

PLANO DE TESTE – EXERCÍCIO PRÁTICO AULA 04

Nome do Grupo: InovaTech

Integrantes:

1. Gabriel Farah de Lima (RA: 822231424)
(farahzerafacul21@hotmail.com)
2. Webster Diogenes Rodrigues (RA:8222242764)
(rdiogenes.webster12@gmail.com)
3. Bianca Alves Ribeiro (RA: 8222240261)
(bialuno4@gmail.com)
4. Luiz Gustavo França de Abreu (RA: 823210075)
(luizgustavo_40@hotmail.com)
5. Fabrício de Barros Narbon (RA: 822227166)
(fabricionarbon50@gmail.com)
6. Rafael Rossetto Guitarrari (RA: 823158602)
(rafaelguitarrari@gmail.com)

Curso: Gestão e Qualidade de Software

Turma: (CCP1AN-MCD3)

Professor: Robson Calvetti

1. Introdução:

O método apresentado com o nome “busca_binaria” é um algoritmo de busca que permite encontrar itens em lista ordenadas. É um método escrito na linguagem de programação Java e tem como resultado otimizar a busca.

2. Requisitos do Teste:

Sem dúvidas, por se tratar de um algoritmo de busca, o principal ponto a ser testado é desempenho. Essa é a principal funcionalidade explorada no método, logo, deve ser o principal recurso a ser testado

3. Estratégias e Ferramentas:

Um teste que foi bem aplicado nesse método foi o teste de caixa branca. Explicando o algoritmo de busca afim de contextualização. Existe uma entrada a ser buscada em uma lista, logo, existe a expectativa de retorno correto. Isso faz o teste de caixa branca ser muito preciso nesse cenário. Pois será validado a entrada, lógica e saída. Cada passo, entrada, estrutura e possível resultado será analisado e esquematizado.

4. Equipe e Infraestrutura:

- **Hardware:**
 - Notebook pessoal com processador Intel i5 ou superior, 8 GB de RAM.
- **Software:**
 - IDE: IntelliJ IDEA / Eclipse / VSCode com suporte a Java.
 - Java JDK: Versão 11 ou superior.
 - Frameworks de teste: JUnit 5 para criação e execução dos testes automatizados.
 - Sistema Operacional: Windows 10 / Linux Ubuntu.
 - Controle de versão: Git

5. Cronograma de atividades:

- Execução dos Testes: Rodar os testes e validar os resultados obtidos. (Principalmente o teste de caixa branca)
- Avaliação dos Resultados: Verificar a eficácia dos testes e identificar possíveis melhorias no código.

ROTEIRO DE TESTE – EXERCÍCIO PRÁTICO AULA 04

ID do Caso	Descrição	Entrada (iVet, iK)	Resultado Esperado	Tipo de Teste
TC001	Elemento no meio do vetor	[1, 3, 5, 7, 9], 5	2	Caminho básico
TC002	Elemento na primeira posição	[1, 3, 5, 7, 9], 1	0	Limite inferior
TC003	Elemento na última posição	[1, 3, 5, 7, 9], 9	4	Limite superior
TC004	Elemento ausente	[1, 3, 5, 7, 9], 4	-1	Busca sem sucesso
TC005	Vetor com apenas um elemento (encontrado)	[5], 5	0	Caso unitário
TC006	Vetor com apenas um elemento (ausente)	[5], 3	-1	Caso unitário
TC007	Vetor vazio	[], 5	-1	Caso extremo
TC008	Vetor com elementos repetidos	[1, 2, 2, 2, 3], 2	2 ou outro índice de 2	Comportamento esperado
TC009	Elemento negativo	[-5, -3, -1, 0, 2], -3	1	Variedade de dados