmo de ordenamiento en ordenar una serie de datos desordenados.

Como se puede observar en el diagrama el programa incrementa las centésimas de segundo y verifica si la centésima llega al valor de 100 y de ser así incrementa en uno el valor de los segundos luego verifica si el valor de los segundos es igual a 60 y de ser así incrementa en uno el valor de los minutos, realizando esto hasta que el valor de la variable “parar” sea igual a verdadero.