

**Tema:**

Proyecto Parcial 2 Arboles B

**Integrantes:**

Carlos Flamenco #62011019

Raul Trochez #61841297

**Asignatura:**

Estructura de datos II

**Docente:**

Ing. Oscar Galeano

**Seccion:**

#851

Fecha:

19/jun/2022

**INDICE**

[**Introducción** 3](#_Toc106533749)

[**Pseudocodigo** 4](#_Toc106533750)

[**Descripcion de codigo** 7](#_Toc106533751)

[**Bitacora** 9](#_Toc106533752)

[**Dificultades econtradas** 11](#_Toc106533753)

[**Enlace de video en Youtube** 12](#_Toc106533754)

[**Conclusion** 13](#_Toc106533755)

# **Introducción**

En este informe daremos a conocer lo que realizamos en el proyecto del segundo parcial como tema Arboles B, Los arboles B tienen ventajas sustanciales sobre otras implementaciones cuando el tiempo de acceso a los nodos excede al tiempo de acceso entre nodos. Este caso se da usualmente cuando los nodos se encuentran en dispositivos de almacenamiento secundario como los discos rígidos.

El lenguaje seleccionado para el desarrollo de este programa de arboles b fue el lenguaje de C++ en el cual trabajamos toda la estructura del mismo.

Los arboles B funcionan de la siguiente manera cada nodo puede almacenar un máximo de m-1 claves. Las características que debe cumplir un árbol-B son: Un parámetro muy importante en los árboles-B es el ORDEN (m). El orden de un árbol-B es el número máximo de ramas que pueden partir de un nodo.

# **Pseudocodigo**

Pseudocodigo de las partes mas importantes del codigo para su funcionamiento.

Int menu

Creacion de arbol

Void para animar el reinicio del programa

Int main

Escribir eleccion

Escribir orden

Escribir numero

Según opcion hacer:

1: agregar

2:agregar por archivo

3:Pintar o mostrar lista

4:Pintar o mostrar la lista de prueba

5:Borrar numero

6:Borrar por archivo

7:Guardar por archivo

8:Reiniciar el programa

9: Estadistica

10: Buscar Numero

11: Salir

Luego contamos con lo que son la creacion de las paginas:

Struct casilla

insertar

Escribir valor

Pagina \*hijoIzquierdo, Hijo derecho

Pagina actual

Casilla \*siguiente, anterior

Clase pagina

Entero n

Casilla \*principio,\*final,\*anterior,\*mitad,\*lugar Encontrado;

Pagina \*padre;

Casilla \*padreCasilla;

Entero cuantos

Leer public:

Paginas

Terminar

Buscar

Booleano agregar

Tipo void acomodar

Entero obtener cuantos

Carilla obtener mitad

Void padre

Casilla obtener final

Casilla obtener padre

Casilla obtener inicio

Ahora nos dirigimos a la parte del arbol:

Leer clase arbolB

Entero orden

Entero minimo

Pagina puntero raiz

Casilla puntero lugar encontrado

Casilla puntero principio

Entero cuantos

Leer public

Arbol B entero4arbolB

Void poner el orden

Void destructor

Booleano busqueda

Booleano pintar

Void agregar

Void pintar arbol

Void dividi para splits

Void intercambiar

Buscar numero

Casilla lugar

Pagina p, actual

P es igual a raiz

Mientras que P

Si el actual es igual es buscar2

Lugar es igual a ObtenerLugarencontrado

# **Descripcion de codigo**

Para este programa contamos con un Menu con 11 opciones, para ello utilizamos lo que es un switch case con 11 cases para el funcionamiento de cada opcion.

“Case 1: agregar por teclado, en esta parte podemo agregar un numero de manera pulsativa con el teclado de nuestra computadora.

“Case 2”: Agregar por archivo, aquí nos tendremos la posibilidad de agregar datos mediante un archivo tercero o archivo ya existente.

“Case 3”: Pintar lista, en esta parte es cuando se muestran los datos que se han agregado al arbol.

“Case 4”:Pintar lista Prueba, aquí es basicamente lo mismo que el 3 a diferencia que nos muestra una prueba de como ira quedando nuestro arbol.

“Case 5”: Borrar Numero, aquí tenemos la posibilidad de eliminar cualquier dato de nuestro arbol b.

“Case 6”: Borrar por archivo, como su nombre lo dice podemos borrar en un archivo externo nuestro datos ya ingresados.

“Case 7”: Guardar por archivo, en esta parte podremos registrar y guardar nuestros datos en un archivo externo.

“Case 8”: Reiniciar programa, para iniciar desde 0 nuestro programa.

“Case 9”: Estadisticas, en esta parte es la implementacion de las estadisticas del programa.

“Case 10”: Buscar Numero, en el cual podremos dar la busqueda de un numero en especifico que este o no en nuestra arbol creado.

“Case 11”:Salir, aquí damos por finalizado la ejecucion de todo el programa..

# **Bitacora**

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Actividad | Participantes | Tiempo |
| 30/May/2022 | Selección de el lenguaje para el desarrollo del proyecto | Carlos, Raul | 2 dias |
| 1/jun/2022 | Inicio de la base del proyecto en C# | Carlos | 1 dia |
| 2/jun/2022 | Creacion de diferentes clases en c# | Raul | 1 dia |
| 4/jun/2022 | Decision de cambio de lenguaje a C++ en vez de c# | Carlos, Raul | 1 dia |
| 6/jun/2022 | Inicio del informe para el proyecto | Carlos, raul | 2 horas |
| 6/jun/2022 | Creacion de Primeras operaciones | Carlos | 3 horas |
| 7/jun/2022 | Reunion via Zoom para agregar y corregir errores | Carlos, raul | 3 horas |
| 8/jun/2022 | Primeras pruebas del codigo | Carlos, raul | 2 horas |
| 8/jun/2022 | Creacion de mas operaciones | Raul | 3 horas |
| 9/jun/2022 | Pruebas y cambio de variables | Carlos | 3 horas |
| 10/jun/2022 | Reunion via Zoom para corregir errores en el codigo y hacer pruebas | Carlos, Raul | 4 horas |
| 11/jun/2022 | Resumir y quitar procesos que no se ocupen | carlos | 1 dia |
| 13/jun/2022 | Reunion via Zoom para observar el codigo y ver sus funciones | Carlos, Raul | 2 horas |
| 14/jun/2022 | Implementacion de estructuras de orden | Carlos | 3 horas |
| 15/jun/2022 | Reunion via zoom para probar todo el codigo | Carlos, raul | 2 horas |
| 18/jun/2022 | Implementacion de estadisticas | carlos | 4 horas |
| 19/jun/22 | Proyecto finalizado y entragado | Carlos,Raul | 1 hora |

# **Dificultades econtradas**

Como primer punto encontramos que tuvimos que cambiar de lenguaje ya que para realziar toda la estructura como queriamos era mas complicado de realizarse en c# por eso decidimos de migrar a c++ ya que observamos que la estructura seria mas facil y entendible.

Tuvimos ciertos problemas con la estructura del arbol ya que al comienzo el arbol no se nos mostraba en pantalla y todo el codigo estaba de manera correcto, lo cual nos llevo a revisar casi la mitad del codigo para hacer los diferentes cambios para que se mostrara en pantalla.

# **Enlace de video en Youtube**

# **Conclusion**

Hemos concluido con la ejecucion y creacion de este progama de arboles B desarrollado por Carlos Flamenco y Raul Trochez en el lenguaje de programacion C++, podemos concluir que los arboles B mantienen en su mayoria del tiempo los datos de forma ordenada y precisa o tambien llamados arboles Paginados, las inserciones y eliminaciones se realizan en tiempo logaritmico amortizado  los nodos internos deben tener un número variable de nodos hijo dentro de un rango predefinido. Cuando se inserta o se elimina un dato de la estructura, la cantidad de nodos hijo varía dentro de un nodo. Logramos comprender de mejor manera el funcionamiento de los arboles B mediante este proyecto realizado.