Universidad Peruana de Ciencias Aplicadas
UPC

PROYECTO
FILEEXP

CURSO DE ALGORITMO Y ESTRUCTURA DE DATOS
Carrera de Ingeniería de software
Carrera de Ciencias de la computación

Sección SW32

Alumno:
Luis Ticona Miranda u201714230
Carolina Villegas u201711783
Sergio Banzan 2016

Monterrico, Noviembre 2019

Índice

1. Introducción

En el presente trabajo se implementará el proyecto FILEEXP el cual busca reducir el tiempo que tarda la búsqueda de archivos, gracias a que cuentan con un orden específico. Este puede ser usado tanto para un sistema operativo window o linux. El programa contará con las funciones como organizar, indexar, buscar por(nombre,extensión,tamaño,fecha) y filtrar por condiciones tales como(empieza por,terminación y por tamaño del archivo). Para la elaboración de este proyecto se utilizó un árboles avl que tiene un costo de búsqueda promedio de O(log(n) por lo que es más eficaz que otro tipo de estructura de datos.

2. Objetivo del estudiante

- Crear un explorador de archivos eficiente mediante el uso del contenido del curso de Algoritmo y Estructura de base de datos.
- Explicar las funcionalidades e implementación de los algoritmos para el desarrollo del FILEEXP Así mismo, mostrar el cumplimiento de los requisitos a través del lenguaje C++.

4. Presentación

El programa tiene por objetivo simplificar la búsqueda de archivos. Se puede buscar por nombre,tamaño,extensión y fecha. Además cuando se hace una búsqueda por nombre el usuario tiene la opción de filtrar por las letras iniciales,terminaciones y contenido. También, en el caso de hacer una busqueda por tamaño se puede filtrar por peso de archivo ya sea mayor que ,menor que o igual a lo requerido por el usuario. Finalmente, FileEXP, usa el método de ordenamiento mergesort para poder ordenar los elementos de la lista de manera ascendente o descendente en orden alfabético.

5. Marco Conceptual

- Programación Orientada a Objetos: Es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, abstracción, polimorfismo y encapsulamiento

- Ordenamiento por árboles: Es un algoritmo de ordenamiento, el cual se basa en ir construyendo poco a poco el árbol binario introduciendo cada

uno de los elementos, los cuales quedarán ya ordenados. Después, se obtiene la lista de los elementos ordenados.

- Archivos: Un archivo o fichero informático es un conjunto de bits que son almacenados en un dispositivo. Un archivo es identificado por un nombre y la descripción de la carpeta o directorio que lo contiene. A los archivos informáticos se les llama así porque son los equivalentes digitales de los archivos escritos en expedientes, tarjetas, libretas, papel o microfichas del entorno de oficina tradicional.

6 capítulo 3: Diseño del proyecto

Para la implementación del proyecto usamos herramientas como github que nos permite el guardado del mismo en un repositorio donde se puede actualizar de manera grupal a través de los commits y pulls, de igual modo se puede establecer issues los cuales nos ayudan a repartirnos el trabajo y ordenar las prioridades para una mejor implementación del proyecto. Junto a esta herramienta se utilizó el entorno de desarrollo visual studio

Diseño del producto

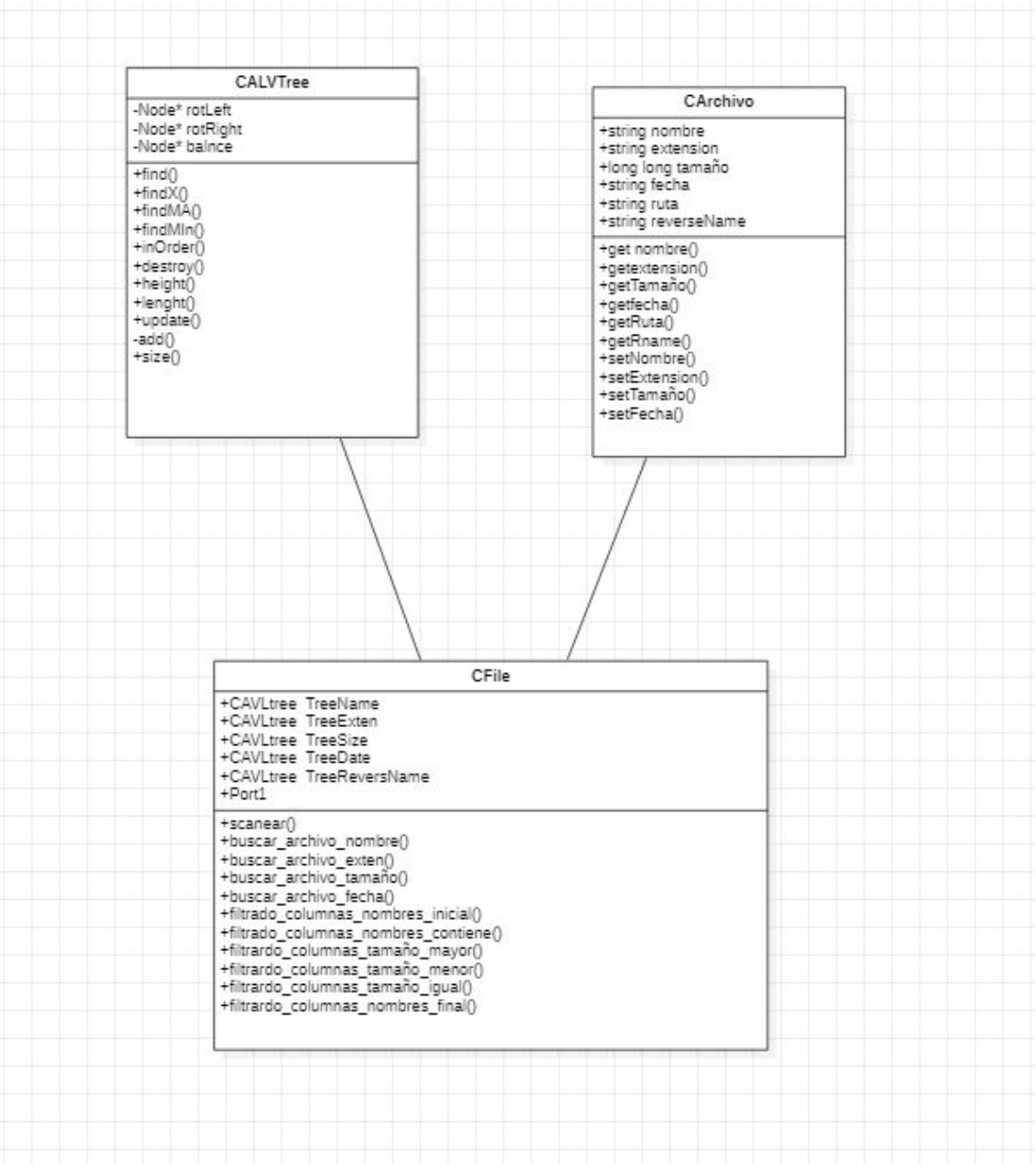
El programa a realizar busca solucionar problemas en un contexto real en la salud pública por ejemplo la gran cantidad de Files que se guardan en un hospital hace complicada la búsqueda de archivos para determinado paciente por lo que genera grandes colas y un un gran tiempo de espera. Sin embargo nuestro proyecto busca agilizar ese proceso al tener una búsqueda de archivos más ordenada y rápida, ya que este software cuenta con diferentes tipos de búsqueda como por ejemplo…()de de todos estos datos y que se puedan ordenar datos de los usuarios de manera que se acelere el proceso para hallar el archivo específico solucionando las grandes colas que se generan por la ineficiencia de este proceso.

b) Requisitos funcionales y no funcionales:

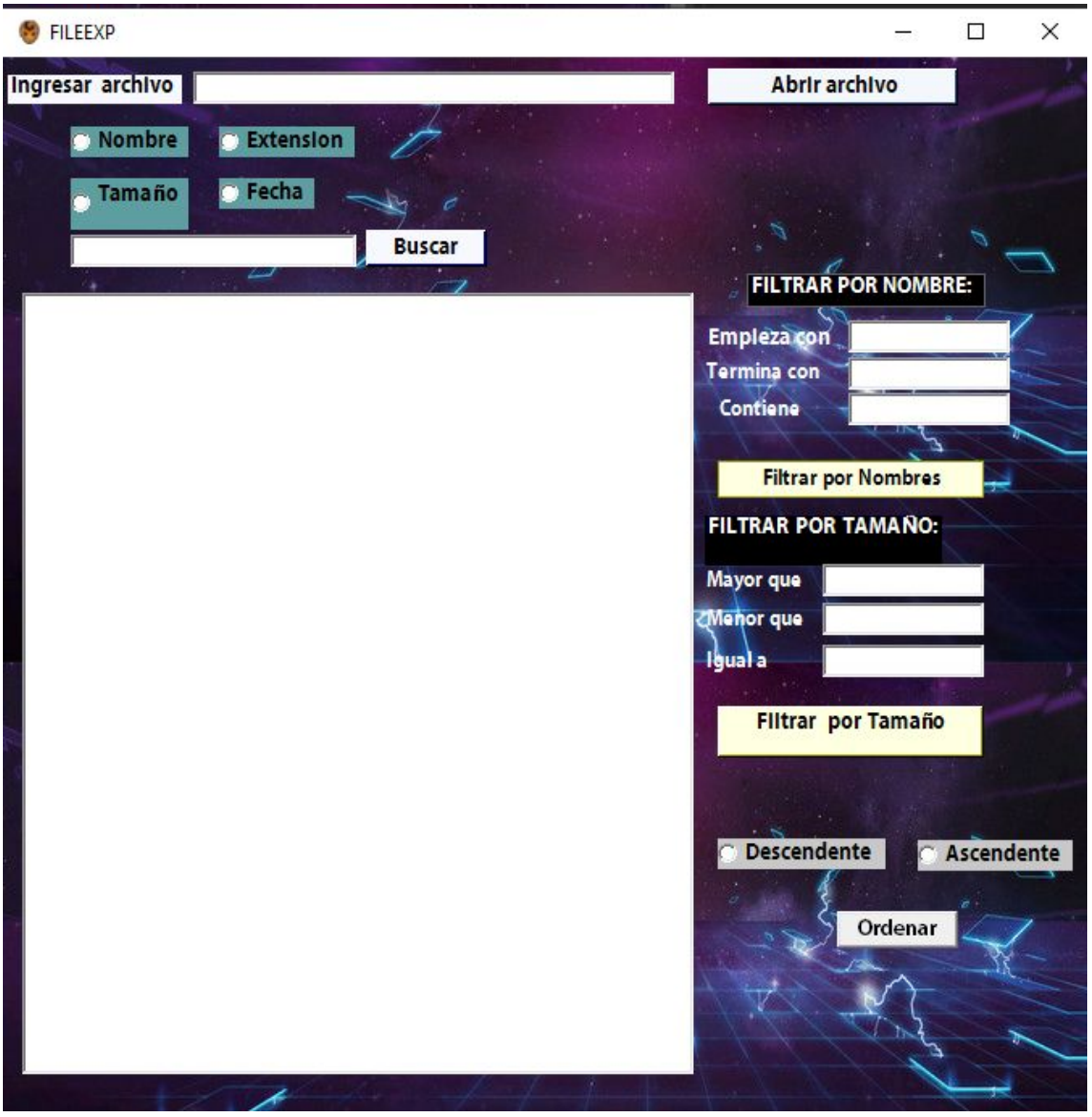
Requerimientos	Descripción del requerimiento
RF1	Se quiere una operación de selección para buscar los

	datos según el campo ingresado
RF2	Se requiere un criterio de búsqueda para empezar.
RF5	El programa debe filtrar para una columna definida, para las operaciones Mayor, Menor, Igual a, Inicia con, Finaliza con, Está contenido en, No está contenido en.
RNF1	Debe construir haciendo uso de Programación Orientada a Objetos
RNF2	lenguaje a utilizar c++
RNF3	La interfaz del usuario es en formulario

c) Elaboración de diagrama de clases



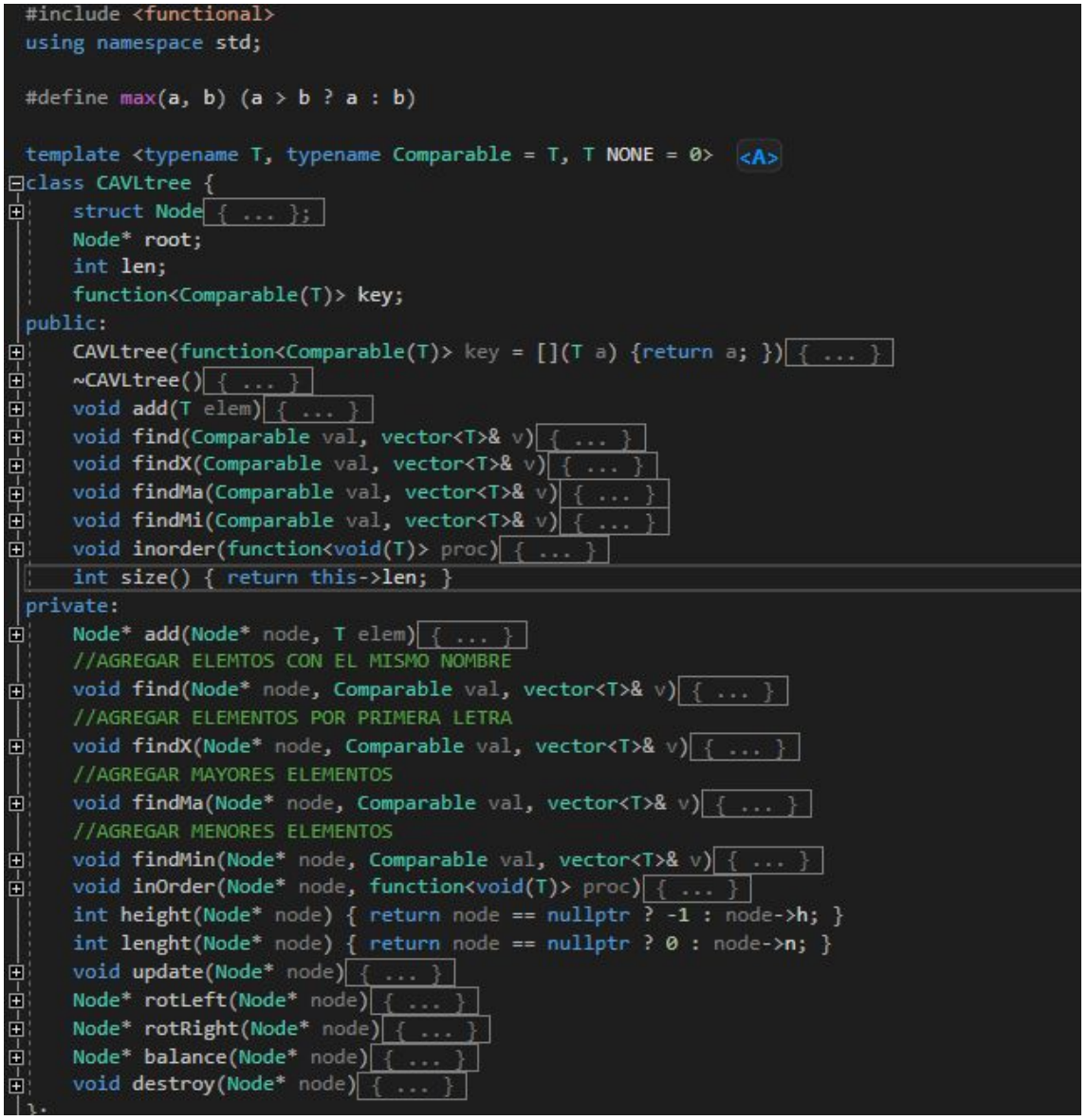
d) Diseño de interfaz de usuario


FILEEXP
Ingresar archivo
Abrir archivo
Nombre
Extension
Tamaño
Fecha
Buscar
FILTRAR POR NOMBRE:
Empieza con
Termina con
Contiene
Filtrar por Nombres
FILTRAR POR TAMAÑO:
Mayor que
Menor que
Igual a
Filtrar por Tamaño
Descendente
Ascendente
Ordenar

f) Selección de estructura de datos

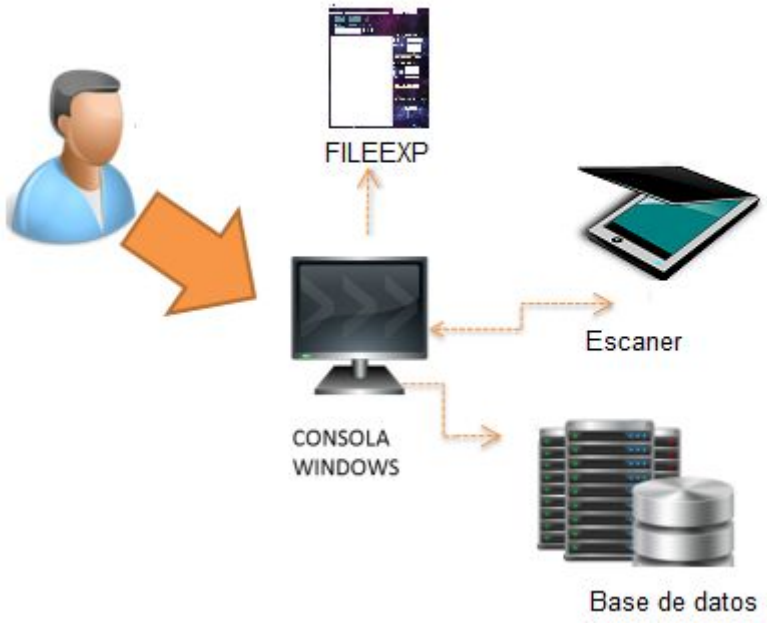
Implementamos un Árbol AVL que es un árbol binario de búsqueda pero este va estar equilibrando al momento de insertar si es necesario, no utilizamos un árbol binario de búsqueda porque su complejidad en el peor de los casos es O(n), y el árbol AVL es O(log n) en el peor de los casos y así se mantiene, nos da una mayor rapidez para acceder a los datos en los nodos en el menor tiempo posible.

I



8. Capítulo 5: Diseño del proceso de desarrollo de la solución

a. Diseño de la arquitectura del software

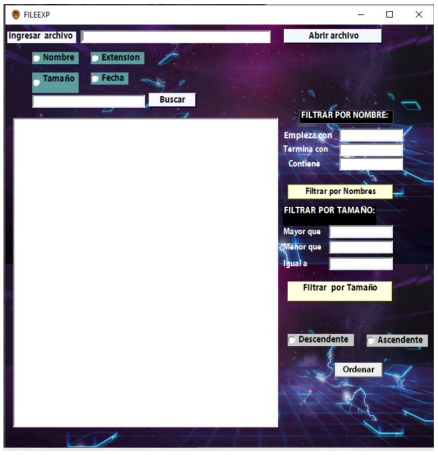


b. Implementación de las funcionalidades del software.

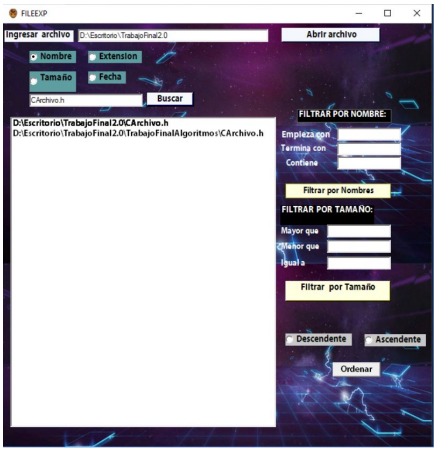
Funcionalidades	Responsable
Buscar por nombre	Sergio Bazan
Buscar por Tamaño	Sergio Bazan
Buscar por Fecha	Sergio Bazan
Buscar por extensión	Carolina Villegas
Filtrar por contenido	Carolina Villegas
filtrar por peso mayor o menor	Luis Ticona
Ordenar ascendente y descendente	Luis Ticona
Filtrar por letras iniciales y finales	Carolina Villegas
Filtrar por peso igual a	Luis Ticona

Diseño interfaz	Luis Ticona
Escaneado	Sergio Bazan
Validación de datos	Carolina Villegas

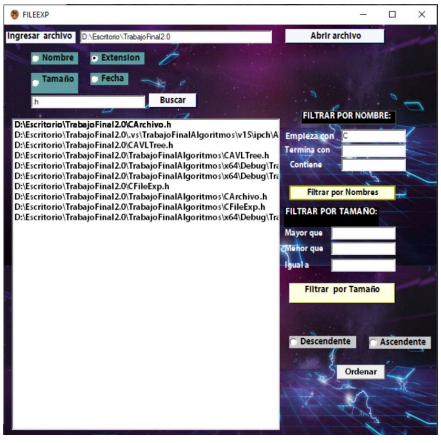
c. Pruebas del software



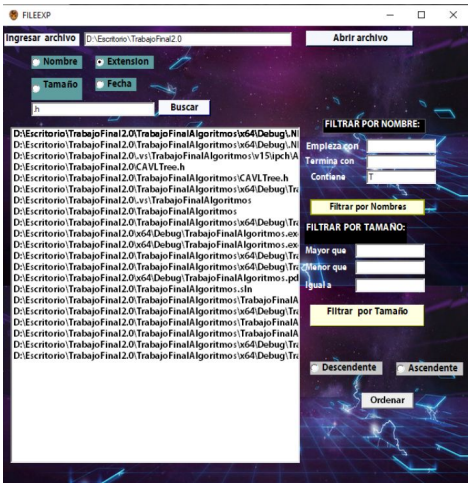
Buscar por nombre:



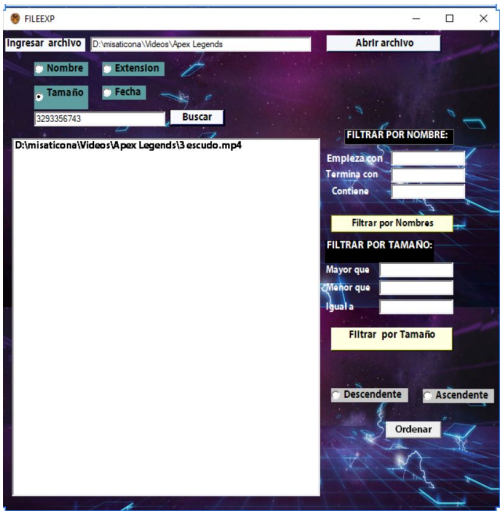
Buscar por extensión



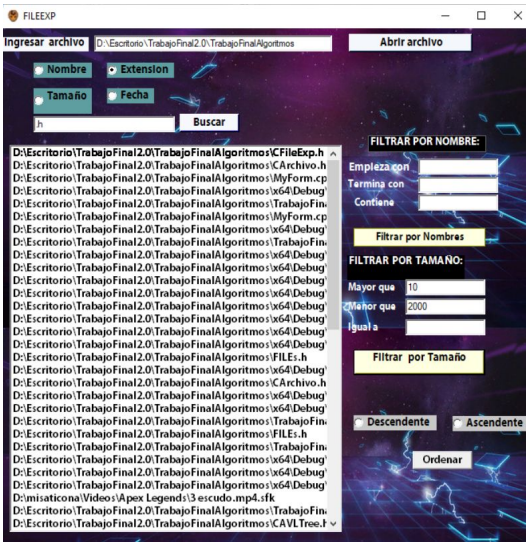
Filtrar por contenido:



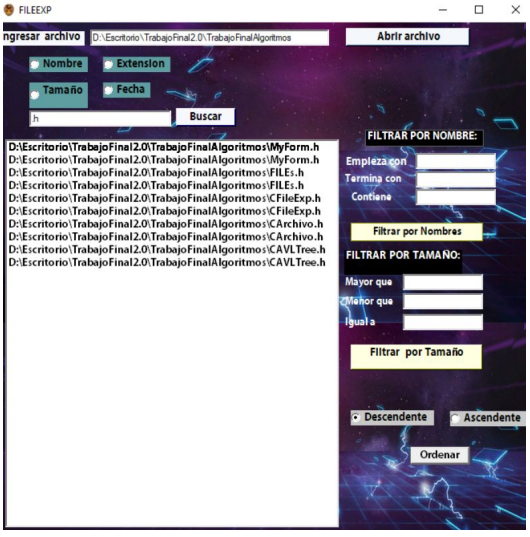
Buscar por Tamaño



Filtrar por tamaños Mayor, menor o igual



Ordenamientos Ascendente y descendente



9. Conclusiones

- El concepto de Archivos es muy importante para escribir o modificar el contenido de cualquier archivo,

también que la lectura y escritura de otros tipos de archivos son diferentes. En este caso la lectura de un archivo .csv utilizamos getline() y otros tipos de como stringstream que el otro tipo de string pero que este nos facilita leer cada carácter de los datos del archivo.

- Con un Árbol AVL podemos acceder más rápido a un dato que un árbol binario de búsqueda, ya que el Árbol AVL mantiene su complejidad en O (log n) en cambio un árbol binario de búsqueda en el peor de los casos es O(n) lo que no sería óptimo para la solución de este problema.

- Crear un FILEEXP con todas sus funciones, nos ayuda a mejorar nuestro conocimiento en el manejo del uso de la memoria mediante la implementación de estructuras de datos como el Árbol AVL que este por lógica utiliza punteros, ya que un FILEEXPse utiliza mucho en bases de datos, el tener un buen conocimiento en el manejo de memoria permite que todas las funciones del FILEEXP ya se importar, filtrar o buscar sean los más rápidos y óptimos posibles,

10. Bibliografía

Sebastián Gurin .2004. Árboles AVL[PDFFILE].http://es.tldp.org/Tutoriales/doc-programacion-arboles-avl/avl-trees.pdf

Exploración del sistema de archivos.Msdn.microsoft.com.https://msdn.microsoft.com/es-es/library/dn986850.aspx

11. Anexos (ppt, video, otros)

https://prezi.com/p/cxl-gkqeoroe/tf-algoritmos/

https://www.youtube.com/watch?v=Vp1t45gJJqs