



Instituto Tecnológico de Costa Rica

Proyecto II

CE - 1103

Algoritmos y Estructuras de Datos

Grupo 03

Prof. Leonardo Andres Araya Martínez

Fabiana Zamora Carvajal 2022000142

Esteban Daniel Secaida Vargas 2019042589

Segundo semestre 2022

Tabla de contenidos

I. Breve descripción del problema	3
II. Diagrama de clases	4
III. Descripción de las estructuras desarrolladas	5
IV. Descripción de los algoritmos desarrollados	6
V. Descripción de los algoritmos desarrollados	7
Link a AzureDevops	8
Recursos	8
AVL tree explained and implemented	8
Árboles Binarios de Búsqueda AVL, Implementación	8

I. Breve descripción del problema

En este segundo proyecto se tratará de realizar una aplicación de escritorio que permite buscar texto en archivos .pdf, .txt, y en .docx. El usuario podrá hacer búsqueda de texto mediante la aplicación. Al igual que se mostrarán los resultados de la búsqueda en todos los documentos y muestra al usuario extractos del documento en el que aparece la palabra. Se implementará una función en donde el usuario podrá ordenar los resultados por Nombre del archivo, fecha de creación y tamaño-cantidad de palabras por medio de diferentes algoritmos desarrollados.

Se espera poder implementar la arquitectura cliente-servidor con una comunicación mediante sockets. En donde el usuario tendrá una sola interfaz en la cuál podrá interactuar con el sistema. El servidor por otro lado será el encargado de la gestión de las estructuras de datos, los algoritmos y el procesamiento de las solicitudes de los usuarios. Esperamos que al finalizar este proyecto tendremos claro los conceptos de árboles implementados, algoritmos de ordenamiento, y demás métodos de servidor-cliente.

II. Diagrama de clases

Diagrama de clases completo

III. Descripción de las estructuras desarrolladas

Algunas de las estructuras desarrolladas vistas en la realización de este proyecto son los árboles binarios de búsqueda y los árboles AVL.

Los árboles binarios de búsqueda son empleados en la representación de expresiones aritméticas. Se suele visualizar de tal manera en que la raíz es el operador y los subárboles sean los operandos. En un árbol binario el valor de cada nodo es mayor que los valores de su subárbol izquierdo y es menor que los valores de su subárbol derecho. Los árboles AVL tienen una dinámica parecida a los binarios de búsqueda y se le añade una condición de equilibrio. En este caso, la indización de la biblioteca se trabajará en árboles binarios de búsqueda y en AVL.

IV. Descripción de los algoritmos desarrollados

Algunos de los algoritmos desarrollados son el quick sort, bubble sort y el radix sort.

Al ser el quicksort un algoritmo que ordena elementos en un tiempo proporcional, se basa en el divide y vencerás. Es un algoritmo de ordenamiento rápido porque va subdividiendo el array en arrays más pequeños con el pivote para luego escoger que tipo de búsqueda haría. Es por eso que la implementación del quicksort servirá para que el usuario pueda ordenar los resultados por nombre del archivo.

El bubblesort al ser un algoritmo que prioriza que la partición ordenada vaya creciendo, revisa cada elemento de las listas y va intercambiandolos de posición por si estuvieran en el orden equivocado. En el proyecto el bubble sort es implementado en la parte de la fecha de creación.

Por último en el radix sort los elementos son ordenados al agrupar primeramente los dígitos individuales del mismo valor posicional. Sigue ordenando los elementos de acuerdo a su orden ya sea creciente o decreciente. Típicamente el radix sort es frecuentado en máquinas secuenciales de acceso aleatorio, donde los registros están codificados por múltiples campos. En nuestra aplicación de Text Finder este algoritmo de ordenamiento será implementado en el tamaño-cantidad de palabras.

V. Problemas encontrados (Issues Github)

[Bugs GitHub Cliente](#)

[<> Code](#) [Issues 1](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

Label issues and pull requests for new contributors

Dismiss

Now, GitHub will help potential first-time contributors [discover issues](#) labeled with [good first issue](#)

Filters [is:issue is:open](#)

Labels 9 Milestones 0

New issue

☐ 1 Open ☒ 0 Closed

Author [Label](#) [Projects](#) [Milestones](#) [Assignee](#) [Sort](#)

☐ [Borrar](#)

#1 opened 42 minutes ago by fabieds23

[Bugs Github Server](#)

[<> Code](#) [Issues 2](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

Label issues and pull requests for new contributors

Dismiss

Now, GitHub will help potential first-time contributors [discover issues](#) labeled with [good first issue](#)

Filters [is:issue is:open](#)

Labels 9 Milestones 0

New issue

☐ 2 Open ☒ 0 Closed

Author [Label](#) [Projects](#) [Milestones](#) [Assignee](#) [Sort](#)

☐ [Problema Bubble sort y Radix Sort](#)

#2 opened 44 minutes ago by fabieds23

☐ [Eliminación de archivos de texto](#)

#1 opened 44 minutes ago by fabieds23

Link a AzureDevops

[TextFinder AzureDevops](#)

Recursos

[AVL tree explained and implemented](#)

[Árboles Binarios de Búsqueda AVL, Implementación](#)

[AVL Tree \(with Java Code\)](#)

[JavaFX and Java Sockets - Client Server Messenger](#)