



Universidade São Judas Tadeu

Ciências da Computação

RELATÓRIO DE TESTE – CAIXA BRANCA

Victor Ignacio - 823125249

Kaue Brito Vieira - 824211851

Giulia Gabriella - 823122979

Kauê Dib de Souza Dias - 823149871

Vinícius Santana Teixeira - 82319112

Murilo Bonuccelli de Oliveira – 823148988

1. Introdução

O teste de caixa branca, também conhecido como teste estrutural, foca na lógica interna do algoritmo. Foram analisados os caminhos lógicos da função de busca binária implementada em Java, com o objetivo de garantir que todos os fluxos foram devidamente testados.

2. Código original

```
public class TesteBuscaBinaria {  
    public static int busca_binaria(int iVet[], int iK) {  
        int iBaixo = 0, iAlto = iVet.length - 1, iMeio;  
  
        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;  
    }  
  
    public static void main(String[] args) {  
        int[] vetor = {1, 3, 5, 7, 9};  
        int chave = 1;  
  
        int resultado = busca_binaria(iVet: vetor, iK: chave);  
        System.out.println("Resultado da busca: " + resultado);  
    }  
}
```

3. Função testada (Código refatorado)

```
public class BuscaBinariaRefatorada {  
    public static int buscaBinaria(int[] vetor, int chave) {  
        if (vetor == null || vetor.length == 0) return -1;  
  
        int inicio = 0;  
        int fim = vetor.length - 1;  
  
        while (inicio <= fim) {  
            int meio = (inicio + fim) / 2;  
  
            if (chave < vetor[meio]) {  
                fim = meio - 1;  
            } else if (chave > vetor[meio]) {  
                inicio = meio + 1;  
            } else {  
                return meio;  
            }  
        }  
  
        return -1;  
    }  
  
    public static void main(String[] args) {  
        int[] vetor = {1, 3, 5, 7, 9};  
        int chave = 3;  
  
        int resultado = buscaBinaria(vetor, chave);  
        System.out.println("Resultado da busca: " + resultado);  
    }  
}
```

4. Casos de Teste de Caixa Branca

Caso	Descrição	Entrada (vetor[], chave)	Saída esperada
CB1	Valor está no meio da primeira tentativa	{1, 3, 5, 7, 9}, 5	2
CB2	Valor está à esquerda	{1, 3, 5, 7, 9}, 1	0
CB3	Valor está à direita	{1, 3, 5, 7, 9}, 9	4
CB4	Valor menor que todos	{2, 4, 6, 8}, 1	-1
CB5	Valor maior que todos	{2, 4, 6, 8}, 10	-1
CB6	Vetor de 1 elemento, valor existe	{5}, 5	0
CB7	Vetor de 1 elemento, valor não existe	{5}, 3	-1
CB8	Vetor vazio	{}, qualquer	-1

CB9	Vetor par, valor no lado esquerdo	{1, 2, 3, 4}, 2	1
CB10	Vetor par, valor no lado direito	{1, 2, 3, 4}, 4	3
CB11	Valor no início	{1, 2, 3, 4, 5}, 1	0
CB12	Valor no fim	{1, 2, 3, 4, 5}, 5	4

4. Conclusão

Com os testes de caixa branca, foi possível validar todos os caminhos internos da função. Os resultados obtidos correspondem às saídas esperadas, garantindo a corretude da lógica aplicada na busca binária.