Plano de Testes & Roteiro de Testes

Integrantes:
Gabriel Neules Gomes Rodrigues Soares - RA 822167394 - Email: 822167394@ulife.com.br Lucas Vasconcellos Ramos de Sousa - RA 8222242697 - Email: 8222242697@ulife.com.br Paloma Lopes de Sousa - RA 822167506 - Email: 822167506@ulife.com.br Wellerson Resende Monteiro - RA 8222243349 - Email: 8222243349@ulife.com.br

2025-1

Plano de Teste

1. Introdução

Projeto: Validação do algoritmo de busca binária iterativa em Java
Objetivo: Garantir que a implementação do algoritmo de busca binária esteja correta e retorne os índices apropriados para diferentes casos de uso.
Público-alvo: Desenvolvedores, testadores e equipe de QA do projeto de algoritmos fundamentais.
Escopo: Este plano cobre testes unitários do método `busca_binaria`, considerando arrays ordenados de inteiros, inclusive casos de borda e entrada inválida.

---

2. Requisitos do Teste

Funcionais:

- O método deve retornar o índice correto do elemento buscado, se ele existir.
- O método deve retornar -1 quando o elemento não estiver presente.

Não funcionais:

- **Desempenho:** O tempo de execução deve ser logarítmico em relação ao tamanho do vetor.
- **Confiabilidade:** O método não deve lançar exceções com entradas válidas.
- **Robustez:** Deve lidar corretamente com arrays vazios.

---

3. Estratégias e Ferramentas

Tipos de Testes:

- Testes unitários
- Testes de borda
- Testes negativos (valor não encontrado)
- Testes com vetores grandes (validação de desempenho)

Técnicas Empregadas:

- Particionamento de equivalência
- Análise de valores limites

Critério de Finalização:

- 100% de cobertura de ramos e decisões
- Todos os testes devem passar sem falhas

Ferramentas:

- JUnit (para execução dos testes)
- IntelliJ IDEA ou Eclipse (IDE)
- JaCoCo (para análise de cobertura de testes)

---

5. Equipe e Infraestrutura

Equipe:

- 1 Engenheiro de Testes
- 1 Desenvolvedor
- 1 Gerente de Qualidade

Infraestrutura:

- Ambiente local com JDK instalado
- IDE com suporte a execução de testes unitários
- Ferramenta de cobertura de código

---

6. Cronograma de Atividades

Atividade	Início	Fim
Projeto e escrita dos testes	01/05/2025	01/05/2025
Execução dos testes	01/05/2025	02/05/2025
Avaliação dos resultados	02/05/2025	03/05/2025

---

7. Documentação

- Especificação do método `busca_binaria`
- Plano de testes
- Código-fonte em Java
- Relatório de execução dos testes
- Relatório de cobertura de código

Roteiro de Testes

Localização:

Módulo de Algoritmos > Método de Busca Binária

Objeto de Teste:

Validação do método `busca_binaria` com diferentes tipos de entrada.

Caso de Teste 1

Descrição: Buscar elemento existente no meio do vetor.
Pré-condição: Vetor ordenado de forma crescente.
Procedimento:

1. Executar `busca_binaria(new int[]{1, 3, 5, 7, 9}, 5)` **Resultado esperado:** Retorna `2` (índice do número 5)

Caso de Teste 2

Descrição: Buscar elemento inexistente no vetor.
Pré-condição: Vetor ordenado.
Procedimento:

1. Executar `busca_binaria(new int[]{2, 4, 6, 8}, 5)` **Resultado esperado:** Retorna `-1`

Caso de Teste 3

Descrição: Buscar elemento na primeira posição.
Pré-condição: Vetor com múltiplos elementos ordenados.
Procedimento:

1. Executar `busca_binaria(new int[]{0, 1, 2, 3}, 0)` **Resultado esperado:** Retorna `0`

Caso de Teste 4

Descrição: Buscar elemento na última posição.
Pré-condição: Vetor com múltiplos elementos ordenados.
Procedimento:

1. Executar `busca_binaria(new int[]{0, 2, 4, 6, 8}, 8)` **Resultado esperado:** Retorna `4`

Caso de Teste 5

Descrição: Vetor vazio.
Pré-condição: Vetor sem elementos.
Procedimento:

1. Executar `busca_binaria(new int[]{}, 3)` **Resultado esperado:** Retorna `-1`

Caso de Teste 6

Descrição: Vetor com um único elemento (presente).
Pré-condição: Vetor com um único valor.
Procedimento:

1. Executar `busca_binaria(new int[]{10}, 10)` **Resultado esperado:** Retorna `0`

Caso de Teste 7

Descrição: Vetor com um único elemento (ausente).
Pré-condição: Vetor com um único valor.
Procedimento:

1. Executar `busca_binaria(new int[]{10}, 5)` **Resultado esperado:** Retorna `-1`