

# UNIVERSIDADE SÃO JUDAS TADEU

-

## CAMPUS BUTANTÃ

### GESTÃO E QUALIDADE DE SOFTWARE – CCP1AN-BUE1

#### Nome do Grupo

TESTE DE CAIXA BRANCA  
E  
TESTE DE CAIXA PRETA

#### Membro(s) do Grupo

822160071 – FABRÍCIO PERES – CCP - [822160071@ulife.com.br](mailto:822160071@ulife.com.br)

824116869 – HERMANO PEREIRA DE SOUSA – CCP - [824116869@ulife.com.br](mailto:824116869@ulife.com.br)

822127136 – JONATA PABLO GARCIA – CCP - [822127136@ulife.com.br](mailto:822127136@ulife.com.br)

823214064 – JÚLIA SILVA PEREIRA – ADS - [823214064@ulife.com.br](mailto:823214064@ulife.com.br)

823126459 – RANGEL RIBEIRO SANTOS – ADS - [823126459@ulife.com.br](mailto:823126459@ulife.com.br)

8222241099 – VÍTOR DE SOUZA – CCP - [8222240199@ulife.com.br](mailto:8222240199@ulife.com.br)

# SUMÁRIO

---

<b>1. TESTE DE CAIXA-BRANCA.....</b>	<b>03</b>
<b>1.1.    TESTE DE CAIXA-BRANCA SOBRE MÉTODO.....</b>	<b>03</b>
<b>2. TESTE DE CAIXA-PRETA.....</b>	<b>08</b>
<b>2.1.    CASO DE USO: LOGIN COM VALIDAÇÃO EM DUAS ETAPAS..</b>	<b>08</b>

## 1 – TESTE DE CAIXA BRANCA

---

### 1.1 – Teste de caixa-branca sobre método

#### Método de busca binária (iterativo)

```
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;
}
```

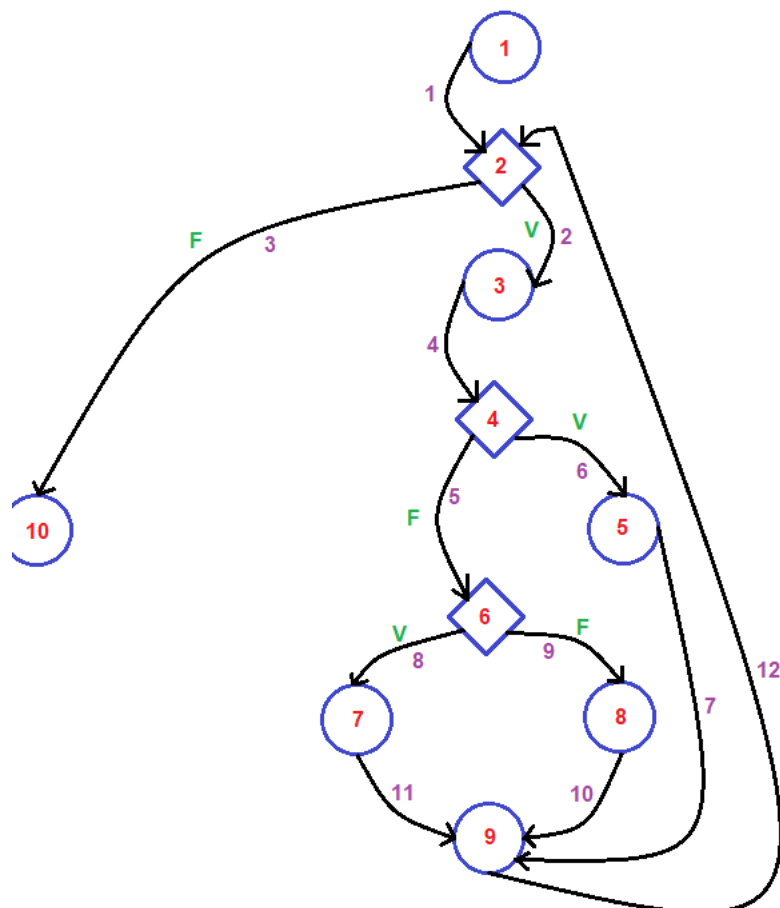
#### Agrupando em diferentes blocos

```

public static int busca_binaria(int iVet[], int iK)
{
    int iBaixo, iAlto, iMeio;
1 | iBaixo=0;
    iAlto=iVet.length-1;
2 | while(iBaixo <= iAlto)
    {
        3 | iMeio=(iBaixo+iAlto)/2;
        4 | if(iK < iVet[iMeio])5 | iAlto=iMeio-1;
        6 | else if(iK > iVet[iMeio])7 | iBaixo=iMeio+1;
            else8 | return iMeio;
    9 | }
10 | return -1;
}

```

### Grafo de fluxo



## 1.2 – Complexidade Ciclomática

$$C = A - N + 2$$

$$C = 12 - 10 + 2$$

$$C = 4 \text{ caminhos independentes}$$

1 . { 1, 2, 10 }	
iVet	{ }
iK	Não importa (nunca será consultado)

2 . { 1, 2, 3, 4, 5, 9, 2, 10 }	
iVet	{3, 4, 5}
iK	4

3 . { 1, 2, 3, 4, 6, 7, 9, 2, 10 }	
iVet	{8}
iK	11

4 . { 1, 2, 3, 4, 6, 8, 9, 2, 10 }	
iVet	{8}
iK	7

## Análise de Caminhos e Condições:

As condições principais são:

- O loop **while** (**iBaixo** <= **iAlto**).
- As condições dentro do loop que comparam **iK** com **iVet[iMeio]**.

## 2. Casos de Teste:

- **Caso 1:** Elemento está presente no meio.
- **Caso 2:** Elemento está no início.
- **Caso 3:** Elemento está no final.
- **Caso 4:** Elemento está ausente (menor que o menor elemento).
- **Caso 5:** Elemento está ausente (maior que o maior elemento).
- **Caso 6:** Elemento está ausente (entre dois elementos).

## Exemplos de Casos de Teste

**Caso 1:** iVet = [1, 3, 5, 7, 9], iK = 5

- Esperado: 2 (índice do elemento 5)

**Caso 2:** iVet = [1, 3, 5, 7, 9], iK = 1

- Esperado: 0 (índice do elemento 1)

**Caso 3:** iVet = [1, 3, 5, 7, 9], iK = 9

- Esperado: 4 (índice do elemento 9)

**Caso 4:** iVet = [1, 3, 5, 7, 9], iK = 0

- Esperado: -1 (elemento ausente)

**Caso 5:** iVet = [1, 3, 5, 7, 9], iK = 10

- Esperado: -1 (elemento ausente)

**Caso 6:** iVet = [1, 3, 5, 7, 9], iK = 4

- Esperado: -1 (elemento ausente)

### **Cobertura de Teste**

- **Caminhos A:**  $iK < iVet[iMeio]$  e ajusta iAlto.
- **Caminhos B:**  $iK > iVet[iMeio]$  e ajusta iBaixo.
- **Caminhos C:**  $iK == iVet[iMeio]$  e retorna iMeio

## 2 – TESTE DE CAIXA PRETA

---

### 2.1 – Caso de uso: login com validação em duas etapas.

#### Análise dos requisitos:

- 1.Receber, via teclado, na janela do sistema, o *login* do usuário no campo “Login”;
- 2.Receber, via teclado, na janela do sistema, a senha do usuário no campo “Senha”;
- 3.Aguardar, via mouse, na janela do sistema, a seleção do botão “Entrar”;
- 4.Validar o *login* e a senha digitados como *login* e a senha previamente armazenados na base de dados do sistema;
- 5.Se o *login* e/ou senha digitados não coincidir em com os previamente cadastrados, apresentar, na janela do sistema, a mensagem “Login e/ou Senha incorretos”;
- 6.Se o *login* e senha digitados coincidirem com os previamente cadastrados, gerar o código dinâmico para a validação em duas etapas;
- 7.Enviar o código de validação em duas etapas gerado pelo sistema, por SMS, para o número do celular do usuário previamente cadastrado no sistema;
- 8.Receber, via teclado, na janela do sistema, o código de validação em duas etapas enviado ao usuário por SMS;



9. Se o código de validação em duas etapas digitado não coincidir com o código gerado, apresentar, na janela do sistema, a mensagem “*Login não autorizado!*”;

10. Se o código de validação em duas etapas digitado coincidir com o código gerado, apresentar, na janela do sistema, a mensagem “*Login realizado com sucesso*” e liberar o acesso ao programa.

### CASOS DE TESTES

- Os campos “Login” e “Senha”, vazios serão inválidos;
- Os campos “Login” e “Senha”, incorretos serão inválidos;
- Os campos “Login” e “Senha”, corretos serão válidos para gerar o código SMS;
- O campo “Código”, vazio, será inválido;
- O campo “Código”, com formato diferente (letras ou caracteres), será inválido;
- O campo “Código”, com o formato correto e código correto, será válido;

CASOS DE TESTE	ENTRADA PREVISTA	SAÍDA PREVISTA
CT01 – login e senha vazios	Login: "" Senha: ""	“Login e /ou Senha incorretos”
CT02 – login e senha incorretos	Login: “alunoincorreto@ulife.com.br” Senha: “senhaincorreta”	“Login e /ou Senha incorretos”
CT03 – login correto e senha	Login: “alunocorreto@ulife.com.br”	“Login e /ou Senha incorretos”

incorreta	Senha: "senhaincorreta"	
CT04 – login incorreto e senha correta	Login: "alunoincorreto@ulife.com.br" Senha: "senhacorreta"	"Login e /ou Senha incorretos"
CT05 – login e senha corretos	Login: "alunocorreto@ulife.com.br" Senha: "senhacorreta" Código SMS: "240819"	Gerar o código dinâmico para a validação em duas etapas e enviar via SMS para o celular do usuário. (SMS correto - "240819")
CT06 – código SMS vazio	Login: "alunocorreto@ulife.com.br" Senha: "senhacorreta" Código SMS: ""	"Login não autorizado!"
CT07 – código SMS com formato inválido	Login: "alunocorreto@ulife.com.br" Senha: "senhacorreta" Código SMS: "240*j/"	"Login não autorizado!"
CT08 – código SMS incorreto	Login: "alunocorreto@ulife.com.br" Senha: "senhacorreta" Código SMS: "918042"	"Login não autorizado!"
CT09 – código SMS correto	Login: "alunocorreto@ulife.com.br" Senha: "senhacorreta" Código SMS: "240819"	"Login realizado com sucesso." Acesso liberado.