

Introducción

En esta Práctica vemos un programa muy práctico para su utilización ya que se usaron Combo Box para la elección, además de las gráficas se muestran de la mejor manera. Además de buscar en los archivos que tenemos y que los botones tengan que hacer cada uno para que el usuario al momento de su ejecución sea de comprensión sencilla para cualquier persona que no sepa que hace el programa o que no esté familiarizada con un IDE además del .Jar es muy sencillo para el uso de los usuarios.

Requerimientos del Sistema

Requerimientos Mínimos

Sistema Operativo: Windows 10+, macOS 10.13+, Linux (64-bit)

Java Runtime Environment (JRE) Versión 8 o superior

Memoria RAM: 2 GB disponibles

Espacio en Disco: 100 MB libres

Dependencias

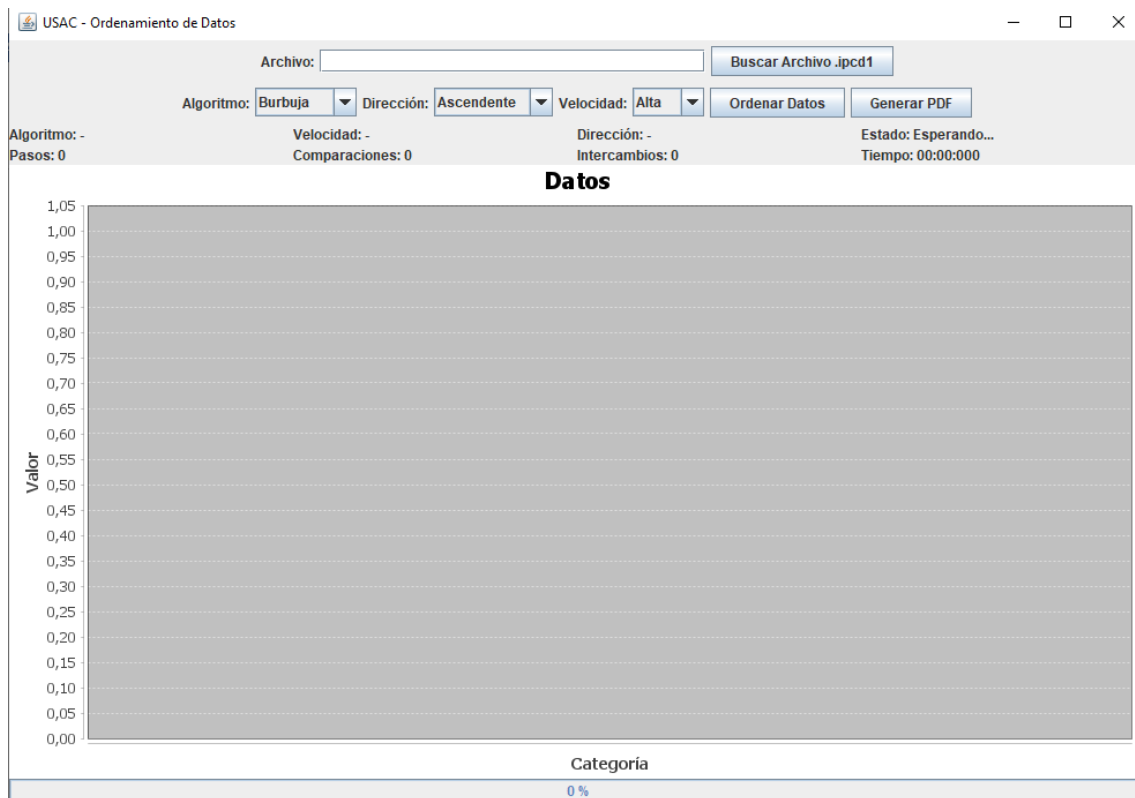
Librerías incluidas en el proyecto:

itextpdf-5.5.10 (generación de PDF).

jfreechart-1.5.0 (gráficas interactivas).

Utilización

Para empezar al iniciar el programa aparece el entorno grafico el View ya que este se inicializa desde el inicio del programa. Y es un entorno muy amigable para el usuario. Tiene que buscar el archivo CSV con la extensión .ipcd1 para que el programa funcione en caso contrario lanza un mensaje de alerta para decir que no puede usar el programa ya que es una extensión no permitida por el programa.



Configuración de Ordenamiento

Selecciona un Algoritmo:

Burbuja, Inserción, Selección, QuickSort, MergeSort, ShellSort.

Dirección:

Ascendente o Descendente

Velocidad:

Alta (100 ms), Media (500 ms), Baja (1000 ms).

En este caso ms significa milisegundo que es lo que va a tardar aproximadamente el programa en la realización del ordenamiento seleccionado por el usuario

Proceso de Ordenamiento

Haz clic en "Ordenar Datos".

Observa en tiempo real:

La gráfica se actualiza paso a paso.

Las estadísticas (pasos, comparaciones, tiempo) se muestran en el panel superior.

Generación de Reporte PDF

Tras finalizar, haz clic en "Generar PDF".

El reporte incluirá:

Portada con los datos del creador y la institución además de la fecha y hora de la creación del PDF.

Configuración usada (algoritmo, velocidad).

Datos originales y ordenados (tablas + gráficas).

Estadísticas completas.

Ejemplos Prácticos

Caso 1: Ordenamiento Didáctico

Objetivo: Enseñar el algoritmo Burbuja.

Carga un archivo con 7 elementos.

Configura:

Algoritmo: Burbuja.

Velocidad: Baja.

Observa cada intercambio en la gráfica.

Luego se da en Descendente y la vuelve a ordenar

Al darle Click a Generar PDF. Crea un PDF que tiene las especificaciones dichas con anterioridad

Solución de Problemas

Error al cargar archivo: Verifica que la extensión sea .ipcd1 y el formato coincida con el ejemplo.

La aplicación se bloquea: Usa algoritmos eficientes (QuickSort/MergeSort) para datasets grandes.

PDF no generado: Asegúrate de tener permisos de escritura en la carpeta.

Recomendaciones

Al ejecutar el programa no meter una cantidad exagerada de datos para que no se sature el programa, además de que se pueda visualizar de una manera clara y precisa todos los datos especificados en el archivo. Usar a lo máximo una cantidad de unos 5-10 datos. Además de no usar mucho tiempo en los sleep porque sino se puede hacer tedioso esperar demasiado tiempo para el ordenamiento del programa.

Conclusión

En este proyecto se aplicaron hilos para la creación de los ordenamientos además de usar `Threads.sleep` para darle tiempo al algoritmo para que se ejecute. Además de la creación del PDF y se ha utilizado todos los conocimiento en el Laboratorio. Además que emplea de manera directa otras librerías para la creación de gráficos a partir de datos ya implementados en un tipo de archivo especial y para la creación de PDF y la implementación modelo MVC.