

WELLERSON RESENDE MONTEIRO | RA: 8222243349

PALOMA LOPES DE SOUSA | RA: 822167506

LUCAS VASCONCELLOS RAMOS DE SOUSA | RA: 8222242709

GABRIEL NEULES GOMES RODRIGUES SOARES | RA: 822167394

# Plano de Testes: Exercício Prático 1 - Busca Binária

## 1. Identificação do Projeto

- **Projeto:** Teste de Caixa-Branca para o Exercício Prático 1.
- **Documento:** Plano de Testes.
- **Autor:** (Seu Nome/Nome do Grupo)
- **Data:** 11/06/2025
- **Versão:** 1.0

## 2. Introdução e Objetivos

O objetivo deste documento é delinear o plano e a estratégia para a execução de testes de caixa-branca (também conhecidos como testes estruturais) no método `busca_binaria`, conforme especificado no "Exercício prático 1" da disciplina de Gestão e Qualidade de Software.

O teste visa garantir que todas as ramificações lógicas e caminhos de execução do método sejam exercitados, identificando possíveis erros na sua implementação.

## 3. Escopo do Teste

Item em Teste: O método `busca_binaria` fornecido em Java.

```
public static int busca_binaria(int iVet[], int iK)
```

```
{  
    int iBaixo, iAlto, iMeio;  
    iBaixo = 0;  
    iAlto = iVet.length - 1;  
    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;
}

```

- 
- **Fora do Escopo:**
  - Testes de performance.
  - Testes de usabilidade.
  - Testes em outros componentes do sistema que não o método especificado.

## 4. Estratégia de Teste

A abordagem utilizada será o Teste de Caminho Básico (Base Path Testing). Esta técnica de teste de caixa-branca garante que cada instrução e decisão no código seja executada pelo menos uma vez.

As etapas da estratégia são:

1. **Desenhar o Grafo de Fluxo:** Mapear o código-fonte em um grafo que represente os fluxos de controle.
2. **Calcular a Complexidade Ciclomática:** Determinar o número mínimo de caminhos independentes necessários para a cobertura completa. A fórmula é  $V(G) = A - N + 2$  (Arestas - Nós + 2) ou  $V(G) = P + 1$  (Predicados + 1).
3. **Identificar os Caminhos Independentes:** Listar os caminhos de execução do início ao fim do método.
4. **Desenvolver Casos de Teste:** Criar casos de teste específicos para forçar a execução de cada caminho independente.

## 5. Análise e Casos de Teste

### 5.1. Grafo de Fluxo e Complexidade Ciclomática

(Nesta seção, você desenharia o grafo de fluxo e calcularia a complexidade).

- **Predicados (Decisões):** while (iBaixo <= iAlto), if (iK < iVet[iMeio]), if (iK > iVet[iMeio]).
- **Complexidade Ciclomática (V(G)):**  $P + 1 = 3 + 1 = 4$ .
- Portanto, são necessários 4 caminhos independentes para testar o método.

### 5.2. Caminhos Independentes e Casos de Teste

ID	Caminho Independente (Nós)	Descrição do Cenário	Dados de Entrada (iVet, iK)	Saída Esperada
CT01	1-2-10	O laço while não é executado. Vetor vazio.	iVet = {}, iK = 10	-1
CT02	1-2-3-4-5-8-2-10	iK é menor que o elemento do meio e não é encontrado.	iVet = {20, 30, 40}, iK = 10	-1
CT03	1-2-3-4-6-7-8-2-10	iK é maior que o elemento do meio e não é encontrado.	iVet = {20, 30, 40}, iK = 50	-1
CT04	1-2-3-4-6-9	iK é encontrado exatamente no meio na primeira tentativa.	iVet = {20, 30, 40}, iK = 30	1
CT05	(Caminho adicional)	iK é encontrado após várias iterações (ex: último elemento).	iVet = {10, 20, 30, 40, 50}, iK = 50	4

*Nota: CT05 é um caso de teste adicional para garantir a robustez do laço, embora a base de caminhos já esteja coberta pelos 4 primeiros.*

## 6. Critérios de Entrada e Saída

- Critério de Entrada: O código-fonte do método busca\_binaria está completo e disponível para teste.
- Critério de Saída: Todos os casos de teste definidos foram executados, os resultados documentados e 100% dos caminhos independentes foram cobertos.

## 7. Recursos Necessários

- **Hardware:** Computador com ambiente de desenvolvimento.
- **Software:** IDE Java (ex: Eclipse, IntelliJ), Kit de Desenvolvimento Java (JDK), JUnit (opcional, para automação).

## 8. Cronograma

Atividade	Responsável	Início Previsto	Fim Previsto
Elaboração do Plano de Testes	(Seu Nome/Grupo)	11/06/2025	11/06/2025
Execução dos Testes	(Seu Nome/Grupo)	12/06/2025	12/06/2025
Relatório de Resultados	(Seu Nome/Grupo)	13/06/2025	13/06/2025

## 9. Entregáveis

1. Plano de Testes (este documento).
2. Relatório de Execução de Testes: Um documento contendo os resultados da execução de cada caso de teste (se passou ou falhou), com evidências (logs, screenshots).
3. Relatório Final de Testes: Um resumo geral do processo, incluindo o grafo de fluxo, o cálculo da complexidade, e quaisquer defeitos encontrados.

# Plano de Testes: Exercício Prático 2 - Login com 2 Etapas

## 1. Identificação do Projeto

- **Projeto:** Teste de Caixa-Preta para o Exercício Prático 2.
- **Funcionalidade:** Login com validação em duas etapas.
- **Autor:** (Seu Nome/Nome do Grupo)
- **Data:** 11/06/2025

- **Versão: 1.0**

## **2. Introdução e Objetivos**

Este documento detalha o plano para a execução de testes de caixa-preta na funcionalidade de "Login com validação em duas etapas", conforme descrito no exercício 2 da aula 04. O teste de caixa-preta foca na validação dos requisitos funcionais do sistema, tratando-o como uma "caixa-preta", sem conhecimento de sua estrutura interna de código.

O objetivo principal é verificar se o sistema se comporta conforme o esperado para todas as entradas e condições descritas nos requisitos, garantindo que o fluxo de login seja seguro e funcional.

## **3. Escopo do Teste**

- **Em Escopo:**
  - Validação do par usuário/senha.
  - Geração e envio (simulado) do código de validação por SMS.
  - Validação do código de duas etapas.
  - Exibição de todas as mensagens de feedback para o usuário (sucesso, erro de credenciais, erro de código).
  - Fluxo completo de login, desde a entrada de credenciais até o acesso ao sistema.
- **Fora do Escopo:**
  - Testes de desempenho ou carga da funcionalidade.
  - Testes de segurança aprofundados (ex: SQL Injection, XSS).
  - Teste da infraestrutura de envio de SMS.
  - Cadastro de novos usuários ou recuperação de senha.

## **4. Estratégia de Teste**

A estratégia será baseada em técnicas de teste de caixa-preta para garantir a cobertura dos requisitos funcionais.

- **Técnicas Utilizadas:**
  1. **Particionamento de Equivalência:** As entradas (login, senha, código) serão divididas em classes de dados válidos e inválidos.
  2. **Análise de Valor Limite:** Será aplicada caso os campos tenham restrições de tamanho (ex: mínimo/máximo de caracteres), embora não especificado nos requisitos.
  3. **Teste Baseado em Caso de Uso:** Serão criados cenários de teste que simulam a interação real do usuário com o sistema, cobrindo o "caminho feliz" (sucesso) e os fluxos de exceção.

## 5. Recursos e Ambiente de Teste

- **Ambiente:** Um ambiente de homologação com a aplicação web disponível.
- **Dados de Teste:** Uma base de dados de teste contendo usuários pré-cadastrados com senhas e números de celular definidos.
  - **Exemplo de Usuário Válido:**
    - Login: **usuario.teste**
    - Senha: **Senha@123**
    - Celular: **(11) 99999-9999**
- **Ferramentas:**
  - Navegador Web (Chrome, Firefox).
  - Ferramenta para simular/interceptar o recebimento de SMS (se disponível) ou um log no sistema que exiba o código gerado.

## 6. Casos de Teste

A seguir, uma lista de casos de teste derivados dos requisitos.

ID	Cenário de Teste	Pré-condições	Passos de Execução	Resultado Esperado
CT01	Caminho Feliz: Login bem-sucedido	Usuário <b>usuario.teste</b> existe no banco de dados.	1. Inserir login: <b>usuario.teste</b> . 2. Inserir senha: <b>Senha@123</b> . 3. Clicar em "Entrar". 4. Receber o código de validação (ex: <b>123456</b> ). 5. Inserir o código <b>123456</b> . 6. Clicar em "Validar".	A mensagem "Login realizado com sucesso" é exibida e o acesso ao programa é liberado.
CT02	Credenciais Inválidas: Senha incorreta	Usuário <b>usuario.teste</b> existe no banco de dados.	1. Inserir login: <b>usuario.teste</b> . 2. Inserir senha: <b>senha_errada</b> . 3. Clicar em "Entrar".	A mensagem "Login e/ou Senha incorretos" é exibida. O fluxo é interrompido.

CT03	Credenciais Inválidas: Login incorreto	Nenhum usuário <b>login.inexistent</b> e existe no banco.	1. Inserir login: <b>login.inexistente</b> . 2. Inserir senha: <b>qualquer_senha</b> . 3. Clicar em "Entrar".	A mensagem "Login e/ou Senha incorretos" é exibida.
CT04	Código de Validação Incorreto	Credenciais válidas foram inseridas.	1. Seguir passos 1-4 do CT01. 2. Inserir um código inválido (ex: <b>654321</b> ). 3. Clicar em "Validar".	A mensagem "Login não autorizado!" é exibida.
CT05	Entrada em Branco: Campo de login vazio	N/A	1. Deixar o campo de login em branco. 2. Inserir senha: <b>Senha@123</b> . 3. Clicar em "Entrar".	O sistema deve exibir uma mensagem de erro indicando que o campo é obrigatório (Comportamento esperado, embora não especificado).
CT06	Entrada em Branco: Campo de senha vazio	N/A	1. Inserir login: <b>usuario.teste</b> . 2. Deixar o campo de senha em branco. 3. Clicar em "Entrar".	O sistema deve exibir uma mensagem de erro indicando que o campo é obrigatório (Comportamento esperado).
CT07	Entrada em Branco: Código de validação vazio	Credenciais válidas foram inseridas.	1. Seguir passos 1-4 do CT01. 2. Deixar o campo de código em branco. 3. Clicar em "Validar".	O sistema deve exibir uma mensagem de erro indicando que o campo é obrigatório

(Comportamento  
esperado).

## **7. Critérios de Sucesso e Saída**

- **Critério de Sucesso:** Todos os casos de teste devem ser executados e seus resultados devem corresponder aos resultados esperados.
- **Critério de Saída:** O processo de teste será considerado concluído quando todos os casos de teste forem executados, os defeitos críticos forem corrigidos e retestados, e a equipe concordar que o sistema atinge o nível de qualidade desejado.

## **8. Entregáveis**

1. **Plano de Testes** (este documento).
2. **Roteiro de Testes:** Documento com o passo a passo detalhado para a execução de cada caso de teste.
3. **Relatório de Execução de Testes:** Documento com o status de cada caso de teste (Passou, Falhou, Bloqueado) e evidências.
4. **Relatório de Defeitos:** Lista de todos os bugs encontrados durante os testes.