

**Nome:** Matheus Goulart Ranzani  
**RA:** 800278

## Busca Sequencial e Busca Binária

### Código do arquivo buscas.h

```
#define TAMANHO 50

// Se encontrou o elemento x, retorna sua posição
int busca_sequencial(int vetor_ordenado[], int x) {
    for (int i = 0; i < TAMANHO; i++) {
        if (vetor_ordenado[i] == x) {
            return i;
        }
    }
}

// Função auxiliar para realizar a Busca Binária
static int _busca_binaria(int vetor_ordenado[], int esq, int dir, int x) {
    if (esq == dir - 1) {
        return dir;
    } else {
        int meio = (esq + dir) / 2;

        if (vetor_ordenado[meio] < x) {
            return _busca_binaria(vetor_ordenado, meio, dir, x);
        } else {
            return _busca_binaria(vetor_ordenado, esq, meio, x);
        }
    }
}

// Se encontrou o elemento x, retorna sua posição
int busca_binaria(int vetor_ordenado[], int x) {
    return _busca_binaria(vetor_ordenado, -1, TAMANHO, x);
}
```

## Código do arquivo usa\_buscas.c

```
#include "buscas.h"

#include <stdio.h>

void inicia_vetor_ordenado(int vetor[]) {
    for (int i = 0; i < TAMANHO; i++) {
        vetor[i] = i + 1;
    }
}

void imprime_vetor(int vetor[]) {
    printf("Vetor ordenado com %d elementos:\n", TAMANHO);
    for (int i = 0; i < TAMANHO; i++) {
        printf("%d ", vetor[i]);
    }
}

int main() {
    int vetor[TAMANHO];

    inicia_vetor_ordenado(vetor);
    imprime_vetor(vetor);

    int posicao1 = busca_sequencial(vetor, 38);
    printf("\n\nElemento 38 foi encontrado na posicao %d\n", posicao1);

    int posicao2 = busca_binaria(vetor, 45);
    printf("Elemento 45 foi encontrado na na posicao %d", posicao2);

    printf("\n");

    return 0;
}
```

## Comparação de desempenho

Tanto a busca sequencial quanto a busca binária são capazes de encontrar elementos solicitados em um vetor ordenado. Entretanto, na busca sequencial, o pior caso (quando o elemento procurado for o último do vetor) pode demandar muito processamento. Assim, a busca sequencial é da ordem de  **$O(n)$** . Já na busca binária, no pior caso, não é necessário percorrer o vetor inteiro, trazendo assim, mais desempenho. Dessa maneira, a busca binária é da ordem de  **$O(\log(n))$** .

## Imagem da execução do programa

```
C:\Users\ranza\OneDrive\Área de Trabalho\Code\AED1\Frequências\F10>gcc usa_busca
s.c -o usa_buscas

C:\Users\ranza\OneDrive\Área de Trabalho\Code\AED1\Frequências\F10>usa_buscas
Vetor ordenado com 50 elementos:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

Elemento 38 foi encontrado na posicao 37
Elemento 45 foi encontrado na na posicao 44
```