# UNIVERSIDADE FEDERAL DE OURO PRETO CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**CRISTIAN ANSELMO DE SOUZA PINTO - 21.2.4005 PEDRO AUGUSTO CARNEIRO DE OLIVEIRA - 21.2.4119 SAMUEL GONÇALVES REIS - 21.2.4092**

# RELATÓRIO DO TRABALHO PRÁTICO 2 DE ESTRUTURA DE DADOS I BCC-202

**OURO PRETO 2022**

# Introdução:

Este documento é o relatório do terceiro trabalho prático da matéria Estrutura de Dados I que busca expor a implementação, impressões gerais, análise e conclusão de um código criado para através de uma pesquisa binária, encontrar uma embarcação entre as digitadas pelo usuário e retornar se foi encontrada a embarcação na posição digitada, senão como alcança-la.

# Implementação

Primeiramente é feita a leitura de C e M que indica quantos tabuleiros serão lidos e quantos dados serão digitados em cada linha (sendo 8 embarcações, índice da posição, embarcação desejada).

Após a leitura é processado cada linha que o usuário digitou e fazendo as verificações necessárias. Nessa hora que entra a busca binária, onde o vetor deve estar ordenado, ele será dividido e irá perguntar se a posição desejada está maior ou menor que a posição do meio, a partir dessa resposta o vetor passa a ter seu tamanho reduzido e isso é perguntado até que ache a chave desejada e é contada quantas divisões foram necessárias.

A função Find retorna se foi encontrada ou não a embarcação.

A função TDcionario\_Binario está contando quantas divisões são necessárias para encontrar a embarcação.

A função mensagem recebe os dados processados para exibir a resposta, onde é preciso imprimir se a embarcação foi encontrada e se precisa de mais divisoẽs para encontra-la ou se a posição digitada foi correta.

Se a embarcação não existe entra na função não existe e é impresso o tipo da embarcação é informa que não foi encontrada.

Após exibir todas as informações é desalocada a memória do programa.

# Análise:

O código apresenta complexidade de tempo O(log n) para realizar a busca binaria recursiva, já que usa o método divisão e conquista, e O(n) para procurar se a chave é existente no vetor, portanto como a função linear domina a função logaritmo a complexidade é O(n). Um detalhe importante é que a complexidade em relação ao uso de memoria da busca binaria recursiva, mais simples de implementar, é logn enquanto a busca binaria iterativa é constante, o que deve ser levado em consideração na escolha do método.

# Impressões Gerais:

A estruturação para a escrita do código foi baseada nos slides da disciplina, foi um código que não apresentou dificuldade, devido a busca binária praticamente retornar tudo que era necessário para o programa, precisando desenvolver logicamente somente na questão do que deveria ser impresso de acordo com a entrada.

# Conclusão:

Ao finalizar o nosso trabalho nos sentimos satisfeitos pelo resultado obtido e por percebermos que conseguimos compreender a matéria introduzida e visto ela em prática por meio desse TP, pudemos utilizar o método de pesquisa binária, ponteiro, alocação dinâmica, e exercer a análise de complexidade do código. Com isso, concluímos que foi efetivo o empenho do nosso trabalho para a fixação de nosso conhecimento.