

Testes baseados na especificação

Baseado em notas de aula da profa. Eliane Martins

Referências

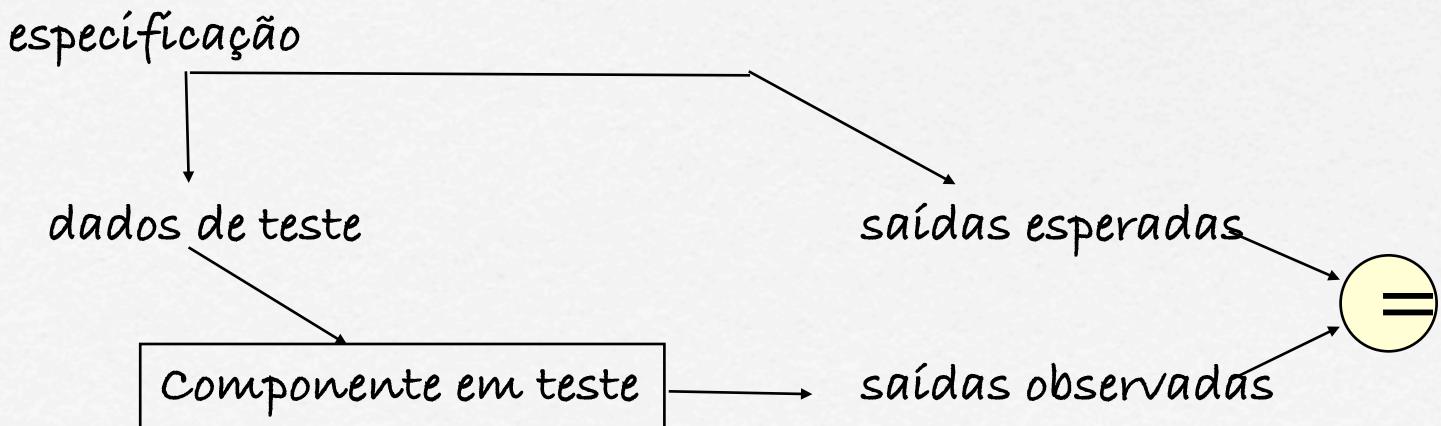
- R.Binder. "Testing OO Systems", 2000.
- P.Jalote. "An Integrated Approach to SW Engineering", 2^a ed., 1997, cap9.2.3
- R.S.Pressman. "Engenharia de Software", 3^a edição, 1995, cap. 18.5.3.
- G.J.Myers. "The Art of Software Testing", 1979, cap.4.

Tópicos

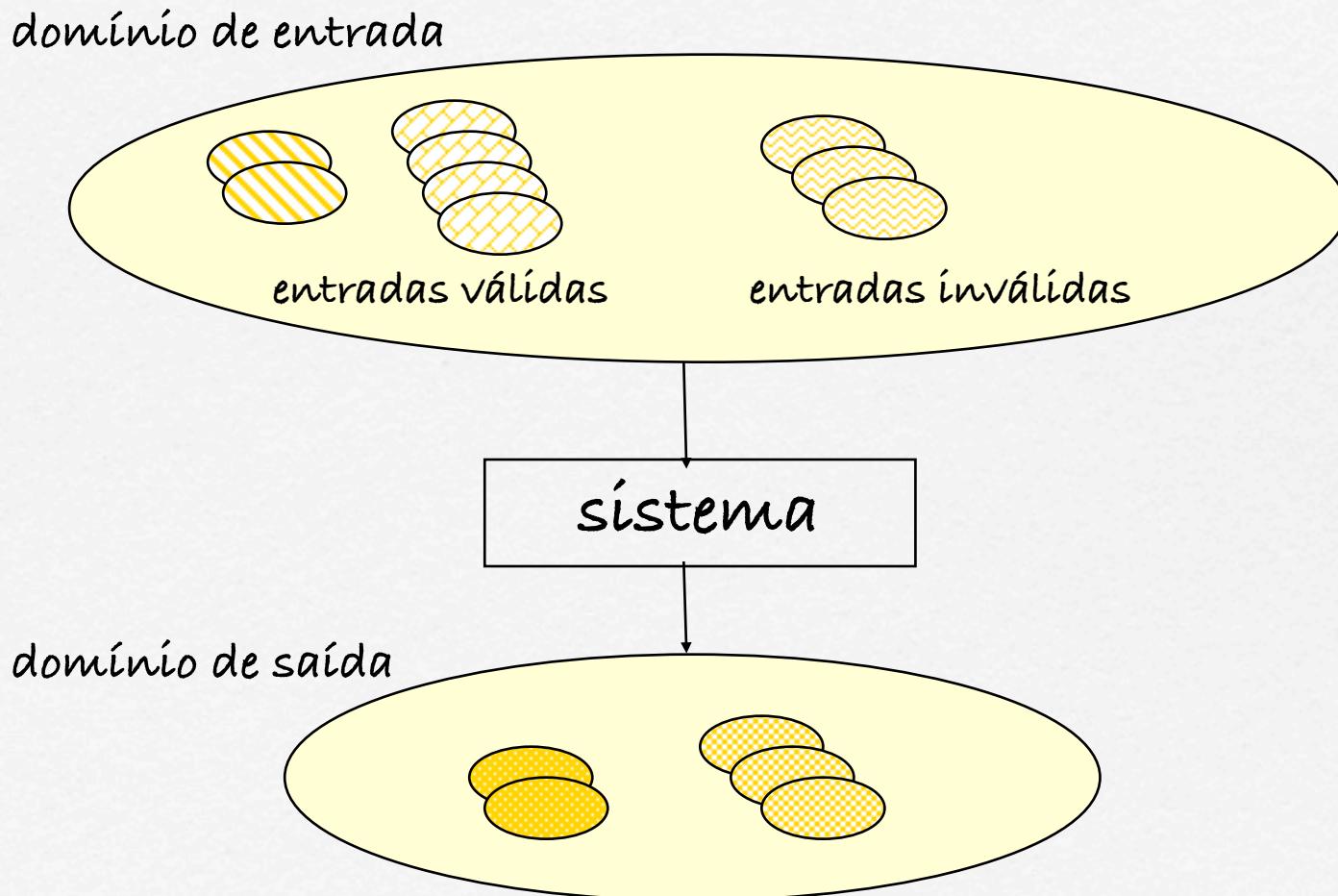
- Partição de equivalência
- Testes de valores-límite
- Grafo causa-efeito

Testes baseados na especificação

- Também chamados de testes caixa preta
- Objetivos dos testes: demonstrar cobertura da especificação
- Modelos de teste obtidos das especificações:
 - requisitos
 - arquitetura



Partições de equivalência: princípio



Partições de equivalência

- O domínio de entrada (ou saída) do programa/função é dividido em um número finito de partições (ou classes) de equivalência
 - supõe-se que dados pertencentes a uma partição revelam as mesmas falhas
 - partições válidas e inválidas são consideradas
- Geração de testes: selecionar um ou mais dados de cada partição
- Critério de cobertura: cada partição deve ser considerada ao menos 1 vez

Partição de equivalência: passos

- Decompor o programa em funções
- Identificar as variáveis que determinam o comportamento de cada função
- Partitionar os valores de cada variável em classes de equivalência (válidas e inválidas)
- Especificar os casos de teste:
 - eliminar as classes impossíveis ou os casos desinteressantes
 - selecionar casos de testes cobrindo as classes válidas das diferentes variáveis
 - para cada classe inválida escolha um caso de teste que cubra 1 e somente 1 de cada vez

Determinação das classes de equivalência

Definição da variável de entrada	Classes de equivalência
Intervalo	<ul style="list-style-type: none">• Uma classe válida para valores pertencentes ao intervalo• Uma classe inválida para valores menores que o limite inferior• Uma classe inválida para valores maiores que o limite superior
Lista de valores válidos	<ul style="list-style-type: none">• Uma classe válida para os valores incluídos na lista• Uma classe inválida para todos os outros valores

Determinação das classes de equivalência

Definição da variável de entrada Número de valores válidos	Classes de equivalência <ul style="list-style-type: none">• Uma classe válida para número de valores igual ao número previsto• Uma classe inválida para número de valores = 0• Uma classe inválida para número de valores maior ou menor que o valor previsto
Restrições (expressão lógica; sintaxe; valor específico; compatibilidade com outras variáveis)	<ul style="list-style-type: none">• Uma classe válida para os valores que satisfazem às restrições• Uma classe inválida para os outros valores

Exemplo 1

- Função:
 - Considere uma função que aceita como entradas de 4 a 6 valores inteiros de 2 dígitos maiores do que 10.
 - Identificação das variáveis de entrada e das condições que estas devem satisfazer:
 - $nro_entradas \in [4, 6]$
 - $valor \in [10, 99]$

Exemplo 1

- Determinação das classes de equivalência:

variável	classes válidas	classes inválidas
nro_entradas	C1. $4 \leq \text{nro_entradas} \leq 6$	C3. $\text{nro_entradas} < 4$ C4. $\text{nro_entradas} > 6$
valor	C2. $10 \leq \text{valor} \leq 99$	C5. $\text{valor} < 10$ C6. $\text{valor} > 99$

Exemplo 1

- casos de teste:
 - selecionar casos de testes cobrindo as classes válidas das diferentes variáveis

nro_entradas	C1	C2	C3	C4	C5	C6
4 ... 6						X
< 4						
> 6						
valor						
10 ... 99						X
< 10						
> 99						

nro_entradas	valores
5	11, 12, 45, 78, 95

Exemplo 1

- casos de teste:
 - para cada classe inválida escolha um caso de teste que cubra 1 e somente 1 de cada vez

nro_entradas	C1	C2	C3	C4	C5	C6
4 ... 6	X					
< 4			X			
> 6						
valor				X		
10 ... 99						
< 10						
> 99						

nro_entradas	valores
1. 5	11, 12, 45, 78, 95
2. 3	11, 12, 45

Exemplo 1

- casos de teste:
 - para cada classe inválida escolha um caso de teste que cubra 1 e somente 1 de cada vez

nro_entradas	C1	C2	C3	C4	C5	C6
4 ... 6	X					
< 4		X				
> 6			X			
valor				X		
10 ... 99		X				
< 10				X		
> 99					X	

nro_entradas	valores
1.	5
2.	11, 12
3.	11, 12, 45, 78, 95, 67, 77, 54
4.	5, 11, 12, 45, 6
5.	110, 45, 78, 340, 95

Exemplo 2

- Considere o teste do procedimento: `Valida_Nova_Senha` que recebe como entrada uma senha e valida-a conforme as seguintes regras:
 - uma senha deve ter de 6 a 10 caracteres
 - o primeiro caracter dever ser alfabético, numérico ou um “?”
 - os outros caracteres podem ser quaisquer, desde que não sejam caracteres de controle
 - a senha não pode existir em um dicionário

Exemplo 2 - Classes de Equivalência

variáveis	válidas	inválidas
tamanho	C1. tamanho $\in [6, 10]$	C7. tamanho < 6 C8. tamanho > 10
1º caracter (car1)	C2. car1 alfabético C3. car1 numérico C4. car1 é "?"	C9. car1 $\in \{\text{caract. controle}\}$ C10. car1 $\notin \{\text{letras, dígitos, ?}\}$
outros caract (Outro)	C5. Outro $\notin \{\text{caract. de controle}\}$	C11. Outro $\in \{\text{caract. de controle}\}$
status	C6. Senha $\notin \text{dicionário}$	C12. Senha $\in \text{dicionário}$

Exemplo 2

Análise de valores-limite

- Critério de seleção que identifica valores nos limites das classes de equivalência
- Exemplos:
 - valor mínimo (máximo) igual ao mínimo (máximo) válido
 - uma unidade abaixo do mínimo
 - uma unidade acima do máximo
 - arquivo vazio
 - arquivo maior ou igual à capacidade máxima de armazenamento
 - cálculo que pode levar a “overflow” (“underflow”)
 - erro no primeiro (último) registro

Exemplo 3

- Casos de teste:
 - para cada classe inválida escolha um caso de teste que cubra 1 e somente 1 de cada vez

nro_entradas	C1	C2	C3	C4	C5	C6
4 ... 6	X					
< 4		X				
> 6			X			
valor				X		
10 ... 99		X				
< 10			X			
> 99				X		

nro_entradas	valores
1.	4
2.	10, 11, 50, 55, 98, 99
3.	10, 11, 98
4.	10, 11, 50, 55, 98, 99, 45
5.	9, 10, 11, 99
6.	10, 11, 100, 101

Limitação

- Testes baseados em partições de equivalência ou análise de valores-límite:
 - consideram cada valor de entrada isoladamente
 - e se existirem combinações de valores que constituam situações interessantes a serem testadas?

Análise causa - efeito

- Necessária quando se deseja testar combinações de entradas
- Utiliza tabelas de decisão e árvores de decisão
- grafo causa-efeito como modelo auxiliar

Definições

Causas:

- condições de entrada (valor lógico)

Efeitos

- ações realizadas em resposta às diferentes condições de entrada

ex.:

causa: $\text{preço} \geq 50 \wedge 0 \leq \text{qtd} \leq 99$

efeito: fornecer 5% de desconto

Árvore de decisão

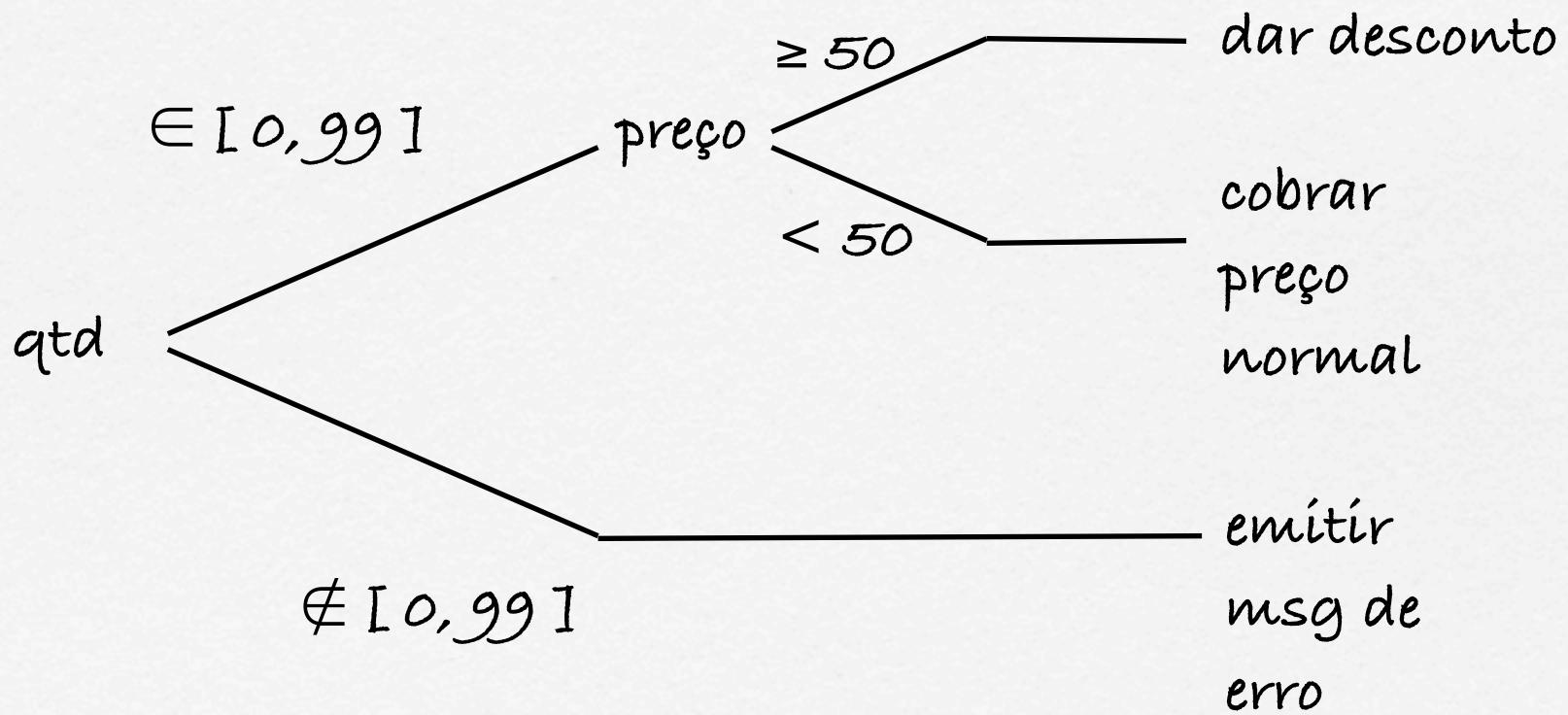


Tabela de decisão

Condições de entrada (causas)

Ações (efeitos)

regra

Construção da tabela de decisão

preço ≥ 50	✓	✗	✗
$0 \leq qtd \leq 99$	✓	✓	✗
dar desconto	✗		
cobrar preço normal			✗
emitir msg de erro			✗

Limitação das tabelas de decisão

- Tamanho!
- $\text{nº de regras} = 2^N$, onde N é o nº de causas

Exemplo

Supor um sistema bancário que trate somente duas transações:

depósito nº da conta quantia

saque nº da conta quantia

Requisitos:

- se o comando é depósito e o nº da conta é válido então a quantia é depositada
- se o comando é saque e o nº da conta é válido e a quantia é válida ($0 < \text{quantia} \leq \text{saldo}$) então a quantia sacada
- se o comando ou nº da conta ou a quantia for inválido então exibir mensagem de erro apropriada

- Causas:

- c1. Comando é depósito
- c2. Comando é saque
- c3. Nº da conta é válido
- c4. Quantia é válida



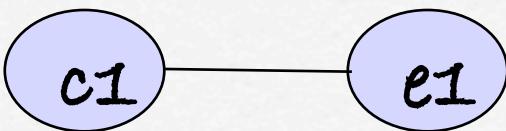
$$\text{nº de regras} = 2^4 = 16$$

será que todas interessam ?

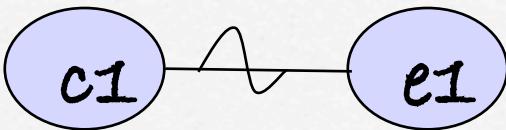
- Efeitos:

- e1. Exibir "comando inválido"
- e2. Exibir "nº da conta inválido"
- e3. Exibir "quantia inválida"
- e4. Depositar a quantia
- e5. Sacar a quantia

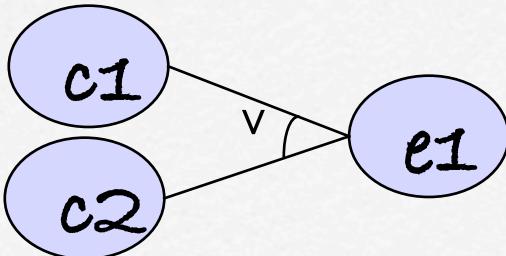
Grafo causa-efeito: notação básica



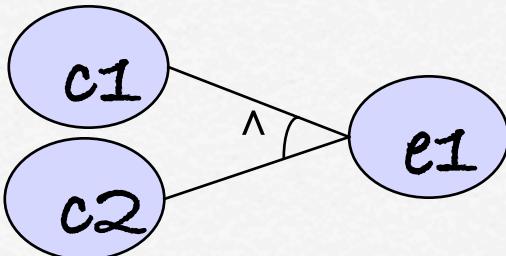
Identidade



Negação

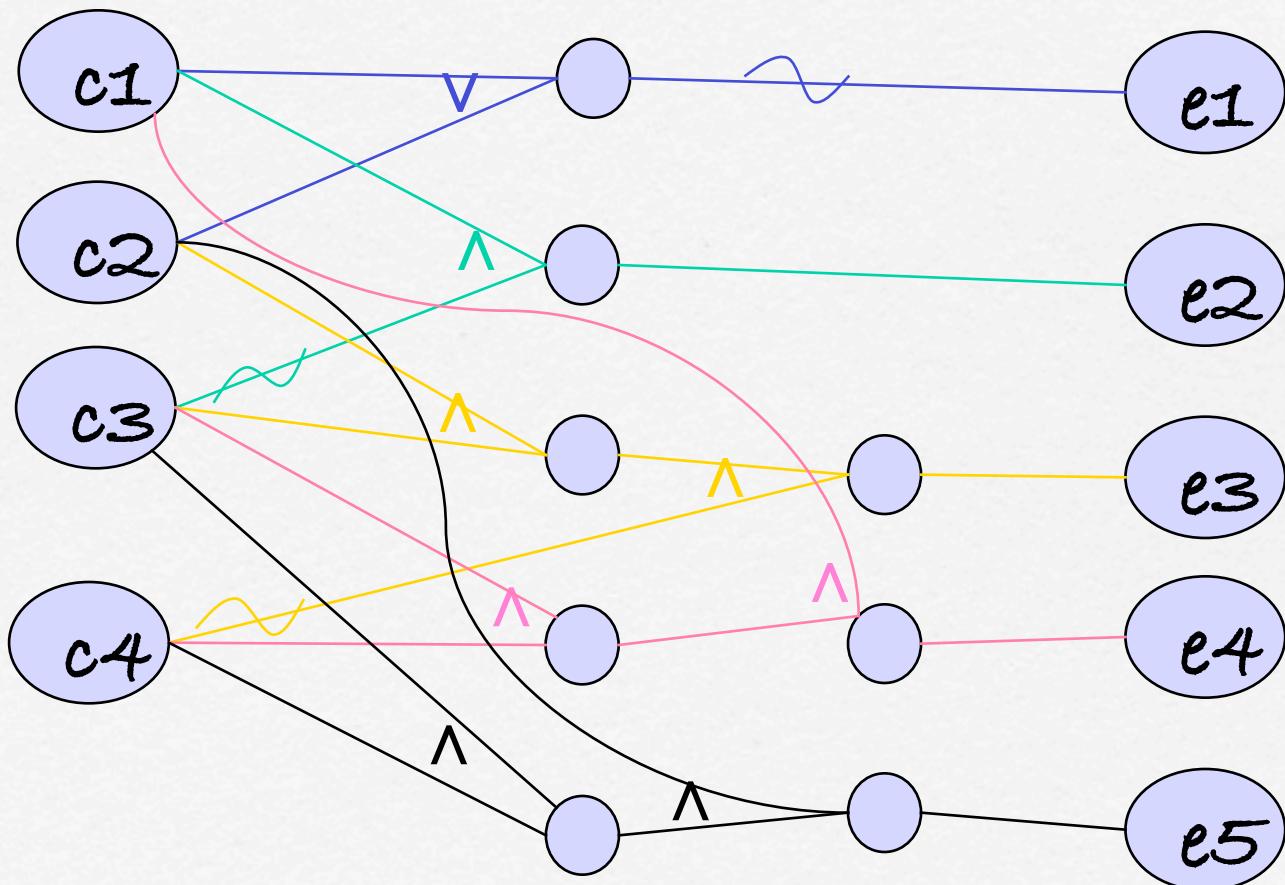


Ou



E

Exemplo: grafo causa-efeito



Conversão em tabela de decisão

- Escolher um efeito como ação a ser executada, i.e., marcar um 'X' na regra correspondente a este efeito.
- Rastrear no grafo quais as combinações de causas que levam a esse efeito e marcar um "V" ou "F" na posição correspondente na tabela
- Para cada combinação criada, verificar se ocorrem ou não os outros efeitos

Exemplo: tabela de decisão

Id.	1	2	3	4	5
c1	F	∨	●	●	∨
c2	F	●	∨	∨	●
c3	●	●	∨	∨	∨
c4	●	●	F	∨	∨
e1	x				
e2		x			
e3			x		
e4				x	
e5					x

Geração de testes

- Tabela de decisão
 - critério: exercitar cada regra pelo menos 1 vez
- Árvore de decisão
 - critério: exercitar cada caminho da raiz até a folha pelo menos 1 vez

Exemplo: casos de teste

Regra 1:

comando $\notin \{\text{depósito, saque}\}$, $\forall \text{nº conta}, \forall \text{quantia}$

Regra 2:

comando = depósito, nº de conta inválido, $\forall \text{quantia}$

Regra 3:

comando = saque, nº de conta válido, quantia
inválida

...

Exercício

- Considere uma função com duas variáveis de entrada: Cliente e Qtd, e uma variável de saída, Desconto. Cliente pode ser do tipo A, B ou C e Qtd pode variar de 1 a 1000. A função calcula Desconto de acordo com as seguintes regras:
 - Clientes do tipo A não recebem desconto se nº de itens comprados for inferior a 10; recebem 5% desconto para compras entre 10 e 99 itens; 10% de desconto acima de 100 itens.
 - Clientes do tipo B recebem 5% de desconto para compras abaixo de 10 itens; 15% de desconto entre 10 e 99 itens; 25% de desconto acima de 100 itens.
 - Clientes do tipo C não recebem desconto se nº de itens comprados for inferior a 10; 20% de desconto entre 10 e 99 itens; 25% de desconto acima de 100 itens.