

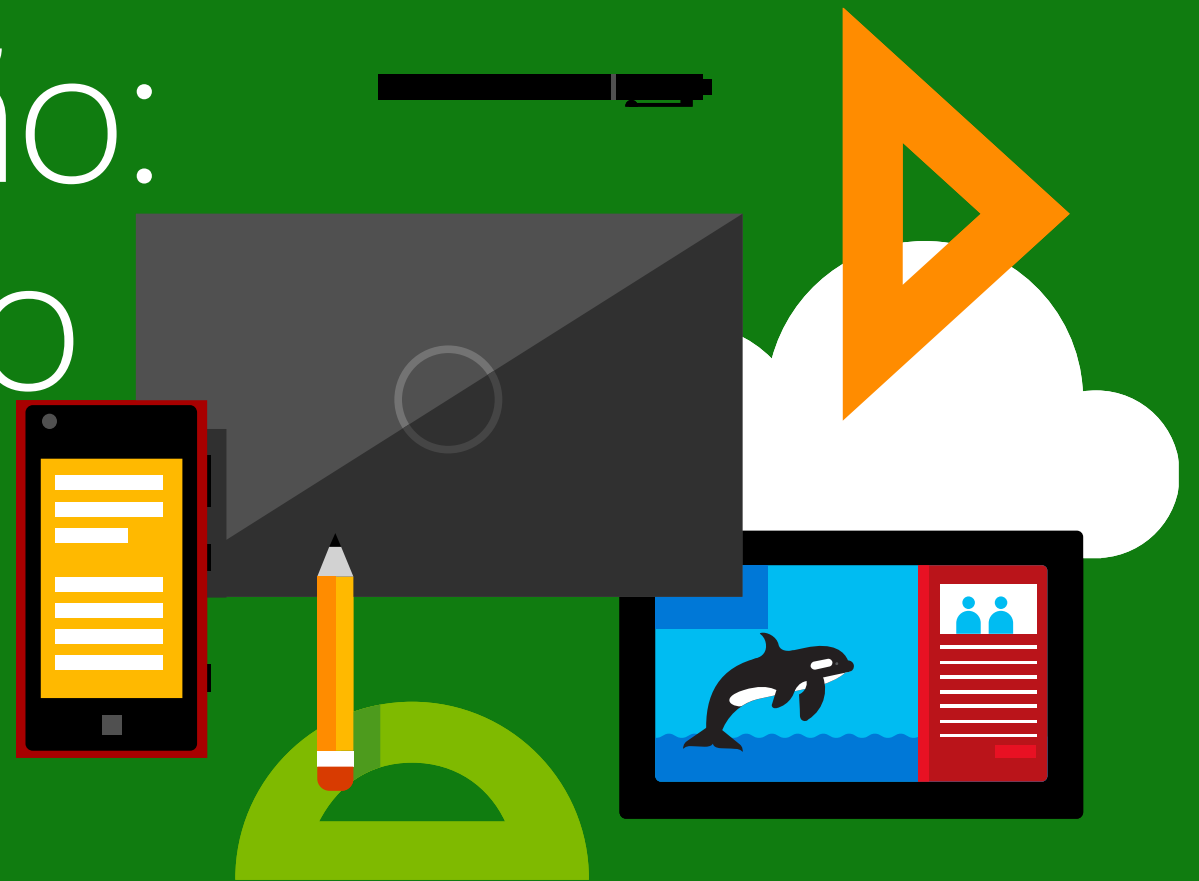
Verificação e Validação

Testes Baseados em Especificação (Particionamento)

Júlio Pereira Machado (julio.machado@pucrs.br)



Testes Baseados em Especificação: particionamento



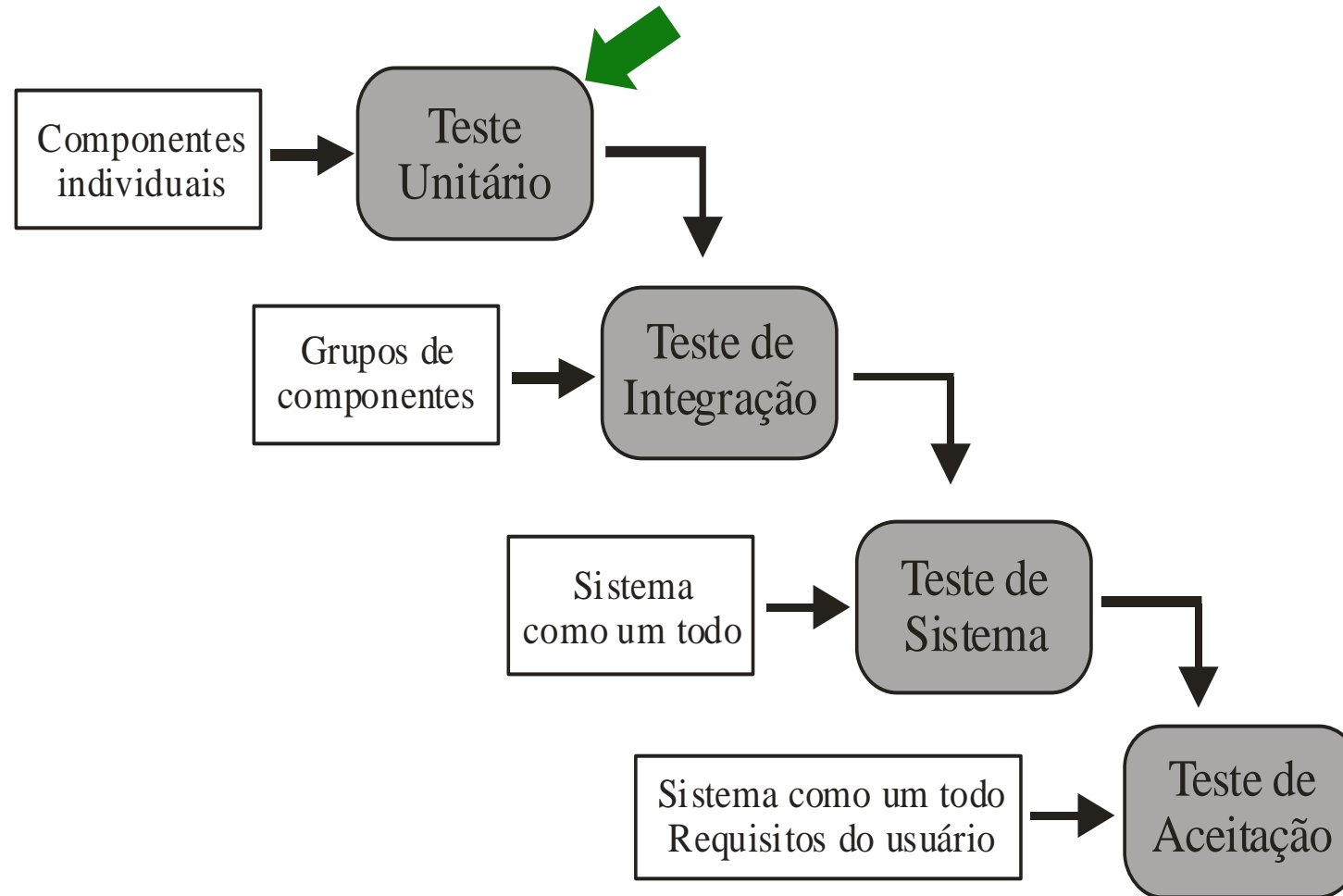
Teste funcional



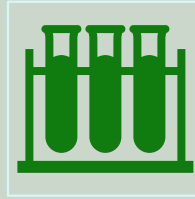
Níveis de testes



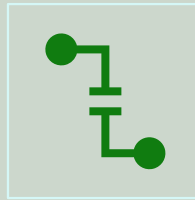
Níveis de testes



Introdução



A técnica de teste baseado em especificação usa apenas a especificação de requisitos como fonte de entrada para os testes



Considera-se um conjunto de entradas onde cada entrada exercita uma parte (ou partição) do programa



Por dividir o espaço de entrada em partições ou classes de equivalência esta técnica também é comumente conhecida como particionamento

Particionamento do espaço de entrada



Os programas normalmente são muito complexos para serem testados com um único caso de teste



Para encontrar um bom conjunto de casos de teste – um “test suíte” – divide-se o espaço de entrada em subgrupos (classes de equivalência)



Cada classe se caracteriza por ser única (duas classes não exercitam a mesma porção de código) e por ser fácil de verificar se o comportamento está correto ou não

Exemplo 1 (variável simples)

O valor dos ingressos de um cinema varia conforme a quantidade de ingressos adquiridos em conjunto. O valor normal do ingresso é de R\$ 15,00. Compras até 5 (inclusive) ingressos pagam o preço normal, compras entre 6 e 10 (inclusive) ingressos tem 10% de desconto e compras de mais de 10 ingressos tem 15% de desconto.

Classes e partições para o Exemplo 1

$ingressos \leq 5$



```
graph TD; A[ingressos ≤ 5] --> B[6 ≤ ingressos ≤ 10]; B --> C[ingressos > 10];
```

$6 \leq ingressos \leq 10$

$ingressos > 10$

Gerando casos de teste

- As asserções que definem as classes de equivalência não são por si só casos de teste, pois podem se tornar verdadeiras para uma infinidade de valores
- Entretanto, se cada uma é única (exercita apenas uma parte do sistema), pode-se assumir que qualquer conjunto de valores que torna a respectiva asserção verdadeira resultará em um comportamento equivalente
- Assim um único caso de teste por classe de equivalência é usualmente suficiente

Casos de teste para o Exemplo 1

Asserção	Quantidade de ingressos (caso de teste)	Resultado esperado
$\text{ingressos} \leq 5$	3	R\$ 45,00
$6 \leq \text{ingressos} \leq 10$	7	R\$ 94,50
$\text{ingressos} > 10$	12	R\$ 153,00
...

Exemplo 2 (variáveis dependentes)

O valor dos ingressos de um cinema varia conforme a quantidade de ingressos adquiridos em conjunto e conforme o dia da semana. O valor normal do ingresso é de R\$ 15,00. Compras até 5 (inclusive) ingressos pagam o preço normal, compras entre 6 e 10 (inclusive) ingressos tem 10% de desconto e compras de mais de 10 ingressos tem 15% de desconto. Esses valores valem de segunda a sexta. Nos sábados e domingos o valor do ingresso é de R\$ 20,00.

Classes e partições para o Exemplo 2

- Como o exemplo 2 tem duas variáveis e elas estão relacionadas, precisamos definir as partições para cada uma
- Na sequência fazemos o produto cartesiano das partições para definir os casos de teste

Classes e partições de ingressos para o Exemplo 2

Q1: $\text{ingressos} \leq 5$



```
graph TD; Q1[Q1: ingressos ≤ 5] --> Q2[Q2: 6 ≤ ingressos ≤ 10]; Q2 --> Q3[Q3: ingressos > 10];
```

Q2: $6 \leq \text{ingressos} \leq 10$

Q3: $\text{ingressos} > 10$

Classes e partições de dia da semana para o Exemplo 2

D1: dia da
semana



D2: dia de
fim de
semana

Casos de teste para o Exemplo 2

Combinação de partições	Valores de entrada	Resultado esperado
Q1xD1	3 ingressos, terça	R\$ 45,00
Q1xD2	4 ingressos, sábado	R\$ 80,00
Q2xD1	7 ingressos, quarta	R\$ 94,00
Q2xD2	8 ingressos, domingo	R\$ 144,00
Q3xD1	12 ingressos quinta	R\$ 153,00
Q3xD2	13 ingressos domingo	R\$ 221,00
...

Técnica de particionamento

1. Identifique os parâmetros ou as entradas do programa
2. Derive as características de cada parâmetro
 - Algumas dessas características podem ser encontradas diretamente na especificação
 - Ex: um "int ano" deve ser um inteiro positivo entre 0 e infinito
 - Outras podem não ser localizadas na especificação
 - Ex: uma entrada não pode ser "null" se determinado método não for capaz de lidar com isso
3. Acrescente restrições de maneira a minimizar o conjunto de casos de teste
 - Identifique combinações inválidas
 - Comportamentos excepcionais nem sempre precisam ser combinados com todos os valores válidos
4. Gere combinações dos valores de entrada
 - Estes são os casos de teste

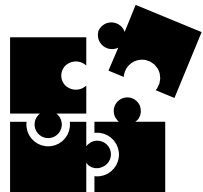
Exemplo 3

A tarifa de energia em uma determinada região tem um custo básico de R\$ 5,00 por MW (Megawatt)

Se o sistema estiver operando em bandeira verde não tem acréscimo, em bandeira amarela o acréscimo é de 2,5% na bandeira vermelha o acréscimo é de 5%

Se o consumo ultrapassar 20 MW, o valor do excedente tem um adicional de 15% depois de calculado o valor com as bandeiras

Se a residência se enquadrar no programa de tarifa social então terá 50% de desconto no valor final



Exercício: identifique as partições e projete os casos de teste

Exemplo 3

Entradas do "programa":

- Cor da bandeira
- Consumo em MW
- Enquadramento na tarifa social

Exemplo 3

Partições:

- B1: bandeira verde
- B2: bandeira amarela
- B3: bandeira vermelha

- C1: consumo ≤ 20 MW
- C2: consumo > 20 MW
- C3: consumo inválido

- R1: residência se enquadra na tarifa social
- R2: residência não se enquadra na tarifa social

Exemplo 3

Partições:

- B1: bandeira verde
- B2: bandeira amarela
- B3: bandeira vermelha

- C1: consumo ≤ 20 MW
- C2: consumo > 20 MW
- C3: consumo inválido

- R1: se enquadra na tarifa social
- R2: não se enquadra na tarifa social

Combinações de partições:

- B1,C1,R1; B1,C1,R2;
- B2,C1,R1; B2,C1,R2;
- B3,C1,R1; B3,C1,R2;

- B1,C2,R1; B1,C2,R2;
- B2,C2,R1; B2,C2,R2;
- B3,C2,R1; B3,C2,R2;

- B3,C3,R1; <-- casos inválidos basta testar um

Exemplo 3

Partições:

- B1: bandeira verde (Vd)
- B2: bandeira amarela (Am)
- B3: bandeira vermelha (Vm)

- C1: consumo ≤ 20 MW
- C2: consumo > 20 MW
- C3: consumo inválido

- R1: se enquadra na tarifa social (T)
- R2: não se enquadra na tarifa social (F)

Casos de teste:

- B1,C1,R1; B1,C1,R2; Vd,10,T; Vd,10,F
- B2,C1,R1; B2,C1,R2; Am,15,T; Am,15,F
- B3,C1,R1; B3,C1,R2; Vm,5,T; Vm,5,F

- B1,C2,R1; B1,C2,R2; Vd,25,T; Vd,25,F
- B2,C2,R1; B2,C2,R2; Am,21,T; Am,21,F
- B3,C2,R1; B3,C2,R2; Vm,30,T; Vm,30,F

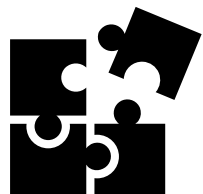
- B3,C3,R1; Vm,-5,T

Exemplo 4

Uma loja possui diversos tamanhos de embalagem para armazenar barras de chocolate. O estoque da loja contém barras de 1 Kg e de 5Kg. Quando o cliente escolhe o tamanho da embalagem, deve-se usar a maior quantidade possível de barras de 5Kg e completar o peso desejado com barras de 1kg.

As entradas do programa são a quantidade de barras de 1Kg e 5Kg disponíveis e o tamanho (capacidade em Kgs) da embalagem.

O resultado é a quantidade de barras de 1Kg e 5Kg que devem ser usadas para completar a embalagem. Deve retornar -1 se não houver no estoque barras suficientes.



Exercício: identifique as partições e projete os casos de teste

Exemplo 4

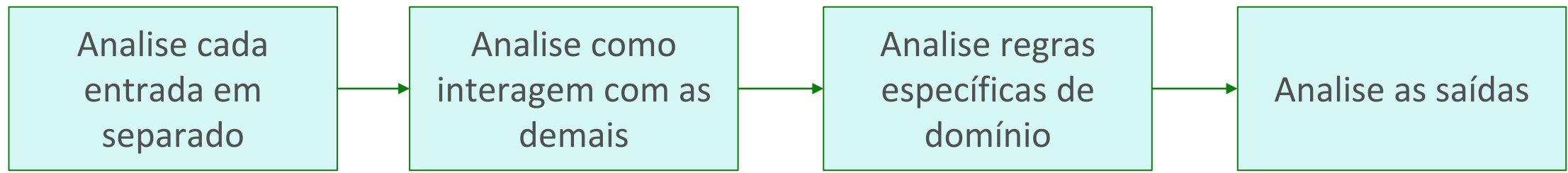
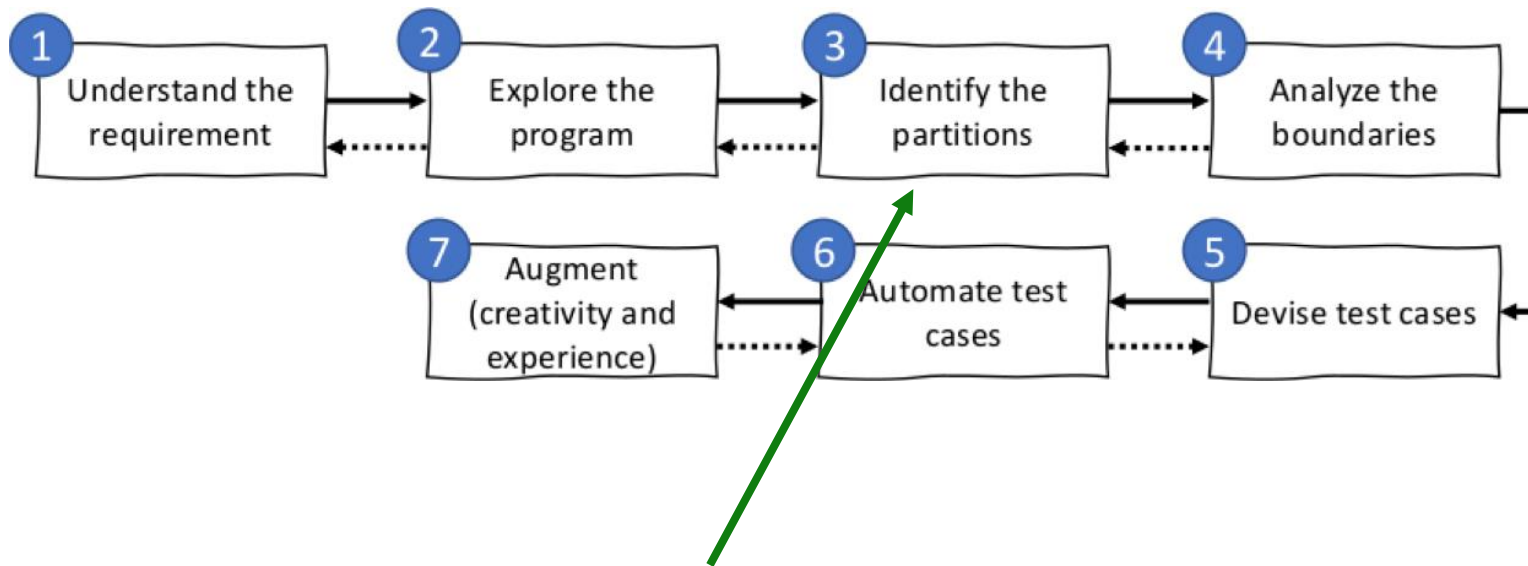
Quais são as partições ou classes de equivalência neste caso?

- **Precisa apenas barras pequenas.** Uma solução que usa apenas barras pequenas.
- **Precisa apenas barras grandes.** Uma solução que usa apenas barras grandes.
- **Precisa barras grandes e pequenas.** Uma solução que usa tanto barras grandes como pequenas.
- **Barras insuficientes.** Um caso que é impossível atender porque não tem barras suficientes.
- **Fora da especificação.** Um caso excepcional.

Exemplo 4

Exemplo de casos de teste (p = estoque pequena, g = estoque grande, n =demanda)

- **Precisa apenas barras pequenas.** $p=4, g=2, n=3$
- **Precisa apenas barras grandes.** $p=5, g=3, n=10$
- **Precisa grandes e pequenas.** $p=5, g=3, n=17$
- **Barras insuficientes.** $p=1, g=1, n=10$
- **Fora da especificação.** $p=4, g=2, n=-1$



Toda análise levando em conta os requisitos do sistema!