

# Trabalho Final da disciplina Algoritmos e Estrutura de Dados

## 2

**Leticia M. Leonel, Mirelle Silva Vieira**

IFMG - Instituto Federal de Minas Gerais - *campus* Bambuí  
Bambuí – MG – Brazil

Bacharelado em Engenharia de Computação

leticiamoreiraleonel0318@gmail.com, mirelle.sv.vieira@gmail.com

**Abstract.** *This article consists of an explication of the use of the project made for the final work of the Algorithms and Data Structure 2 discipline. It consists of exposing the libraries, implementations, tree and TADS used.*

**Resumo.** *Este artigo consiste na explicação da utilização do projeto feito para o trabalho final da disciplina de Algoritmos e Estrutura de Dados 2. Nele consiste a exposição das bibliotecas, implementações, árvores e TADS utilizados.*

## 1. INTRODUÇÃO

Código criado para cadastramento de notas de alunos de uma determinada escola.

### 1.1. Fundamentação

- TADs: Lista por Arranjo e Fila por Ponteiro;
- Bibliotecas: Criadas em Função dos TADs usados;
- Estruturas Homogêneas de dados: Laços de repetições e condições e vetores.
- Estrutura Heterogênea de dados: Structs
- Modularização e Recursividade: Funções criadas para utilização de operações e procedimentos;

### 1.2 Objetivo Geral:

O objetivo que queremos alcançar ao final do trabalho é criar um programa para auxiliar os professores de qualquer escola, a inserir notas dos alunos e receber prontamente as médias das notas de cada aluno, a média de todas as notas finais do alunos, e receber os registros cadastrais dos alunos que seriam aprovados depois dos cálculos feitos.

### 1.3 Objetivos Específicos

O programa consiste em um cadastramento e um menu. A principal funcionalidade do menu é: dar opções de operações ao usuário e dentro da operação selecionada, mostrar a resposta desejada.

## 1. DESENVOLVIMENTO

Para dar início a todo processo, o trabalho foi decidido ser feito em dupla. Primeiramente, escrevemos em uma folha de papel quais seriam as resoluções dos problemas, quais técnicas de programação que íamos usar, quais conceitos aprendidos no decorrer da disciplina seriam úteis para a resolução do problema e por fim decidimos também não dividir as tarefas e ir fazendo aos poucos. Criamos uma pasta no site do Google Drive e a cada alteração feita por alguém da dupla, adicionava um novo arquivo lá para poder ficar ciente da atualização.

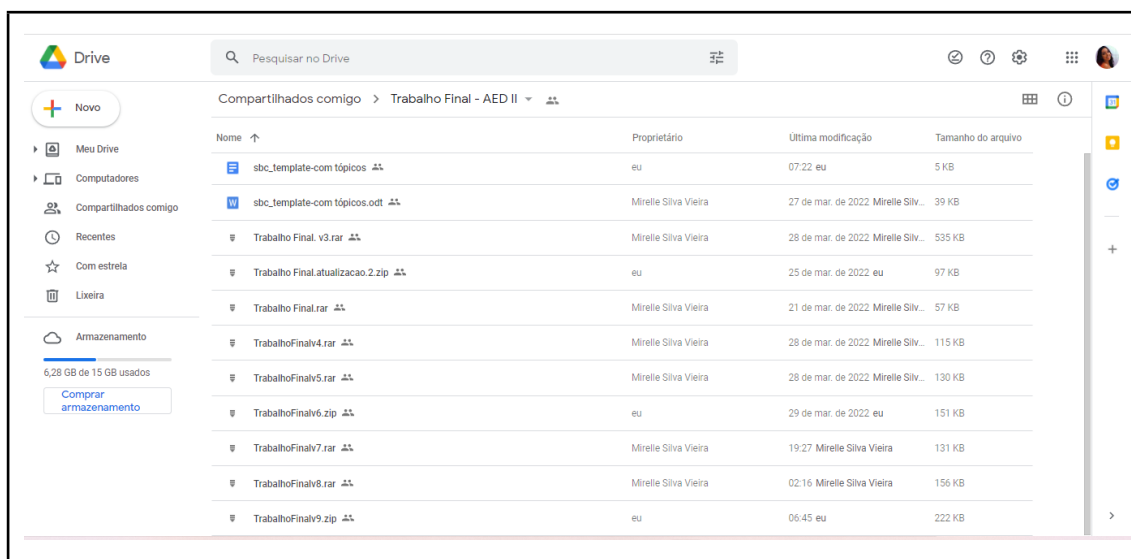


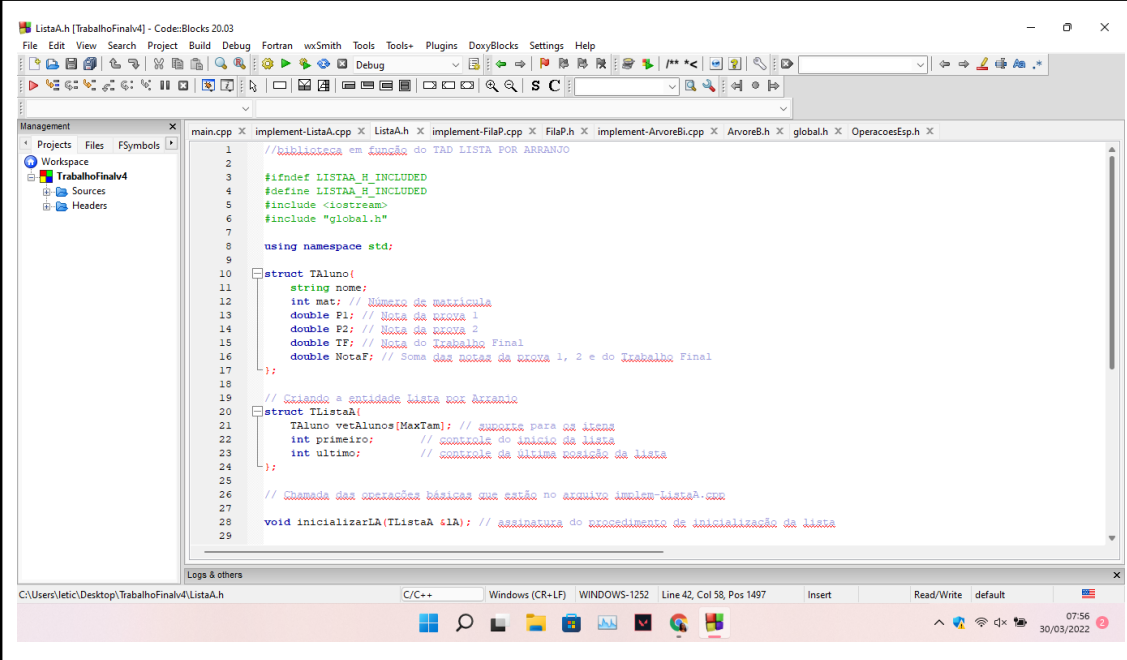
Imagem 1 - Printscreen da tela do Google drive.

- Computadores utilizados para a resolução do trabalho foram :
  - 1 - Notebook Lenovo Ideapad S145, processador Ryzen 5, 8GB RAM, SSD 240GB.
  - 2 - Notebook Multilaser Legacy PC134, processador Intel Quad Core Atom, 2GB RAM.
- Softwares utilizados:
  - 1 - Codeblocks: Criação e compilação do código.
  - 2 - Google Chrome: Pesquisas e Drive.
- Linguagem de programação:
  - 1 -A linguagem de programação aprendida na disciplina e utilizada na criação do código foi C++.
- Fontes de consulta:
  - 1 - AVA. Sistema de acesso às matérias, apostilas e listas aprendidas no curso disponibilizado pelo professor Gabriel Silva.

## 2. RESULTADOS

Decidimos utilizar as seguinte técnicas de programação para resolução do problema:

1 - TADs e Bibliotecas: Para utilização dos TADs, foi preciso criar bibliotecas no programa. Criamos uma biblioteca para usar o TAD Lista por arranjo, onde seriam inicialmente inseridos os dados cadastrais dos alunos com o objetivo de inserção de apenas 30 alunos. E uma biblioteca para utilizar o TAD Fila por ponteiro, que seria utilizado para transferência de dados e manipulação das operações. Nesta parte do código, ficam todas as assinaturas das operações e procedimentos utilizados na implementação (parte do projeto a ser explicado mais abaixo).



```
1 //biblioteca em função do TAD LISTA POR ARRANJO
2
3 #ifndef LISTA_H_INCLUDED
4 #define LISTA_H_INCLUDED
5 #include <iostream>
6 #include "global.h"
7
8 using namespace std;
9
10 struct TAluno{
11     string nome;
12     int mat; // Número de matrícula
13     double P1; // Nota da prova 1
14     double P2; // Nota da prova 2
15     double TF; // Nota do Trabalho Final
16     double NotaF; // Soma das notas da prova 1, 2 e do Trabalho Final
17 };
18
19 // Criando a entidade lista por arranjo
20 struct TListaA{
21     TAluno vetAlunos[MaxTam]; // array para as listas
22     int primeiro; // ponteiro do início da lista
23     int ultimo; // ponteiro da última noção da lista
24 };
25
26 // Chamada das operações básicas que estão no arquivo implem-ListaA.cpp
27
28 void inicializarLA(TListaA &LA); // assinatura do procedimento de inicialização da lista
29
```

Imagem 2 - Biblioteca Lista

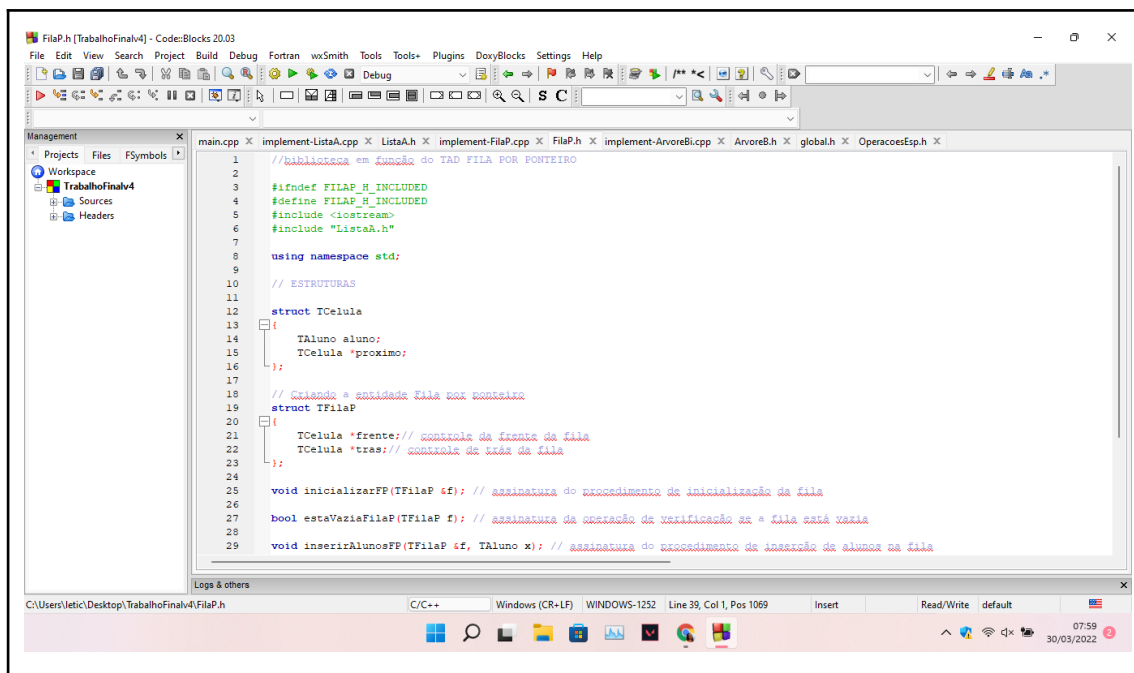


Imagem 3 - Biblioteca Fila

2 - Laços de repetição e condições: utilizados nas operações.

3 - Árvore binária: Utilizada para transferência de dados. Essa parte, os dados transferidos e retornados seriam de acordo com as notas dos alunos. Do lado esquerdo, os alunos reprovados e do lado direito, os alunos reprovados.

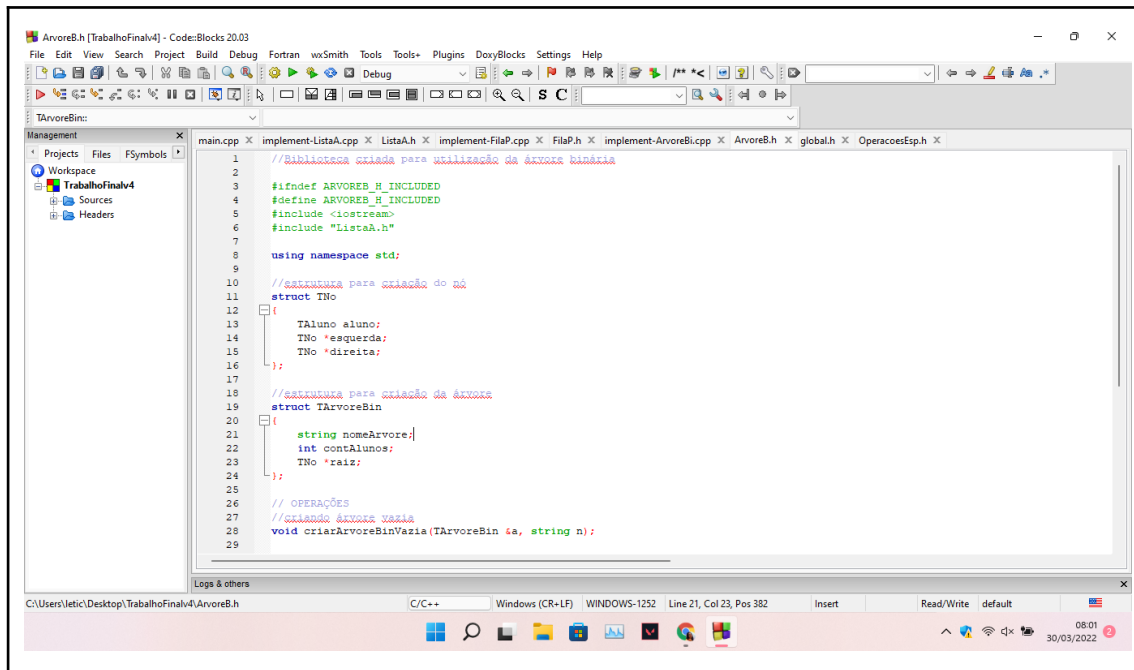
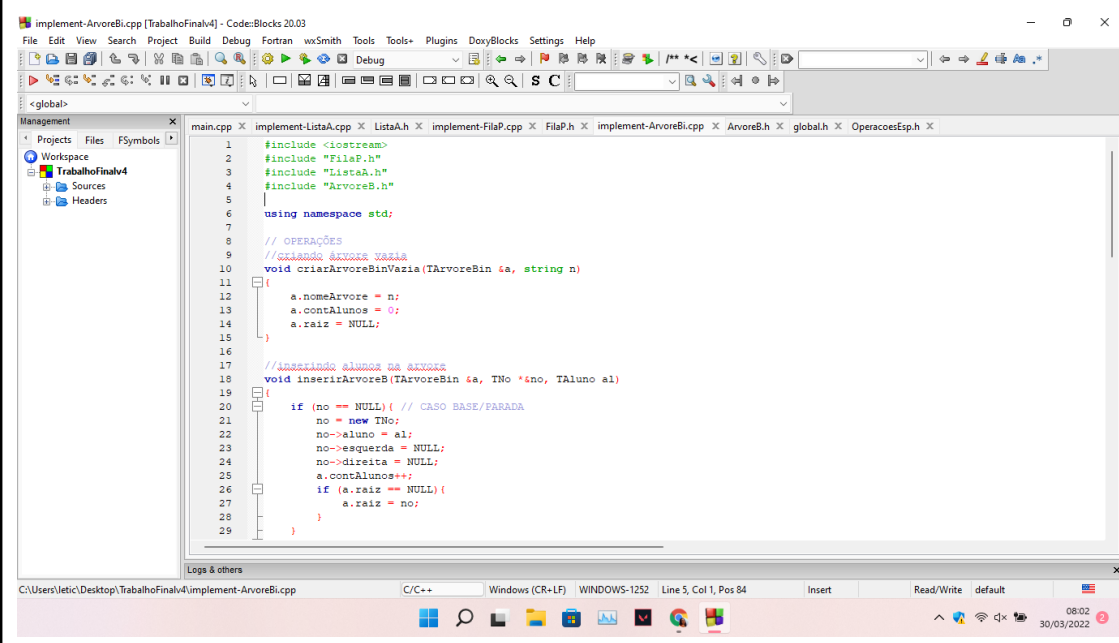


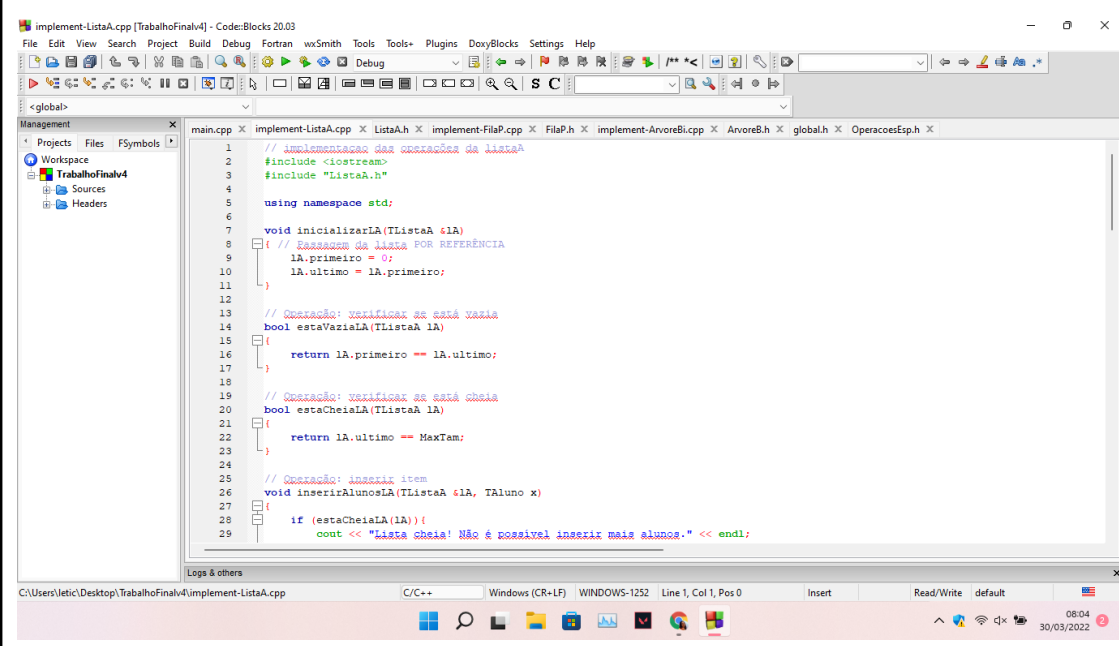
Imagem 4 - Biblioteca árvore

4 - Implementações: Foi criado um arquivo de implementação para cada biblioteca e árvore. Assim, poderíamos alterar facilmente as operações caso precisássemos. Aqui é a parte do código que ficam todas as operações e procedimentos usados de cada biblioteca.



```
1 #include <iostream>
2 #include "FileA.h"
3 #include "ListaA.h"
4 #include "ArvoreB.h"
5
6 using namespace std;
7
8 // OPERAÇÕES
9 // Criando árvore vazia
10 void criarArvoreBinVazia(TArvoreBin &a, string n)
11 {
12     a.nomeArvore = n;
13     a.contAlunos = 0;
14     a.raiz = NULL;
15 }
16
17 // Inserindo alunos na árvore
18 void inserirArvoreB(TArvoreBin &a, TNo &no, TAluno al)
19 {
20     if (no == NULL) { // CASO BASE/PARADA
21         no = new TNo;
22         no->aluno = al;
23         no->esquerda = NULL;
24         no->direita = NULL;
25         a.contAlunos++;
26         if (a.raiz == NULL) {
27             a.raiz = no;
28         }
29     }
```

Imagem 5 - Implementação árvore



```
1 // Implementação das operações da ListaA
2 #include <iostream>
3 #include "ListaA.h"
4
5 using namespace std;
6
7 void inicializarLA(TListaA &LA)
8 {
9     // Inicialização da lista POR REFERÊNCIA
10     LA.primeiro = 0;
11     LA.ultimo = LA.primeiro;
12 }
13
14 // Operação: verificar se está vazia
15 bool estaVaziaLA(TListaA LA)
16 {
17     return LA.primeiro == LA.ultimo;
18 }
19
20 // Operação: verificar se está cheia
21 bool estaCheiaLA(TListaA LA)
22 {
23     return LA.ultimo == MaxTam;
24 }
25
26 // Operação: inserir item
27 void inserirAlunosLA(TListaA &LA, TAluno x)
28 {
29     if (estaCheiaLA(LA)) {
30         cout << "Lista cheia! Não é possível inserir mais alunos." << endl;
31     }
```

Imagem 6 - Implementação Lista

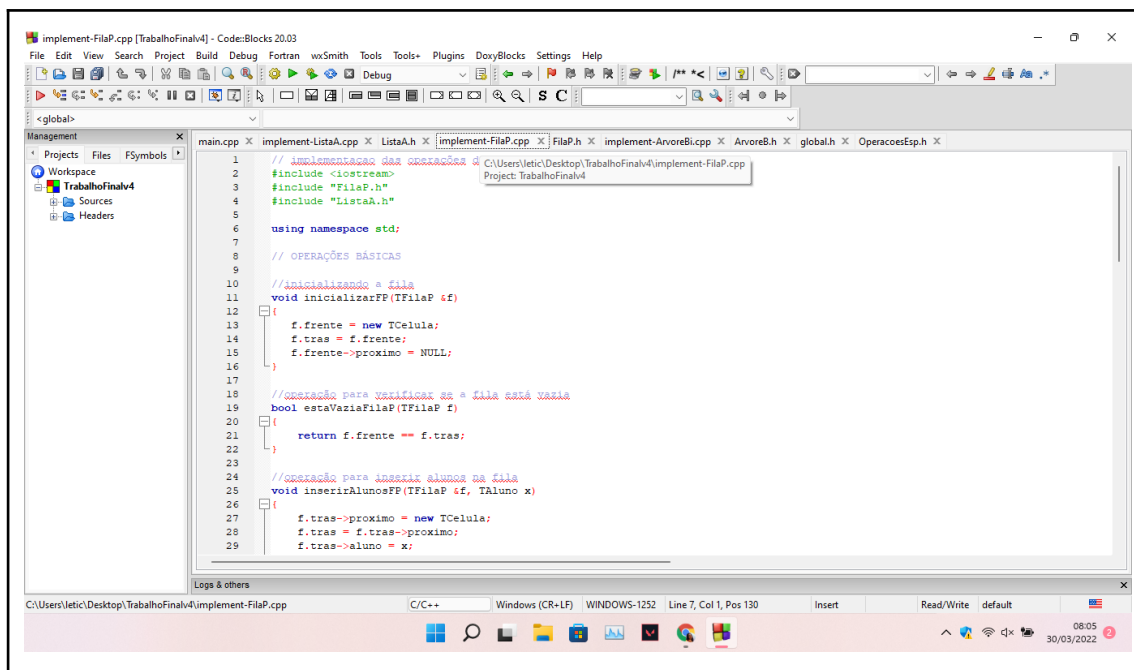


Imagem 7 - Implementação Fila

5 - Biblioteca e implementação de operações especiais: Decidimos deixar aqui, todas as assinaturas(biblioteca) e operações(implementação) que utilizamos para transferir os dados de uma Lista para a Fila, e da Fila para a árvore. Alocamos também a função do menu (próximo tópico a ser explicado).

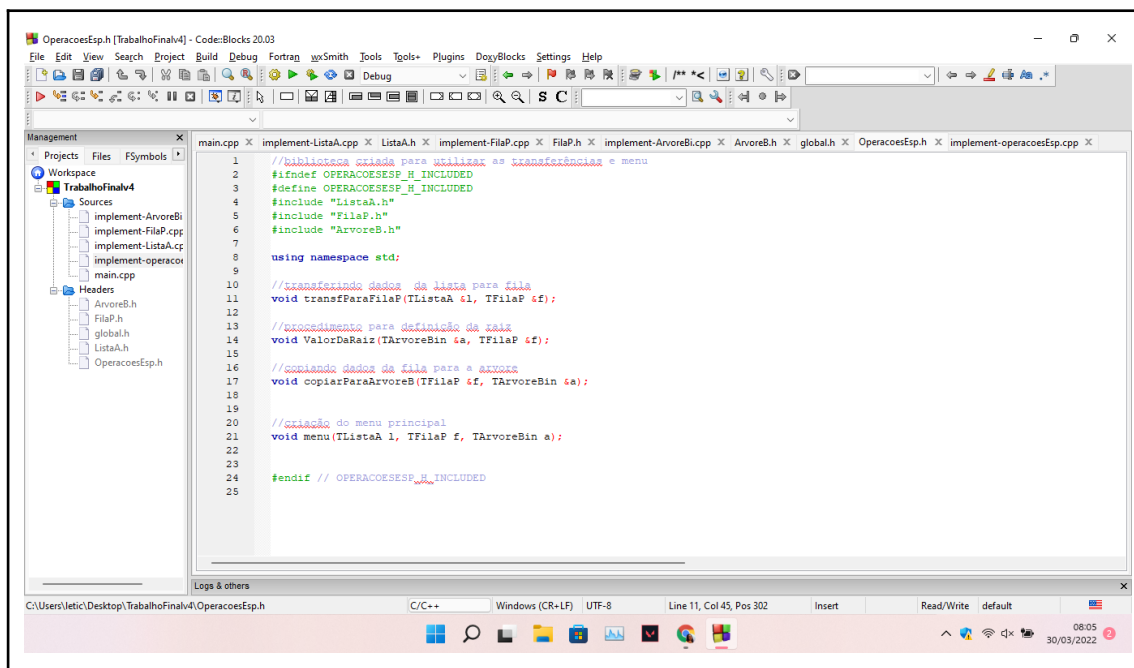


Imagem 8 - Biblioteca Operações especiais

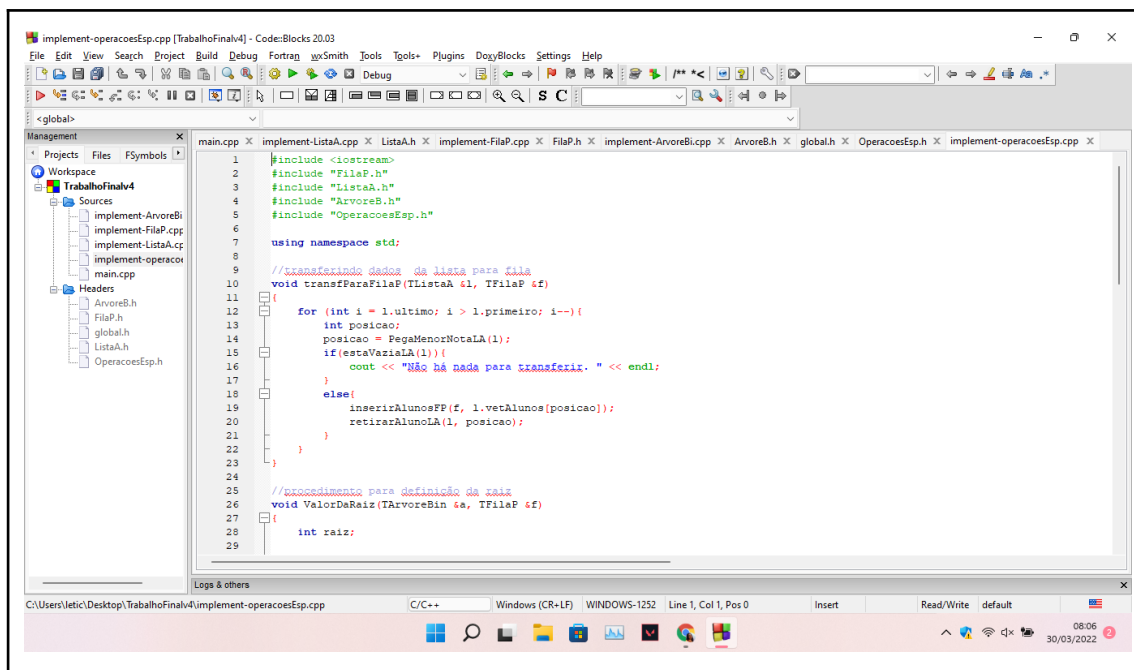


Imagem 9 - Implementação Operações especiais

6- Menu: Aqui ficariam todas as opções que os usuários poderiam escolher sobre determinadas operações.

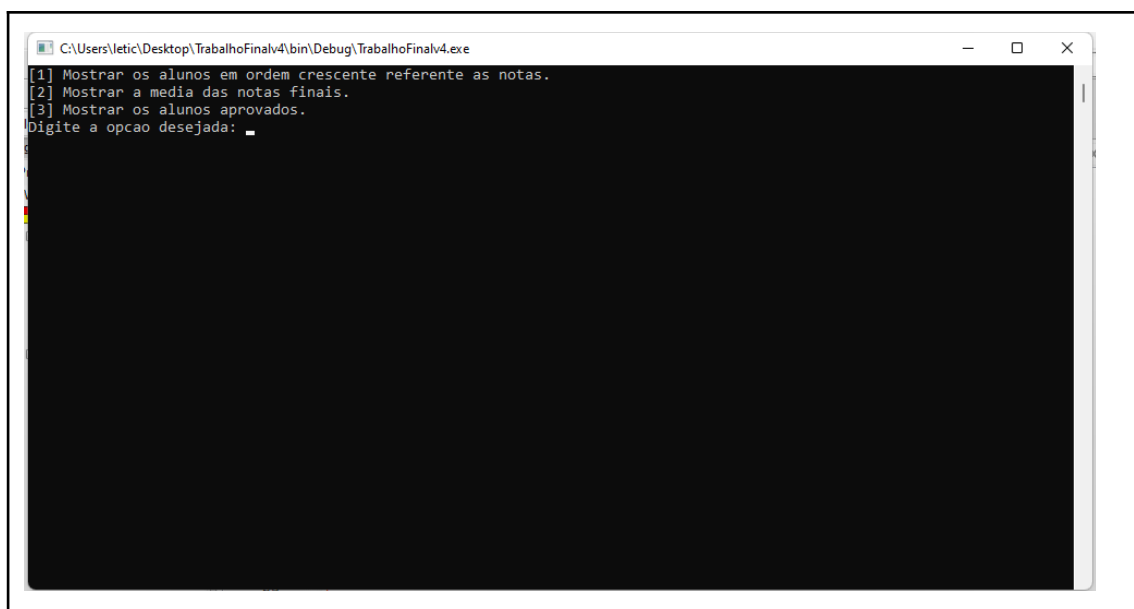


Imagem 10 - Exemplo do menu após a compilação

### 3. CONCLUSÃO

Entendemos este trabalho como uma forma de colocar em prática toda a matéria



apresentada na disciplina de Algoritmos e Estrutura de Dados 2. De certa forma, foi uma contribuição grande para o nosso aprendizado pois tivemos dificuldades.

Essas dificuldades estavam relacionadas à matéria de Árvores Binárias. Como o assunto foi “corrido”, não tinha ficado muito claro sobre a maneira de construir e manipular a árvore, e com o trabalho, pesquisamos e estudamos até encontrar a solução para este problema.

Apesar deste trabalho ter dado “trabalho”, foi de grande importância para o nosso aprendizado em relação a disciplina.

Algoritmos e estrutura de Dados 2, bem como Algoritmos e Estruturas de Dados 1, são as disciplinas bases e se não umas das mais importantes para o Curso de Engenharia de Computação. Sendo assim, a partir delas, conseguimos começar a enxergar o longo caminho que temos pela frente, pois a computação tem infinitas áreas para se aprender e não conseguiremos alcançá-las se não soubermos a base da programação.

## REFERÊNCIAS

- **Linguagem C++ - Notas de Aula Prof. Armando Luiz N. Delgado** baseado em revisão sobre material de Profa Carmem Hara e Prof. Wagner Zola - Março de 2018.
- A.G. Ana Fernanda Gomes. **Estrutura de dados: algoritmos, análise da complexidade e implementações em Java e C + +/ Ana Fernanda Ascencio, Graziela Santos de Araújo.** São Paulo: Pearson Prentice Hall, 2010.