

JUEVES 9 DE NOVIEMBRE DEL 2019

FILEEX

INFORME DE TRABAJO FINAL

Luciano Lopez Duran, Edwar Zevallos Rojas

UPC: UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS

Contenido

[Introducción 2](#_Toc25346567)

[Objetivo del estudiante 2](#_Toc25346568)

[Capítulo 1: Presentación 3](#_Toc25346569)

[Capítulo 2: Marco conceptual 3](#_Toc25346570)

[Capítulo 3: Diseño del proyecto 4](#_Toc25346571)

[● Diseño de un plan de proyecto de desarrollo 4](#_Toc25346572)

[Capítulo 4: Diseño del producto 5](#_Toc25346573)

[● Definición de las necesidades 5](#_Toc25346574)

[● Definición de requisitos funcionales y no funcionales 6](#_Toc25346575)

[o Requisitos Funcionales 6](#_Toc25346576)

[o Requisitos No funcionales 6](#_Toc25346577)

[● Elaboración de diagrama de clases 8](#_Toc25346578)

[● Diseño de la interfaz de usuario 9](#_Toc25346579)

[● Diseño de tipos de datos abstractos 11](#_Toc25346580)

[● Selección de estructuras de datos 12](#_Toc25346581)

[Capítulo 5: Diseño del proceso de desarrollo de la solución 13](#_Toc25346582)

[● Diseño de la arquitectura del software 13](#_Toc25346583)

[● Implementación de las funcionalidades del software. 13](#_Toc25346584)

[● Pruebas del software 14](#_Toc25346585)

[Conclusiones 19](#_Toc25346586)

[Bibliografía 20](#_Toc25346587)

[Anexos 20](#_Toc25346588)

## Introducción

Durante la presentación del proyecto se definida los objetivos del alumno con respecto al proyecto realizado con la finalidad de mostrar la utilidad de lo aprendido durante el curso de Algoritmo de Estructura y Datos.

Se realizará una presentación sobre el proyector en el cual se mostrará las herramientas usada para poder cumplir el requerido en el curso.

## Objetivo del estudiante

Los objetivos del curso que debe tener en cuenta los alumnos son:

* El estudiante debe reconocer el enunciado antes de programar
* El estudiante debe conocer los usos de las herramientas que usará durante el proyecto que se realizada
* La implementación de nodos.
* El uso correcto y adecuado de templates.
* La manera correcta de implementar los árboles binarios, ya sea en un arbolAVL, Binary Heap, Binary Max.
* Investigar e implementar de manera correcta la biblioteca System::IO, la cual permitirá usar Directory, File y Path, esenciales para determinar los nombres, años, dia, mes entro otros atributos de los archivos a escanear.
* Implementar nodos en los árboles binarios, estos nodos deberán contener información de los archivos escaneados de la carpeta indicada por el usuario.
* Usar las herramientas de GitHub, repositorios, bash y administrador de proyectos para el desarrollo del trabajo.
* Hacer un uso correcto de los filtradores al momento de mostrar en la pantalla.
* Motivar al estudiante a realizar investigaciones dependiendo con respecto a su carrera

## Capítulo 1: Presentación

En el presente ciclo 2019-2 en el curso de Algoritmos y Estructura de datos, se propone el siguiente proyecto con la intención de que el estudiante desarrolle y amplíe sus conocimientos sobre los árboles binarios. El proyecto involucra el conocimiento teórico y la implementación en código de los árboles AVL para el almacenamiento de información.

Durante el desarrollo del proyecto el alumno tendrá que investigar e implementar árboles AVL y las bibliotecas necesarias para su ejecución de manera organizada apoyado en las herramientas que proporciona GitHub, como los repositorios y el administrador de proyectos.

Se realizará un diseño de solución con las asignaciones de recursos, milestones y issues con la finalidad de realizar un plan de solución en la ingeniería.

Posteriormente, se hará un diseño de producto con las definiciones en el impacto de salud pública, seguridad, bienestar, cultura, sociales, ambientales y económicos. Además, se definida los requisitos funcionales y no funciones. También, mostrará el uso de diagramas de clases usados en el proyecto, la interfaz de usuario, diseño de datos abstractos y selección de estructura de datos.

Luego se mostrará el diseño de arquitectura del software, implementación del software y la prueba de compilación sobre el proyecto realizado.

Finalmente, se llegará a una conclusión en el cual el alumno dará las opiniones sobre la utilidad del trabajo y como este le ayuda al estudiante a poder dominar las diversas maneras de lo estudiado durante el ciclo.

## Capítulo 2: Marco conceptual

Arboles Binarios:

En ciencias de la computación, un árbol binario es una estructura de datos en la cual cada nodo puede tener un hijo izquierdo y un hijo derecho. No pueden tener más de dos hijos (de ahí el nombre "binario"). Si algún hijo tiene como referencia a null, es decir que no almacena ningún dato, entonces este es llamado un nodo externo. En el caso contrario el hijo es llamado un nodo interno. Usos comunes de los árboles binarios son los árboles binarios de búsqueda, los montículos binarios y Codificación de Huffman.

**Arboles AVL**

Un árbol AVL es un tipo especial de árbol binario ideado por los matemáticos rusos Adelson-Velskii y Landis. Fue el primer árbol de búsqueda binario auto-balanceable que se ideó.

El árbol AVL toma su nombre de las iniciales de los apellidos de sus inventores, Georgii Adelson-Velskii y Yevgeniy Landis. Lo dieron a conocer en la publicación de un artículo en 1962, «An algorithm for the organization of information» («Un algoritmo para la organización de la información»).

Los árboles AVL están siempre equilibrados de tal modo que, para todos los nodos, la altura de la rama izquierda no difiere en más de una unidad de la altura de la rama derecha o viceversa. Gracias a esta forma de equilibrio (o balanceo), la complejidad de una búsqueda en uno de estos árboles se mantiene siempre en orden de complejidad O(log n). El factor de equilibrio puede ser almacenado directamente en cada nodo o ser computado a partir de las alturas de los subárboles.

Para conseguir esta propiedad de equilibrio, la inserción y el borrado de los nodos se ha de realizar de una forma especial. Si al realizar una operación de inserción o borrado se rompe la condición de equilibrio, hay que realizar una serie de rotaciones de los nodos.

## Capítulo 3: Diseño del proyecto

### Diseño de un plan de proyecto de desarrollo

**Hito: En este hito se tuvo pensado crear un plan para poder cumplir lo requerido en el corto plazo que nos queda en el curso:**

El trabajo tiene como finalidad hacer la creación de un explorador de archivos. Con la finalidad, de hacer uso de lo estudiado en la clase de Estructura de Algoritmo y Datos los cuales son el uso de nodos y árboles, ya sea binario heap o binario search. Se tiene planeado hacer una repartición para cumplir con el trabajo asignado. La cual está estructurada de la siguiente manera, con el cual tenemos una idea a programar y con el tiempo establecido:

**Durante la semana 11 y 12 se hará los siguiente**

* **Se ira avanzando la estructura de informe lo siguiente**

1. Introducción (Lopez)

2. Índice (Zevallos)

3. Objetivo del Estudiante (Student Outcome) (Zevallos)

4. Capítulo 1: Presentación (Zevallos)

5. Capítulo 2: Marco conceptual (Lopez)

6. Capítulo 3: Diseño del proyecto (Lopez)

a) Diseño de un plan de proyecto de desarrollo de soluciones en ingeniería con las restricciones establecidas considerando asignación de recursos, milestones, issues.

* **Se avanzará el proyecto “FILEEXP”**
  + Le creara un primer uso con listas para ver si se almacena el enunciado (Zevallos)
  + La creación de un Nodo (Lopez)
  + Se creará la interfaz del proyector (Lopez)
  + Se hará uso de bibliotecas IO para poder encontrar los archivos de la carpeta a explorar (Lopez)
  + Se creará una clase almacenadora la cual guarda todos los datos que pide el enunciado (Zevallos)
  + Se hará uso de una Lista para ver si está funcionando de almacenamiento está funcionando correctamente (Lopez/Zevallos)

**Durante la semana 13 y 14**

* **Se tiene planeado completar el informe**

**Semana 13**

7. Capítulo 4: Diseño del producto (Lopez)

a. Definición de las necesidades específicas considerando el impacto en salud pública, seguridad, bienestar y factores globales, culturales, sociales, ambientales y económicos.

b. Definición de requisitos funcionales y no funcionales

c. Elaboración de diagrama de clases

d. Diseño de la interfaz de usuario

e. Diseño de tipos de datos abstractos

f. Selección de estructuras de datos

**Semana 14**

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

a. Diseño de la arquitectura del software

b. Implementación de las funcionalidades del software.

c. Pruebas del software

9. Conclusiones 4 V1.0 /5 (Zevallos)

10. Bibliografía 11. Anexos (ppt, video, otros) (Lopez / Zevallos)

* **Proyecto “FILEEXP”**
  + Se tiene pensado crear un árbol AVL para que devuelva los datos de forma ordenada (Zevallos)
  + Implementado en junto al formulario (Zevallos)
  + Finalmente, el proyecto finalizado durante la semana 13 a 14

## Capítulo 4: Diseño del producto

### Definición de las necesidades

Tras los múltiples intentos de reformas, no cabe duda que existe una deuda que se tiene con la sociedad en materia de salud pública. Se requiere un cambio estructural en el que el estado aborda la salud y los medios con los que nos brinda. Hasta ahora solo se han logrado un conjunto de esfuerzos aislados que no logran una mejora tangible.

La complejidad y dificultad de la tarea es muy alta. Para ello se debe dividir en tareas de menor complejidad que tengan logros tangibles y medibles.

Unas de las primeras tareas a solucionar en la calidad de la atención al público. Por ello desde el curso Algoritmos y Estructuras de datos proponemos el uso de FILEEXP como software para la atención al público en los establecimientos de salud pública. Con el fin de buscar archivos, historiales e información sobre los pacientes, médicos, y servicios de una forma rápida y eficiente. Se demuestra este deficit de atencion, ya que en un reporte periodístico de la Gestión se puede apreciar que aun con las historias clínicas digitales existe una demora de 16 días hábiles para la atención (2018, Gestion) lo cual demuestra un mal manejo en el sector salud.

En el sector económico, de determinar que existe una deficiencia en la suma de dinero y este muchas veces es usado para otros fines de lucro y no por el bien de la sociedad. Por ello, con el uso del FILEEXP se podrá determinar los costos de cada proyecto con el fin de tener un control más establecido y no generar pérdidas o sobre costos.

### Definición de requisitos funcionales y no funcionales

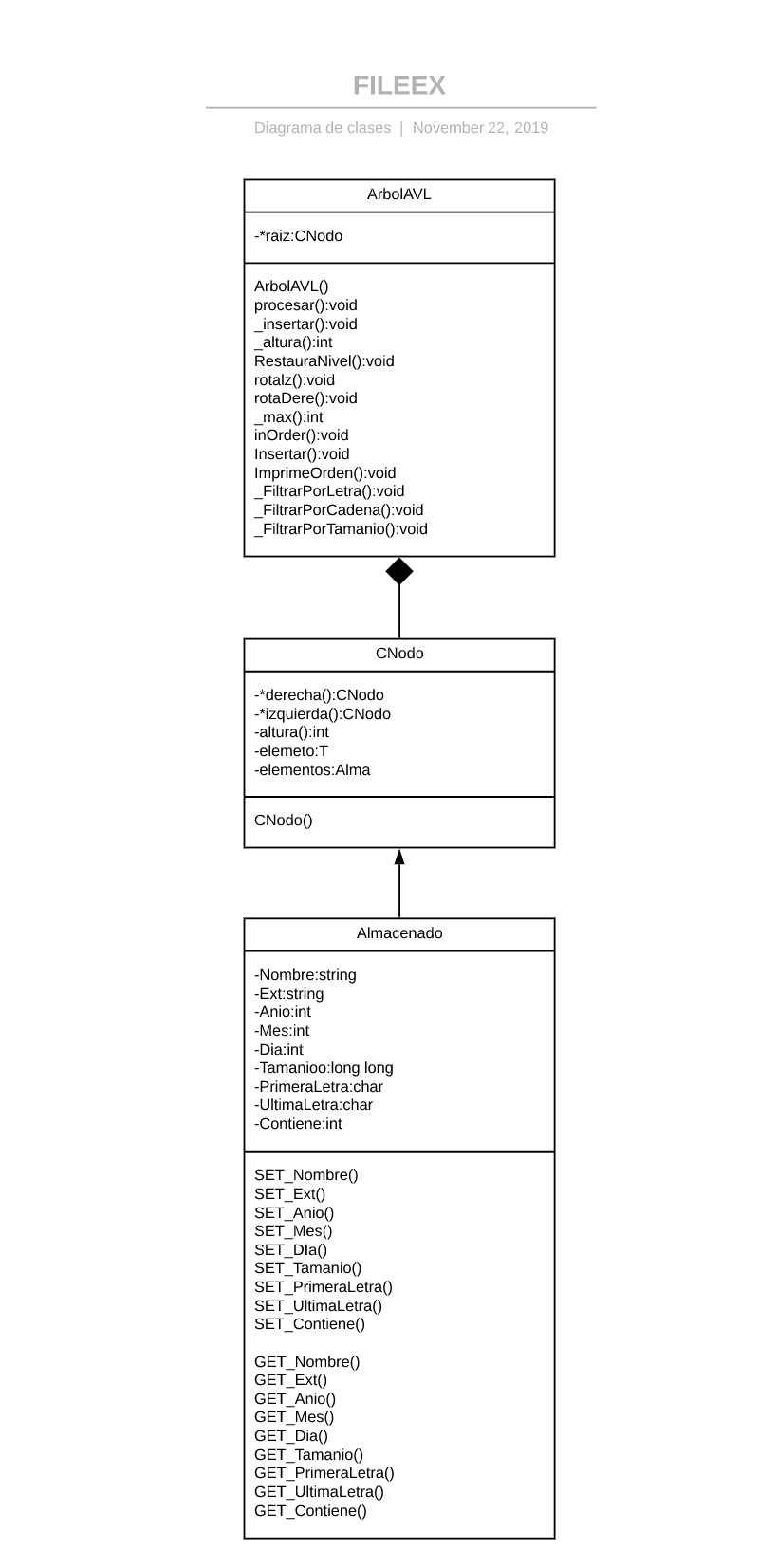
### Requisitos Funcionales

* + - El software debe mostrar los archivos de la carpeta seleccionada.
    - El durante la ejecución del programa el usuario debe designar una carpeta establecidas donde no exista ningún archivo protegido.
    - El software debe catalogar los archivos por Nombre, Tamaño, Extensión e Iniciales del nombre del archivo.
    - Es un software que puede ordenar los diversos datos de forma ascendente y descendente. Este software ordena dependiendo el peso, primera letra, última letra y cantidad de caracteres de la carpeta.

### Requisitos No funcionales

* + - El software debe ser capaz de procesar y responder la búsqueda del operador en 3 segundo o menos, esto se medirá por medio de herramientas de Software Testing
    - El software debe ser capaz de funcionar en equipos de pocas capacidades o computadoras de gama baja.
    - El software debe ser compatible con los sistemas operativos Windows XP y superiores
    - El software debe desarrollarse aplicando patrones de diseño.
    - No Funcionales
    - El software debe ser capaz de procesar y responder la búsqueda del operador en 3 segundo o menos, esto se medirá por medio de herramientas de Software Testing
    - El software debe ser capaz de funcionar en equipos de pocas capacidades o computadoras de gama baja.
    - El software debe ser compatible con los sistemas operativos windows XP y superiores
    - El software debe desarrollarse aplicando patrones de diseño
    - El tiempo de aprendizaje del operador debe ser menor a 4 horas.
    - Debe contar con manual de uso breve y estructurado o un módulo de ayuda.
    - El software debe tener un diseño entendible con el fin de garantizar la adecuada visualización en múltiples computadoras.

### Elaboración de diagrama de clases

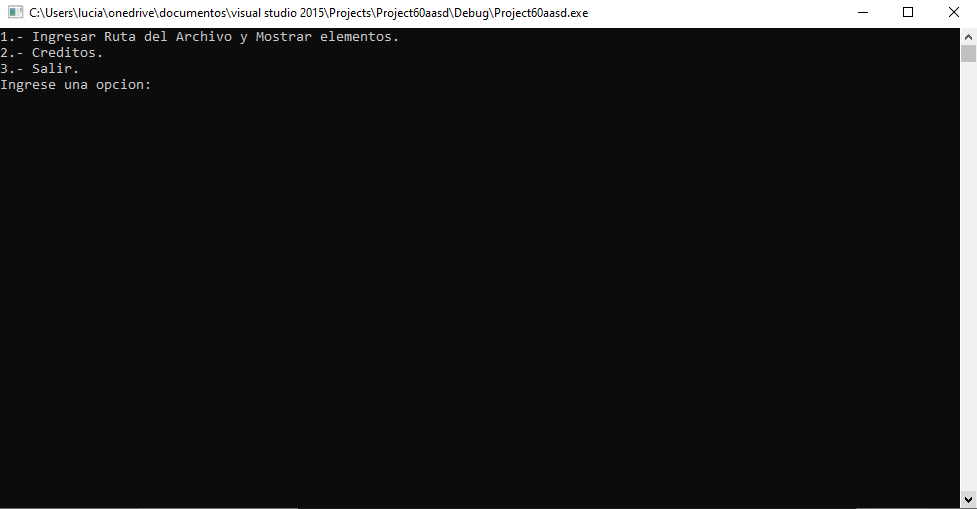


### Diseño de la interfaz de usuario

El programa se desarrollará en consola por lo que la interfaz deberá responder a inputs numéricos mediante el comando cin.

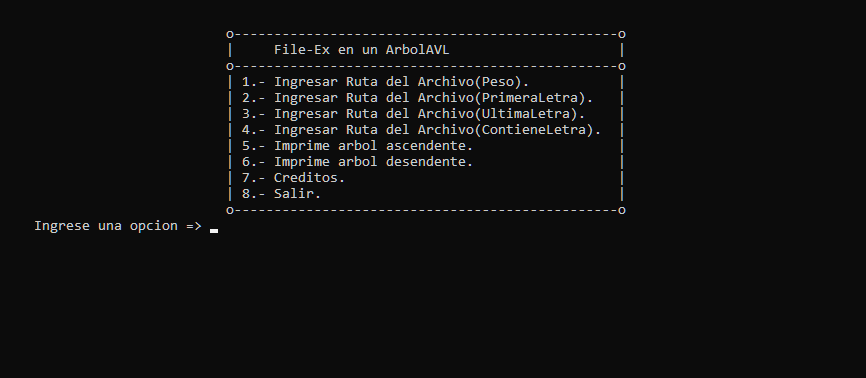
Los primeros diseños que diseñamos para el programa fueron:



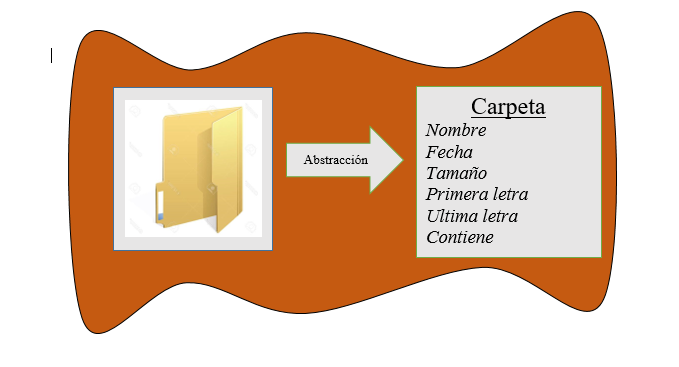


Cuando nos reunimos determinamos que nuestra interfaz de inicio era muy simple y sin forma. Por lo cual nosotros decidimos crear una con una mejor presentación para el usuario. Por consiguiente, conseguimos lo siguiente:





### Diseño de tipos de datos abstractos



### Selección de estructuras de datos

Las estructuras de datos que implementamos en nuestro código fueron:

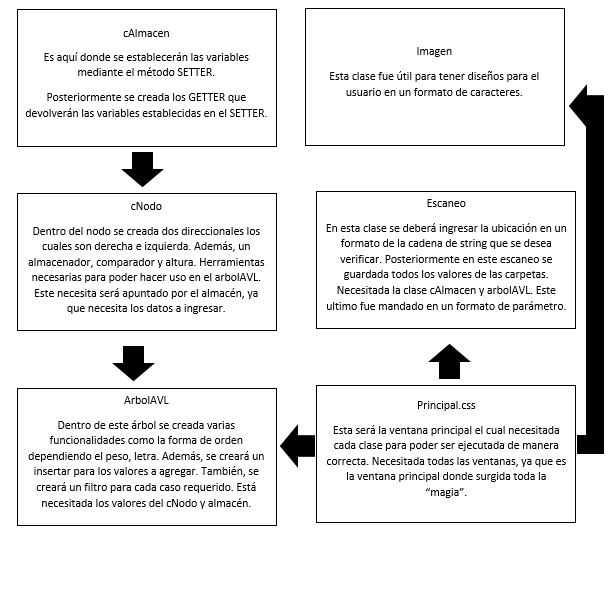
Clases:

* Imagenes: Esta estructura de dato nos fue útil, ya que no ayudar a disminuir el código en el main. Su funcionalidad de esta clase es crear un comentario con relleno de caracteres para que el programa tenga un buen diseño al momento de la interfaz.
* Escaneo: En esta clase utilizamos las librerías del IO donde nos ayudó con detectar las variables que tiene cada carpeta a la que escaneara. Sin embargo, no puede escanear archivo protegidos si es que no está activado como modo administrador.
* cAlmacenamiento: En esta variable se guarda de forma temporal los valores obtenidos por la librería IO que a través de un for each guardada los datos en un árbol con el insertar.
* cNodo: Esta clase es donde se guarda cada puntero que serán usados en el árbol determinado. En nuestro caso, hemos puesto dos tipos de template. El primer template consiste en los datos que se guardan al momento de introducir una variable, clase o asignación. Por otra parte, el segundo template se encarga de hacer la comparativa en el árbol con la finalidad de tener un arbolAVL de forma ordenada.
* ArbolAVL: En esta clase se guardan de manera ordenada dependiendo la segunda variable del template. Además, en este árbol se guardarán todos los datos. Además, creamos 4 tipos de void en la consola la cual serán nuestros filtros el cual verificada si un archivo es mayor, menor, igual al que el usuario ingreso.

## Capítulo 5: Diseño del proceso de desarrollo de la solución

### Diseño de la arquitectura del software

### 



### Implementación de las funcionalidades del software.

La funcionalidad del software consiste en poder leer un determinado archivo. Al momento de recibir la dirección a leer del archivo este detectara las variables como se especifica en el diseño de datos abstractos donde serán insertados a través de un nodo y guardados. Posteriormente el usuario para poder hallar una carpeta podrá escoger entre los 6 tipos de filtros que existen los cuales son:

Filtro por primera letra. En este caso se realiza un recorrido por el árbol para determinar si en el nombre de la variable existe la letra.

Filtro por última letra. Lo mismo que en el caso anterior solo que comparada la última letra

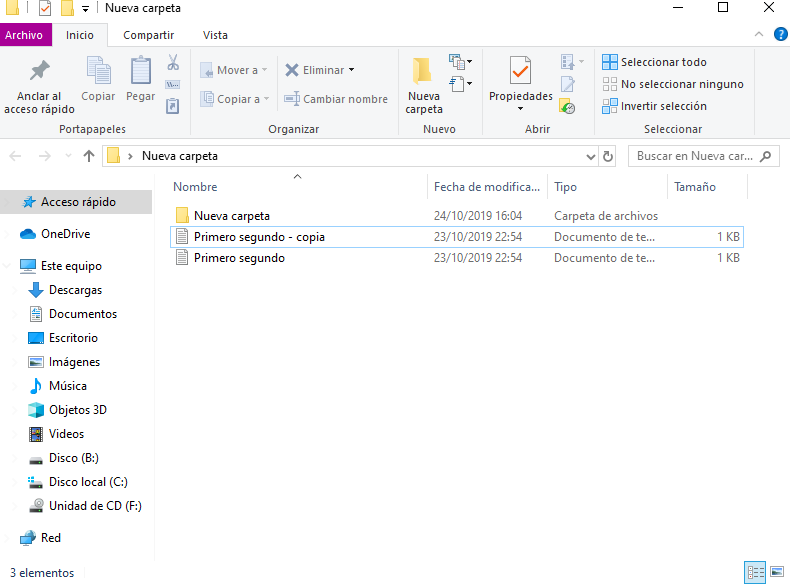
Filtro por contenido. En este caso usamos una cadena para que pueda comparar si existe un valor igual dentro del árbol.

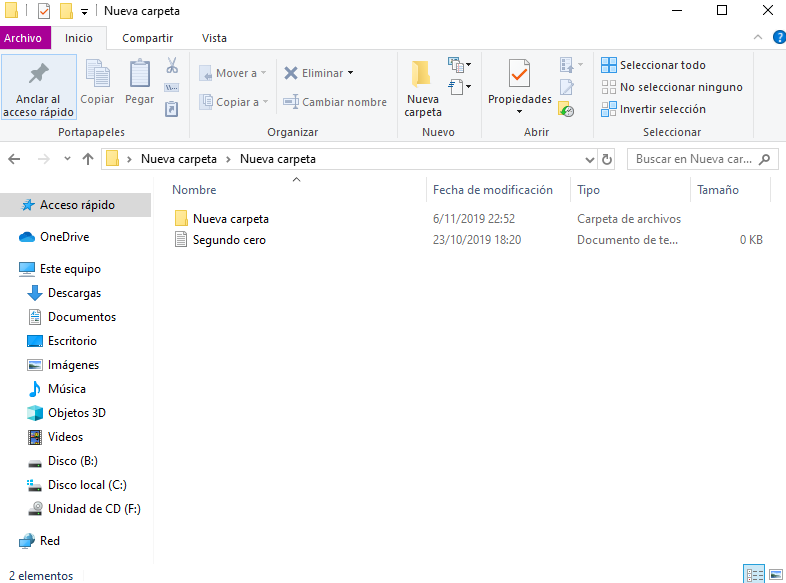
Filtro por tamaño menor. El programa recibido un valor del usuario y este debe devolver los valores menores a este.

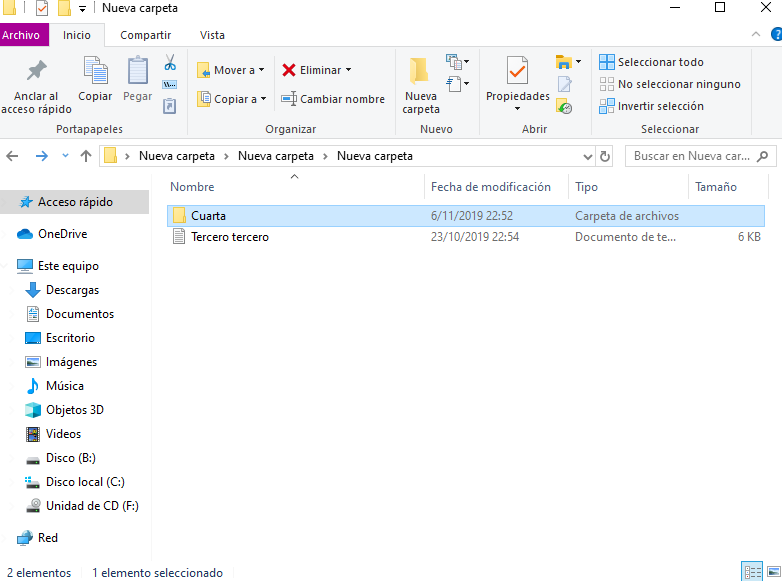
Filtro por tamaño mayor. El usuario al ingresar los valores el programa debe devolver el peso mayor al que el usuario ingreso.

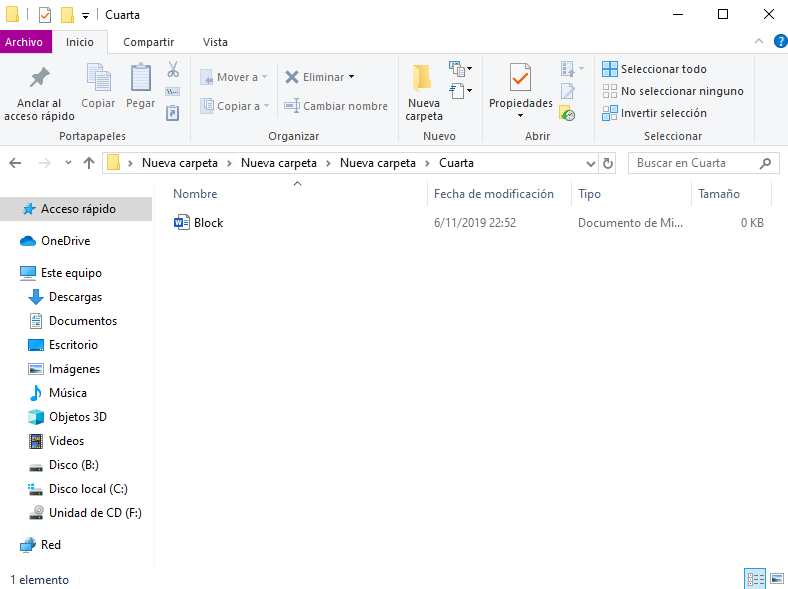
Filtro por igual. El programa recibe la variable del usuario y verificará en cada nodo del árbol si existe un semejante.

### Pruebas del software

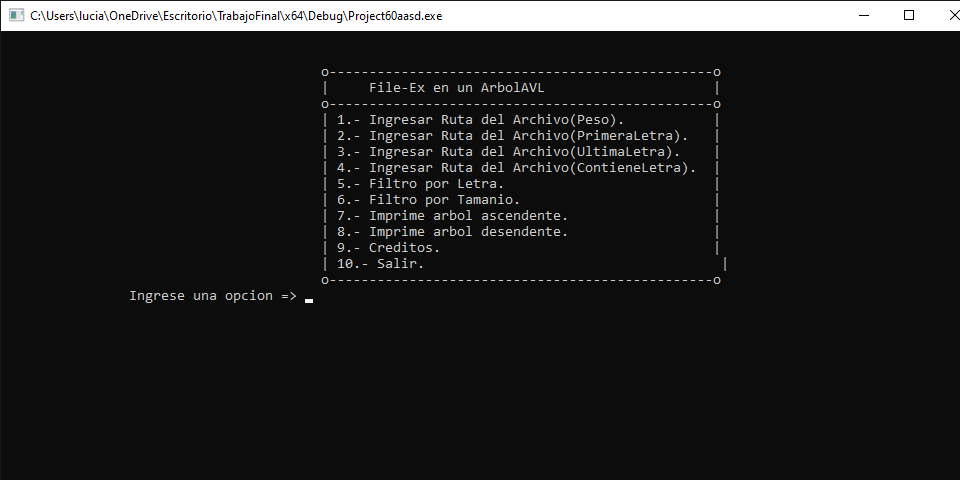




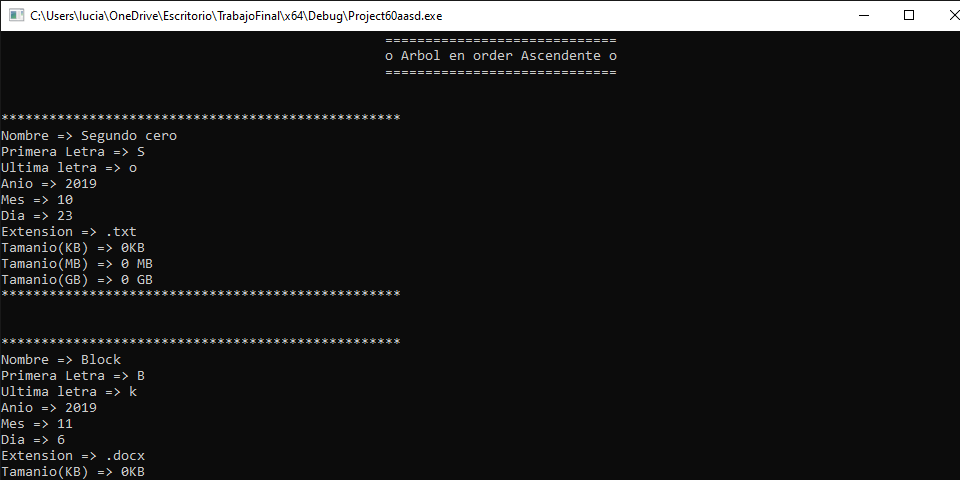




Este es nuestra pantalla de inicio con el usuario:



Este es el árbol ordenado con respecto al peso de forma ascendente:



Este la opción 5 el cual es nuestro filtrador según las letras:



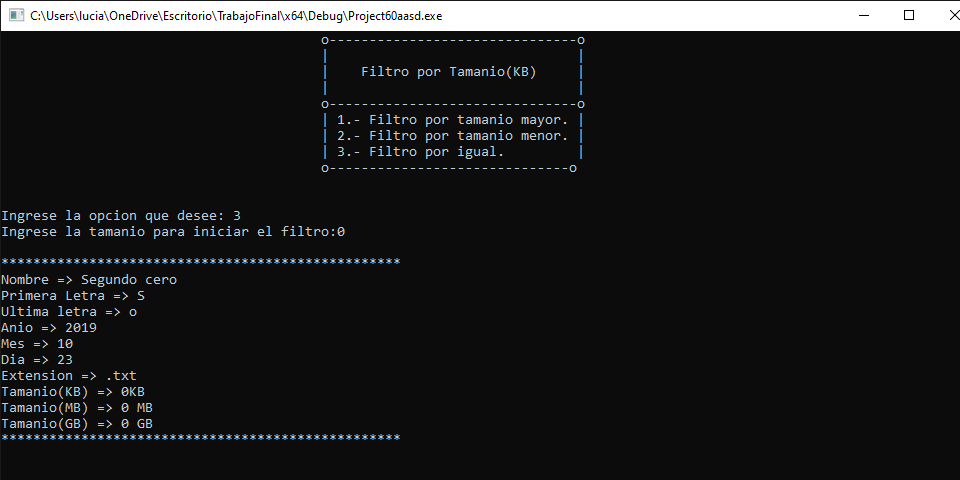
Ejecutando con la primera opción:



Ejecutando con el filtro por contenido:



Ejecutando con el filtro por tamaño igual:



## Conclusiones

Luego del desarrollo de los Hitos propuestos para este trabajo. Como alumnos de Algoritmos y Estructuras de datos concluimos que los árboles binarios sirven para realizar pequeñas bases de datos donde la búsqueda y almacenamiento sean óptimos. Ya que un árbol binario no almacena los datos de forma lineal como una lista, un arreglo o vector donde los datos que entran solo se pueden leer en el respectivo orden de entrada.

Un árbol binario resulta ser una estructura de datos útil y fácil de implementar. Además, permite hacer búsquedas, inserciones y eliminaciones de elementos a muy altas velocidades. Ya que su función de eficiencia tiene un tiempo promedio de O(log2N) donde N es el número de elementos presentes en el árbol.

Además, podemos concluir que este trabajo nos ayudó a poder trabajar grupalmente. Además, el informe nos ayudó a tener una perspectiva de cómo este trabajo apoya a las entidades económicas, educativas y salud.

## Bibliografía

* Gabriel del Castillo Mory. Siguiente reforma: La impostergable reingeniería del sistema de salud pública en el Perú. Pacifico Business School. <http://pbs.edu.pe/noticias/siguiente-reforma-impostergable-reingenieria-sistema-salud-publica-peru-gabriel-castillo-mory/>. [Consultado el 17 de noviembre del 2019].
* Programa de prioridades de la Salud. Instituto Nacional de Estadística e Informática (INEI). <https://proyectos.inei.gob.pe/endes/endes2007/1.%20Introducci%C3%B3n/1.7%20Per%C3%BA%20Programa%20y%20Prioridades%20de%20Salud.html>. [Consultado el 18 de noviembre del 2019]
* Requerimientos no funcionales, PMOinformatica. <http://www.pmoinformatica.com/2015/05/requerimientos>. [Consultado el 18 de noviembre del 2019]
* Gestion. (2018). Historias clínicas digitales disminuirán hasta en 16 días la demora en atención EsSalud. Recuperado de: <https://gestion.pe/economia/historias-clinicas-digitales-disminuirian-16-dias-demora-atencion-essalud-252808-noticia/>. [Consultado el 19 de noviembre del 2019]

## Anexos

* Projecto Gihub: https://github.com/users/EdwarZevallos/projects/1
* Repositorio GitHub: <https://github.com/EdwarZevallos/AlgoFinal> (Edwar Zevallos)
* Repositoria GitHub: <https://github.com/FelCreepsXT/AlgoritmoEstructuraDeDatos> (Luciano Lopez)