

Avance 1 - Manual Técnico

Integrantes

- Burga Rodriguez Janluvi Gabdiel (U22202217)
- Huanca Fernández Kevin (U22244540)
- Llacsahuanga Buques Diego Alexis (U22203352)

Carrera:

Ingeniería de sistemas

Curso:

Algoritmos y Estructura de datos

Docente:

Jorge Martín Rodríguez Castro

Sección:

29924

Año:

2024

ÍNDICE

Introducción	3
Algorithms.....	4
CSV	6
Controllers	7
Utilities	8

INTRODUCCIÓN

Este manual técnico está hecho con el objetivo de documentar las clases implementadas, así como su importancia y el porqué de cada una de ellas. Cada clase desempeña un papel fundamental en el funcionamiento del programa contribuyendo de manera significativa a su operatividad y eficiencia, permitiendo así que el programa funcione correctamente. Esto permitirá que el usuario pueda tener una visión mucho más clara y completa de cómo interactúan las diferentes clases entre sí para lograr los objetivos del programa. Además, se brinda la opción de que los usuarios escojan el método de ordenamiento que deseen, permitiéndoles organizar los datos según diferentes criterios.

En resumen, este manual técnico servirá como una guía completa para aquellos que deseen comprender el funcionamiento interno del programa, facilitando a su vez su procesamiento mediante métodos de ordenamiento y búsqueda avanzados o para aquellos que necesiten realizar modificaciones en el futuro.

➤ Algorithms

● Search

BinarySearch: Este método implementa el algoritmo de búsqueda binaria para encontrar un elemento en un arreglo, devolviendo el índice donde se encuentra el elemento o -1 si no se encuentra.

InterpolationSearcher: Este método implementa el algoritmo de búsqueda por interpolación para encontrar un elemento en un arreglo, devolviendo, al igual que en la clase anterior el índice donde se encuentra el elemento o -1 si no se encuentra.

SearchAlgorithm: Esta interfaz define un contrato para las clases que implementan algoritmos de búsqueda, especificando cómo deben ser los métodos de búsqueda para poder ser utilizados de manera general en diferentes contextos.

SequentialSearch: Este método implementa un algoritmo de búsqueda secuencial, que busca el elemento deseado uno por uno en el arreglo. Si encuentra el elemento, devuelve su índice; de lo contrario, devuelve -1.

● Sort

BogoSort : Algoritmo de ordenamiento que funciona mezclando aleatoriamente los elementos del arreglo hasta que estén ordenados.

BubbleSort : Esta clase proporciona una implementación del algoritmo de ordenamiento de burbuja que puede ser utilizada para ordenar arreglos de elementos comparables en orden ascendente.

HeapSort: Esta clase proporciona un método estático llamado “sort”, que toma un arreglo de elementos comparables y lo ordena utilizando el algoritmo de ordenamiento HeapSort, que es eficiente y tiene una complejidad de tiempo de ejecución de “ $O(n \log n)$ ” (en el peor de los casos)

MergeSort: Esta clase ordena un arreglo de elementos comparables en orden ascendente. Este algoritmo tiene una complejidad de tiempo de ejecución de “ $O(n \log n)$ ” (en el peor de los casos), permitiendo ordenar arreglos grandes.

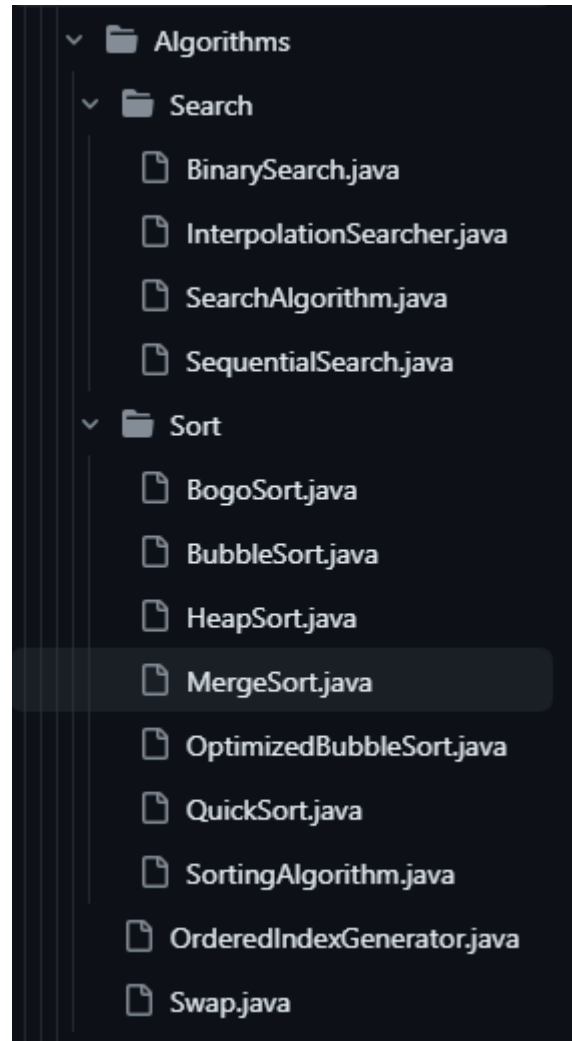
OptimizedBubbleSort: Esta clase proporciona una implementación del algoritmo de ordenamiento de burbuja optimizado, capaz de ordenar arreglos de elementos comparables de una manera eficiente.

QuickSort : Esta clase proporciona una implementación del algoritmo QuickSort , que ordena arreglos de elementos de una manera eficiente.

SortingAlgorithm: Esta interfaz proporciona un estándar para las implementaciones de algoritmos de ordenamiento al definir un método “sort” que toma un arreglo de elementos comparables y lo ordena.

OrderedIndexGenerator: Esta clase proporciona funcionalidad para trabajar con arreglos y sus índices, particularmente para generar índices basados en un arreglo original y otro ordenado.

Swap: Esta clase proporciona funcionalidad utilizada para intercambiar elementos en arreglos, ya sea en genéricos o específicos para tipos de datos enteros y enteros primitivos.



➤ CSV

CSVHeader: Esta clase proporciona una estructura para representar el encabezado de una columna en un archivo CSV. Además, permite almacenar el nombre y el tipo de datos del encabezado.

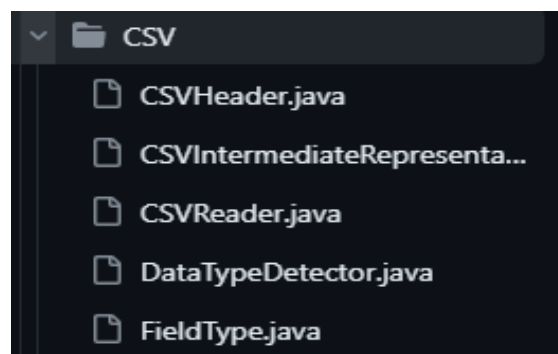
CSVIntermediateRepresentation:

Esta clase proporciona una forma de representar y manipular los datos de un archivo CSV antes de ser procesados o mostrados en la interfaz, permitiendo establecer los encabezados, los datos de las columnas y obtener información sobre ellos.

CSVReader: Esta clase permite leer un archivo CSV, incluida la generación de encabezados y la verificación de tipos de datos, y devuelve una representación intermedia de los datos leídos en la clase “CSVIntermediateRepresentation”.

DataTypeDetector: Esta clase brinda un conjunto de métodos estáticos para detectar el tipo de datos de un valor dado.

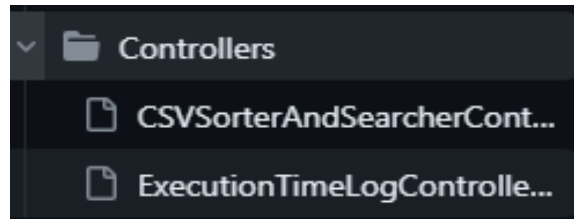
FieldType: Este enum proporciona una forma conveniente de representar los diferentes tipos de datos que pueden ser detectados durante el análisis de valores en un archivo. Ayuda a mantener el código claro y legible al manipular diferentes tipos de datos de manera consistente.



➤ Controllers

CSVSorterAndSearcherController: Esta clase actúa como el intermediario entre la lógica de la aplicación y la interfaz de usuario, nos ayuda gestionando la interacción del usuario y coordinando las acciones

ExecutionTimeLogController: Este controlador se encarga de inicializar y gestionar una tabla de registro de tiempos de ejecución, permitiendo la adición de nuevas filas y la limpieza de la tabla cuando sea necesario.



➤ Utilities

ExportModes: Esta clase proporciona una forma conveniente de exportar datos en diferentes formatos sin tener que repetir el código para cada tipo de formato

TableV: Esta clase encapsula la lógica para manejar los datos y la visualización en una tabla JavaFX, proporcionando métodos para poder establecer encabezados y elementos y limpiar la tabla.

Utilities: Esta clase encapsula métodos para leer y escribir registros de archivos recientes, mostrar alertas al usuario y guardar datos en archivos.

CSVSorterAndSearcher: Esta clase inicia la aplicación JavaFX y carga la interfaz de usuario desde un archivo FXML llamado MainWindow.fxml. Además, se asegura de que la aplicación se cierre correctamente cuando se cierra la ventana.

