**INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA DE MINAS GERAIS – CAMPUS SÃO JOÃO EVANGELISTA**

**BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

RHANNA RAMOS DE QUADROS

**TRABALHO PRÁTICO III**

**SÃO JOÃO EVANGELISTA**

**2022**

RHANNA RAMOS DE QUADROS

Sistema de gerenciamento de uma UPA

**SÃO JOÃO EVANGELISTA**

**2022**

SUMÁRIO

[1. INTRODUÇÃO 4](#_Toc119804164)

[1.1. Objetivo Geral 4](#_Toc119804165)

[1.2. Objetivos Específicos 4](#_Toc119804166)

[1.3. Justificativa 4](#_Toc119804167)

[2. DESENVOLVIMENTO 6](#_Toc119804168)

[2.1. Conceitos: Pilha, Lista sequencial, TADs e arquivos 6](#_Toc119804169)

[2.2. Implementação 7](#_Toc119804170)

[3. CONCLUSÃO 15](#_Toc119804171)

[4. REFERÊNCIAS 16](#_Toc119804175)

[5. APÊNDICES 17](#_Toc119804176)

[5.1. APÊNDICE A – TADs de pilha 17](#_Toc119804177)

# INTRODUÇÃO

Esta documentação tem cunho avaliativo, exigido como forma de avaliação para a disciplina de Algoritmos e Estruturas de Dados I, porém também tem cunho expositivo, onde é descrito todo o caminho percorrido para realização do trabalho prático.

## Objetivo Geral

Este trabalho tem como objetivo geral apresentar na prática os conhecimentos adquiridos nas aulas de Algoritmos e Estruturas de Dados I, a respeito de fila com ponteiro e prioridade, manipulação de tempo, utilizando a linguagem de C/C++.

## Objetivos Específicos

Esse trabalho tem como objetivos específicos:

* Aplicar conhecimentos sobre fila com ponteiro, integrado a fila de prioridade;
* Aplicar conhecimentos a respeito das bibliotecas chrono e vector.
* Aplicar e aprimorar os conhecimentos com a biblioteca time.
* Aplicar conhecimentos adquiridos, em um minissistema de delivery.

## Justificativa

O código inteiro foi escrito na linguagem C/C++, com maior foco na linguagem C++. Quando se aprende sobre Estruturas de Dados, vemos o conteúdo a respeito de Ponteiros como base para todas as estruturas que vamos utilizar ao longo do curso, aprendemos sobre o uso de bibliotecas, como a time.h, chrono e vector, sendo essas 2 primeiras para manipulação de tempo dentro do C++, e última para manipulação de vetores. Posteriormente vimos conteúdos de Estruturas de Dados mesmo, e por último foi visto o conteúdo a respeito de fila, que seria semelhante a uma pilha pela simplicidade, a fila, porém, é horizontal, e ao invés de ter o conceito de “primeiro que entra último que sai”, temos que “o primeiro que entra é o primeiro que sai”. Inicialmente nos é apresentado 3 tipos de fila, a com arranjo que é com alocação estática, ou seja, possui um limite pré-definido, podendo ter um número limitado de itens, a com ponteiro que seria a com alocação dinâmica, ou seja, não possui um limite pré-definido podendo ter um número infinito de itens, e a fila com prioridade, bem semelhante a que existe no mundo real, e pode ser aplicada tanto na fila com arranjo quanto na com ponteiro, onde inserimos itens na fila de acordo com a prioridade do item.

No trabalho irei apresentar os conceitos de fila de prioridade com ponteiro, junto às novas bibliotecas apresentadas a chrono e time.h(ambas para manipulação de tempo) e a vector(manipular vetores). Todos esses conceitos foram aplicados para realização do trabalho, e assim montar um sistema para gerir uma UPA.

# DESENVOLVIMENTO

Nesta seção do documento é apresentado o desenvolvimento do trabalho.

## Conceitos: Fila, TADs e classes

Aqui falo de forma bem superficial o que contém no meu programa.

Sobre as TADs, Tipos Abstratos de Dados, em C++ às declaramos como *structs*, ou estruturas de Dados acompanhados com funções para manipular a mesma. *Structs* são representações de qualquer coisa no mundo real, sendo ela lógica ou física, como por exemplo uma pessoa, que é algo físico, ou um filme digital, cada um com suas características específicas. Uma pessoa tem sexo, idade, nome, altura, dentre outras, e um filme tem o elenco, personagens, duração, categoria, ano de lançamento, dentre outros, e tudo isso poderia estar dentro de uma *struct* definida para cada um deles. Resumindo uma *Struct* é uma espécie de modelo para cadastrar diferentes itens, que tem características comuns, em uma lista, dentro de um software escrito em C/C++. Acompanhado das *structs* temos as funções para manipulação dos dados dessa lista, e desses itens. Juntos eles compõem o que chamamos de TAD.

Figura 1

(Fila)

Na figura 1, vemos um desenho esquemático de como seria uma fila com ponteiro. Ela é bem semelhante a um vetor, porém a tipagem de dados a ser inserida podem ser as structs que são usadas para representar coisas do mundo real, podendo armazenar mais de um tipo de variável dentro de um único “quadradinho”.

Os elementos são inseridos dentro de um espaço do vetor, ao se inserir novos elementos, eles são alocados no Ultimo, e assim sucessivamente, porém quando se trata de sair, utilizamos o conceito de FIFO - *First in, First out*, ou seja, o primeiro que entra é o primeiro que sai. Quando se trata de uma lista de prioridade, alguns serão inseridos na frente de outros, e isso muda, mas aí podemos definir que a primeira posição sempre é a primeira a sair.

Agora temos nossas bibliotecas, a chrono e time.h, para manipulação de tempo e a vector, para manipular vetores.

Chrono foi usada para...

Time.h foi usada para...

Vector foi usada para...

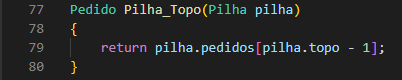
## Implementação

Na figura 3, vemos como foi feita a divisão de arquivos por modularização.

Figura 3

Foram utilizadas as TADs padrão de pilha (Apêndice A) e de lista (Apêndice B) adaptadas para o que o trabalho exigia, e funções que utilizam essas TADs. Foram adicionadas duas novas funções, apresentadas na figura 4 e figura 5, sendo elas:

Figura 4



# CONCLUSÃO

Ao longo dos trabalhos que tenho feito tenho aplicado bastante coisas que aprendi no 1° período como a modularização. Apliquei modularização separando o trabalho por arquivos, à título de organização por conselho de colegas de classe, dividindo em 6 arquivos, 1 arquivo \*.cpp para a fila e seu arquivo cabeça(\*.hpp), 1 arquivo \*.cpp para o sistema e seu arquivo cabeça(\*.hpp) e as main’s (\*.cpp e \*.exe).

Tenho dificuldade em Algoritmos e Estruturas de Dados, como sempre falo e com ajuda de colegas eu consigo desenvolver o trabalho tranquilamente, e o desenvolvimento é sempre bom, porém gostaria de melhorar isso mais, e não precisar de alguém pra ajudar, creio que até ano que vem eu aprenda bem mais. Estou satisfeita com o resultado do programa, com o funcionamento, minhas expectativas foram atendidas, é sempre bom ver o que é dito em sala de aula aplicada na prática.



# REFERÊNCIAS

TRINDADE. Eduardo. Algoritmos e Estrutura de Dados – Fila com ponteiro. 2022. Apresentação PDF. Disponível em: <https://ead.ifmg.edu.br/saojoaoevangelista/pluginfile.php/149127/mod_resource/content/1/Aula%2011%20-%20Fila%20com%20Ponteiro.pdf>. Acesso em: 18 de novembro de 2022.

TRINDADE. Eduardo. Algoritmos e Estrutura de Dados – Data, hora e tempo em C++. 2022. Apresentação PDF. Disponível em: <https://ead.ifmg.edu.br/saojoaoevangelista/pluginfile.php/149625/mod_resource/content/1/Aula%2013%20-%20Data%2C%20hora%20e%20tempo%20em%20C%2B%2B.pdf> . Acesso em: 18 de novembro de 2022.

TRINDADE. Eduardo. Algoritmos e Estrutura de Dados – Classe Vector. 2022. Apresentação PDF. Disponível em: <https://ead.ifmg.edu.br/saojoaoevangelista/pluginfile.php/149748/mod_resource/content/2/Aula%2014%20-%20Classe%20Vector.pdf> . Acesso em: 18 de novembro de 2022.

Meu código no GitHub: <https://github.com/RhannaQuadros/Algoritmos-e-Estruturas-de-Dados-I->

# APÊNDICES

* 1. APÊNDICE A – TADs de pilha