

Facultad: Ingeniería en Sistemas

Universidad: Da Vinci

Curso: Estructura De Datos

Tema: Implementación de Algoritmos de Ordenamiento y Búsqueda

Nombre: Ezry Adilia Maribeth Leal Franco

Carnet: 202200202

“PROYECTO ESTRUCTURA DE DATOS: IMPLEMENTACIÓN DE ALGORITMOS DE ORDENAMIENTO Y BÚSQUEDA”

INTRODUCCIÓN

Este proyecto desarrolla una plataforma interactiva en consola, el cual permitirá a los usuarios visualizar un menú con múltiples opciones para trabajar con algoritmos de ordenamiento y búsqueda en java. Este esta diseñado para proporcionar una experiencia practica en el uso de estructuras de datos, permitiendo al usuario explorar y comprender diferentes métodos de ordenamiento y búsqueda. Teniendo en cuenta lo importante y fundamental que son los algoritmos de ordenamiento y búsqueda.

EXPLICACIÓN DEL PROYECTO

1. ESTRUCTURA DE ARCHIVOS:

Se encontrarán varios archivos como lo son **ProyectoOrdenamiento.java** el cual contiene la clase principal para gestionar el menú y poder correr la consola. **CargaDatosCSV.java** este es una clase para cargar los datos que necesitamos ingresar, **datoscsv** en este ingresamos los numero que deseamos ordenar. **BubbleSort.java** contiene el algoritmo de BubbleSort para ordenar. Y el **EnhancedBubbleSort** la implementación para el algoritmo y cada archivo como su nombre lo indica contiene el algoritmo y la implementación para la consola logre correr bien.

ESTRUCTURA DEL MENÚ

1. INICIO DEL PROGRAMA:

Se utiliza el método `main` el cual permite llamar a mostrar información, este solicitará el nombre del estudiante y luego de ser ingresado mostrará la información de catedrático, universidad y nombre de quien creó la consola, así como las tareas pendientes.

El método `mostrarmenu` muestra las varias opciones con la que el usuario puede interactuar, teniendo en cuenta que primero se debe registrar el `datoscsv` el listado a trabajar.

Al momento de ingresar en el archivo `datoscsv` el listado a trabajar, en la consola en **ProyectoOrdenamiento** al iniciar el programa nos solicitará la ruta del archivo csv en donde la consola llama a `cargardatoscsv` el cual para leer el archivo y cargar los datos en `listadatos` esto para que la información se pueda cargar exitosamente y se muestre en la consola.

El menú es interactivo por lo que se utiliza un bucle `do-while` y un `switch` para así poder manejar varias opciones con el usuario. El menú es muy intuitivo al momento de utilizarlo y permite al usuario seleccionar la opción que desee por medio de el número correspondiente. Esto luego de haber cargado los datos del archivo `datoscsv`. Cada opción en el menú llamará a una función específica para la ejecución correspondiente.

Primero la consola solicitará el nombre del estudiante, y dar enter automáticamente se imprime los datos de la Universidad, catedrático, etc.

```

Ingrese su nombre: Adilia Franco

=====
Universidad Da Vinci de Guatemala
Estructura de Datos
Docente Ing. Brandon Chitay
Realizado por: Ezry Leal
Carnet: 202200202

Nombre del estudiante: Adilia Franco

1. Información del Desarrollador
2. Menú Principal

Tareas pendientes:
[PENDIENTE] 3. Carga de Datos desde un CSV
[PENDIENTE] 4. Algoritmo de Ordenamiento - Bubble Sort
[PENDIENTE] 5. Algoritmo de Ordenamiento - Enhanced Bubble Sort
[PENDIENTE] 6. Algoritmo de Ordenamiento - Quick Sort
[PENDIENTE] 7. Algoritmo de Ordenamiento - Selection Sort
[PENDIENTE] 8. Algoritmo de Ordenamiento - Merge Sort
[PENDIENTE] 9. Algoritmo de Búsqueda - Binary Search
[PENDIENTE] 10. Presentación Final en YouTube
=====
Presione Enter para continuar...

```

Y luego enter ya nos da acceso al menú principal para iniciar la interacción

```

===== MENÚ PRINCIPAL =====
1. Cargar datos desde un archivo CSV
2. Ordenar datos usando Bubble Sort
3. Ordenar datos usando Enhanced Bubble Sort
4. Ordenar datos usando Quick Sort
5. Ordenar datos usando Selection Sort
6. Ordenar datos usando Merge Sort
7. Buscar un número con Binary Search
8. Salir
Ingrese una opción: 1
Ingrese la ruta del archivo CSV: datoscsv
[EXITO] Datos cargados correctamente.
[DATOS] Datos cargados: [23, 56, 12, 75, 43, 9, 83]

===== MENÚ PRINCIPAL =====
1. Cargar datos desde un archivo CSV
2. Ordenar datos usando Bubble Sort
3. Ordenar datos usando Enhanced Bubble Sort
4. Ordenar datos usando Quick Sort
5. Ordenar datos usando Selection Sort
6. Ordenar datos usando Merge Sort
7. Buscar un número con Binary Search
8. Salir
Ingrese una opción: █

```

2. ALGORITMO DE ORDENAMIENTO:

BubbleSort: Iniciamos con BubbleSort es un algoritmo simple de ordenamiento, compara pares adyacentes y los intercambia si estos están en el orden incorrecto, esto se repite una y varias veces hasta que la lista esta ordenada. El archivo con el que trabajamos cuenta con Bubblesort_opcionBubblesort esta llama al listado de datos que tenemos, mientras que opcionBubblesort verificara que los datos ya se encuentren cargados para así poder llamar al ordenador así ordena los datos utilizando el BubbleSort, permitiendo que se muestre en la consola de manera ya ordenada. el enhaced BubbleSort permite al algoritmo una mejor optimización, esto para que, si en una pasada no hay cambios, significa que la lista esta ordenada y se pueda detener antes de implementar todas las interacciones necesarias reduciendo su tiempo y permitiendo la optimización deseada.

QuickSort: la opción QuickSort verificara que los datos ya se encuentren cargados, luego los ordena utilizando el método QuickSort, este algoritmo es eficiente porque sigue el paradigma de divide y vencerás, este se basa en el pivote, de dividir la lista en dos partes una menor y una mayor al pivote y así puede ordenar recursivamente cada sub lista y reduce el tiempo de ordenamiento.

SelectionSort: Esta utiliza la opcionSelectionSort para verificar los datos cargados y así luego ordenarlos usando selectionsort. Este algoritmo funciona seleccionando repetidamente el elemento mínimo de la lista no ordenada, luego los coloca en la posición correspondiente de la lista, presentando una lista ordenada.

MergeSort: Este utiliza la opcionMergeSort para verificar que los datos se encuentren cargados y luego los inicia a ordenar para presentarlos en la consola ya

ordenados. Este algoritmo de ordenamiento es muy eficiente, tal como QuickSort; divide la lista en dos mitades, las ordena recursivamente y luego las inicia a combinar de una manera ya ordenada para presentarla a la consola.

BinarySearch: Luego de que opcionBinarySearch verifico la carga de los datos, el programa solicitara al usuario que numero desea buscar, esto llamara a buscar para encontrar el numero deseado, así se presentara una explicación de la ubicación del numero deseado. Este algoritmo es de búsqueda, funciona solo en listas ya ordenadas, buscando el valor en la lista dividiendo el rango de búsqueda a la mitad repetidamente hasta encontrar el valor que se buscar o determinar sino se encuentra presente.

Cargar datos CSV: Se utiliza la clase CargarDatosCSV que lee el archivo línea por línea y convierte cada línea en un numero entero. Estos se almacenan en una lista para ser utilizados por el algoritmo ya sea de ordenamiento y búsqueda.

```
4 public class CargarDatosCSV {
5     public static List<Integer> cargarDesdeCSV(String archivo) {
6         List<Integer> lista = new ArrayList<>();
7
8         try (BufferedReader br = new BufferedReader(new FileReader(archivo))) {
9             String linea;
10            while ((linea = br.readLine()) != null) {
11                String[] numeros = linea.split(regex:",");
12                for (String num : numeros) {
13                    try {
14                        lista.add(Integer.parseInt(num.trim())); // Convertir y agregar a la lista
15                    } catch (NumberFormatException e) {
16                        System.out.println("[ERROR] Error al procesar el número: " + num);
17                    }
18                }
19            }
20        } catch (IOException e) {
21            System.out.println("[ERROR] Error al leer el archivo: " + e.getMessage());
22        }
23
24        return lista;
25    }
}
```

RECOMENDACIONES Y MEJORAS

Interfaz de Usuario: mejorar la interfaz del usuario en la consola, incluso desarrollar una interfaz gráfica para hacer más interactivo el programa.

validación de Datos: Agregar una validación donde se muestren errores para asegurar que los datos ingresados por el usuario son correctos.

CONCLUSION

Se ha logrado desarrollar una plataforma interactiva la cual permite al usuario explorar diferentes tipos de ordenamientos y búsqueda, esto no solo refuerza los conceptos teóricos con los que ya se cuentan, sino que también me a ayudado a aplicar la practica de como puedo utilizar esto en problemas reales.

LINK

Video Youtube

<https://youtu.be/uWQPXLqr-mk>

GitHub

<https://github.com/EzryLeal/Proyecto-Estructura-de-Datos-Implementaci-n-de-Algoritmos-de-Ordenamiento-y-B-squeda.git>