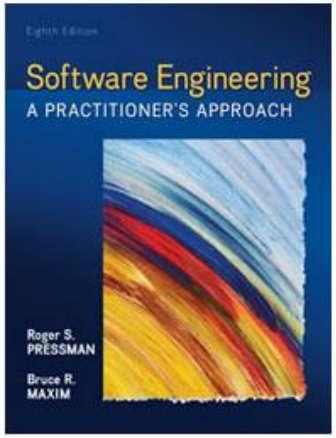
Unidade 14
Técnicas de Modelagem de Teste

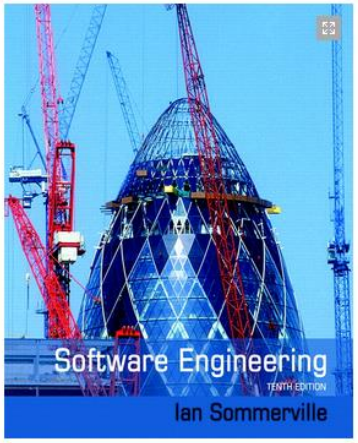
Prof. Aparecido V. de Freitas
Doutor em Engenharia
da Computação pela EPUSP

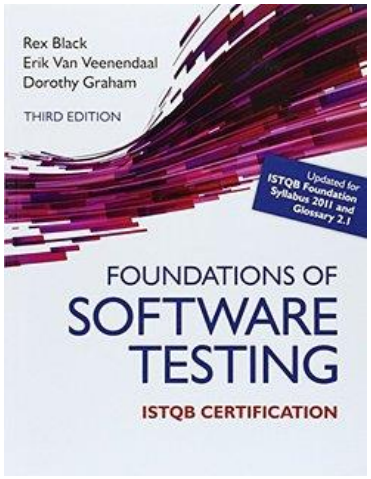
Bibliografia

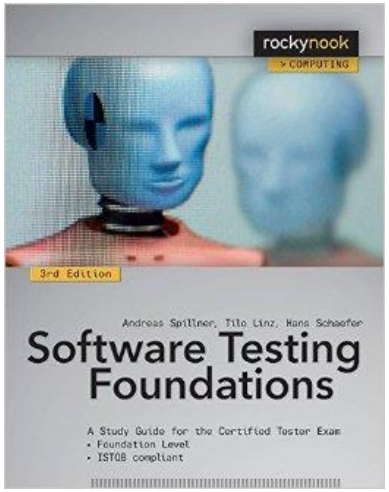
- **Software Engineering – A Practitioner's Approach – Roger S. Pressman – Eight Edition – 2014**
- **Software Engineering – Ian Sommerville – 10th edition - 2015**
- Engenharia de Software – Uma abordagem profissional – Roger Pressman - McGraw Hill, Sétima Edição - 2011
- Engenharia de Software – Ian Sommerville – Nona Edição – Addison Wesley, 2007
- **Software Testing Foundations – 4th edition – Andreas Spillner, Tito Linz, Hans Schaefer – 2014**
- **Foundations of Software Testing – Third Edition – Rex Black – ISTQB Certification**
- **Syllabus – ISTQB – International Software Testing Qualifications Board**



Software Engineering: A Practitioner's Approach, 8/e

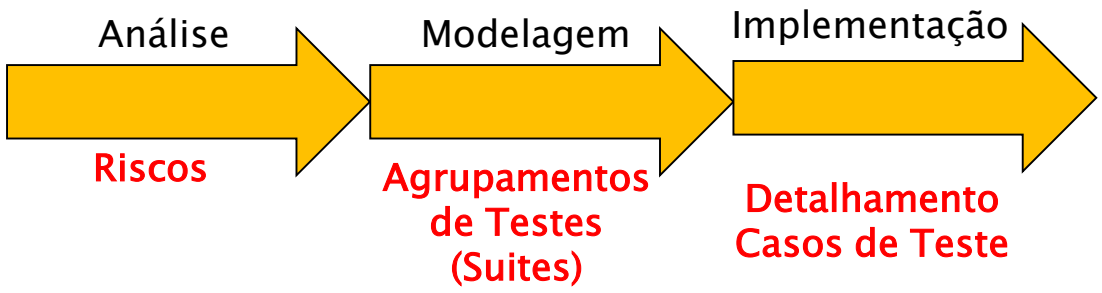






Fases do Desenvolvimento do Teste

- Desenvolvimento do Teste geralmente progride em fases:
  - **Análise** (**riscos** de qualidade) – Determina-se o que deve ser testado.
  - **Projeto de teste em alto nível** (Modelagem do Teste) – Define-se os nomes dos casos de teste e suítes de teste (conjuntos de casos de teste).
  - **Projeto de teste em baixo nível** (**implementação**) – **Detalhamento** dos casos de teste, dados de teste e scripts de teste.



Risco de Qualidade

- Risco é algo que **pode** dar errado. Representa a possibilidade de um resultado indesejado ou inesperado;
- Corresponde a um valor entre **0** e **1**;
- Risco é sempre projetado num tempo futuro;
- Risco de qualidade corresponde à possibilidade da ocorrência de uma **falha** durante a execução do programa, gerando – por consequência – insatisfação do cliente;

Como se analisa Riscos de Qualidade

- **Informal** – Baseia-se na experiência do profissional de testes;
- **ISO 9126** – Baseia-se em seis características de qualidade FRUEMP (funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade);
- **Stakeholder** – Baseia-se em consenso com o **cliente**.
- Pode ser determinado por meio de uma reunião de **brainstorming** da equipe de projeto.

Exemplo de Matriz de Risco

- Riscos de qualidade comprometem a satisfação do usuário;
- **Risco Técnico**: Possibilidade da ocorrência do problema;
- **Risco de Negócio**: Impacto do problema;
- **Prioridade do Risco** = **Risco Técnico** * **Risco de Negócio**

1–5 = Extensivo (**muito grave**)
6 –10 = Abrangente
11–15 = Ocasional
16–20 = Oportuno
21–25 – Relata bugs

1 = Muito alto
2 = Alto
3 = Médio
4 = Baixo
5 = Muito Baixo

IEEE 829

- Norma útil para a **documentação de testes de software**;
- Utilizada como base para a criação da **Especificação de Modelagem de Teste**, documento que descreve uma **coleção** de casos de teste (suíte) incluindo os recursos a serem testados, a identificação do teste, critérios passa/falha, etc
- A norma também auxilia na escrita da **Especificação de Caso de Teste**, documento que descreve os detalhes de um caso de teste, incluindo itens de teste (**o que** deve ser entregue e testado), especificações de entrada, especificações de saída, etc.
- A norma também auxilia na escrita da **Especificação de Procedimento de Teste**, documento que descreve **como** executar um ou mais casos de teste, incluindo basicamente os **passos** do procedimento.

Cobertura do Caso de Teste (Rastreabilidade)

- Corresponde a uma matriz no qual se mede ou se correlaciona testes mediante áreas de preocupação;
- Usada como meio de medir e melhorar os testes.
- Esta é uma técnica comumente usada para garantir cobertura completa do teste.

Matriz de Rastreabilidade

				Caso de teste									
	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	2.1	2.2	2.3	2.4	2.5	Total
Espec														
1.1	1	1		1		1		2						5
1.2	2			2	1				1		2			5
1.3														0
1.4		2									2			2
1.5	1			1				1	1					4
1.6						2								1
1.7			2		1		2			2				4
1.8					2							1		2
1.9	1			1							2			3
1.10		1					1	1			1			4
Total	4	3	1	4	3	2	2	3	2	1	4	1	0	

0: nada; 1 = indireto ; 2 = direto

Exemplo: Tela de Login (1)

Matriz de Rastreabilidade

	Caso de teste													
	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	2.1	2.2	2.3	2.4	2.5	Total
Espec														
1.1	1	1		1		1		2						5
1.2	2			2	1				1		2			5
1.3														0
1.4		2									2			2
1.5	1			1				1	1					4
1.6						2								1
1.7			2		1		2			2				4
1.8					2							1		2
1.9	1			1							2			3
1.10		1					1	1			1			4
Total	4	3	1	4	3	2	2	3	2	1	4	1	0	

0: nada; 1 = indireto ; 2 = direto

Qual o requisito que não foi testado ?

Qual o caso de teste que ainda não testou nenhum requisito ?

Qual o requisito que tem uma maior cobertura de testes ?

Exemplo: Tela de Login (1)

Modelagem de Teste

- O propósito da Modelagem de Teste é identificar as **condições de teste** e os **casos de teste**.

- Uma **condição de teste** é um item ou evento de um componente ou sistema que pode ser verificado por um ou mais casos de teste, por exemplo: função, transação, característica, atributo de qualidade ou elemento estrutural.

- **Caso de Teste** é um conjunto de valores de entrada, precondições de execução, resultados esperados e pós-condições de execução desenvolvidas para um determinado objetivo ou condição de teste, tais como para exercitar o caminho de um determinado programa ou verificar o atendimento a um requisito especifico.

Categorias das Técnicas de Modelagem de Teste

- Técnica baseada em **Especificação**: Condições de teste e Casos de Teste são derivados com base na análise da Documentação (Base de Teste). Isto inclui testes funcionais e não-funcionais. Não se leva em conta a estrutura interna do código. Também é chamada **Técnica de Caixa Preta**.

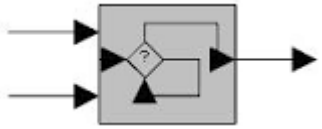
- Técnica baseada em **Estrutura**: Considera a estrutura interna do código do software. Também é chamada **Técnica de Caixa Branca**. Há a necessidade de acesso ao código fonte da aplicação.

- Técnica baseada em **Experiência**: Testes são criados com base no entendimento do sistema e experiência anterior das pessoas envolvidas no teste.

Teste baseado em Especificação ou Caixa Preta

- Casos de Teste são derivados de **modelos de especificação do software** a ser construído;
- Teste muito empregado pelas empresas de software;
- Utilizam os seguintes conceitos:
  - **Partição de Equivalência;**
  - **Análise do Valor Limite;**
  - **Diagramas de Transição de Estados;**
  - **Tabelas de Decisão;**
  - **Diagramas de Caso de Uso.**

Teste baseado em Estrutura ou Caixa Branca



- Casos de Teste são derivados a partir da estrutura do software (**código**).
- Consideram as técnicas de **cobertura de sentenças**, **cobertura de decisão** e outras técnicas baseadas em estrutura.
- Podem estar associados a:
  - **Nível de Componente**: A estrutura é o próprio código (comandos, decisões e desvios);
  - **Nível de Integração**: A estrutura pode ser uma árvore de chamadas (Diagrama em que um módulo chama outro);
  - **Nível de Sistema**: A estrutura pode ser uma estrutura de menu, processos de negócios ou estrutura das páginas Web.


```
if   (a == b) {
        x=1;
}
```

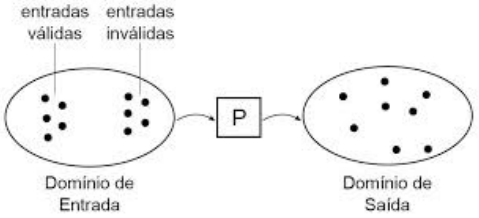
Sentença (comando): 1 teste
Decisão (desvio): 2 testes

Teste baseado na Experiência

- São testes derivados da **intuição** e **conhecimento** dos testadores através de sua experiência em aplicações e tecnologias similares.
- Podem produzir amplas variedades e graus de eficiência, dependendo da experiência do testador;
- Possivelmente a técnica mais amplamente aplicada é supor (**adivinhar**) onde estão os erros.
- Uma abordagem estruturada da técnica de dedução de erros é enumerar uma lista de possíveis erros e construir testes com objetivo de atacar/cobrir estes erros. Esta abordagem é chamada de **Ataque de Falha**.
- É uma abordagem muito usual, em locais onde a especificação é rara ou inadequada e existe uma grande pressão por conta de prazos.
- Exemplo: **Teste Exploratório**

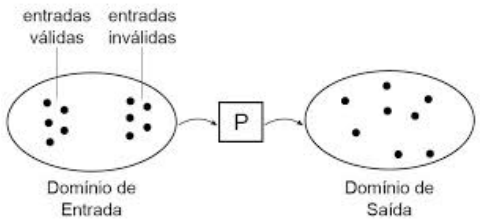
Partição de Equivalência

- As entradas do software são divididas em **grupos** que tenham um **comportamento similar**, podendo ser tratados na mesma forma.
- **Partições** ou **classes de equivalência** podem ser encontradas em dados **válidos** e **inválidos**. Podem ser aplicáveis à todos os níveis de teste



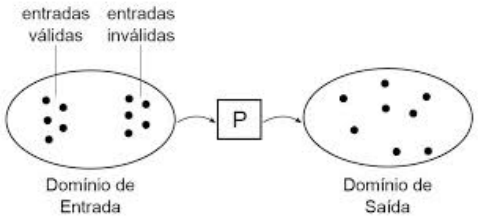
Partição de Equivalência - Exercício

- Um titular de um convênio de seguros deve ter entre **18** e **65** anos. Avalie os casos de teste abaixo:
  - Caso de Teste 1: 15 , 33 , 70
  - Caso de Teste 2: 15, 17, 33 , 70
  - Caso de Teste 3: 35 , 70



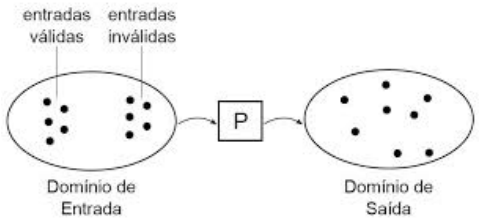
Partição de Equivalência - Exemplo

- Uma câmara frigorífica precisa manter medicamentos sensíveis entre -9,5ºC e -3,0ºC. Avalie os casos de teste abaixo:
  - Caso de Teste 1: -10,0 -7,0 -1,0
  - Caso de Teste 2: -10,0 -11,0 -2,0 0,0
  - Caso de Teste 3: -8,0 +4,0



Partição de Equivalência - Exemplo

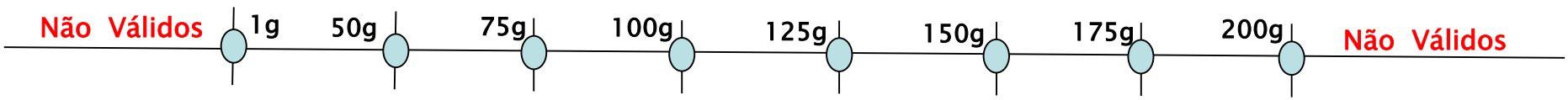
- Produtos são classificados em uma escala crescente de risco que vai de J a P. Avalie os casos de teste abaixo:
  - Caso de Teste 1: A , K , R
  - Caso de Teste 2: B , L
  - Caso de Teste 3: M , Q , R , S
  - Caso de Teste 4: I , J , Q
  - Caso de Teste 5: I, J , P



Partição de Equivalência - Exemplo

Partição de Equivalência - Exemplo

- A Agência dos Correios cobra R$ 0,25 centavos por carta até 50g. A cada 25g adicionais são cobrados R$ 0,50 centavos. As cartas podem ser enviadas com no máximo 200g. Quais testes você faria por partição de equivalência ?
  - Caso de Teste 1: -10g, 45g, 60g, 80g, 110g, 130g, 160g, 190g, 210g



Testes ocultos: 0g e 2000g (limite da balança). Enviar mensagem de erro!

Partição de Equivalência - Exemplo

- Um funcionário é isento de imposto até R$ 2.000,00. Nos próximos R$ 3.500,00 paga 10% de imposto. Acima disto, paga 27,5%. Quais testes você faria por partição de equivalência ?
  - Caso de Teste 1: R$ -50,00 ; R$ 1500,00 ; R$ 2300,00 ; R$ 8000,00

Análise do Valor Limite

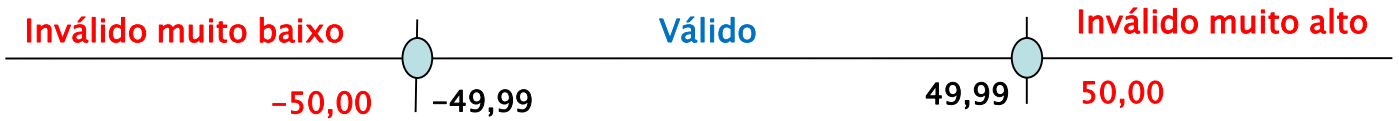
- Um refinamento da Partição de Equivalência que seleciona as **margens** ou **pontos terminais** de cada partição para teste.
- Também chamado Teste de **Fronteira**, Teste **Limítrofe** ou Teste de **Limites**.
- O comportamento nos limites de uma Partição de Equivalência corresponde à maior probabilidade de estar **incorreto**. Portanto, limites são áreas onde testes são mais propensos a indicar defeitos.
- Os valores limite de uma partição são seu **máximo** e seu **mínimo**.
- Testes podem ser projetados para cobrir tanto valores válidos quanto inválidos.
- Um valor limite para uma partição válida é um valor limite válido.
- Um valor limite para uma partição inválida é um valor limite inválido.

Análise do Valor Limite - Exemplo

- Considere um sistema que permite a compra de até 99 unidades.
- Portanto, o intervalo válido é de 1 a 99;
- Intervalo inválido <= 0 ou >=100 ;
- Os valores limites são 0, 1, 99 e 100 ;
- Os limites válidos são 1 e 99;
- Os limites inválidos são 0 e 100.

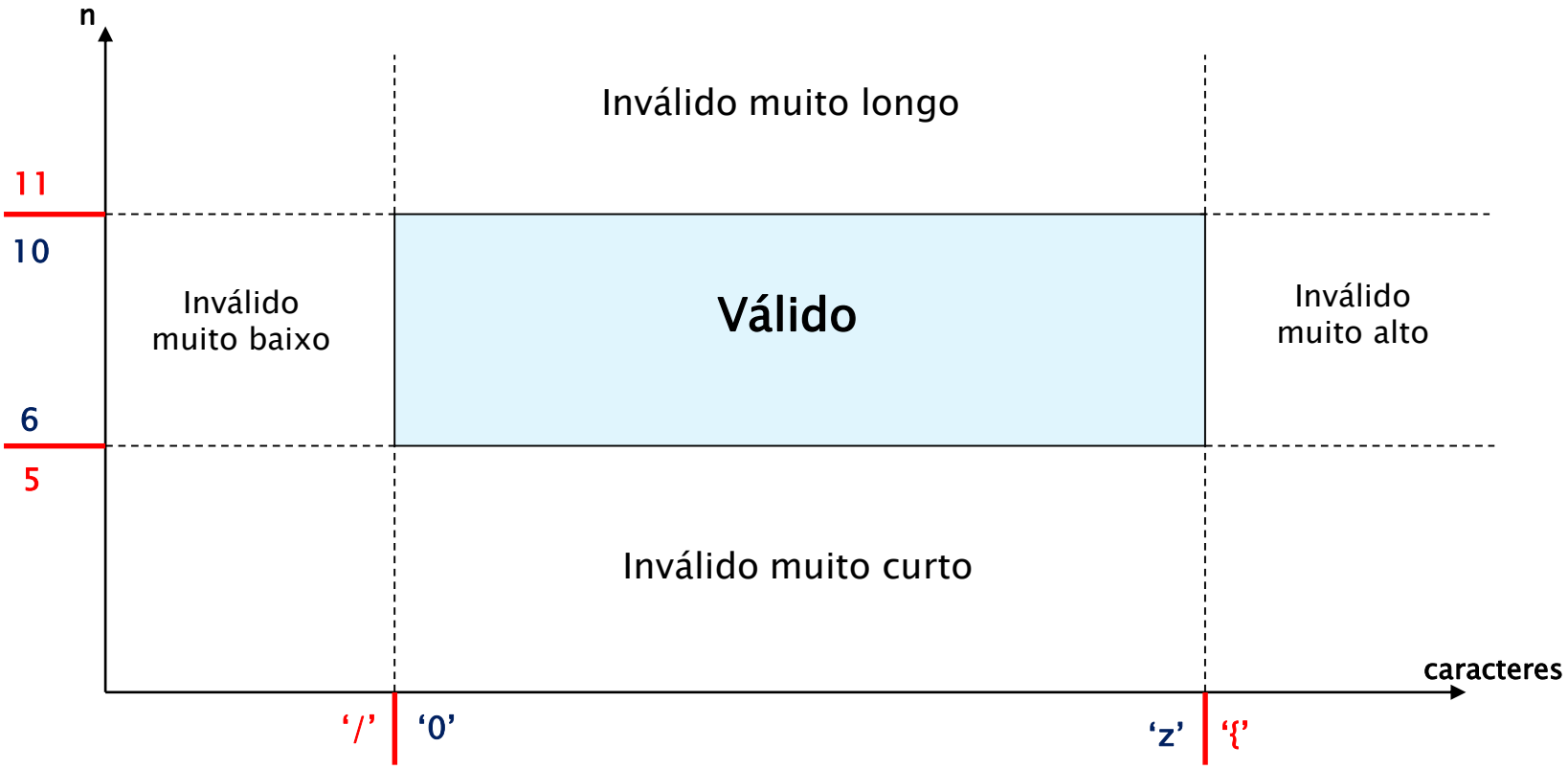
Análise do Valor Limite - Exemplo

- Qual a média de temperatura ?
- A temperatura pode oscilar de -49,99 até + 49.99;
- Limites válidos: -49,99 e 49,99;
- Limites inválidos: -50,00 e 50,00.



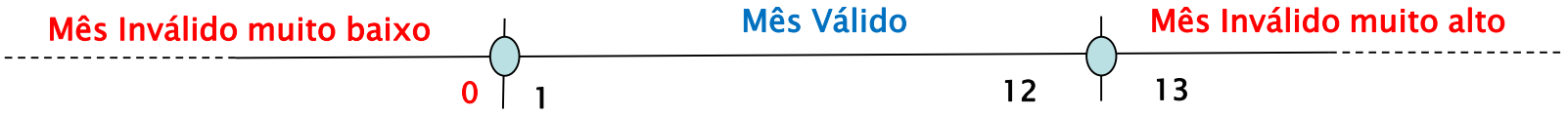
Análise do Valor Limite - Exemplo

- Senha permitida deve ser de 6 – 10 caracteres alfanuméricos



Análise do Valor Limite - Exemplo

- Entre com a data de partida do vôo: (DD/MM/AA)

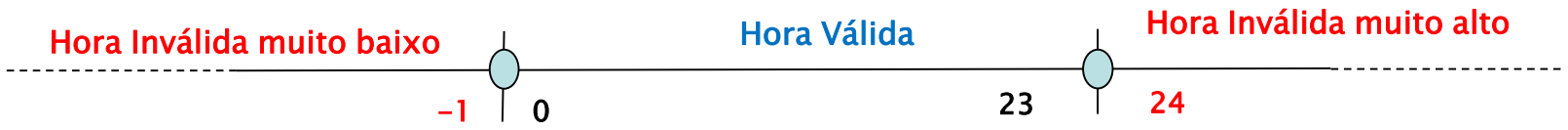


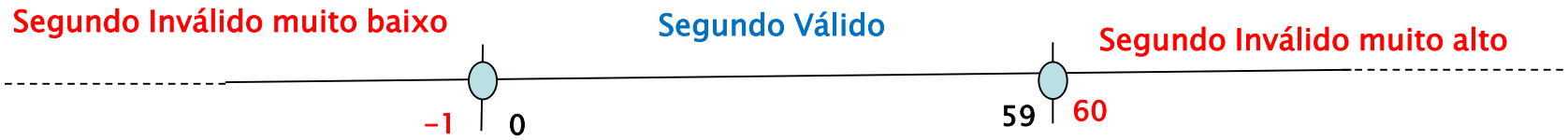
n depende do mês em onze casos e do ano em 1 caso

n e m dependem da aplicação em muitos casos

Análise do Valor Limite - Exemplo

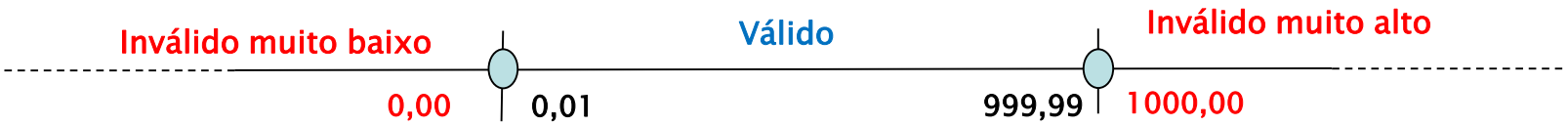
- Entre com o horário de partida do vôo: (HH:MM:SS)





Análise do Valor Limite - Exemplo

- Entre com uma oferta de preço (menor que R$ 1000,00):



Análise do Valor Limite - Exemplo

- Um titular de um convênio de seguros deve ter entre 18 e 65 anos. Avalie os casos de teste abaixo:

Valores Limites: 17 , 18, 65 e 66

Análise do Valor Limite - Exemplo

- Uma câmara frigorífica precisa manter medicamentos sensíveis entre -8,5ºC e -4,0ºC. Quais os valores limites ?

Valores Limites: -8,6 -8,5 -4,0 -3,9

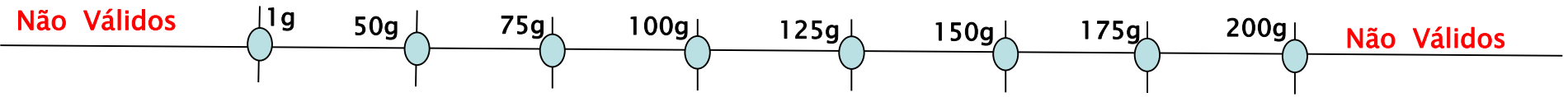
Análise do Valor Limite - Exemplo

- Produtos são classificados em uma escala crescente de risco que vai de J a P. Quais os valores limites ?

Valores Limites: I , J , P, Q

Análise do Valor Limite- Exemplo

- A Agência dos Correios cobra R$ 0,25 centavos por carta até 50g. A cada 25g adicionais são cobrados R$ 0,50 centavos. As cartas podem ser enviadas com no máximo 200g. Quais os valores limite ?

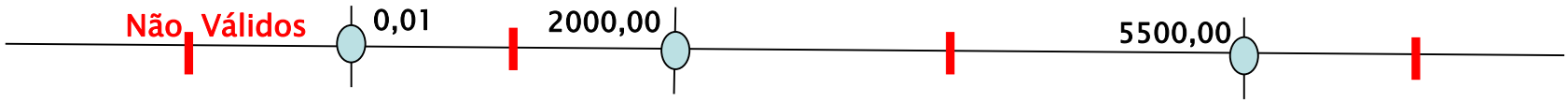


Valores Limites: -10,1,50,51,75,76,100,101,125,126,150,151,175,176,200,201

Testes ocultos: 0g e 2000g (limite da balança). Enviar mensagem de erro!

Análise do Valor Limite - Exemplo

- Um funcionário é isento de imposto até R$ 2.000,00. Nos próximos R$ 3.500,00 paga 10% de imposto. Acima disto, paga 27,5%. Quais os valores limite ?



Valores Limites: 0,00 0,01 2000,00 2001,00 5500,00 5501,00

Exercício

- Uma fábrica produz chapas de aço de **10** a **15** mm, **20** a **25**mm e **30** a **35** mm.

A) Quantas partições de equivalência deveriam ser testadas ?

B) Que valores você testaria ?

C) Que valores você testaria por valor limite ?

Exercício

- Uma fábrica produz chapas de aço de **10** a **15** mm, **20** a **25**mm e **30** a **35** mm.

Não Válidos	Válidos	Não Válidos	Válidos	Não Válidos	Válidos	Não Válidos
9	10 15	16 19	20 25	26 29	30 35	36

A) Quantas partições de equivalência deveriam ser testadas ?

Partições Válidas: [10,15] , [20,25] , [30,35]
Partições Inválidas: <10 , [16,19] , [26,29] , > 35

Resposta: **7**

B) Que valores você testaria ?

5, 11 , 18, 22 , 28, 33 , 40

C) Que valores você testaria por valor limite ?

9, 10 , 15, 16 , 19, 20, 25 , 26 , 29, 30, 35 , 36

Tabela de Decisão

- Boa alternativa para capturar requisitos de softwares que contém condições lógicas e para documentar o comportamento interno do software;
- As condições de entrada e ações são declaradas de uma forma que possam ser entendidas, como verdadeiro ou falso (Booleano);
- A tabela de decisão contém as condições que disparam as ações;
- Cada coluna da tabela corresponde a uma regra de negócio que define uma única combinação de condições que resulta na execução de ações associadas com aquela regra;

		1	2	3	4	5	6
Condições	Mais de 25% de faltas (S : sim / N : não)	S	S	S	N	N	N
	Média (1 : >=7,0 / 2 : entre 3,0 e 6,9 / 3 : < 3,0)	1	2	3	1	2	3
Ações	Aprovado direto				X		
	Exame final					X	
	Reprovado	X	X	X			X
	Matrícula para o ano seguinte	X	X	X	X		X

Tabela de Decisão - Exemplo

Condição	1	2	3	4	5
Cartão válido	N	S	S	S	S
Senha válida	-	N	N	S	S
Senha inválida=3	-	N	S	N	N
Saldo OK	-	-	-	N	S
Ação					
Rejeita cartão	S	N	N	N	N
Redigita senha	N	S	N	N	N
Prende cartão	N	N	S	N	N
Redigita opção	N	N	N	S	N
Fornece dinheiro	N	N	N	N	S

Obs. Na tabela acima será necessário um teste para cada coluna da tabela

Teste de Transição de Estado

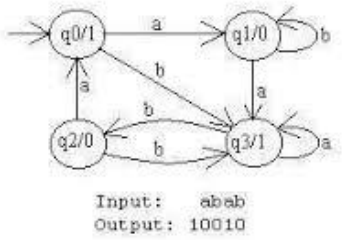
- O comportamento de um software pode ser representado por um Diagrama de Transição de Estados (**DTE**), que por sua vez pode ser representada por uma **Tabela de Transição de Estado**.

- Uma tabela de transição de estados exibe a relação entre estados e entradas, e pode destacar possíveis transições inválidas.

- Os testes podem ser construídos para cobrir uma sequência típica de status, cobrir todos os estados, exercitar todas as transições, exercitar uma sequência específica de transições ou testar transições inválidas.

- A técnica é adequada para testar fluxos de dados de telas de diálogos, por exemplo em aplicações web.

- O teste se resume a verificar os caminhos e respectivas ações especificadas no **DTE**.

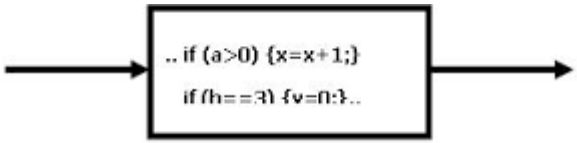


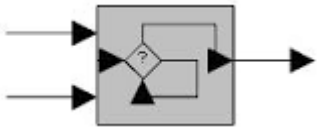
DTE - Exemplo

Caminho Feliz...

Técnicas Baseadas na Estrutura (Caixa Branca)

- Teste de estrutura ou **caixa-branca** baseia-se na estrutura do software;
- Podem ser:
  - **Nível de Componente**: A estrutura é o próprio código. Exemplo: comandos, desvios e decisões.
  - **Nível de Integração**: A estrutura pode ser uma árvore de chamadas (um diagrama em que um módulo efetua chamadas em outros módulos).
  - **Nível de Sistema**: A estrutura pode ser uma estrutura de menus, processos de negócios ou estruturas de páginas Web.





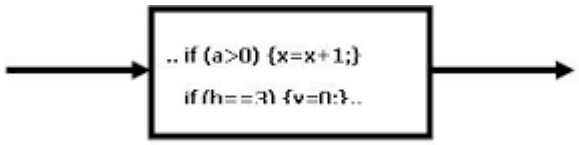
Cobertura de Código

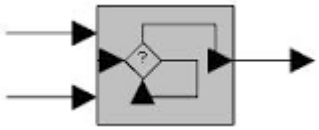
- É uma medida, referente ao número de itens (código) testados, pelo total de itens, multiplicado por 100%.

$$\text{Cobertura} = \frac{\text{número de itens exercitados}}{\text{total de itens}} \times 100\%$$

Níveis de Cobertura de Código

- **Cobertura de comandos ou sentenças**: Todo **comando** (sentença) executado;
- **Cobertura de desvio (decisão)**: Todo desvio (decisão) tomado em cada direção, verdadeiro ou falso;
- **Cobertura de condição**: Toda combinação de condições (e/ou) verdadeiras ou falsas avaliadas (isto é, a tabela verdade inteira);
- **Cobertura de laço**: Todos os caminhos de laço executados zero, uma, e múltiplas (idealmente, máximo) vezes;

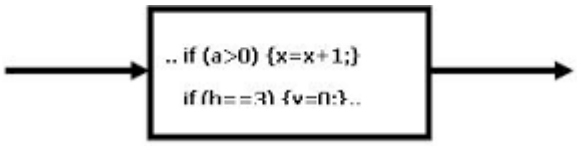


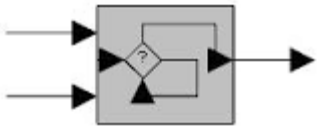


Teste e Cobertura de Comandos

- Definir casos de teste que executem todos os comando pelo menos uma vez **(deve passar por todos os comandos do grafo)**
- **Métrica**: Número de comandos cobertos;
- Cobertura de comando é medida, de acordo com o número de comandos exercitados dividido pelo total de comandos, multiplicado por 100%.

$$\text{Cobertura de comando} = \frac{\text{número de comandos exercitados}}{\text{total de comandos}} \times 100\%$$

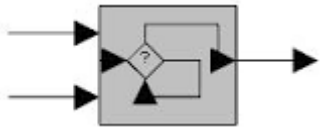




Exercício – Cobertura de Comandos

- Quantos testes são necessários para a cobertura total de **Comandos** ?

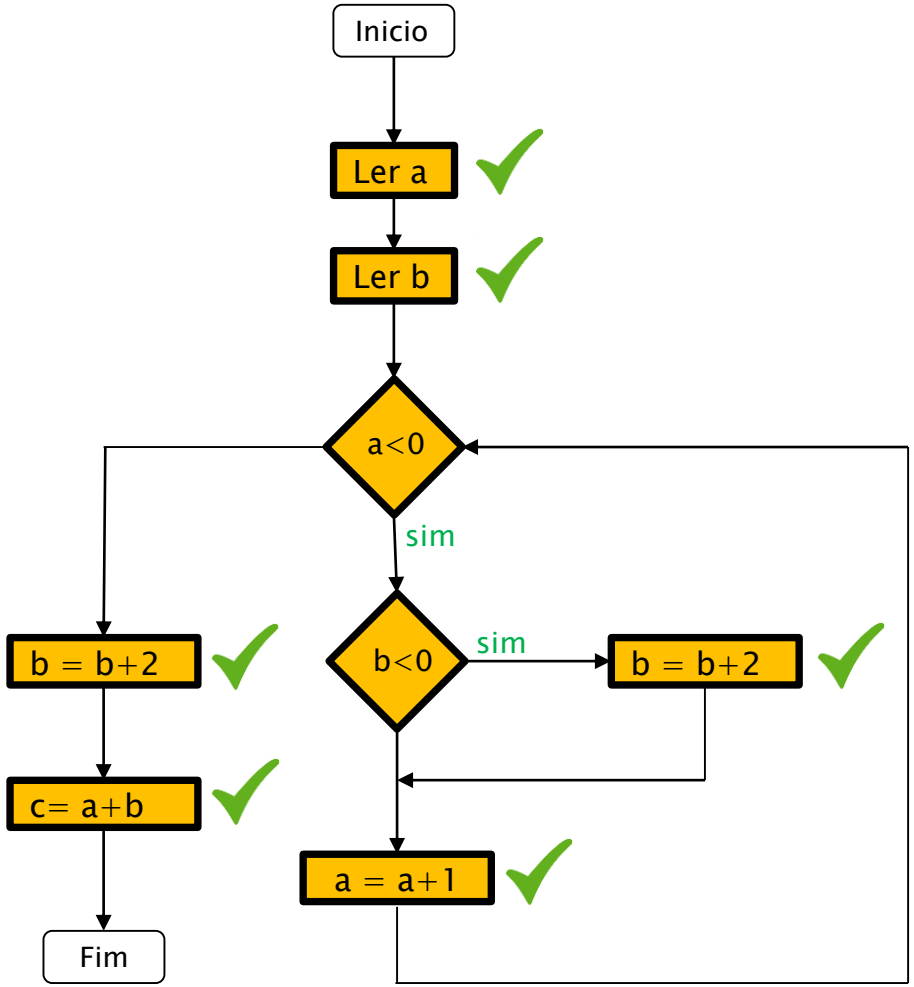
```
package maua;
public class ISTQB_01 {
        public static void main(String[] args) {
                Integer a = Integer.parseInt(args[0]);
                Integer b = Integer.parseInt(args[1]);
                Integer c;
                while ( a < 0 )  {

                        if ( b < 0 ) {

                                b = b + 2;

                        }

                        a = a + 1;

                }

                c = a + b;

                System.out.println(c);

        }

}
```



Exercício – Cobertura de Comandos

- Quantos testes são necessários para a cobertura total de Comandos ?

```
package maua;
public class ISTQB_01 {
        public static void main(String[] args) {
                Integer a = Integer.parseInt(args[0]);
                Integer b = Integer.parseInt(args[1]);
                Integer c;
                while ( a < 0 )  {

                        if ( b < 0 ) {

                                b = b + 2;

                        }

                        a = a + 1;

                }

                c = a + b;

                System.out.println(c);

        }

}
```



Total de Comandos = 6

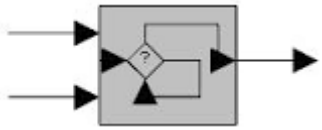
Exercício – Cobertura de Comandos

- Quantos testes são necessários para a cobertura total de Comandos ?

```
package maua;
public class ISTQB_01 {
        public static void main(String[] args) {
                Integer a = Integer.parseInt(args[0]);
                Integer b = Integer.parseInt(args[1]);
                Integer c;
                while ( a < 0 )  {

                        if ( b < 0 ) {

                                b = b + 2;

                        }

                        a = a + 1;

                }

                c = a + b;

                System.out.println(c);

        }

}
```

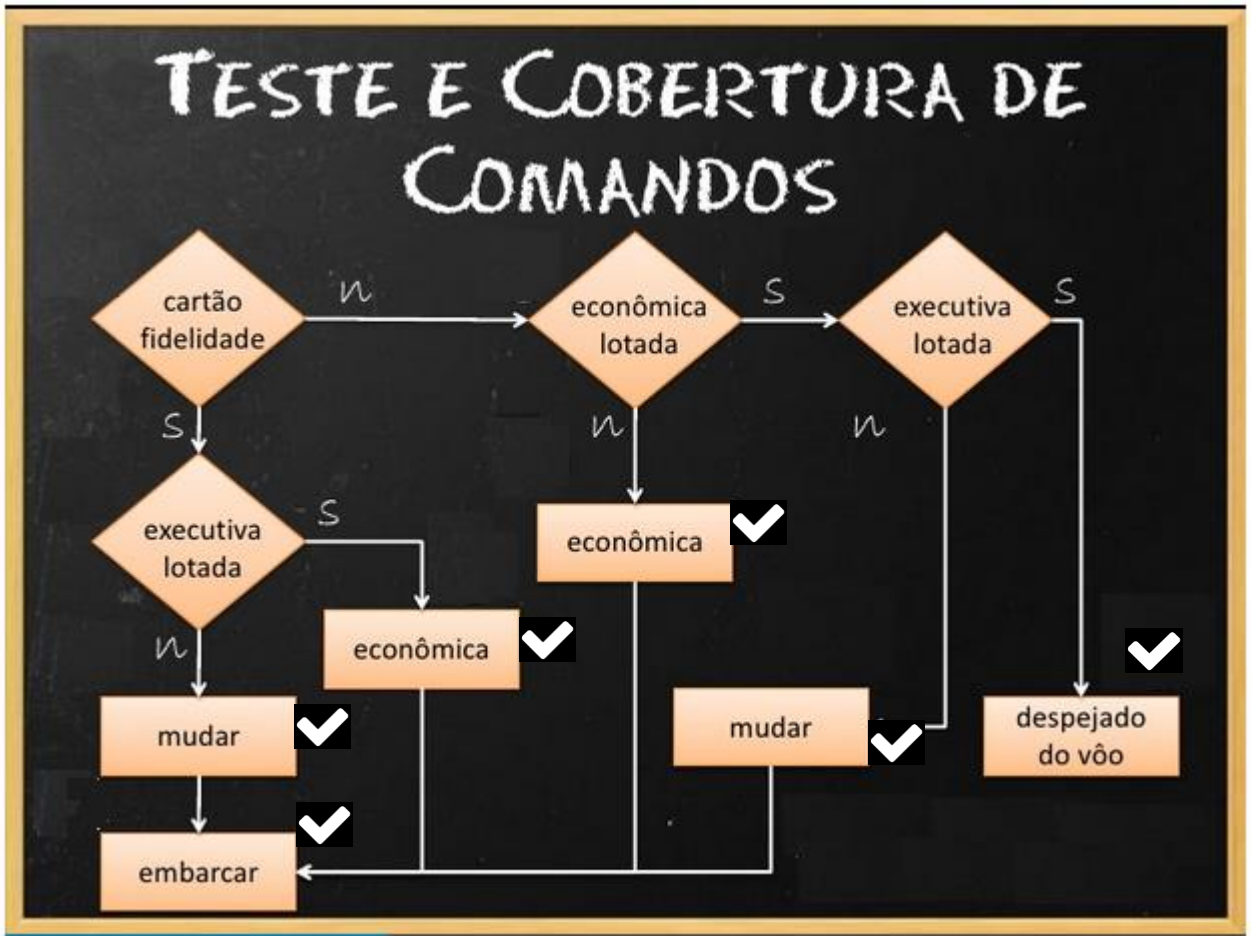
a	b	c
-2	-2	0

Resposta: Um teste é suficiente !



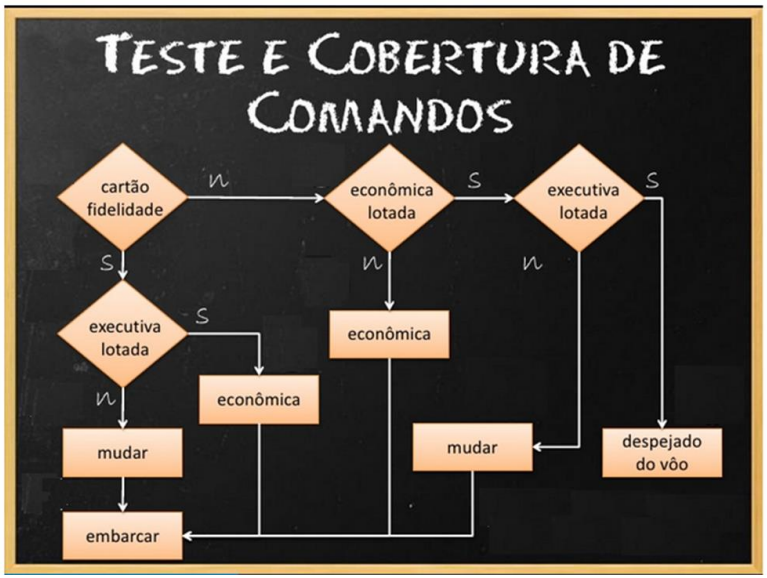
Questão – CTFL – ISTQB

- Se você estiver voando com um bilhete da classe econômica, há uma possibilidade de você conseguir mudar para a classe executiva. Principalmente se você tiver um cartão fidelidade da companhia aérea. Se você não tiver o cartão fidelidade (CF), há a possibilidade de você ser "despejado" do vôo se ele estiver lotado e você chegar atrasado.



Questão – CTFL - ISTQB

- Três testes foram executados:
  - ✓ **Teste 1**: O passageiro tem o CF e mudou para a classe executiva e embarcou;
  - ✓ **Teste 2**: O passageiro não tem o CF e permaneceu na classe econômica;
  - ✓ **Teste 3**: O passageiro foi "despejado" do vôo.
- Qual é a cobertura de comandos obtida para esses três testes ?



Teste de Cobertura de Decisão

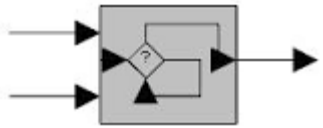
- Uma decisão é um **IF**, um loop (por exemplo: **do-while** ou **repeat-until**), ou um **CASE**, no qual existem duas ou mais possibilidades de saídas ou resultados a partir de uma decisão.

- Teste de decisão é uma forma de teste de Controle de Fluxo, já que gera um fluxo específico através dos pontos de decisões.

- Os testes devem cobrir **cada** saída possível de um nodo que tenha uma condição. (**Métrica**: número de arestas cobertas).

$$\text{Cobertura de decisão} = \frac{\text{número de decisões exercitados}}{\text{total de decisões}} \times 100\%$$

Exercício – Cobertura de Decisão



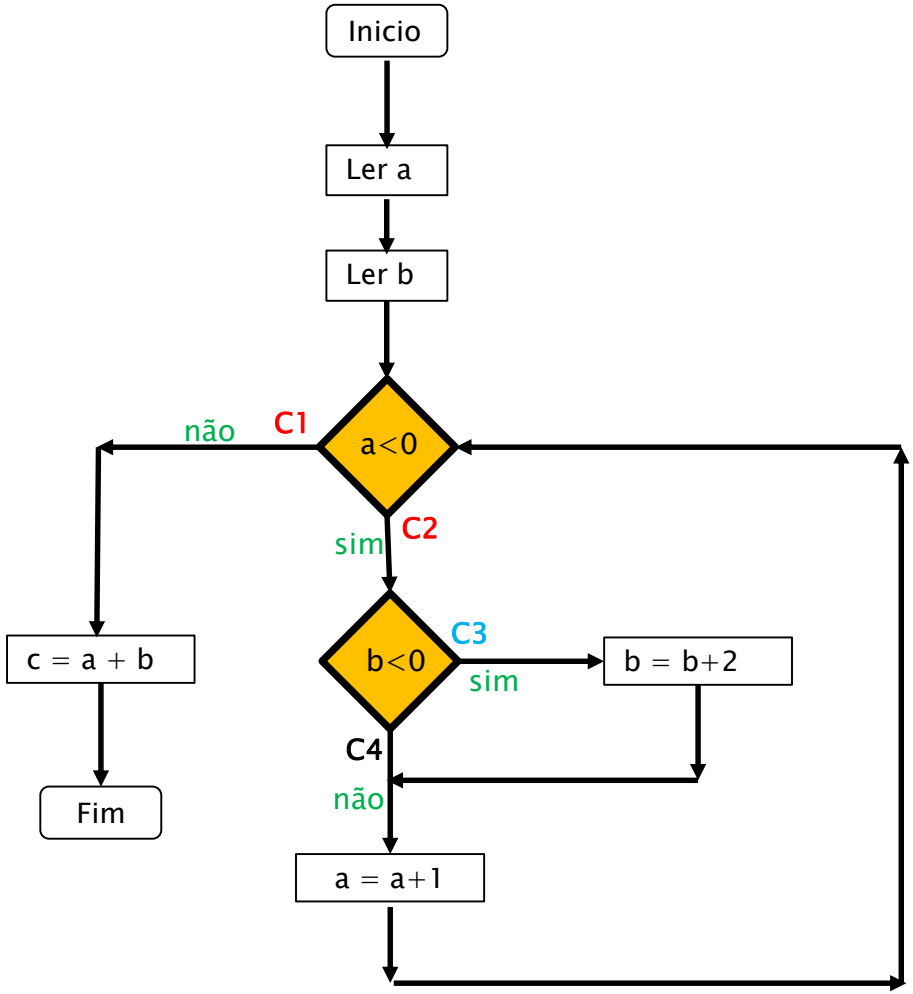
```
package maua;
public class ISTQB_01 {
        public static void main(String[] args) {
                Integer a = Integer.parseInt(args[0]);
                Integer b = Integer.parseInt(args[1]);
                Integer c;
                while ( a < 0 )  {

                        if ( b < 0 ) {

                                b = b + 2;

                        }

                        a = a + 1;

                }

                c = a + b;

                System.out.println(c);

        }

}
```

- Quantos testes são necessários para a cobertura total de **Decisão** ?

Exercício – Cobertura de Decisão

- Um testador executou um teste com os casos de teste: a = **-4** e b = **5**
- Qual a cobertura de decisão do teste?

```
package maua;
public class ISTQB_01 {
        public static void main(String[] args) {
                Integer a = Integer.parseInt(args[0]);
                Integer b = Integer.parseInt(args[1]);
                Integer c;
                while ( a < 0 )  {

                        if ( b < 0 ) {

                                b = b + 2;

                        }

                        a = a + 1;

                }

                c = a + b;

                System.out.println(c);

        }

}
```



Questão ISTQB – Cobertura de Decisão

- Tendo como base o código abaixo, quantos testes são necessários para se atingir 100% da cobertura de desvio/decisão ?

a) 1
b) 2
c) 3
d) 4

```
package maua;
public class ISTQB_2 {
public static void main(String[] args) {

        Integer x = Integer.parseInt(args[0]);
        Integer y = Integer.parseInt(args[1]);

        if   (x == 3)   {
                System.out.println("X");
                System.out.println("C1");
                if (y == 2)  {
                        System.out.println("Y");
                        System.out.println("C3");
                }
                else {
                        System.out.println("Z");
                        System.out.println("C4");
                }
        }
        else {
                System.out.println("Z");
                System.out.println("C2");
                }
        }

}
```

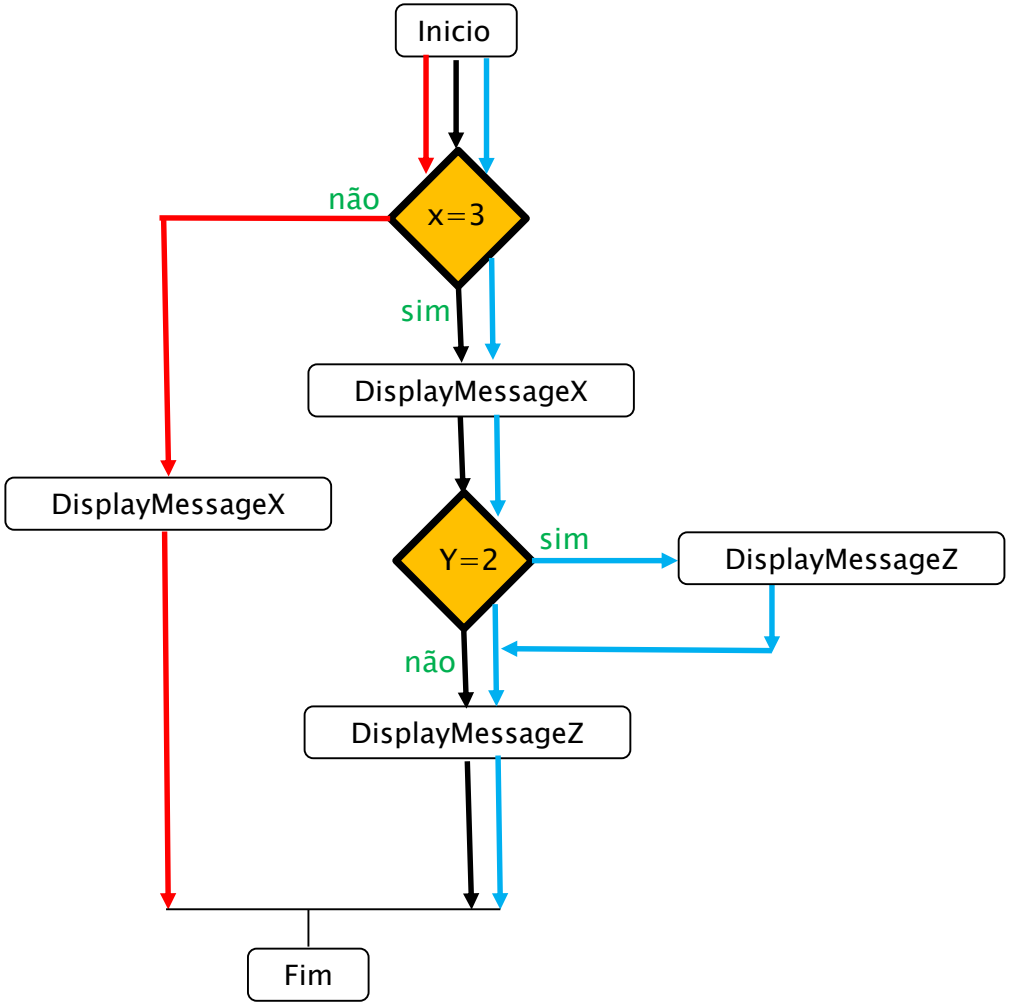
Questão ISTQB – Cobertura de Decisão

- Tendo como base o código abaixo, quantos testes são necessários para se atingir 100% da cobertura de desvio/decisão ?

a) 1
b) 2
c) 3
d) 4

x	y
3	2
0	0
3	0

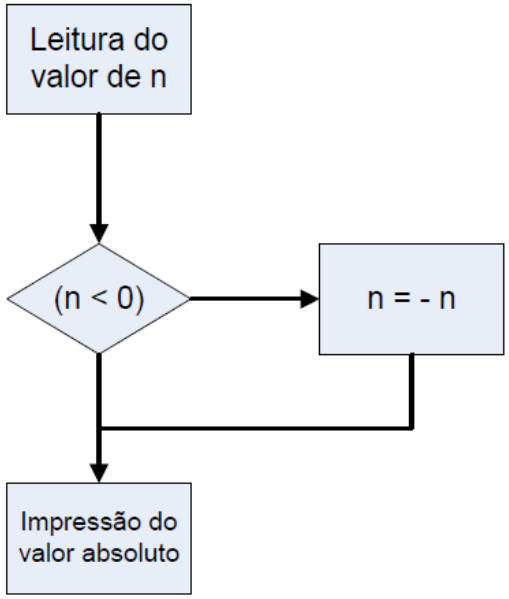
```
package maua;
public class ISTQB_2 {
public static void main(String[] args) {

        Integer x = Integer.parseInt(args[0]);
        Integer y = Integer.parseInt(args[1]);

        if   (x == 3)   {
                System.out.println("X");
                System.out.println("C1");
                if (y == 2)  {
                        System.out.println("Y");
                        System.out.println("C3");
                }
                else {
                        System.out.println("Z");
                        System.out.println("C4");
                }
        }
        else {
                System.out.println("Z");
                System.out.println("C2");
                }
        }

}
```



Exemplo de Cobertura de Código: Valor Absoluto

Considere o seguinte código, no qual é lido um número inteiro e realizada a impressão do seu valor absoluto.

```
#include <stdio.h>

int main () {
   int n;
   printf ("Digite o numero: ");
   scanf ("%d", &n);

   if (n < 0)
      n = -n;

   printf ("Valor absoluto: %d\n",n);
   return 0;
}
```

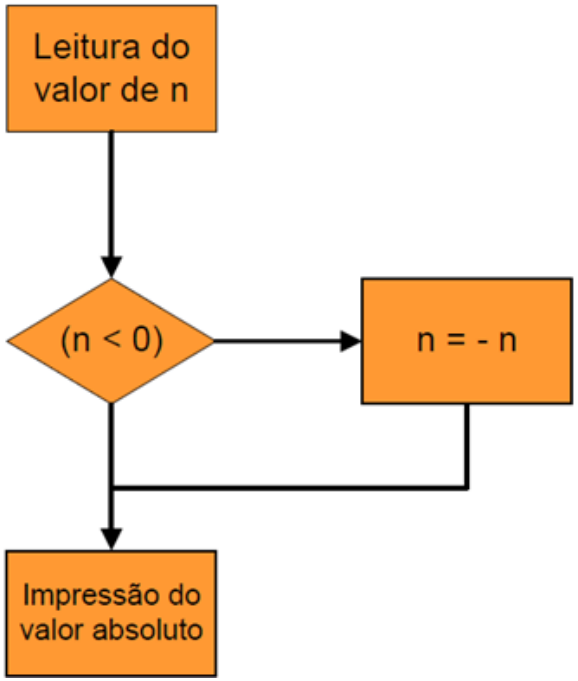


Cobertura de Comando: 1 teste
Cobertura de Decisão: 2 testes

Exemplo de Cobertura de Código: Valor Absoluto

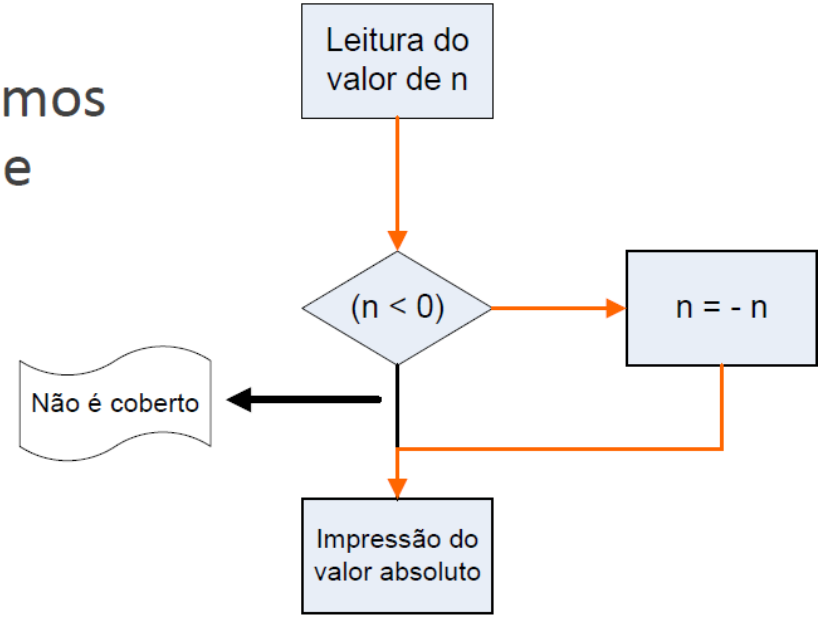
Para esse código, quais valores de teste para n precisamos para 100% de cobertura de comandos?

- Basta somente um teste com valor n < 0.



Exemplo de Cobertura de Código: Valor Absoluto

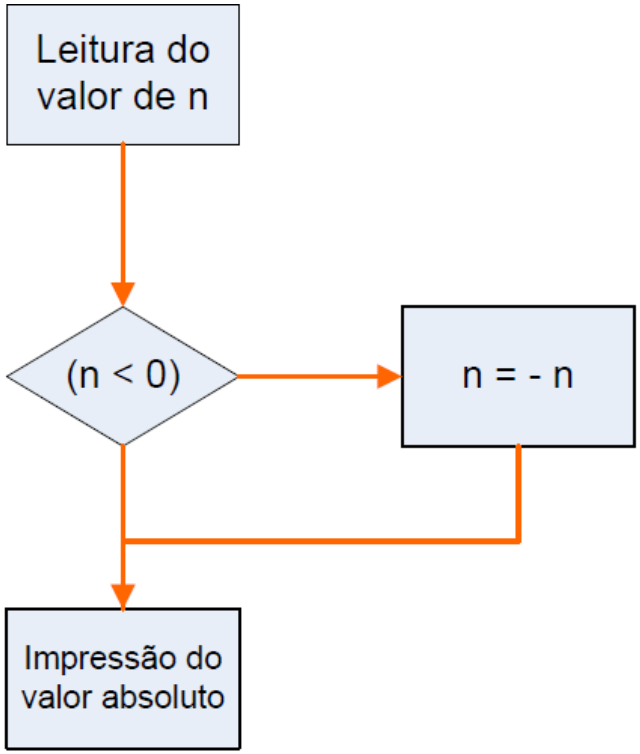
Com o teste anterior (n < 0), não conseguimos 100% de cobertura de desvio.



Exemplo de Cobertura de Código: Valor Absoluto

Quais valores de teste para n precisamos para 100% de cobertura de desvios?

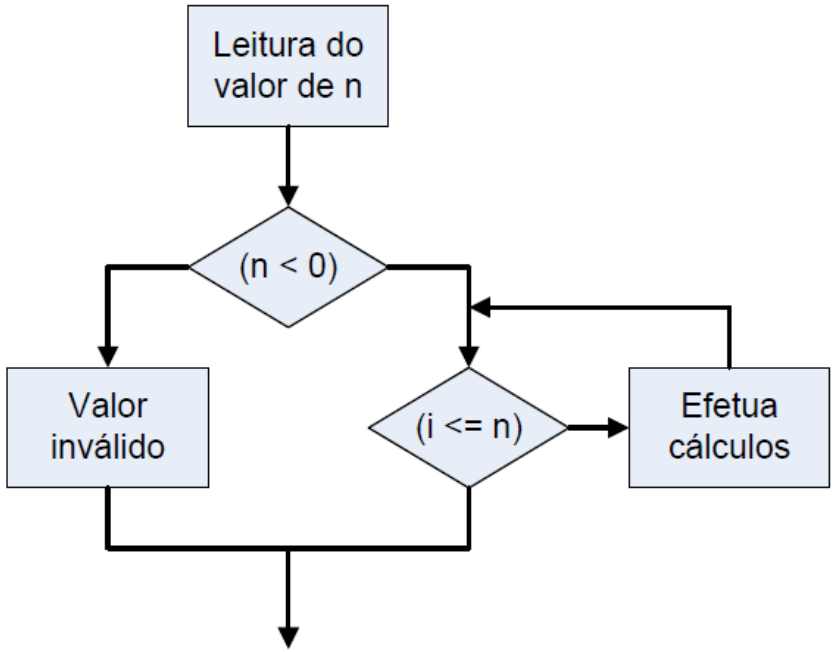
- Precisamos de dois testes, n < 0 e n >= 0



Exemplo de Cobertura de Código: Fatorial

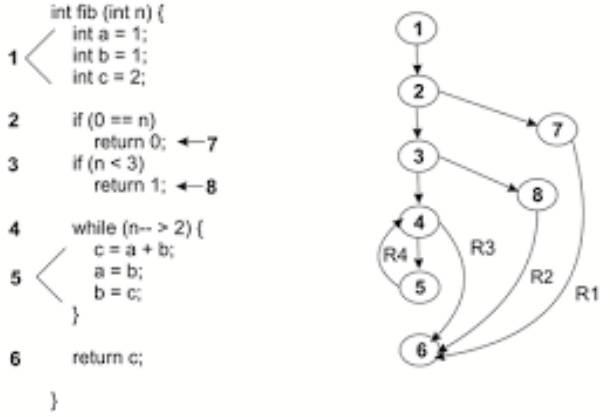
Considere o seguinte código para calcular o fatorial de um número inteiro dado.

```
#include <stdio.h>

int main () {
  int n, fatorial, i;
  printf ("Digite o numero: ");
  scanf ("%d", &n);

  if (n < 0) {
    printf ("Invalido: %d\n", n);
    n = -1;
  }

  else {
    fatorial = 1;
    for (i = 1; i <= n; i++)
      fatorial *= i;
    printf ("%d! = %d\n", n, fatorial);

  }
  return n;
}
```



Complexidade Ciclomática de McCabe

- É uma métrica de software usada para indicar a **complexidade** de um código;
- Mede a quantidade de **caminhos de execução independentes** a partir de um código fonte;
- A complexidade é computada pelo grafo de fluxo de controle do programa;
- Os nós representam entradas, saídas, decisões;
- Arestas representam comandos que não realizam desvios;
- **Estratégia de teste de McCabe**: A quantidade de casos de teste a serem testados corresponde à Complexidade Ciclomática de McCabe.

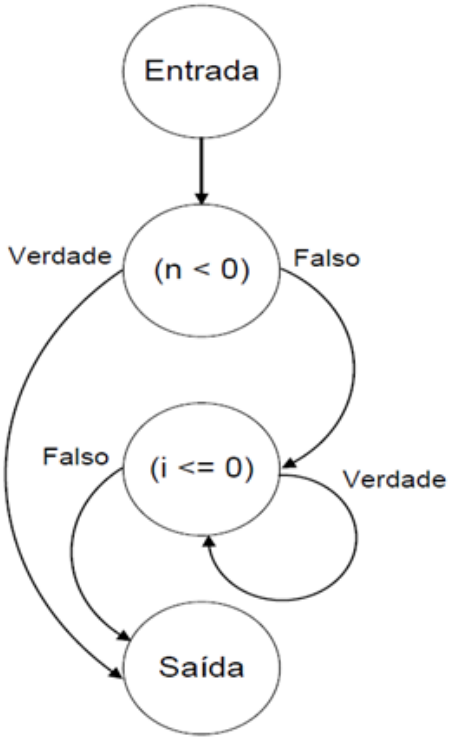


Complexidade Ciclomática de McCabe

Programa

```
#include <stdio.h>

int main () {
  int n, fatorial, i;
  printf ("Digite o numero: ");
  scanf ("%d", &n);

  if (n < 0) {
    printf ("Invalido: %d\n", n);
    n = -1;
  }

  else {
    fatorial = 1;
    for (i = 1; i <= n; i++)
      fatorial *= i;
    printf ("%d! = %d\n", n,
            fatorial);

  }
  return n;
}
```

Diagrama de Fluxo



Complexidade Ciclomática

C = #R + 1
C = 2 + 1 = 3

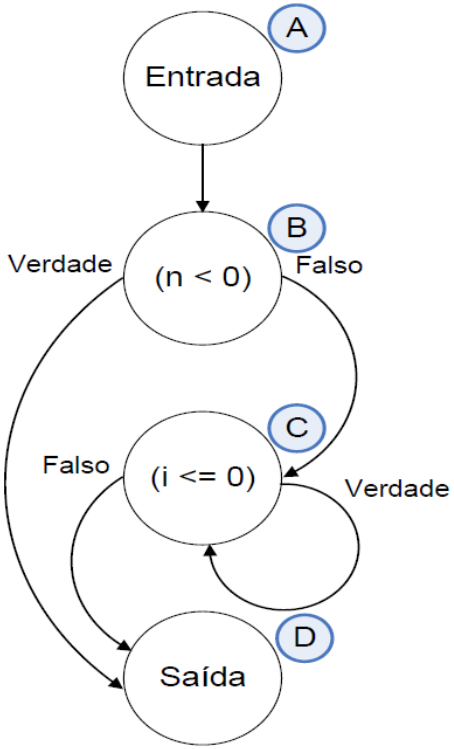
C = Complexidade ciclomática
R = Regiões fechadas

Complexidade Ciclomática de McCabe

Programa

```
#include <stdio.h>

int main () {
  int n, fatorial, i;
  printf ("Digite o numero: ");
  scanf ("%d", &n);

  if (n < 0) {
    printf ("Invalido: %d\n", n);
    n = -1;
  }

  else {
    fatorial = 1;
    for (i = 1; i <= n; i++)
      fatorial *= i;
    printf ("%d! = %d\n", n,
            fatorial);

  }
  return n;
}
```

Diagrama de Fluxo



Caminhos Básicos

1. ABD
2. ABCD
3. ABCCD

Testes Base

	Entrada	Esperado
1.	-1	(inválido -1)
2.	0	0! = 1
3.	1	1! = 1

Fatores na Escolha das Técnicas

- ✓ Tipo de sistema;
- ✓ Normas regulatórias;
- ✓ Requisitos do cliente ou contratuais;
- ✓ Nível e tipo do risco;
- ✓ Objetivos do teste;
- ✓ Documentação disponível;
- ✓ Conhecimento dos testadores;
- ✓ Tempo e orçamento;
- ✓ Ciclo de vida do desenvolvimento;
- ✓ Experiência prévia nos tipos de defeitos encontrados.