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

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

JOÃO VITOR MENDES CAMPOS

GABRIEL KÁICON BATISTA HILÁRIO

**TRABALHO PRÁTICO II**

SÃO JOÃO EVANGELISTA

2023

JOÃO VITOR MENDES CAMPOS

GABRIEL KÁICON BATISTA HILÁRIO

SISTEMA DE RH DE UMA EMPRESA

SÃO JOÃO EVANGELISTA

2023

SUMÁRIO

[1. INTRODUÇÃO 5](#_Toc134541714)

[1.1. Objetivo Geral 5](#_Toc134541715)

[1.2. Objetivos Específicos 5](#_Toc134541716)

[1.3. Justificativa 5](#_Toc134541717)

[2. DESENVOLVIMENTO 6](#_Toc134541718)

[2.1. Árvore Binária de Busca 6](#_Toc134541719)

[2.2. Implementação 6](#_Toc134541720)

[3. CONCLUSÃO 9](#_Toc134541721)

[4. REFERÊNCIAS 10](#_Toc134541722)

[5. APÊNDICES 11](#_Toc134541723)

[APÊNDICE A – (Cria No) 11](#_Toc134541724)

[APÊNDICE B – (Insere No) 11](#_Toc134541725)

[APÊNDICE C – (Removê No) 12](#_Toc134541726)

[APÊNDICE D – (Cria Pessoa) 13](#_Toc134541727)

[APÊNDICE E – (Imprime Item) 14](#_Toc134541728)

[APÊNDICE F – (Busca Matrícula) 14](#_Toc134541729)

[APÊNDICE G - (Busca CPF) 14](#_Toc134541730)

[APÊNDICE H - (Busca Nome) 15](#_Toc134541731)

[APÊNDICE I - (Remove Matrícula) 15](#_Toc134541732)

[APÊNDICE J - (Imprime Ordem) 15](#_Toc134541733)

[APÊNDICE K - (Imprime Pré-Ordem) 15](#_Toc134541734)

[APÊNDICE L - (Imprime Pós-Ordem) 16](#_Toc134541735)

[APÊNDICE M - (Busca) 16](#_Toc134541736)

[APÊNDICE N - (Imprime) 16](#_Toc134541737)

[APÊNDICE O - (Menu Busca) 17](#_Toc134541738)

[APÊNDICE P - (Menu) 17](#_Toc134541739)

[APÊNDICE Q - (Menu Imprimir) 18](#_Toc134541740)

[APÊNDICE R - (TAD) 18](#_Toc134541741)

# INTRODUÇÃO

Este trabalho prático foi documentado para que seja avaliado em conjunto com os códigos na linguagem C/C++, exigido pelo docente Eduardo Augusto da Costa Trindade, dentro da disciplina de Algoritmos e Estruturas de Dados II, ministrada pelo mesmo. Porém a documentação tem cunho expositivo, onde é descrito as funcionalidades do programa, com testes, e desenvolvimento de novas linhas de raciocínio lógico 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 II, a respeito das Árvores Binária de Busca (BST), utilizando a linguagem C++ para escrita dos códigos.

## Objetivos Específicos

Esse trabalho tem como objetivos específicos:

* Fixar conceitos sobre manipulação de árvores binárias.
* Realizar a implementação de árvore binária e suas TAD’S
* Compreender os métodos de inserção, busca e remoção em uma árvore.
* Estimular o raciocínio lógico para a resolução de problemas.

## Justificativa

Ao iniciar os estudos de Algoritmos e Estruturas de Dados II, vemos os conteúdos Análise de Complexidade, Custo da função, Big O, Recursividade, Divisão e conquista Métodos de Ordenação, Árvore Binária de Busca como uma nova Estrutura de Dados, até então desconhecida, porém o que nos interessa nesse momento é a Árvore Binária de Busca (BST).

O objetivo principal de uma BST é permitir uma busca eficiente de dados, a recuperação rápida e eficiente de elementos com base em uma chave de pesquisa, geralmente uma chave numérica ou alfabética. Essa estrutura de dados é ótima para desenvolver algoritmos com bom desempenho simultâneo para a busca e inserção, tendo em vista que a mesma utiliza recursividade. Além disso, as árvores binárias de busca podem ser usadas para realizar outras operações como: remoção, mínimo, máximo, antecessor e sucessor. O trabalho foi exigido para que seja possível desenvolver o raciocínio lógico quanto a aplicação dessa estrutura, e com o bônus de novamente aplicá-la em conjunto.

# DESENVOLVIMENTO

Nesta seção do documento é apresentado, os conceitos aprendidos e o desenvolvimento do trabalho em si, na linguagem C++.

## Árvore Binária de Busca

A propriedade fundamental que define uma Árvore Binária de Busca é que, para cada nó, todos os valores nos filhos à esquerda do nó são menores que o valor do próprio nó, e todos os valores nos filhos à direita do nó são maiores. Isso permite uma busca binária eficiente, em que a busca é iniciada a partir do nó raiz, e o valor da chave é comparado com o valor do nó atual. Dependendo do resultado da comparação, a busca continua na sub-árvore à esquerda ou sub-árvore à direita.

## Implementação

A modularização feita foi a main.cpp, BST.hpp, BST.cpp e um main.exe. Na parte de implementação foram utilizadas as funções fornecidas pelo professor.

**Apêndice A** - Nele é apresentada a função (Cria No) o objetivo dessa função é criar um nó na árvore, essa função é do tipo No e ao final de sua execução retornará um novo nó.

**Apêndice B** - Nele é apresentada a função (Insere No) o objetivo dessa função é inserir um nó na árvore, essa função é do tipo No, ela verifica se já existe uma raiz, se a raiz existe ela continua a executar normalmente, mas se a raiz não existir a função criará essa raiz, seguindo a função verifica se a o objeto a ser inserido é maior que a raiz se assim ocorrer esse objeto será inserido à direita da raiz, mas se o objeto for menor que a raiz será inserida à esquerda da mesma, após o término da execução é retornado a raiz.

**Apêndice C** - Nele é apresentada a função (Remove No) o objetivo é remover um nó, essa função é do tipo No, ela verifica a existência da raiz se ela existir o programa continua a ser executado, mas se a raiz não existir será retornada a raiz, seguindo a verificado se o objeto a ser removido é menor que a raiz assim removendo o objeto da parte esquerda da árvore, mas se o objeto a ser removido é maior que a raiz o objeto será removido da parte direita da árvore, ao final será retornado a raiz.

**Apêndice D** - Nele é apresentada a função (Cria Pessoa) o objetivo é criar o cadastro da pessoa, essa função é do tipo Pessoa, nela o usuário vai passar os dados da pessoa a serem cadastrada, alterando todos os campos da mesma ao final será retornado um “objeto” pessoa para ser inserida na árvore.

**Apêndice E** - Nele é apresentada a função (Imprime Item) o objetivo é imprimir os dados de uma pessoa já cadastrada, essa função é do tipo Void, nela serão imprimidos os dados de uma pessoa, para fazermos uso do reaproveitamento de código, já que sempre vamos pedir para imprimir em diferentes funções.

**Apêndice F** - Nele é apresentada a função (Busca Matrícula) o objetivo é buscar por meio da matrícula uma pessoa, essa função é do tipo No, nela o usuário passa o número de matrícula da pessoa desejada, verifica- se a raiz da árvore existe ou se o número de matrícula é foi encontrado, se assim ocorrer será imprimindo os dados da pessoa e retornado a raiz, mas se isso não ocorrer, será verificado se a número de matrícula escolhido pelo usuário é menor que os números da matrícula contidos na árvore, se assim acontecer a função será executada novamente só que agora levando em conta a esquerda e direita da raiz como parâmetros.

**Apêndice G** - Nele é apresentada a função (Busca CPF) o objetivo é buscar por meio do CPF da pessoa, essa função é do tipo No, nela o usuário passa o CPF da pessoa desejada, verifica- se a raiz da árvore existe ou se o CPF foi encontrado, se assim ocorrer será imprimindo os dados da pessoa e retornado a raiz, mas se isso não ocorrer, será verificado se o CPF escolhido pelo usuário é diferente dos CPFs contidos na árvore, se assim acontecer a função será executada novamente só que agora levando em conta a esquerda e direita da raiz como parâmetros.

**Apêndice H** - Nele é apresentada a função (Busca Nome) o objetivo é buscar por meio do nome da pessoa, essa função é do tipo No, nela o usuário passa o nome da pessoa desejada, verifica- se a raiz da árvore existe ou se o nome foi encontrado, se assim ocorrer será imprimindo os dados da pessoa e retornado a raiz, mas se isso não ocorrer, será verificado se o nome escolhido pelo usuário é diferente dos nomes contidos na árvore, se assim acontecer a função será executada novamente só que agora levando em conta a esquerda e direita da raiz como parâmetros.

**Apêndice I** - Nele é apresentada a função (Remove Matrícula) o objetivo é remover o cadastro de uma pessoa por meio do número de matrícula da pessoa que o usuário escolheu, essa função é do tipo Void, nela o usuário informa o número da matrícula da pessoa a ser removida, a função (Remove No) recebe o número de matrícula e remove o cadastro da árvore.

**Apêndice J** - Nele é apresentada a função (Imprime Ordem) o objetivo é imprimir a árvore em ordem, que seria a impressão começando da sub-árvore à esquerda, raiz e depois a sub-árvore à direita. Essa função é do tipo Void.

**Apêndice K** - Nele é apresentada a função (Imprime Pré-Ordem) o objetivo é imprimir a árvore em pré-ordem, que seria a impressão partindo da Raiz, depois as árvores e/ou sub-árvores à esquerda e depois e/ou sub-árvores à direita. Essa função é do tipo Void.

**Apêndice L** - Nele é apresentada a função (Imprime Pós-Ordem) o objetivo é imprimir a árvore em pós-ordem, isso seria a impressão começando a partir da sub-árvore esquerda, depois a sub-árvore direita, e por fim a raiz. Essa função é do tipo Void.

**Apêndice M** - Nele é apresentada a função (Busca) o objetivo é buscar os dados de uma pessoa na árvore, essa função é do tipo Void, nela o usuário escolhe se a busca será por CPF, Matrícula ou Nome, se selecionado CPF será executado a função (Busca CPF), se selecionado Nome será executado a função (Busca Nome) e se selecionado Matrícula será executado a função (Busca Matrícula), assim retornando os dados da pessoa.

**Apêndice N** - Nele é apresentada a função (Imprime) o objetivo é imprimir a árvore a partir do modo escolhido pelo usuário, essa função é do tipo Void, nela o usuário escolhe se a impressão da árvore será em Ordem, Pré-Ordem ou Pós-ordem, após a escolha a árvore será impressa segundo o modo selecionado.

**Apêndice O** - Nele é apresentado a função (Menu Busca) o objetivo é imprimir um menu de busca, essa função é do tipo Int, neste menu estão os modos de busca que podem ser escolhidos, após a escolha do modo de busca será retornada a opção escolhida.

**Apêndice P** - Nele é apresentado a função (Menu) o objetivo é imprimir o menu principal, essa função é do tipo Void, será apresentado na tela o menu com as opções 1- Cadastrar Funcionário, 2- Buscar Funcionário, 3- Remover Funcionário, 4- Imprimir, 5- Sair e opção:.

**Apêndice Q** - Nele é apresentada a função (Menu Imprimir) o objetivo é imprimir um menu dos modos de impressão, essa função é do tipo Int, nela é apresentado os modos de impressão, é pedido que o usuário escolha qual modo de impressão ele deseja.

**Apêndice R** - Nele é apresentada a TAD escolhida para a execução do trabalho.

# CONCLUSÃO

Ao longo do trabalho tivemos várias ideias de como realizar a implementação do trabalho, já que a Árvore Binária de Busca pode ser feita de várias formas. Utilizamos os códigos padrões da criação de árvores, e damos um foco maior nas funções de inserção, remoção, impressão, e busca por diferentes tipos de dados, também fizemos uso dos slides disponibilizados pelo professor para sanar dúvidas a respeito, e usuários externos para tirar dúvidas sobre a criação e uso de novas funções. Creio que o desenvolvimento foi bem satisfatório nesse trabalho, estávamos bem receosos com o código, depois vimos a simplicidade no desenvolvimento. Nós desenvolvemos novas formas de raciocínio e tivemos que abstrair e diminuir muitas coisas como de costume, para que assim o trabalho ficasse leve, e modularizado, sem ser muito extenso. Tivemos que buscar ajuda externa e conhecimento externo para realização do trabalho, mas mesmo assim conseguimos atingir nossos objetivos estipulados no início do trabalho, explicar bem o que estava fazendo dentro de cada função, apresentar conhecimentos em Árvore Binária de Busca, e com confiança aprimorar nossos conhecimentos. Estamos felizes e satisfeitos com o resultado, e ele atendeu às nossas expectativas além do que era esperado.

# REFERÊNCIAS

TRINDADE. Eduardo. Algoritmos e Estrutura de Dados II – ÁRVORE BINÁRIA DE BUSCA (BST). 2023. Apresentação PDF. Disponível em: [https://ead.ifmg.edu.br/sje-presencial/pluginfile.php/158749/mod\_resource/content/1/Aula 13 - Árvore Binária de Busca (BST).pdf](https://ead.ifmg.edu.br/sje-presencial/pluginfile.php/158749/mod_resource/content/1/Aula%2013%20-%20%C3%81rvore%20Bin%C3%A1ria%20de%20Busca%20(BST).pdf) . Acesso em: 08 de maio de 2023.

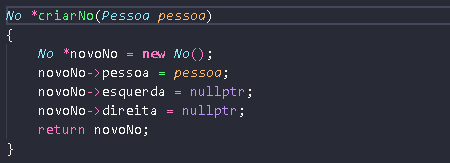
TRINDADE. Eduardo. Algoritmos e Estrutura de Dados II – ÁRVORES. 2023. Apresentação PDF. Disponível em: <https://ead.ifmg.edu.br/sje-presencial/pluginfile.php/158748/mod_resource/content/1/Aula%2012%20-%20%C3%81rvores.pdf> . Acesso em: 08 de maio de 2023.

Link no github <https://github.com/gKaicon/Arquivos_C_and_C-withClasses/tree/main/AEDs%20II%2FTP_%C3%81rvoreBuscaBin%C3%A1ria>

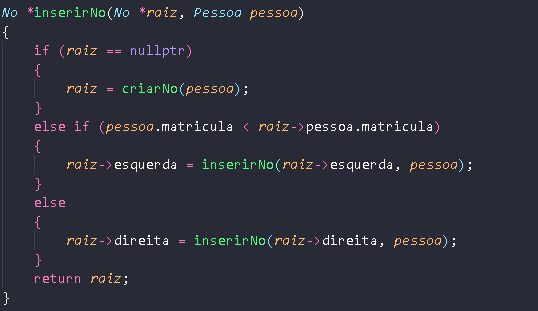
# 

# APÊNDICES

## APÊNDICE A – (Cria No)

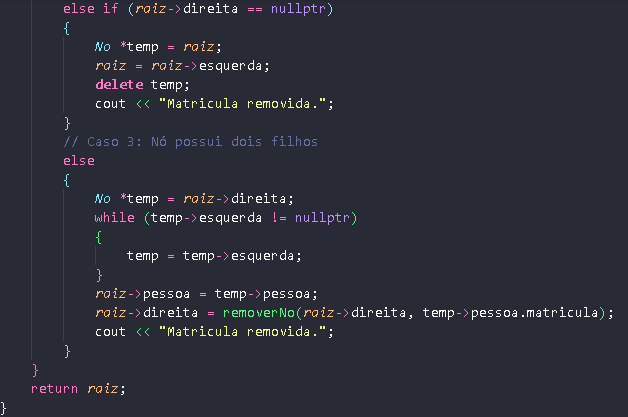


## APÊNDICE B – (Insere No)

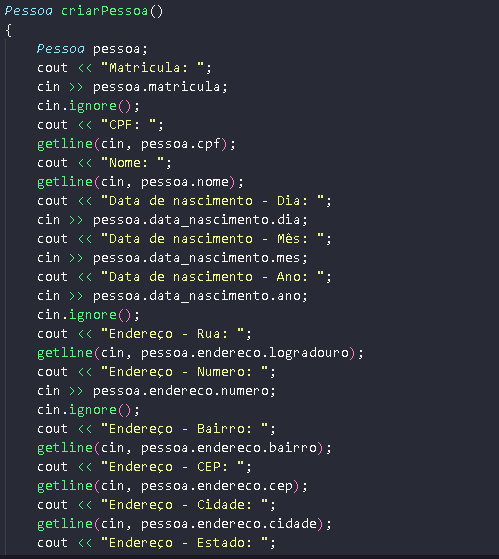
****

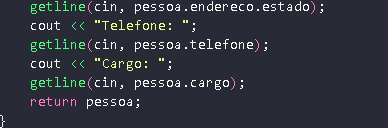
## APÊNDICE C – (Removê No)



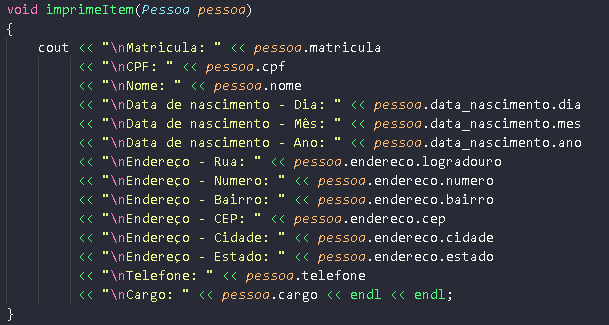


## APÊNDICE D – (Cria Pessoa)

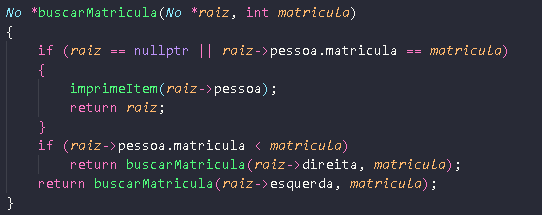




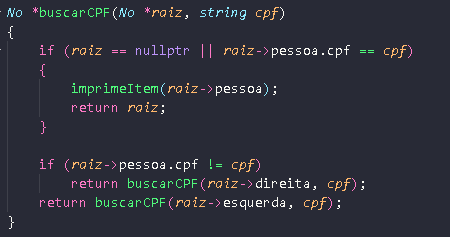
## APÊNDICE E – (Imprime Item)



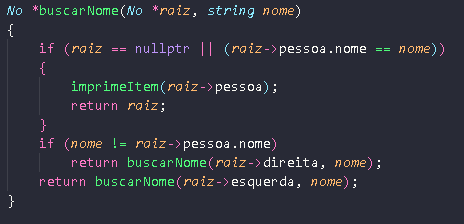
## APÊNDICE F – (Busca Matrícula)



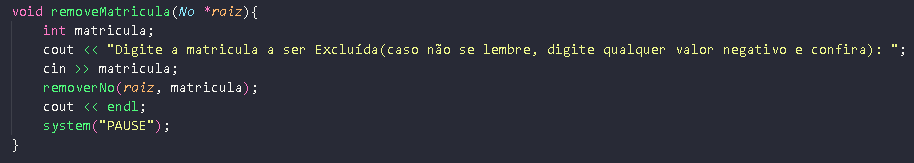
## APÊNDICE G - (Busca CPF)



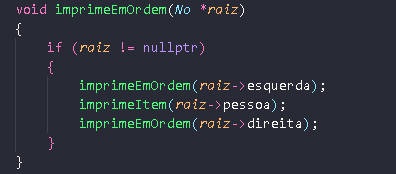
## APÊNDICE H - (Busca Nome)

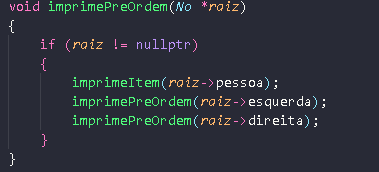


## APÊNDICE I - (Remove Matrícula)

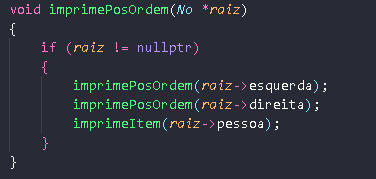


## APÊNDICE J - (Imprime Ordem)

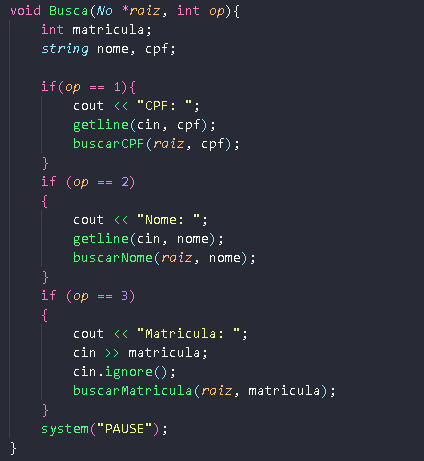


APÊNDICE K - (Imprime Pré-Ordem)

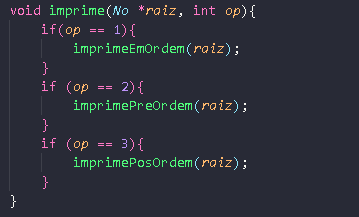
## APÊNDICE L - (Imprime Pós-Ordem)



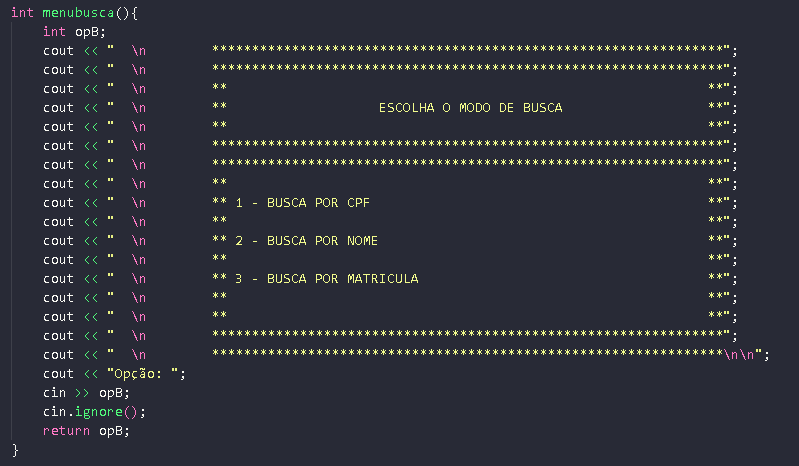
## APÊNDICE M - (Busca)



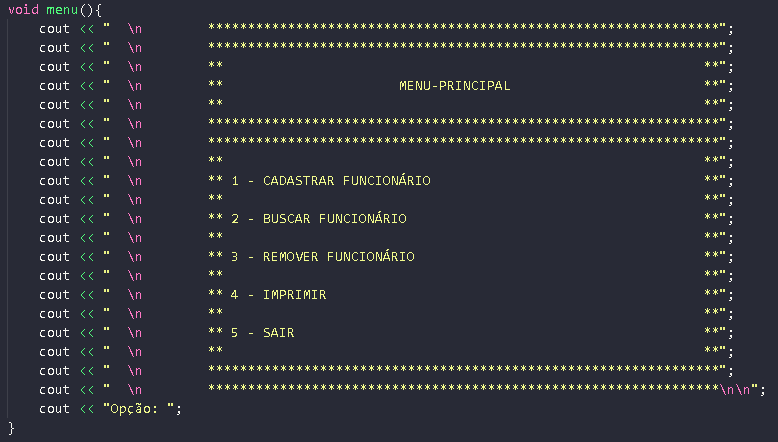
## APÊNDICE N - (Imprime)



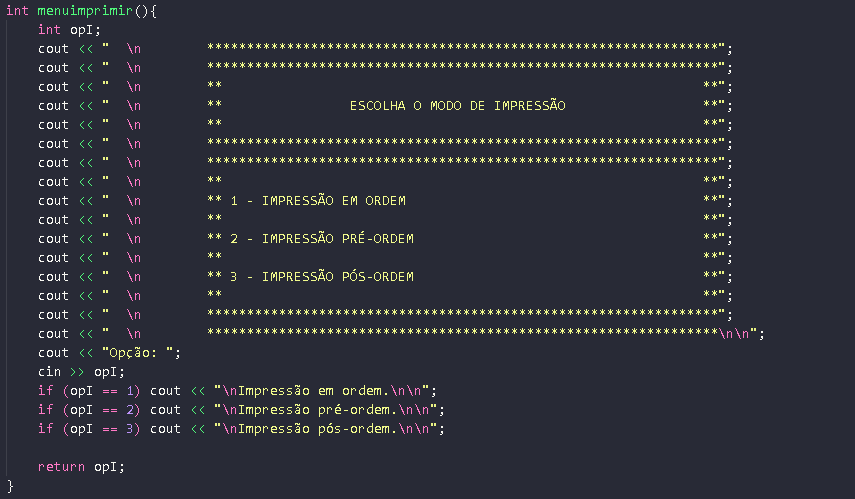
## APÊNDICE O - (Menu Busca)



## APÊNDICE P - (Menu)



## APÊNDICE Q - (Menu Imprimir)



## APÊNDICE R - (TAD)

