

**UNIVERSIDADE SÃO JUDAS TADEU**

**SISTEMAS DE INFORMAÇÃO**

**GESTÃO E QUALIDADE DE SOFTWARE**

**JOÃO VITOR GOMES PEREIRA**

**82329432**

**ATIVIDADE DE AULA**

**PROF.º DOCENTE – ROBSON CALVETTI**

**SÃO PAULO – SP**

**2025**

## ***SUMÁRIO***

EXERCICIO DE EXEMPLO .....	3
APLICANDO BDD .....	4
APLICANDO TDD .....	5

## EXERCICIO DE EXEMPLO

```
public static int busca_binaria(int iVet[], int iK)
{
    int iBaixo, iAlto, iMeio;

    iBaixo=0;
    iAlto=iVet.length-1;
    while(iBaixo <= iAlto)
    {
        iMeio=(iBaixo+iAlto)/2;
        if(iK < iVet[iMeio]) iAlto=iMeio-1;
        else if(iK > iVet[iMeio]) iBaixo=iMeio+1;
        else return iMeio;
    }
    return -1;
}
```

# APLICANDO BDD

## 1- Encontrar um elemento existente

Given: um vetor [1, 3, 5, 7, 9],

When: eu buscar pelo número 7,

Then: o sistema deve retornar o índice 3.

## 2- Não encontrar um elemento inexistente

Given: um vetor [2, 4, 6, 8, 10],

When: eu buscar pelo número 5,

Then: o sistema deve retornar -1, indicando que o número não está presente.

## 3- Buscar o primeiro elemento do vetor

Given: um vetor [10, 20, 30],

When: eu buscar pelo número 10,

Then: o sistema deve retornar o índice 0.

## 4- Buscar o último elemento do vetor

Given: um vetor [1, 2, 3, 4, 5],

When: eu buscar pelo número 5,

Then: o sistema deve retornar o índice 4.

## 5- Buscar em vetor vazio

Given: um vetor vazio [ ],

When: eu buscar por qualquer número,

Then: o sistema deve retornar -1, indicando que não há elementos no vetor.

# APLICANDO TDD

## 1- RED - (Criando os Casos de Testes)

- Teste 1: Buscar um número que está no meio do vetor.
- Esperado: Retornar o índice correto.
- Teste 2: Buscar um número que não existe no vetor.
- Esperado: Retornar -1.
- Teste 3: Buscar o primeiro número do vetor.
- Esperado: Retornar o índice 0.
- Teste 4: Buscar o último número do vetor.
- Esperado: Retornar o último índice.
- Teste 5: Buscar um número em um vetor vazio.
- Esperado: Retornar -1.

## 2- GREEN - (Implantando a funcionalidade)

Após os testes estarem definidos, a função de busca binária foi implementada para atender aos critérios estabelecidos nos testes. A lógica usada percorre o vetor dividindo-o pela metade até encontrar (ou não) o número desejado.

## 3- REFACTOR - (Refatorando o Código)

- Renomeação de variáveis para tornar o código mais legível;
- Inserção de comentários para melhor entendimento da lógica;
- Separação da lógica de busca em métodos distintos, se necessário;
- Adição de validações (ex: tratamento de vetor nulo ou vazio).