



## **Gestão e Qualidade de Software**

### **Sistemas de Informação**

Integrantes do grupo:

Kayky Cerquiaro Prado - 822155538

Enrico Aguiar Vrunski - 82210618

Thiago Ferreira Lima Gonçalves - 824156179

Matheus Tognon Siqueira - 824141731

Felipe Soares de Oliveira – 824156311

# **Plano de Teste e Roteiro de Teste**

## **Exercício Prático 1 e 2 (Caixa Branca):**

### **Plano de Teste:**

#### **1. Introdução:**

- Projeto: Teste estrutural de algoritmo de busca binária.
- Objetivo: Validar o funcionamento correto da busca binária usando a técnica de teste de caixa branca.
- Público-alvo: Equipe de testes e desenvolvimento.
- Escopo: Função de busca binária implementada em linguagem de programação (ex: Python, C, Java).

#### **2. Requisitos do teste:**

- O algoritmo deve localizar corretamente um valor dentro de um vetor ordenado.
- O algoritmo deve retornar -1 se o valor não estiver presente.
- O teste deve cobrir casos como: vetor vazio, um único elemento, valor no meio, valor no início, valor no fim e valor ausente.

#### **3. Estratégias e ferramentas:**

- Estratégia: Teste de Caixa Branca com base em análise de fluxo (nós e arestas).
- Técnica: Tabela de decisão e complexidade ciclomática.
- Critério de finalização: Cobertura de todos os caminhos possíveis (100% de nós e arestas).
- Ferramentas: Editor de código (ex: VS Code), terminal de execução, ferramenta de depuração (opcional).

#### **4. Equipe e infraestrutura:**

- Equipe composta por alunos responsáveis pelo trabalho.
- Infraestrutura: Computadores com ambiente de desenvolvimento instalado.

- Softwares: Compilador/interpretador da linguagem escolhida, editor de texto/código.

#### **5. Cronograma de atividades:**

- Elaboração do código da função de busca binária: 08/04
- Identificação de nós e arestas e cálculo da complexidade ciclomática: 09/04
- Montagem da tabela de casos de teste: 10/04
- Execução dos testes e validação dos resultados: 11/04

#### **6. Documentação:**

- Documento de especificação do algoritmo (código-fonte).
- Tabela de casos de teste estruturais (presente neste plano).
- Roteiro de execução e resultados esperados.

### **Roteiro de Teste:**

Localização: Função: Busca Binária no vetor ordenado.

Objetos de Teste:

- Localização do valor no vetor (meio, início, fim).
- Respostas quando o valor não está presente ou vetor está vazio.

### **Descrição dos Casos de Teste:**

- Onde serão inseridos os procedimentos necessários para a execução de um determinado teste do sistema.
- É composto por:

**Descrição:** onde se descreve a ideia específica do caso de teste;

**Pré-condição:** onde se descreve o requisito para o comportamento do sistema antes da execução do caso de teste;

**Procedimento:** onde se descrevem os passos para a execução do caso de teste, não fugindo do foco apresentado na descrição;

**Resultado esperado:** onde se descreve como o sistema deveria se comportar após a execução do procedimento.

# Casos de Teste:

Caso	Entrada (iVet)	Valor (ik)	Descrição	Saída Esperada	Nós Percorridos
CT01	[1, 2, 3, 4, 5]	3	Valor está no meio do vetor	2	1 → 2 → 3 → 4 (F) → 6 (F) → 8 → 9
CT02	[1, 2, 3, 4, 5]	5	Valor está no fim	4	1 → 2 → 3 → 4 (F) → 6 (V) → 7 → 2 → ... → 8 → 9
CT03	[1, 2, 3, 4, 5]	1	Valor está no início	0	1 → 2 → 3 → 4 (V) → 5 → 2 → ... → 8 → 9
CT04	[1, 2, 3, 4, 5]	6	Valor não está no vetor	-1	1 → 2 → 3 → 4 (F) → 6 (V) → 7 → ... → 2 (F) → 10
CT05	[1, 2, 3, 4, 5]	0	Valor abaixo do menor número do vetor	-1	1 → 2 → 3 → 4 (V) → 5 → ... → 2 (F) → 10
CT06	[ ]	1	Vetor vazio	-1	1 → 2 (F) → 10
CT07	[1]	1	Vetor de um elemento, valor correto	0	1 → 2 → 3 → 4 (F) → 6 (F) → 8 → 9
CT08	[1]	2	Vetor de um elemento, valor incorreto	-1	1 → 2 → 3 → 4 (F) → 6 (V) → 7 → 2 (F) → 10