

UNIVERSIDADE SÃO JUDAS TADEU  
CIÊNCIA DA COMPUTAÇÃO

GABRIEL FORNICOLA AMORIM - 824148690  
GIOVANNI RIBEIRO IANNACE - 82421986  
GIOVANNA FONTES DA SILVA – 823148980  
LUCAS GASPARETTO NARCIZO DE MORAIS - 82426494  
RAPHAEL MIGUEL FOLEGO - 822163593

Gestão e Qualidade de Software

São Paulo  
2025

## **Resumo**

Os exercícios práticos da UC de Gestão e Qualidade de Software têm como foco a elaboração de planos e roteiros de testes baseados nas atividades da aula 04.

## **Plano de Testes (Exercício Prático 1)**

### **1. Identificação do Sistema**

- Nome do sistema: Algoritmo de Busca Binária (Iterativa)
- Função testada: busca binária (int iVet[], int iK)
- Linguagem: Java
- Fonte: Aula de Gestão e Qualidade de Software – Universidade São Judas
- Modulo: Algoritmos de busca
- Função testada: busca binária (int iVet[], int iK)

### **2. Objetivo dos Testes:**

- Validar se o método busca binária retorna corretamente o índice do elemento iK dentro do vetor ordenado iVet.
- Confirmar também se, caso o elemento não exista, o método retorna -1.

### **3. Escopo do Teste:**

- Inclui: Vetores ordenados de inteiros, com diferentes tamanhos (vazios, unitários, pares e ímpares), com valores duplicados e extremos.
- Não inclui: Vetores não ordenados, valores nulos, vetores de tipos não inteiros.

### **4. Critérios de Aceitação**

- O vetor deve estar ordenado de forma crescente.
- A busca deve retornar o índice correto caso o valor esteja presente.
- A busca deve retornar -1 caso o valor não esteja presente.
- O algoritmo deve lidar com casos limites como vetor vazio, um único elemento, ou elementos repetidos.

### **5. Requisitos Funcionais Testados**

RF01: O sistema deve retornar corretamente a posição de um elemento existente em um vetor ordenado.

RF02: O sistema deve retornar -1 se o elemento não existir no vetor.

## 6. Técnicas de Teste Utilizadas

- Particionamento de Equivalência
- Análise do Valor Limite
- Teste de Caixa Preta
- Teste de Caixa Branca (opcional)

## 7. Ambiente de Teste

- SO: Windows/Linux/Mac
- IDE: Eclipse / IntelliJ / VSCode
- JDK: 8 ou superior
- Ferramentas de teste: JUnit (opcional)

## 8. Critérios de Saída

O teste será considerado bem-sucedido se:

- Todas as saídas esperadas forem obtidas conforme definido.
- O sistema não lançar exceções imprevistas.

## 9. Cronograma:

Atividade	Data de Início	Data de Término
Criação do plano	14/04/2025	15/04/2025
Elaboração dos testes	14/04/2025	14/04/2025
Execução dos testes	15/04/2025	15/04/2025
Análise dos resultados	15/04/2025	16/04/2025

## **Roteiro de Testes – Busca Binária (Exercício prático 2)**

### **CT01: Vetor vazio**

- Entrada: iVet = [], iK = 5
- Saída esperada: -1
- Objetivo: Verificar se a função lida corretamente com vetores vazios.

### **CT02: Vetor com um único elemento (elemento presente)**

- Entrada: iVet = [5], iK = 5
- Saída esperada: 0
- Objetivo: Verificar se a função encontra o único elemento corretamente.

### **CT03: Vetor com um único elemento (elemento ausente)**

- Entrada: iVet = [10], iK = 5
- Saída esperada: -1
- Objetivo: Verificar se a função retorna -1 quando o elemento único não corresponde.

### **CT04: Vetor com mais de um elemento (elemento presente)**

- Entrada: iVet = [2, 5, 6, 9], iK = 6
- Saída esperada: 2
- Objetivo: Verificar busca com vetor de tamanho maior que um elemento.

### **CT05: Vetor com mais de um elemento (elemento ausente)**

- Entrada: iVet = [2, 5, 6, 9], iK = 7
- Saída esperada: -1
- Objetivo: Verificar se a função retorna -1 quando o elemento não está presente em um vetor com mais de um elemento.

**CT06: Vetor com elementos duplicados (elemento presente)**

- Entrada: iVet = [1, 2, 2, 2, 3, 4], iK = 2
- Saída esperada: 1, 2, ou 3 (qualquer índice de uma ocorrência válida)
- Objetivo: Confirmar que a função retorna algum índice válido em caso mais de um índice com o mesmo valor.

**CT07: Vetor com elementos duplicados (elemento ausente)**

- Entrada: iVet = [1, 1, 3, 3, 5, 5], iK = 2
- Saída esperada: -1
- Objetivo: Verificar ausência de valor mesmo com mais de um índice igual no vetor.

**CT08: Elemento menor que todos os elementos do vetor**

- Entrada: iVet = [10, 20, 30], iK = 5
- Saída esperada: -1
- Objetivo: Verificar busca de valor abaixo do mínimo.

**CT09: Elemento maior que todos os elementos do vetor**

- Entrada: iVet = [10, 20, 30], iK = 35
- Saída esperada: -1
- Objetivo: Verificar busca de valor acima do máximo.

**CT10: Vetor com valores negativos e positivos**

- Entrada: iVet = [-10, -5, 0, 5, 10], iK = -5
- Saída esperada: 1
- Objetivo: Verificar se a função lida com inteiros negativos corretamente.

**CT11: Vetor com apenas valores negativos (elemento presente)**

- Entrada: iVet = [-20, -15, -10, -5], iK = -10
- Saída esperada: 2
- Objetivo: Validar funcionamento com inteiros negativos.

**CT12: Vetor grande com 1000 elementos (elemento presente)**

- Entrada: iVet = [0, 1, 2, ..., 999], iK = 678
- Saída esperada: 678
- Objetivo: Verificar desempenho e precisão em vetores grandes.

## **Identificação do Projeto (Exercício Prático 3)**

Sistema de autenticação de usuários com verificação em duas etapas (2FA), com base em login, senha e código enviado via SMS.

### **1.2. Definições, Siglas e Abreviações:**

2FA: Two-Factor Authentication (Autenticação em Duas Etapas)

SMS: Short Message Service

QA: Quality Assurance

CT: Caso de Teste

BDD: Base de Dados do sistema

### **1.3. Referências:**

- Documento de requisitos do sistema – Versão 1.0
- Guia de segurança para autenticação – OWASP
- Padrões internos de qualidade da empresa

## **2. Escopo**

Este plano de teste cobre o processo de login com autenticação em duas etapas, garantindo a verificação de credenciais do usuário e validação do código temporário enviado via SMS. O escopo se limita à interface de login e etapas de autenticação.

## **3. Objetivos**

- Validar a funcionalidade de login com autenticação em duas etapas;
- Verificar a resposta do sistema a entradas válidas e inválidas;
- Garantir que mensagens de erro sejam exibidas de forma clara;
- Assegurar que os fluxos alternativos (erros e exceções) estejam cobertos;
- Verificar a integridade do envio de código 2FA via SMS.



#### 4. Requisitos a Serem Testados

ID	Requisito	Descrição
REQ01	Entrada de login e senha	Usuário deve digitar credenciais corretamente
REQ02	Validação de credenciais	O sistema compara com dados armazenados
REQ03	Geração de código 2FA	Código é gerado após validação correta
REQ04	Envio de SMS	Código é enviado ao número registrado
REQ05	Validação do código 2FA	Sistema verifica o código digitado
REQ06	Mensagens de erro	Mensagens apropriadas devem ser exibidas em caso de falha
REQ07	Expiração do código 2FA	Código deve expirar após tempo determinado

#### 5. Estratégias, Tipos de Testes e Ferramentas

##### 5.1 Estratégias de Teste:

- Testes manuais funcionais baseados em casos de uso
- Testes exploratórios para identificar comportamentos inesperados
- Testes de regressão sempre que houver alteração no código de autenticação

##### 5.2 Tipos de Testes:

- Teste funcional
- Teste de caixa preta
- Teste de usabilidade (mensagens, fluxo, feedback ao usuário)
- Teste de integração com serviço de SMS

##### 5.3 Ferramentas Utilizadas:

- Postman: para testes de API de autenticação/SMS
- Selenium: para testes automatizados da interface (se aplicável)
- Jira / TestRail: para gestão de testes e bugs

- Emulador de SMS / Serviço de sandbox: para simular o envio de mensagens

## 6. Recursos a Serem Empregados

Recurso	Quantidade	Responsável
Analistas de Teste	2	Equipe QA
Desenvolvedores para suporte	1	Equipe de Dev
Ambiente de homologação	1	DevOps
Ferramentas de teste	conforme seção 5	QA

## 7. Cronograma e Marcos do Projeto

Fase	Atividade	Início	Fim	Responsável
Planejamento	Criação do plano de teste	15/04	16/04	QA
Preparação	Configuração de ambiente de testes	17/04	18/04	DevOps
Execução	Execução dos casos de teste	19/04	22/04	QA
Validação	Report de resultados e bugs	22/04	23/04	QA
Reexecução	Reteste após correções	24/04	25/04	QA
Fechamento	Entrega do relatório final	26/04	26/04	QA

### Marcos principais:

- Entrega da primeira rodada de testes: 22/04
- Entrega final com retestes: 26/04

## **Roteiro de teste (Exercício Prático 4)**

Login com validação em duas etapas

### **Cenário 1 – Login e senha incorretos**

Pré-condição: Usuário tenta acessar o sistema.

- Passos:
  1. Informar login inválido.
  2. Informar senha inválida.
  3. Clicar no botão “Entrar”.

Resultado Esperado: Mensagem "Login e/ou senha incorretos".

### **Cenário 2 – Login correto, senha incorreta**

Pré-condição: Usuário cadastrado previamente no sistema.

- Passos:
  4. Informar login válido.
  5. Informar senha inválida.
  6. Clicar no botão “Entrar”.

Resultado Esperado: Mensagem "Login e/ou Senha incorretos".

### **Cenário 3 – Login e senha corretos, código em duas etapas incorreto**

Pré-condição: Login e senha válidos, número de celular válido cadastrado.

- Passos:
  7. Informar login válido.
  8. Informar senha válida.
  9. Clicar em “Entrar”.
  10. Receber código de autenticação por SMS.
  11. Informar que o código está incorreto.

Resultado Esperado: Mensagem "Login não autorizado!".

#### **Cenário 4 – Login, senha e código em duas etapas corretos**

Pré-condição: Usuário cadastrado com número de celular válido.

- Passos:
  12. Informar login válido.
  13. Informar senha válida.
  14. Clicar em “Entrar”.
  15. Receber código de autenticação por SMS.
  16. Informar o código corretamente.

Resultado Esperado: Mensagem "Login realizado com sucesso" e acesso liberado ao sistema.

#### **Cenário 5 – Campos vazios (validação de formulário)**

Pré-condição: Nenhum dado informado.

- Passos:
  17. Deixar os campos "Login" e "Senha" vazios.
  18. Clicar em “Entrar”.

Resultado Esperado: Mensagem solicitando o preenchimento dos campos obrigatórios (ex: "Preencha todos os campos").

#### **Cenário 6 – Código de duas etapas não recebido**

Pré-condição: Login e senha válidos.

- Passos:
  19. Informar login válido.
  20. Informar senha válida.
  21. Clicar em “Entrar”.
  22. Não receber código via SMS (simulação de falha).

Resultado Esperado: Mensagem de erro ou opção para reenviar o código.

## **Cenário 7 – Expiração do código de duas etapas**

Pré-condição: Código de validação com tempo limitado.

- Passos:
  23. Informar login e senha válidos.
  24. Receber o código de autenticação.
  25. Aguardar o tempo limite de expiração do código.
  26. Informar o código expirado.

Resultado Esperado: Mensagem "Código expirado" ou "Login não autorizado!".