



**Universidad de San Carlos de Guatemala**

**Facultad de ingeniería**

**Escuela de Ciencias y Sistemas**

**Introducción a la programación y computación 1**

# **Manual Técnico**

**Julio Alfredo Fernández Rodríguez**

**201902416**

## Practica 2

Para ejecutar el programa haga clic en la aplicación llamada Ordenamientos.

En esta práctica se utilizaron hilos para cada ordenamiento y el timer que se activaran desde la clase principal según la elección del usuario.

Para esta práctica se emplearon las siguientes clases:

- **Principal:** Esta clase es un JFrame que es la encargada de la lectura del archivo csv y de indicarle a las demás clases que orden seguir. Para la lectura del documento csv se utilizó la clase JFileChooser que abre un cuadro de dialogo que nos permite ubicar y seleccionar de manera fácil y rápida un archivo o carpeta que se encuentre dentro del sistema de archivos de nuestro PC , guardando la ruta en un JTextField. Seguidamente se crearon grupos de RadioButtons para que se pueda marcar una sola opción a la vez. Dependiendo de los RadioButtons seleccionados se llamarán a las clases correspondientes. Tambien se utilizaron dos JLabel públicos llamados "txttiempo" y "contador" que como su nombre lo indica muestran dicha información. Tambien se creó un JPanel público para mostrar el grafico de barras desde las distintas clases.

Opciones de los RadioButtons:

Algoritmo1=método de ordenamiento burbuja

Algoritmo2=método de ordenamiento por inserción.

Algoritmo3=método de ordenamiento Shell (inserción mejorada).

- **Timer:** En esta clase se creó un variable tiempo de la clase Timer, con el formato horas:minutos:segundos:centésimas de segundo. Esta clase es una extensión de la clase Thread que se activa cuando empieza el ordenamiento sin importar que algoritmo y orden seleccione el usuario.
- **Orden\_Burbuja:** Esta clase es una extensión de la clase Thread. Aquí se realiza el ordenamiento burbuja ascendentemente. Inicialmente se creó un constructor de la clase para recibir la ruta del archivo y el título del JFreechart, posteriormente se creó un arreglo llamado "Datos" que almacenara las filas leídas del documento csv por medio de un BufferedReader, seguidamente se crearon dos arreglos llamados "nombres" y "números", para llenar estos arreglos se realizó un split con el separador "," al arreglo "Datos" y se fue almacenando su información, también se crearon otros dos arreglos llamados "nombresdesordenados" y "numerosdesordenados" que igualmente se rellenaron haciendo split al arreglo Datos, estos últimos dos arreglos sirven únicamente para rellenar la tabla

desordenada en el reporte. Seguidamente se realizó la estructura del algoritmo burbuja y se le aplicó al arreglo “números” y con el mismo iterado al arreglo nombres. Durante el ordenamiento se creó una variable del tipo entera que aumenta tras cada movimiento del ordenamiento, dicha información de la variable es enviada al JLabel “contador” de la clase principal. Al terminar el ordenamiento se detiene el hilo encargado del timer. Finalmente, en esta clase se creó un método con la clase JFreeChart para la creación del gráfico de barras que es enviado al panel encargado de mostrar el gráfico en la clase principal, este método se reimprime dentro del ciclo del ordenamiento.

- **Ins:** Esta clase es una extensión de la clase Thread. Aquí se realiza el ordenamiento inserción. Inicialmente se creó un constructor de la clase para recibir la ruta del archivo y el título del JFreechart, posteriormente se creó un arreglo llamado “Datos” que almacenara las filas leídas del documento csv por medio de un BufferedReader, seguidamente se crearon dos arreglos llamados “nombres” y “números”, para llenar estos arreglos se realizó un split con el separador “,” al arreglo “Datos” y se fue almacenando su información, también se crearon otros dos arreglos llamados “nombresdesordenados” y “numerosdesordenados” que igualmente se rellenaron haciendo split al arreglo Datos, estos últimos dos arreglos sirven únicamente para rellenar la tabla desordenada en el reporte. Seguidamente se realizó la estructura del algoritmo burbuja y se le aplicó al arreglo “números” y con el mismo iterado al arreglo nombres. Durante el ordenamiento se creó una variable del tipo entera que aumenta tras cada movimiento del ordenamiento, dicha información de la variable es enviada al JLabel “contador” de la clase principal. Al terminar el ordenamiento se detiene el hilo encargado del timer. Finalmente, en esta clase se creó un método con la clase JFreeChart para la creación del gráfico de barras que es enviado al panel encargado de mostrar el gráfico en la clase principal, este método se reimprime dentro del ciclo del ordenamiento.
- **Shell:** Esta clase es una extensión de la clase Thread. Aquí se realiza el ordenamiento inserción mejorada(Shell). Inicialmente se creó un constructor de la clase para recibir la ruta del archivo y el título del Afrechar, posteriormente se creó un arreglo llamado “Datos” que almacenara las filas leídas del documento csv por medio de un BufferedReader, seguidamente se crearon dos arreglos llamados “nombres” y “números”, para llenar estos arreglos se realizó un split con el separador “,” al arreglo “Datos” y se fue almacenando su información, también se crearon otros dos arreglos llamados “nombresdesordenados” y “numerosdesordenados” que igualmente se rellenaron haciendo split al arreglo Datos, estos últimos dos arreglos sirven únicamente para rellenar la tabla desordenada en el reporte. Seguidamente se realizó la estructura del algoritmo burbuja y se le aplicó al arreglo “números” y con el mismo iterado al arreglo nombres. Durante el ordenamiento se creó una



variable del tipo entera que aumenta tras cada movimiento del ordenamiento, dicha información de la variable es enviada al JLabel “contador” de la clase principal. Al terminar el ordenamiento se detiene el hilo encargado del timer. Finalmente, en esta clase se creó un método con la clase JFreeChart para la creación del grafico de barras que es enviado al panel encargado de mostrar el grafico en la clase principal, este método se reimprime dentro del ciclo del ordenamiento.

Para ordenar ascendentemente o descendentemente los arreglos solo se cambia el símbolo dentro del condicionar ya sea “<” o “>”.