

UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERIAS



ESTRUCTURA DE DATOS II

Árbol binario de búsqueda

Angel Josue Quezada Lujan

Código: 217452746

24/02/2023

Profesor: Ortega Sánchez Noe
División: DIVTIC

Clave: I5888
Sección: D01
NRC: 59812

Introducción:

En esta actividad se implementó un árbol de búsqueda binaria, en el cual se implementaron funciones para canción y además eliminar canciones, esto mediante el uso de sobrecargas en las que se comparan las variables y objetos de las clases como los de “name” los cuales almacenan el nombre del interprete y del artista, existe la función de “findData” que busca por el nombre de la canción. También se agregó la función de cargar a disco y leer de disco, los tres tipos de recorridos in, pos, pre. Y algunas otras funciones para complementar el árbol. Este código fue reutilizado de un árbol que había realizado el semestre pasado el cual funcionaba con templates, la única diferencia era que el otro árbol trabajaba con una clase de enteros ya que tiene templates y este en vez de ser con una clase de enteros se utilizó la clase que utilizaba en las listas que era la de “song” la cual permite realizar diversas funciones con lo que suele componer una canción, como nombre de la canción, artista, interprete, ranking, formato en mp3 etc, esta clase también tiene sus sobrecargas para el manejo adecuado del archivo. Estas son las sobrecargas que se utilizan en las clases que utiliza el proyecto, “name” y “song” solo que a diferencia de esta que es de “name” la de “song” tiene a su clase en el apartado que le corresponde.

```
bool operator == (const Name&) const;
bool operator != (const Name&) const;
bool operator > (const Name&) const;
bool operator >= (const Name&) const;
bool operator < (const Name&) const;
bool operator <= (const Name&) const;
```

Así los declaramos en la clase de “name”, y así los implementamos:

```
bool Name::operator == (const Name& n) const {
    return first == n.first and last == n.last;
}

bool Name::operator != (const Name& n) const {
    return first != n.first and last != n.last;
}

bool Name::operator > (const Name& n) const {
    return first > n.first and last > n.last;
}

bool Name::operator >= (const Name& n) const {
    return first >= n.first and last >= n.last;
}

bool Name::operator < (const Name& n) const {
```

```

    return first < n.first and last < n.last;
}

bool Name::operator <= (const Name& n) const {
    return first <= n.first and last <= n.last;
}

```

Así los declaramos en la clase de “song”:

```

bool operator == (const Song&) const;
bool operator != (const Song&) const;
bool operator > (const Song&) const;
bool operator >= (const Song&) const;
bool operator < (const Song&) const;
bool operator <= (const Song&) const;

```

y así los implementamos:

```

bool Song::operator == (const Song& c) const {
    return cancion == c.cancion;
}

bool Song::operator != (const Song& c) const {
    return !(*this == c);
}

bool Song::operator > (const Song& c) const {
    return !(*this <= c);
}

bool Song::operator >= (const Song& c) const {
    return !(*this < c);
}

bool Song::operator < (const Song& c) const {
    return cancion < c.cancion;
}

bool Song::operator <= (const Song& c) const {
    return *this < c or *this == c;
}

```

Prácticamente de lo que se encargan es de comparar los nombres que se les dé, por ejemplo si estuviéramos buscando una canción por el nombre del artista:

```

case 'C':
    cout << myArbol.parsePreOrder() << endl << endl;

    cin.ignore();
    cout << "Write the song to search: ";
    getline(cin, myCadena);
    myCancion.setCancion(myCadena);
    system("cls");

```

Aquí recibiríamos el nombre de la canción y se guardaría en la variable “myCadena” que es de tipo string, para posteriormente enviarla a el “setter” de canción y luego a la función “findData”

Aquí es donde inicia todo, utiliza la función `findData...`, y en caso de que no la encuentre ósea que la posición sea nula, mandara un mensaje y en caso de que la encuentre la imprimirá en pantalla usando la función `retrieve`.

```
pos = myArbol.findData(myCancion);

cout<< "The song with name "<<myCadena;
if(pos == nullptr) {
    cout<<" isn't in the list "<<endl<<endl;
} else {
    cout<< " is in the position "<<pos<< " of the tree"<<endl<<endl;
    cout<< " Song: ";
    cout<< myArbol.retrieve(pos).getMP3()<<endl<<endl;
    cout<< " Full record: "<<endl;
    cout<< myArbol.retrieve(pos).toString()<<endl<<endl;
}
pos = nullptr;

break;
```

“`findData`” recibe dos parámetros, un template de la clase que se está usando, ósea `song` y la posición.

```
template<class T>
typename BTree<T>::Position& BTree<T>::findData(Position& r, const T& e)
{
    if(r == nullptr or r->getData() == e) {
        return r;
    }

    if(e < r->getData()) {
        return findData(r->getLeft(), e);
    }

    else {
        return findData(r->getRight(), e);
    }
}
```

Se verifica si la posición es nula o si tiene algún elemento se iguala al template y se retorna, si el dato que está en la posición “e” es menor que el de la posición en la que esta “r” se retorna utilizando otra función de “`findData`” la cual recibe la raíz izquierda y el elemento, y en caso de que no se retorna con la misma función pero el lado derecho.

```
template<class T>
typename BTree<T>::Position& BTree<T>::findData(const T& e) {
    return findData(root, e);
}
```

En la función de “`getData`” se crea un apuntador del objeto nodo llamado `aux` el cual esta inicializado en el ancla, y en la función `while` comienza a recorrer la lista usando “`getData()`” que es otra función para comparar el parámetro que se solicita con el del nodo en el que se encuentra y así cuando lo encuentre

esta función lo devuelve.

```
template<class T>
T BTree<T>::Node::getData() const {
    if (dataPtr == nullptr) {
        throw Exception("Dato inexistente, T BTree<T>::Node::getData");
    }
    return *dataPtr;
}
```

Esta función es muy simple ya que es de la clase nodo, la cual trabaja en conjunto con la clase árbol, se podría decir que esta anidada en esta misma.

Básicamente lo que hace es retornar el dato si es que no es nulo, pero lo regresa en un apuntador. “*dataPtr” es un template.

Para eliminar un elemento del árbol.

```
template<class T>
void BTree<T>::deleteData(const T& e) {
    deleteData(root, e);
}
```

Esta función se utiliza para la recursividad de la siguiente función, pero lo que hace es recibir el objeto y lo manda a la siguiente función inicializado desde la raíz.

```
template<class T>
void BTree<T>::deleteData(Position& r, const T& e) {
    if (r == nullptr) {
        return;
    }

    if (r->getData() == e) {
        if (isLeaf(r)) {
            delete r;
            r = nullptr;
            return;
        }

        if (r->getLeft() != nullptr and r->getRight() != nullptr) {
            T myData(getHighest(r->getLeft())->getData());

            deleteData(r, myData);
            r->setData(myData);

            return;
        }
    }
}
```

```

        Position aux(r->getLeft() == nullptr ? aux = r->getLeft() : aux =
r->getRight());

        delete r;

        r = aux;
    }

    if(e < r->getData()) {
        deleteData(r->getLeft(), e);
    }

    else {
        deleteData(r->getRight(), e);
    }
}

```

Primero se verifica que la posición recibida no sea nula, en caso de que si sea se retornara. Se verifica si el dato que está en la posición que recibe junto con la función de getData es igual al que se recibió por parte del usuario, en caso de que si, se hace una validación para verificar si es una hoja, se elimina y la posición se hace nula y retorna. Si la posición izquierda es nula al igual que la derecha, se solicita la altura del árbol, se repite recursivamente todo con deleteData, y lo que hay en la posición “r” se envía a myData, Se crea un auxiliar de tipo *Nodo, el cual es “Position aux...”, y se verifica si la posición en la izquierda es nula, en caso de que si, el auxiliar se iguala a la posición de la izquierda, si no a la posición derecha, finalmente se elimina la posición y la posición se iguala al auxiliar. Para terminar se validará si el objeto es menor al objeto que está en la posición, si es así, se eliminara de la posición derecha, sino de la izquierda. Esta función es un poco agresiva y al inicio difícil de comprender porque utiliza otra función para realizar la recursividad. Además de que se realizan los religados.

También hemos utilizado una función que es la de retrieve la cual prácticamente lo que hace es que si encuentra al dato en la posición que recibe lo devuelve.

```

template<class T>
T BTree<T>::retrieve(Position& r) const {
    return r->getData();
}

```

Otra función muy importante que se utiliza demasiado es la de “toString()” la cual se encarga de imprimir.

```
template<class T>
string BTree<T>::toString(Position& r) {
    string result;

    if(r == nullptr) {
        return result;
    }

    result+= r->getData().toString();
    result+= '\n';
    result+= toString(r->getLeft());
    result+= toString(r->getRigth());

    return result;
}
```

Las clases que se utilizan en el proyecto tienen implementados sus “toString”, la clase “Name”, la clase “Song” y la que se muestra anteriormente que es la de la clase “Btree”.

```
string Song::toString() const {
    string result;

    result += "|Song: ";
    result += cancion;
    result.resize(25, ' ');
    result += " |Artist: ";
    result += artista.toString();
    result.resize(65, ' ');
    result += " |Rank: ";
    result += ranking;
    result.resize(75, ' ');
    result += " |Interpeter: ";
    result.resize(85, ' ');
    result += nameInterpeter.toString();
    result.resize(110, ' ');
    result += " | ";
    result += MP3;
    result.resize(135, ' ');
    result += " | ";

    return result;
}
```


Esta función es muy útil ya que organiza de una manera más optima los datos, hace uso de funciones de librerías del lenguaje, como la función “resize” para organizar con espacios los datos, esto hace que se vea de la siguiente manera.

Song: Passionfruit	Artist: Drake Inc	Rank: 04	InterpetNana Rogues	Passionfruit.MP3
Song: Money Trees	Artist: Kendrick Lamar	Rank: 01	InterpetJay Rock	MoneyTrees.MP3
Song: It Was A Good Day	Artist: Ice Cube	Rank: 17	InterpetDJ Pooh	ItWasAGoodDay.MP3
Song: Big Poppa	Artist: Carl Chucky Thompson	Rank: 07	InterpetConrad Rad Diamanche	BigPoppa.MP3
Song: Do I Wanna Know?	Artist: Alex Turner	Rank: 18	InterpetArtic Monkeys	DoIWannaKnow?.MP3
Song: JGL	Artist: Luis R Conriquez	Rank: 34	InterpetLa Adictiva	JGL.MP3
Song: Me Porto Bonito	Artist: Benito Antonio Martinez Ocasio	Rank: 12	InterpetBad Bunny	MePortoBonito.MP3
Song: Titi Me Pregunto	Artist: Benito Antonio Martinez Ocasio	Rank: 13	InterpetBad Sunny	TitiMePregunto.MP3
Song: Tarot	Artist: Benito Antonio Martinez Ocasio	Rank: 14	InterpetBad Sunny	Tarot.MP3
Song: Neverita	Artist: Benito Antonio Martinez Ocasio	Rank: 15	InterpetBad Sunny	Neverita.MP3
Song: La Corriente	Artist: Benito Antonio Martinez Ocasio	Rank: 22	InterpetBad Sunny	LaCorriente.MP3
Song: Ojos Lindos	Artist: Benito Antonio Martinez Ocasio	Rank: 29	InterpetBad Sunny	OjosLindos.MP3
Song: Aguacero	Artist: Benito Antonio Martinez Ocasio	Rank: 45	InterpetBad Sunny	Aguacero.MP3
Song: Un Coco	Artist: Benito Antonio Martinez Ocasio	Rank: 1	InterpetBad Sunny	UnCoco.MP3
Song: Andrea	Artist: Benito Antonio Martinez Ocasio	Rank: 9	InterpetBad Sunny	Andrea.MP3
Song: Party	Artist: Rauw Alejandro	Rank: 38	InterpetBad Sunny	Party.MP3
Song: AM REMIX	Artist: Nio Garcia	Rank: 27	InterpetJ Balvin Bad Bunny	AmRemix.MP3
Song: Imagination	Artist: Foster The People	Rank: 26	InterpetColumbia Records	Imagination.MP3
Song: After Dark	Artist: Mr Kitty	Rank: 10	InterpetAndrew Stans	AfterDark.MP3
Song: Purple Lamborghini	Artist: Colin Tiller	Rank: 05	InterpetAndrew Larios Skrillex	PurpleLamborghini.MP3
Song: Starboy	Artist: The Weekend	Rank: 06	InterpetDaft Punk	Starboy.MP3
Song: Call Out My Name	Artist: The Weekend	Rank: 08	InterpetSaul Germaine	CallOutMyName.MP3

SPOTIFY PREMIUM
MAIN MENU

Y así se implementa en “Name”.

```
string Name::toString() const {
    return first + " " + last;
}
```

Prácticamente lo que hace es colocar la cadena del primer nombre antes del apellido.

Song: Tell Good Inc	Artist: Gorillas Inc	Rank: 28	Interpet: Jale Hewlett	TellIncGood.MP3
Song: After The Storm	Artist: The Creator	Rank: 10	Interpet: Kali Uchis	AfterTheStorm.MP3
Song: AM REMIX	Artist: J Balvin Bad Bunny	Rank: 27	Interpet: Nio Garcia	AmRemix.MP3
Song: After Dark	Artist: Andrew Stans	Rank: 10	Interpet: Mr Kitty	AfterDark.MP3
Song: Big Poppa	Artist: Conrad Rad Diamanche	Rank: 07	Interpet: Carl Chucky Thompson	BigPoppa.MP3
Song: Aguacero	Artist: Bad Bunny	Rank: 45	Interpet: Benito Antonio Martinez Ocasio	Aguacero.MP3
Song: Andrea	Artist: Bad Bunny	Rank: 9	Interpet: Benito Antonio Martinez Ocasio	Andrea.MP3
Song: Do I Wanna Know?	Artist: Artic Monkeys	Rank: 18	Interpet: Alex Turner	DoIWannaKnow?.MP3
Song: Call Out My Name	Artist: Saul Germaine	Rank: 08	Interpet: The Weekend	CallOutMyName.MP3
Song: Without Me	Artist: Kevin Bell	Rank: 16	Interpet: Eminem Inc	WithoutMe.MP3
Song: Heathens	Artist: Lin Man	Rank: 21	Interpet: Twenty One Pilots	Heathens.MP3
Song: Passionfruit	Artist: Nana Rogues	Rank: 04	Interpet: Drake Inc	Passionfruit.MP3
Song: Money Trees	Artist: Jay Rock	Rank: 01	Interpet: Kendrick Lamar	MoneyTrees.MP3
Song: It Was A Good Day	Artist: DJ Pooh	Rank: 17	Interpet: Ice Cube	ItWasAGoodDay.MP3
Song: Imagination	Artist: Columbia Records	Rank: 26	Interpet: Foster The People	Imagination.MP3
Song: JGL	Artist: La Adictiva	Rank: 34	Interpet: Luis R Conriquez	JGL.MP3
Song: Me Porto Bonito	Artist: Bad Bunny	Rank: 12	Interpet: Benito Antonio Martinez Ocasio	MePortoBonito.MP3
Song: La Corriente	Artist: Bad Bunny	Rank: 22	Interpet: Benito Antonio Martinez Ocasio	LaCorriente.MP3
Song: Neverita	Artist: Bad Bunny	Rank: 15	Interpet: Benito Antonio Martinez Ocasio	Neverita.MP3
Song: Ojos Lindos	Artist: Bad Bunny	Rank: 29	Interpet: Benito Antonio Martinez Ocasio	OjosLindos.MP3
Song: Party	Artist: Bad Bunny	Rank: 38	Interpet: Rauw Alejandro	Party.MP3
Song: Titi Me Pregunto	Artist: Bad Bunny	Rank: 13	Interpet: Benito Antonio Martinez Ocasio	TitiMePregunto.MP3
Song: Tarot	Artist: Bad Bunny	Rank: 14	Interpet: Benito Antonio Martinez Ocasio	Tarot.MP3
Song: Purple Lamborghini	Artist: Andrew Larios Skrillex	Rank: 05	Interpet: Colin Tiller	PurpleLamborghini.MP3
Song: Starboy	Artist: Daft Punk	Rank: 06	Interpet: The Weekend	Starboy.MP3
Song: Un Coco	Artist: Bad Bunny	Rank: 1	Interpet: Benito Antonio Martinez Ocasio	UnCoco.MP3

SPOTIFY PREMIUM
MAIN MENU

Add a new song [A]
Delete song [B]
Find song [C]
Retrieve song [D]
Parse inOrder [I]

[Song: Fell Good Inc	[Artist: Gorillaz Inc	[Rank: 28	[Interpeter: Jale Hewlett	FellIncGood.MP3
[Song: After The Storm	[Artist: The Creator	[Rank: 18	[Interpeter: Kali Uchis	AfterTheStorm.MP3
[Song: AM REMIX	[Artist: J Balvin Bad Bunny	[Rank: 27	[Interpeter: Mía García	AmRemix.MP3
[Song: After Dark	[Artist: Andrew Stans	[Rank: 18	[Interpeter: Mr Kitty	AfterDark.MP3
[Song: Big Poppa	[Artist: Conrad Bad Dismanche	[Rank: 87	[Interpeter: Carl Chucky Thompson	BigPoppa.MP3
[Song: Aguacero	[Artist: Bad Bunny	[Rank: 45	[Interpeter: Benito Antonio Martinez Ocasio	Aguacero.MP3
[Song: Andrea	[Artist: Bad Bunny	[Rank: 9	[Interpeter: Benito Antonio Martinez Ocasio	Andrea.MP3
[Song: Do I Wanna Know?	[Artist: Arctic Monkeys	[Rank: 18	[Interpeter: Alex Turner	DoIWannaKnow?.MP3
[Song: Call Out My Name	[Artist: Saul Germaine	[Rank: 88	[Interpeter: The Weeknd	CallOutMyName.MP3
[Song: Without Me	[Artist: Kevin Bell	[Rank: 16	[Interpeter: Eminem Inc	WithoutMe.MP3
[Song: Heathens	[Artist: Lin Han	[Rank: 21	[Interpeter: Twenty One Pilots	Heathens.MP3
[Song: Passionfruit	[Artist: Nana Rogues	[Rank: 84	[Interpeter: Drake Inc	Passionfruit.MP3
[Song: Money Trees	[Artist: Jay Rock	[Rank: 81	[Interpeter: Kendrick Lamar	MoneyTrees.MP3
[Song: It Was A Good Day	[Artist: DJ Pooch	[Rank: 17	[Interpeter: Ice Cube	ItWasAGoodDay.MP3
[Song: Imagination	[Artist: Columbia Records	[Rank: 26	[Interpeter: Foster The People	Imagination.MP3
[Song: JGL	[Artist: La Adictiva	[Rank: 34	[Interpeter: Luis R Conriquez	JGL.MP3
[Song: Me Porto Bonito	[Artist: Bad Bunny	[Rank: 12	[Interpeter: Benito Antonio Martinez Ocasio	MePortoBonito.MP3
[Song: La Corriente	[Artist: Bad Bunny	[Rank: 22	[Interpeter: Benito Antonio Martinez Ocasio	LaCorriente.MP3
[Song: Neverita	[Artist: Bad Bunny	[Rank: 15	[Interpeter: Benito Antonio Martinez Ocasio	Neverita.MP3
[Song: Ojos Lindos	[Artist: Bad Bunny	[Rank: 29	[Interpeter: Benito Antonio Martinez Ocasio	OjosLindos.MP3
[Song: Party	[Artist: Bad Bunny	[Rank: 38	[Interpeter: Rauw Alejandro	Party.MP3
[Song: Titi Me Pregunto	[Artist: Bad Bunny	[Rank: 13	[Interpeter: Benito Antonio Martinez Ocasio	TitiMePregunto.MP3
[Song: Tarot	[Artist: Bad Bunny	[Rank: 14	[Interpeter: Benito Antonio Martinez Ocasio	Tarot.MP3
[Song: Purple Lamborghini	[Artist: Andrew Larins Skrillex	[Rank: 85	[Interpeter: Colin Tiller	PurpleLamborghini.MP3
[Song: Starboy	[Artist: Daft Punk	[Rank: 86	[Interpeter: The Weeknd	Starboy.MP3
[Song: Un Coco	[Artist: Bad Bunny	[Rank: 1	[Interpeter: Benito Antonio Martinez Ocasio	UnCoco.MP3

Write the name of the song to delete: Un Coco

Deleting the song: Un Coco

[Song: Un Coco	[Artist: Bad Bunny	[Rank: 1	[Interpeter: Benito Antonio Martinez Ocasio	UnCoco.MP3
----------------	--------------------	----------	---	------------

[Song: Fell Good Inc	[Artist: Gorillaz Inc	[Rank: 28	[Interpeter: Jale Hewlett	FellIncGood.MP3
[Song: After The Storm	[Artist: The Creator	[Rank: 18	[Interpeter: Kali Uchis	AfterTheStorm.MP3
[Song: AM REMIX	[Artist: J Balvin Bad Bunny	[Rank: 27	[Interpeter: Mía García	AmRemix.MP3
[Song: After Dark	[Artist: Andrew Stans	[Rank: 18	[Interpeter: Mr Kitty	AfterDark.MP3
[Song: Big Poppa	[Artist: Conrad Bad Dismanche	[Rank: 87	[Interpeter: Carl Chucky Thompson	BigPoppa.MP3
[Song: Aguacero	[Artist: Bad Bunny	[Rank: 45	[Interpeter: Benito Antonio Martinez Ocasio	Aguacero.MP3
[Song: Andrea	[Artist: Bad Bunny	[Rank: 9	[Interpeter: Benito Antonio Martinez Ocasio	Andrea.MP3
[Song: Do I Wanna Know?	[Artist: Arctic Monkeys	[Rank: 18	[Interpeter: Alex Turner	DoIWannaKnow?.MP3
[Song: Call Out My Name	[Artist: Saul Germaine	[Rank: 88	[Interpeter: The Weeknd	CallOutMyName.MP3
[Song: Without Me	[Artist: Kevin Bell	[Rank: 16	[Interpeter: Eminem Inc	WithoutMe.MP3
[Song: Heathens	[Artist: Lin Han	[Rank: 21	[Interpeter: Twenty One Pilots	Heathens.MP3
[Song: Passionfruit	[Artist: Nana Rogues	[Rank: 84	[Interpeter: Drake Inc	Passionfruit.MP3
[Song: Money Trees	[Artist: Jay Rock	[Rank: 81	[Interpeter: Kendrick Lamar	MoneyTrees.MP3
[Song: It Was A Good Day	[Artist: DJ Pooch	[Rank: 17	[Interpeter: Ice Cube	ItWasAGoodDay.MP3
[Song: Imagination	[Artist: Columbia Records	[Rank: 26	[Interpeter: Foster The People	Imagination.MP3
[Song: JGL	[Artist: La Adictiva	[Rank: 34	[Interpeter: Luis R Conriquez	JGL.MP3
[Song: Me Porto Bonito	[Artist: Bad Bunny	[Rank: 12	[Interpeter: Benito Antonio Martinez Ocasio	MePortoBonito.MP3
[Song: La Corriente	[Artist: Bad Bunny	[Rank: 22	[Interpeter: Benito Antonio Martinez Ocasio	LaCorriente.MP3
[Song: Neverita	[Artist: Bad Bunny	[Rank: 15	[Interpeter: Benito Antonio Martinez Ocasio	Neverita.MP3
[Song: Ojos Lindos	[Artist: Bad Bunny	[Rank: 29	[Interpeter: Benito Antonio Martinez Ocasio	OjosLindos.MP3
[Song: Party	[Artist: Bad Bunny	[Rank: 38	[Interpeter: Rauw Alejandro	Party.MP3
[Song: Titi Me Pregunto	[Artist: Bad Bunny	[Rank: 13	[Interpeter: Benito Antonio Martinez Ocasio	TitiMePregunto.MP3
[Song: Tarot	[Artist: Bad Bunny	[Rank: 14	[Interpeter: Benito Antonio Martinez Ocasio	Tarot.MP3
[Song: Purple Lamborghini	[Artist: Andrew Larins Skrillex	[Rank: 85	[Interpeter: Colin Tiller	PurpleLamborghini.MP3
[Song: Starboy	[Artist: Daft Punk	[Rank: 86	[Interpeter: The Weeknd	Starboy.MP3

Do you want to delete other? (y/n):

Song: Ojos Lindos	Artist: Bad Bunny	Rank: 29	Interpeter: Benito Antonio Martinez Ocasio	OjosLindos.MP3
Song: Party	Artist: Bad Bunny	Rank: 30	Interpeter: Rauw Alejandro	Party.MP3
Song: Titi Me Pregunto	Artist: Bad Bunny	Rank: 13	Interpeter: Benito Antonio Martinez Ocasio	TitiMePregunto.MP3
Song: Tarot	Artist: Bad Bunny	Rank: 14	Interpeter: Benito Antonio Martinez Ocasio	Tarot.MP3
Song: Purple Lamborghini	Artist: Andrew Lerios Skrillex	Rank: 05	Interpeter: Colin Tiller	PurpleLamborghini
Song: Starboy	Artist: Daft Punk	Rank: 06	Interpeter: The Weekend	Starboy.MP3

Do you want to delete other? (y/n): y

Write the name of the song to delete: Andrea

Deleting the song: Andrea

Song: Andrea	Artist: Bad Bunny	Rank: 9	Interpeter: Benito Antonio Martinez Ocasio	Andrea.MP3
--------------	-------------------	---------	--	------------

Song: Fell Good Inc	Artist: Gorillaz Inc	Rank: 20	Interpeter: Jale Hewlett	FellIncGood.MP3
Song: After The Storm	Artist: The Creator	Rank: 10	Interpeter: Kali Uchis	AfterTheStorm.MP3
Song: AM REMIX	Artist: J Balvin Bad Bunny	Rank: 27	Interpeter: Nio Garcia	AMRemix.MP3
Song: After Dark	Artist: Andrew Stans	Rank: 10	Interpeter: Mr Kitty	AfterDark.MP3
Song: Big Poppa	Artist: Conrad Bad Diamanche	Rank: 07	Interpeter: Carl Chucky Thompson	BigPoppa.MP3
Song: Aguacero	Artist: Bad Bunny	Rank: 45	Interpeter: Benito Antonio Martinez Ocasio	Aguacero.MP3
Song: Do I Wanna Know?	Artist: Arctic Monkeys	Rank: 18	Interpeter: Alex Turner	DoIWannaKnow?.MP3
Song: Call Out My Name	Artist: Saul Germaine	Rank: 60	Interpeter: The Weekend	CallOutMyName.MP3
Song: Without Me	Artist: Kevin Bell	Rank: 16	Interpeter: Eminem Inc	WithoutMe.MP3
Song: Heathens	Artist: Lin Man	Rank: 21	Interpeter: Twenty One Pilots	Heathens.MP3
Song: Passionfruit	Artist: Nana Rogues	Rank: 04	Interpeter: Drake Inc	Passionfruit.MP3
Song: Money Trees	Artist: Jay Rock	Rank: 01	Interpeter: Kendrick Lamar	MoneyTrees.MP3
Song: It Was A Good Day	Artist: DJ Pooh	Rank: 17	Interpeter: Ice Cube	ItWasAGoodDay.MP3
Song: Imagination	Artist: Columbia Records	Rank: 26	Interpeter: Foster The People	Imagination.MP3
Song: JGL	Artist: La Adictiva	Rank: 34	Interpeter: Luis R Conriquez	JGL.MP3
Song: Me Porto Bonito	Artist: Bad Bunny	Rank: 12	Interpeter: Benito Antonio Martinez Ocasio	MePortoBonito.MP3
Song: La Corriente	Artist: Bad Bunny	Rank: 22	Interpeter: Benito Antonio Martinez Ocasio	LaCorriente.MP3
Song: Neverita	Artist: Bad Bunny	Rank: 15	Interpeter: Benito Antonio Martinez Ocasio	Neverita.MP3
Song: Ojos Lindos	Artist: Bad Bunny	Rank: 29	Interpeter: Benito Antonio Martinez Ocasio	OjosLindos.MP3

Do you want to delete other? (y/n): y

Write the name of the song to delete: Aguacero

Deleting the song: Aguacero

Song: Aguacero	Artist: Bad Bunny	Rank: 45	Interpeter: Benito Antonio Martinez Ocasio	Aguacero.MP3
----------------	-------------------	----------	--	--------------

Song: Fell Good Inc	Artist: Gorillaz Inc	Rank: 20	Interpeter: Jale Hewlett	FellIncGood.MP3
Song: After The Storm	Artist: The Creator	Rank: 10	Interpeter: Kali Uchis	AfterTheStorm.MP3
Song: AM REMIX	Artist: J Balvin Bad Bunny	Rank: 27	Interpeter: Nio Garcia	AMRemix.MP3
Song: After Dark	Artist: Andrew Stans	Rank: 10	Interpeter: Mr Kitty	AfterDark.MP3
Song: Big Poppa	Artist: Conrad Bad Diamanche	Rank: 07	Interpeter: Carl Chucky Thompson	BigPoppa.MP3
Song: Do I Wanna Know?	Artist: Arctic Monkeys	Rank: 18	Interpeter: Alex Turner	DoIWannaKnow?.MP3
Song: Call Out My Name	Artist: Saul Germaine	Rank: 60	Interpeter: The Weekend	CallOutMyName.MP3
Song: Without Me	Artist: Kevin Bell	Rank: 16	Interpeter: Eminem Inc	WithoutMe.MP3
Song: Heathens	Artist: Lin Man	Rank: 21	Interpeter: Twenty One Pilots	Heathens.MP3
Song: Passionfruit	Artist: Nana Rogues	Rank: 04	Interpeter: Drake Inc	Passionfruit.MP3
Song: Money Trees	Artist: Jay Rock	Rank: 01	Interpeter: Kendrick Lamar	MoneyTrees.MP3
Song: It Was A Good Day	Artist: DJ Pooh	Rank: 17	Interpeter: Ice Cube	ItWasAGoodDay.MP3
Song: Imagination	Artist: Columbia Records	Rank: 26	Interpeter: Foster The People	Imagination.MP3
Song: JGL	Artist: La Adictiva	Rank: 34	Interpeter: Luis R Conriquez	JGL.MP3
Song: Me Porto Bonito	Artist: Bad Bunny	Rank: 12	Interpeter: Benito Antonio Martinez Ocasio	MePortoBonito.MP3
Song: La Corriente	Artist: Bad Bunny	Rank: 22	Interpeter: Benito Antonio Martinez Ocasio	LaCorriente.MP3
Song: Neverita	Artist: Bad Bunny	Rank: 15	Interpeter: Benito Antonio Martinez Ocasio	Neverita.MP3
Song: Ojos Lindos	Artist: Bad Bunny	Rank: 29	Interpeter: Benito Antonio Martinez Ocasio	OjosLindos.MP3
Song: Party	Artist: Bad Bunny	Rank: 30	Interpeter: Rauw Alejandro	Party.MP3
Song: Titi Me Pregunto	Artist: Bad Bunny	Rank: 13	Interpeter: Benito Antonio Martinez Ocasio	TitiMePregunto.MP3
Song: Tarot	Artist: Bad Bunny	Rank: 14	Interpeter: Benito Antonio Martinez Ocasio	Tarot.MP3
Song: Purple Lamborghini	Artist: Andrew Lerios Skrillex	Rank: 05	Interpeter: Colin Tiller	PurpleLamborghini
Song: Starboy	Artist: Daft Punk	Rank: 06	Interpeter: The Weekend	Starboy.MP3

una que utilice en una de mis listas anteriores, y esta requiere que esté programada la sobrecarga de los operadores de flujo en cada una de las clases.

```
friend std::ostream& operator << (std::ostream&, const Song&);  
friend std::istream& operator >> (std::istream&, Song&);
```

Esta es de la clase “Song” y prácticamente es similar a las demás, el “ostream” que es el que se encarga de escribir la información del archivo. Y “istream” es la que se encarga de leer en el archivo.

```
ostream& operator << (ostream& os, const Song &s) {  
    os << s.artista      << '|'  
    << s.nameInterpeter << '|'  
    << s.cancion         << '|'  
    << s.MP3             << '|'  
    << s.ranking         << '*' << endl;  
    return os;  
}  
  
istream& operator >> (istream& is, Song& s) {  
    is >> s.artista;  
    is >> s.nameInterpeter;  
    getline(is, s.cancion, '|');  
    getline(is, s.MP3, '|');  
    getline(is, s.ranking, '*');  
    return is;  
}
```

Aquí podemos observar como en el “ostream” que es el que se encarga de guardar la información “<<” se le asignan los delimitadores de campo y de registro al igual que en el “istream”, el cual da lectura a los datos “>>”, en ambos como podemos observar

indican como se guardará y leerá el archivo y como se separan por sus delimitadores.

Como delimitador de campo se utilizará “|” y como delimitador de registro se utilizará “*”.

Así es como se ven las canciones en el archivo:

DJ Pooh|Ice Cube|It Was A Good Day|ItWasAGoodDay.MP3|17*
 Conrad Rad Diamanche|Carl Chucky Thompson|Big
 Poppa|BigPoppa.MP3|07* Artic Monkeys|Alex Turner|Do I Wanna
 Know?|DoIWannaKnow?.MP3|18*

Song: Fell Good Inc	Artist: Gorillaz Inc	Rank: 28	Interpeter: Jaie Hewlett	FellIncGood.MP3
Song: After The Storm	Artist: The Creator	Rank: 10	Interpeter: Kali Uchis	AfterTheStorm.MP3
Song: AM REMIX	Artist: J Balvin Bad Bunny	Rank: 27	Interpeter: Nio Garcia	AmRemix.MP3
Song: After Dark	Artist: Andrew Stams	Rank: 10	Interpeter: Mr Kitty	AfterDark.MP3
Song: Big Poppa	Artist: Conrad Rad Diamanche	Rank: 07	Interpeter: Carl Chucky Thompson	BigPoppa.MP3
Song: Aguacero	Artist: Bad Bunny	Rank: 45	Interpeter: Benito Antonio Martinez Ocasio	Aguacero.MP3
Song: Andrea	Artist: Bad Bunny	Rank: 9	Interpeter: Benito Antonio Martinez Ocasio	Andrea.MP3
Song: Do I Wanna Know?	Artist: Artic Monkeys	Rank: 18	Interpeter: Alex Turner	DoIWannaKnow?.MP3
Song: Call Out My Name	Artist: Saul Germaine	Rank: 60	Interpeter: The Weekend	CallOutMyName.MP3
Song: Without Me	Artist: Kevin Bell	Rank: 16	Interpeter: Eminem Inc	WithoutMe.MP3
Song: Heathens	Artist: Lin Man	Rank: 21	Interpeter: Twenty One Pilots	Heathens.MP3
Song: Passionfruit	Artist: Nana Rogues	Rank: 04	Interpeter: Drake Inc	Passionfruit.MP3

Y así se muestran al leer el archivo.

Estas son las funciones de escritura y lectura, en la de escritura, es necesario crear un objeto ofstream, (myFile). Después abrir el archivo e indicar el tipo de apertura, en este caso se utilizó trunc el cual, si el archivo se abre para operaciones de salida y ya existía, su contenido anterior se elimina y se reemplaza por el nuevo. Por esta razón también se implementó en el menú el “Write to disk”. En caso de que el archivo no se puede abrir, lanzara una excepción con el mensaje de que no se pudo abrir el archivo y terminará el programa. Se inicializa la función de “writeToDisk” a la raíz. En este momento es cuando se utiliza el “ostream”, “<<”.

```
template <class T>
void BTree<T>::writeToDisk() {
    string fileName("file01.txt");
    myFile.open(fileName, ios_base::trunc);

    if(!myFile.is_open()) {
        throw Exception("No se pudo abrir " + fileName + " para escritura");
    }

    writeToDisk(root);
    myFile.close();
}
```

Y despues esta funcion de writeToDisk, recibe la raíz/posición y si esta misma no es nula, comienza a realizar algo parecido a un recorrido en preOrder, pero para escribir el archivo.

```
template <class T>
void BTree<T>::writeToDisk(const Position& r) {
    if(r == nullptr) {
        return;
    }

    myFile << r->getData();
    writeToDisk(r->getLeft());
    writeToDisk(r->getRight());
}
```

Ahora en la función “readFromDisk” que se encarga de leer la información del archivo, primero se creara un objeto de “ifstream” (orientado a ficheros de entrada). Después de indicar el nombre del archivo y haber creado el “ifstream”, al igual que en el caso anterior se le indicara la bandera que utilizara para la apertura, “in” para operaciones de entrada, si el archivo no se pudo abrir, se lanzará una excepción y terminará el programa. Antes de comenzar a leer crearemos un template “myObjeto”, además es importante asegurarnos de que la lista está vacía, utilizaremos la función de “deleteAll()”:

```
template<class T>
void BTree<T>::deleteAll() {
    deleteAll(root);
}

template<class T>
void BTree<T>::deleteAll(Position& r) {
    if(r == nullptr) {
        return;
    }

    deleteAll(r->getLeft());
    deleteAll(r->getRight());
    delete r;

    r = nullptr;
}
```


Prácticamente primero en una función, se le manda a la otra función inicializada en la raíz y en caso de que las posiciones sean nulas se retorna pero además si esta bifurcación no se cumple, se elimina recursivamente lo que hay en la izquierda y derecha, la posición también se elimina y además se apunta a nulo.

```
template <class T>
void BTree<T>::readFromDisk() {
    ifstream myFile;
    string fileName("file01.txt");

    myFile.open(fileName);

    if(!myFile.is_open()) {
        throw Exception("No se pudo abrir " + fileName + " para
lectura");
    }

    T myObjet;
    deleteAll();

    while((myFile >> myObjet) and myFile.ignore()) {
        insertData(myObjet);
    }

    myFile.close();
}
```

También se realizaron los diferentes tipos de recorridos que existen, el preOrder, InOrder y PostOrder, estos funcionan igual así que la explicación de cada uno se puede sobre entender explicando solo uno, por ejemplo en el PreOrder, se crea una función la cual se inicializa en la posición de la raíz, en la otra función se creara una cadena llamada “result”, y se verificara como en la mayoría de las funciones de la posición no sea nula, y con la cadena que creamos se obtendrá el dato que esta en la posición, se realizara un salto de línea y se realizara la recursividad en la izquierda y luego en la derecha, pero cada que se realiza una de las recursividades se vuelve a obtener el dato que se encuentra en la nueva posición dependiendo la raíz en la que se encuentre, de esta manera funcionan los demás, solo que como habíamos visto en la teoría cambia donde se obtiene el dato, por ejemplo en InOrder el dato se obtiene después de obtener la posición izquierda, en el PostOrder primero se realizan las recursividades de izquierda y derecha y finalmente se obtiene el dato.

```
template<class T>
string BTree<T>::parsePreOrder() {
    return parsePreOrder(root);
}

template<class T>
string BTree<T>::parsePreOrder(Position& r) {
    string result;
    if(r == nullptr) {
        return result;
    }
}
```

```

    }

    result+= r->getData().toString();
    result+= '\n';
    result+= parsePreOrder(r->getLeft());
    result+= parsePreOrder(r->getRight());
}

template<class T>
string BTree<T>::parseInOrder() {
    return parseInOrder(root);
}

template<class T>
string BTree<T>::parseInOrder(Position& r) {
    string result;
    if(r == nullptr) {
        return result;
    }

    result+= parseInOrder(r->getLeft());
    result+= r->getData().toString();
    result+= '\n';
    result+= parseInOrder(r->getRight());
}

template<class T>
string BTree<T>::parsePostOrder() {
    return parsePostOrder(root);
}

template<class T>
string BTree<T>::parsePostOrder(Position& r) {
    string result;
    if(r == nullptr) {
        return result;
    }

    result+= parsePostOrder(r->getLeft());
    result+= parsePostOrder(r->getRight());
    result+= r->getData().toString();
    result+= '\n';
}

```

Conclusion:

En este Proyecto principalmente se desarrollaron las sobrecargas para lograr realizar el manejo de archivos y también aprovechar y comparar para de esta manera encontrar los nombres de canciones, además de que se implementó la función para buscar con las sobrecargas. Las cuales funcionan

de una manera más eficiente, logre aprender en este trabajo ya que el profesor explico en clase su cómo insertar los datos en el árbol y me hizo recordar y desempolvar el funcionamiento, además de que en la investigación realice una búsqueda larga y vi un par de videos sobre las funciones de los árboles esto me hizo recordar como lo había desarrollado para tener una idea más clara, aún faltan cosas por aprender y perfeccionar. El siguiente reto será el grafo.