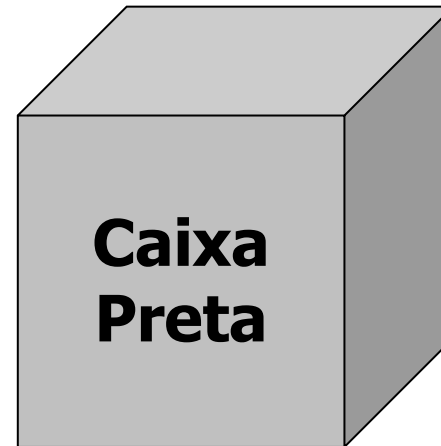
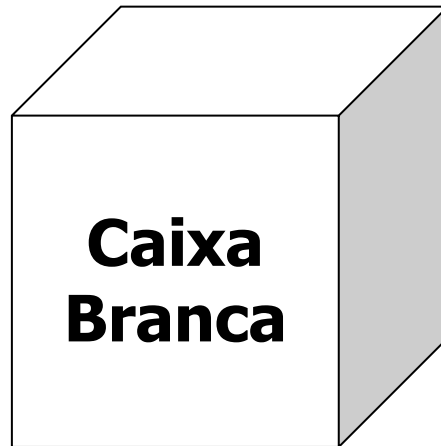


TESTES DE SOFTWARE

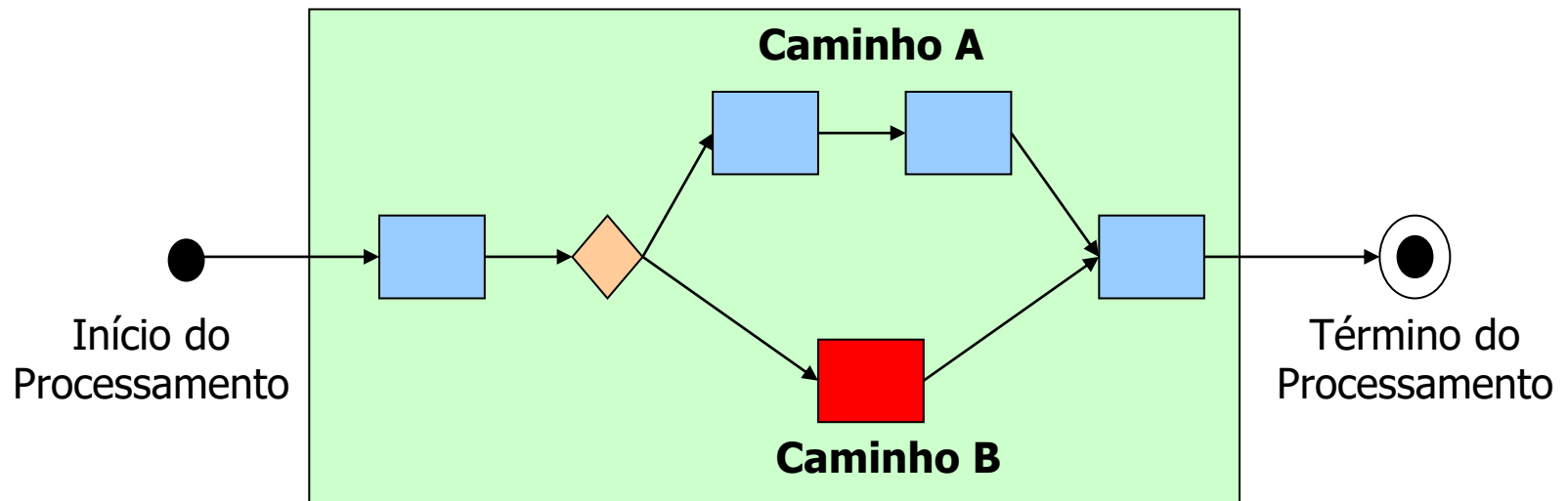
TESTES DE SOFTWARE

O QUE É TESTE DE SOFTWARE?

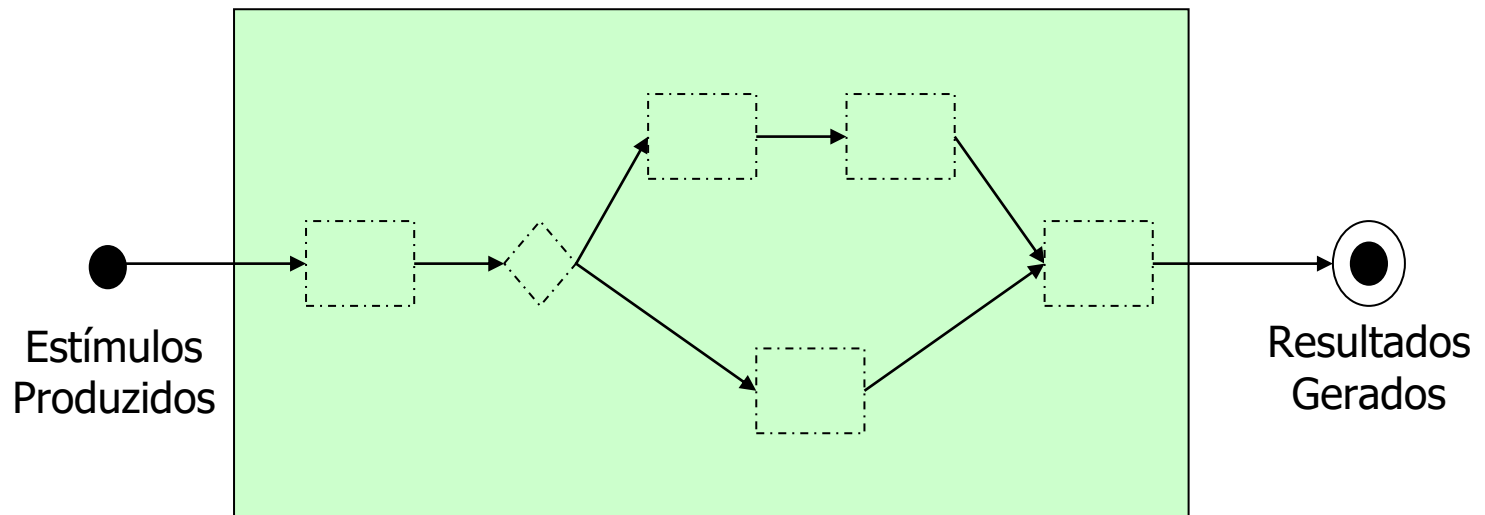
- O teste do software é a **investigação** do **software** a fim de fornecer informações sobre sua **qualidade** em relação ao contexto em que ele deve operar, se relaciona com o conceito de verificação e validação. Isso inclui o processo de utilizar o produto para encontrar seus defeitos.
- O teste é um processo realizado pelo testador de software, que permeia outros processos da **engenharia de software**, e que envolve ações que vão do **levantamento de requisitos** até a execução do teste propriamente dito.



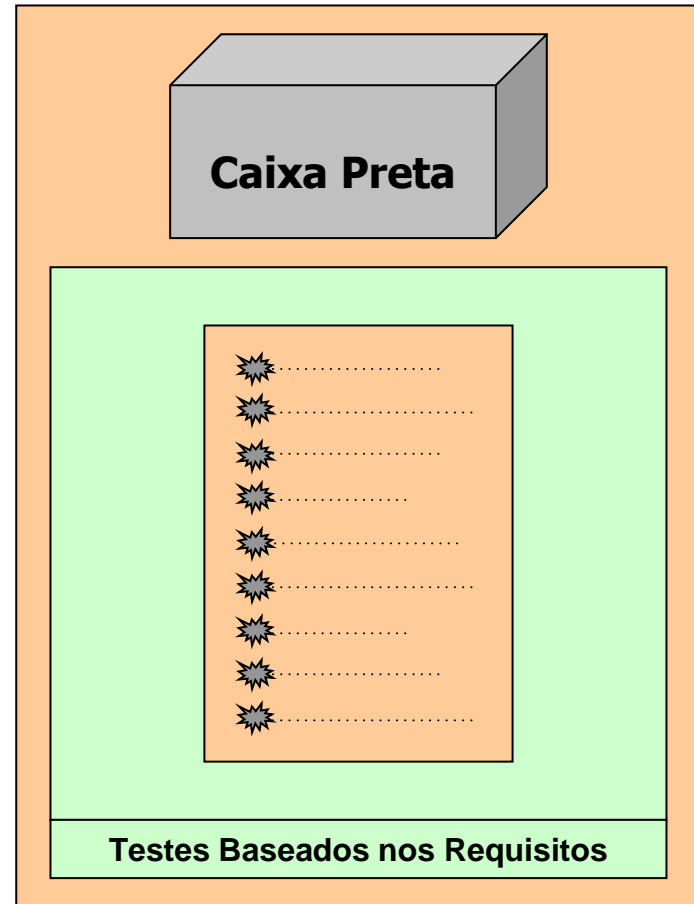
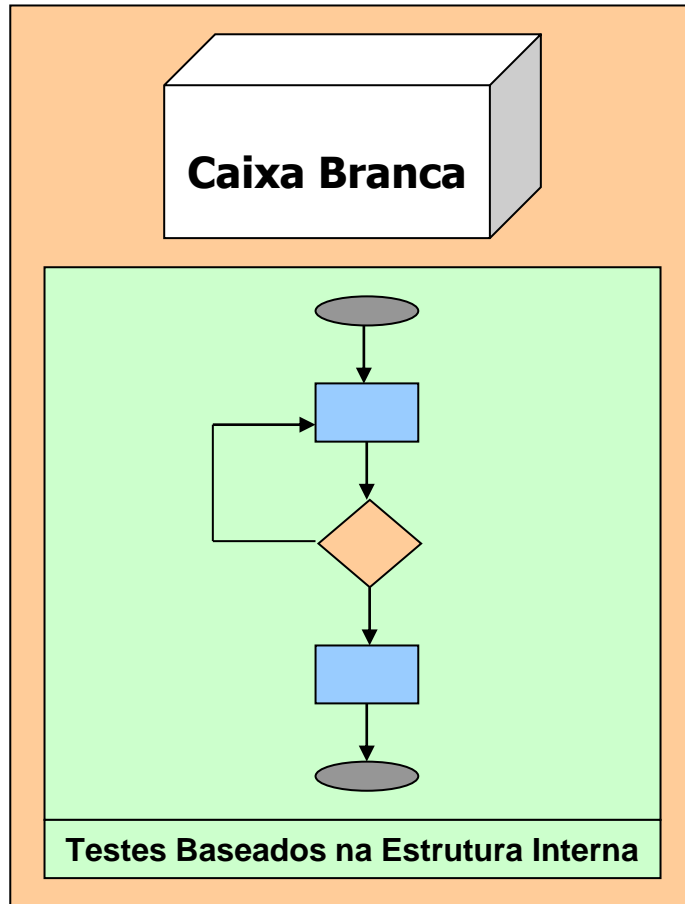
- Tem o objetivo é identificar defeitos nas estruturas internas dos programas, como por exemplo:
 - Código inalcançável
 - Loop infinito
 - Erros de Lógica



- Tem o objetivo de garantir que os requisitos do sistema são plenamente atendidos
 - Não interessa como isso acontece internamente
- Não exige conhecimento tecnológico
- Maior número de profissionais capacitados

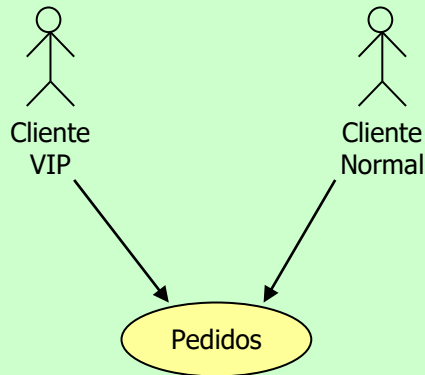


Abordagens Fundamentais dos Testes



Progressividade e Regressividade dos Testes

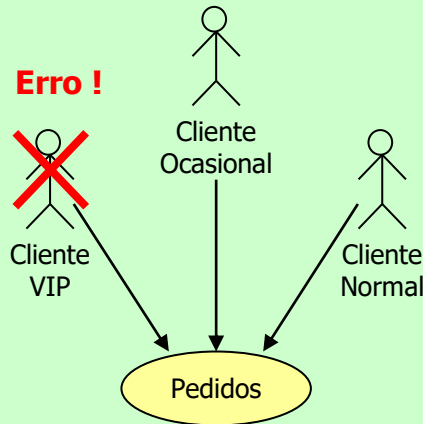
Cenário Versão "A"



Nesta versão, o sistema atende duas categorias de clientes, sendo que o Cliente VIP responde por 75% do faturamento.

A necessidade de políticas de negociação para Clientes Ocasiais gera demanda para uma nova funcionalidade.

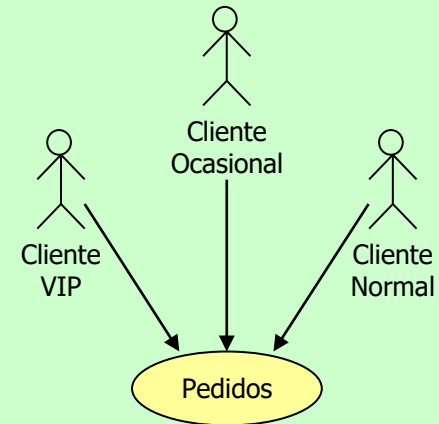
Cenário Versão "B"



Foi gerada uma nova versão e somente os testes progressivos foram aplicados.

Infelizmente, a política de negociação de Clientes VIP foi indevidamente afetada pela mudança, provocando reduções nos preços de linhas inteiras de produtos.

Cenário Versão "B.1"



As reduções de preços aumentaram repentinamente a requisição de pedidos. A área de vendas comemorou por apenas quatro horas essa nova situação.

O erro foi identificado e a solução só foi disponibilizada após seis horas.

- Testes progressivos
 - Elaborados de acordo com a evolução do produto
 - A medida que o software ganha novas funcionalidades, um novo conjunto de testes deve ser criado
 - Testa-se somente as inovações do software
- Testes regressivos
 - Reexecuta um subconjunto de testes previamente executados
 - Assegurar que alterações não afetaram outras partes do produto

Definições Importantes: Cenário

- Define um contexto inicial para facilitar a visualização, entendimento ou análise de um problema ou situação.
- Dentro de um cenário de teste podemos ter diversos Casos de Teste

Identifique os possíveis cenários de Teste

1. Tela de Login

O diagrama representa a interface de uma tela de login. No topo, centralizado, está o título "Tela de Identificação". Abaixo dele, há dois campos de entrada: o primeiro é rotulado "Código do Usuário" e o segundo "Senha". Ambos os campos são representados por retângulos cinza. No canto inferior direito da tela, há um botão retangular com o texto "OK".

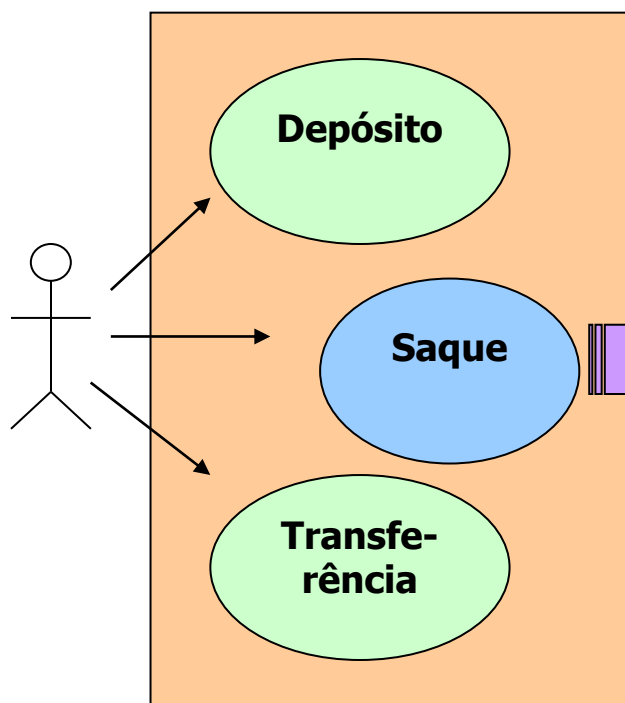
```
graph TD; Title[Tela de Identificação]; User[Código do Usuário]; Pass[Senha]; OK[OK]; Title --- User; Title --- Pass; User --- Pass; Pass --- OK;
```

Definições Importantes: Caso de Teste

- Especificação mais detalhada do teste, composta por:
 - Conjunto de entradas
 - Condições de execução
 - Resultados esperados
- Tem como objetivo verificar os requisitos especificados do sistema

CATEGORIAS DE TESTES

Categorias dos Testes



Cenários de Testes

- simular saques acima do saldo disponível;
- simular saques com cartão vencido;
- avaliar se a duração do saque dura até 30 seg. num universo de 5 milhões de correntistas e 100 milhões de movimentação bancária;
- simular saque com defeito no “cash-dispenser”;
- simular saque com impressora do fornecedor A, B e C;
- avaliar se a senha do cartão esta sendo requisitada antes e depois da transação;
- simular 2 saques simultâneos na mesma conta corrente;
- simular saque na conta-poupança;
- avaliar se a senha adicional e randômica esta sendo requisitada no início da operação.
- simular saques no Windows 95, 98, NT e 2000;
- avaliar se todas as telas possuem ajuda;

Organizando em Categorias

Funcional	Segurança	Usabilidade	Performance
<ul style="list-style-type: none"> - simular saques acima do saldo disponível; - simular saque na conta-poupança; - simular saque acima do valor do limite da conta; - simular saque com valores não múltiplos das notas; - simular saque com valores não múltiplos das notas; 	<ul style="list-style-type: none"> - simular saques com cartão vencido; - avaliar se a senha do cartão esta sendo requisitada antes e depois da transação; - avaliar se a senha adicional e randômica esta sendo requisitada no início da operação; - simular saque noturno acima do valor permitido; 	<ul style="list-style-type: none"> - avaliar se todas as telas possuem ajuda; - avaliar se mensagens são claras e objetivas; - avaliar se o padrão visual é mantido em todos os momentos; - avaliar se todas as operações possuem caminhos de fuga; 	<ul style="list-style-type: none"> - avaliar se a duração do saque dura até 30 seg. num universo de 5 milhões de correntistas e 100 milhões de movimentação bancária; - garantir que manipulação com dispositivos físicos no saque não ultrapassem 10 seg. da operação;
Carga e Concorrência	Configuração	Recuperação	Contingência
<ul style="list-style-type: none"> - simular 2 saques simultâneos na mesma conta-corrente; - simular 10.000 saques simultâneos; 	<ul style="list-style-type: none"> - simular saque com impressora do fornecedor A, B e C; - simular saques no Windows 95, 98, NT e 2000; - simular saque com impressora do fornecedor X, Y e Z; 	<ul style="list-style-type: none"> - simular saque com defeito no "cash-dispenser"; - simular saque com defeito na impressora; - simular saque com falha de conexão com a central; - simular saque com queda de energia; 	<ul style="list-style-type: none"> - disparar processo de instalação emergencial;

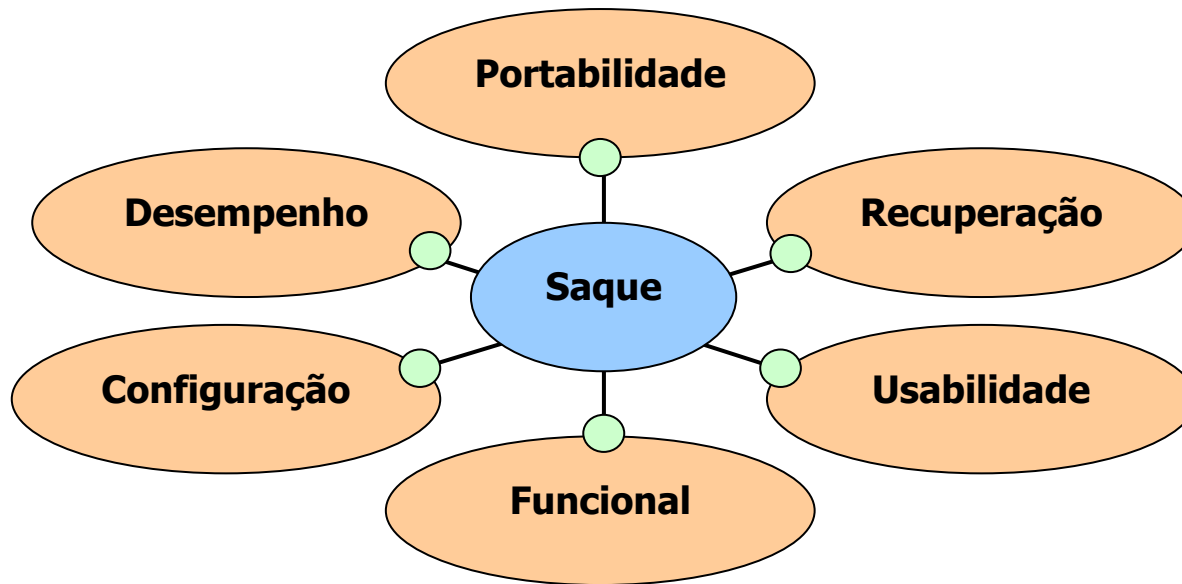
Os cenários em AZUL só foram descobertos durante o processo de categorização.

Fatores da Qualidade



Corretude	Ele faz aquilo que eu preciso?
Confiabilidade	Ele se comporta com precisão 100% tempo?
Manutenibilidade	Posso consertar o programa?
Flexibilidade	Posso mudar o programa? (novo requisito)
Testabilidade	Posso testar o programa?
Eficiência	Ele rodará no meu hardware tão bem quanto possível?
Integridade	O programa é seguro?
Usabilidade	O programa foi projetado para este usuário? É fácil de usar?
Portabilidade	Poderei utilizá-lo em outro ambiente?
Reusabilidade	Poderei utilizá-lo novamente em outro sistema?
Interoperabilidade	Poderei compor uma interface com outro sistema?

Entendendo as Categorias

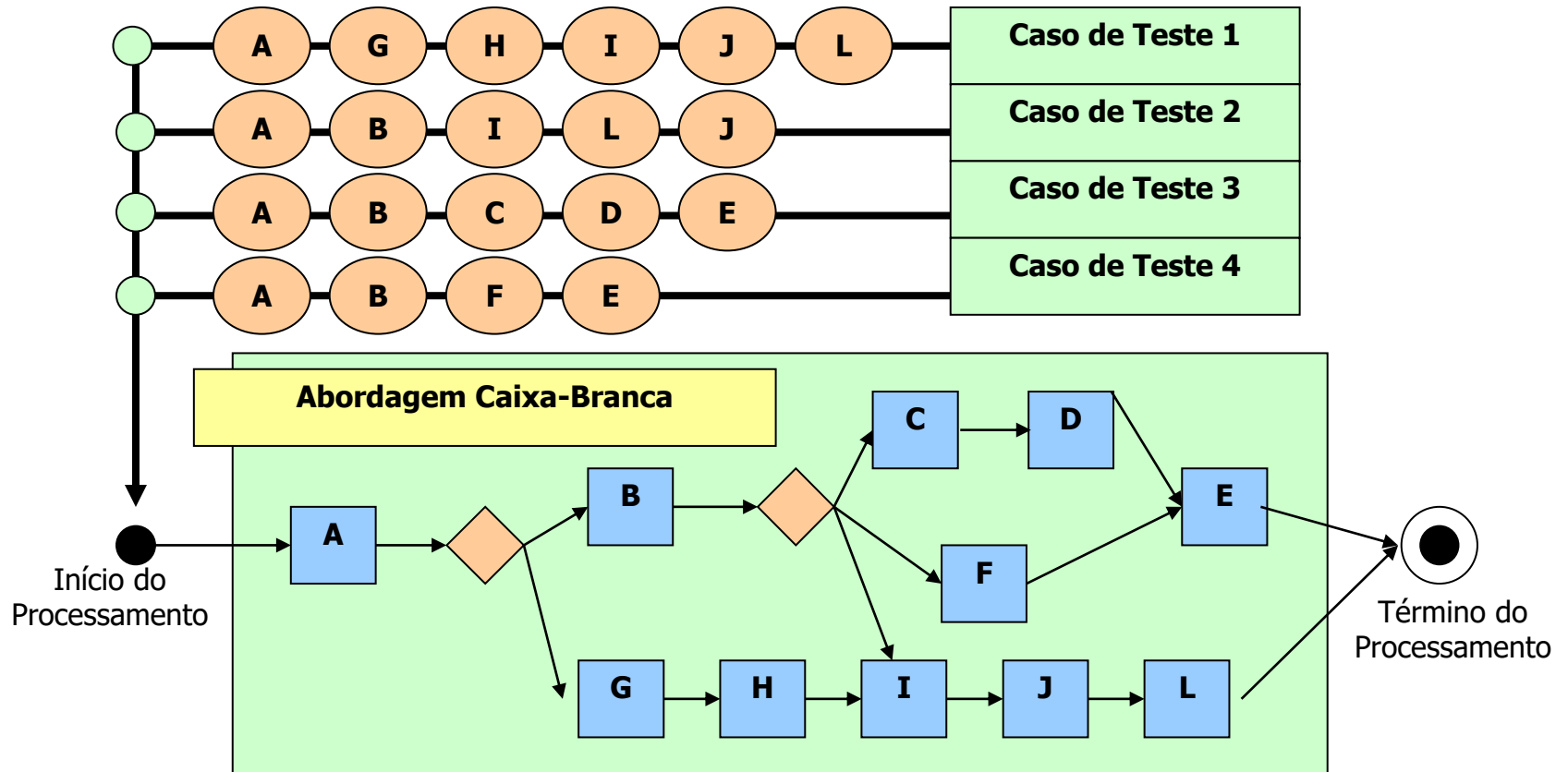


Priorizando as Categorias

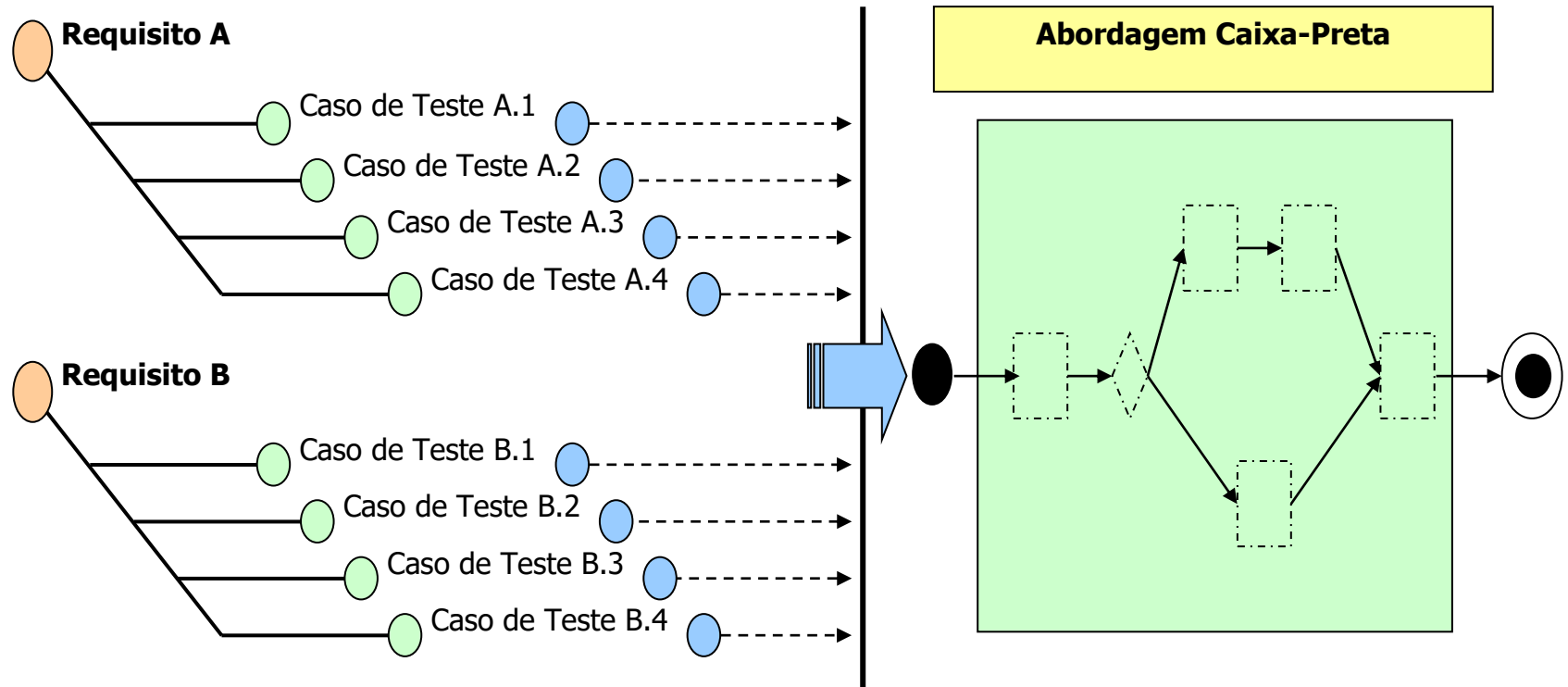
Características da Aplicação	Importância
01. Funcional	<i>Essencial</i>
02. Desempenho	<i>Médio Impacto</i>
03. Confiabilidade/Disponibilidade	<i>Alto Impacto</i>
04. Segurança	<i>Essencial</i>
05. Carga e Concorrência	<i>Alto Impacto</i>
06. Usabilidade	<i>Médio Impacto</i>
07. Compatibilidade	<i>Essencial</i>
08. Portabilidade	<i>Baixo Impacto</i>
09. Contingência	<i>Alto Impacto</i>
10. Instalação	<i>Médio Impacto</i>
11. Distribuição	<i>Alto Impacto</i>
12. Recuperação	<i>Alto Impacto</i>

CASOS DE TESTES

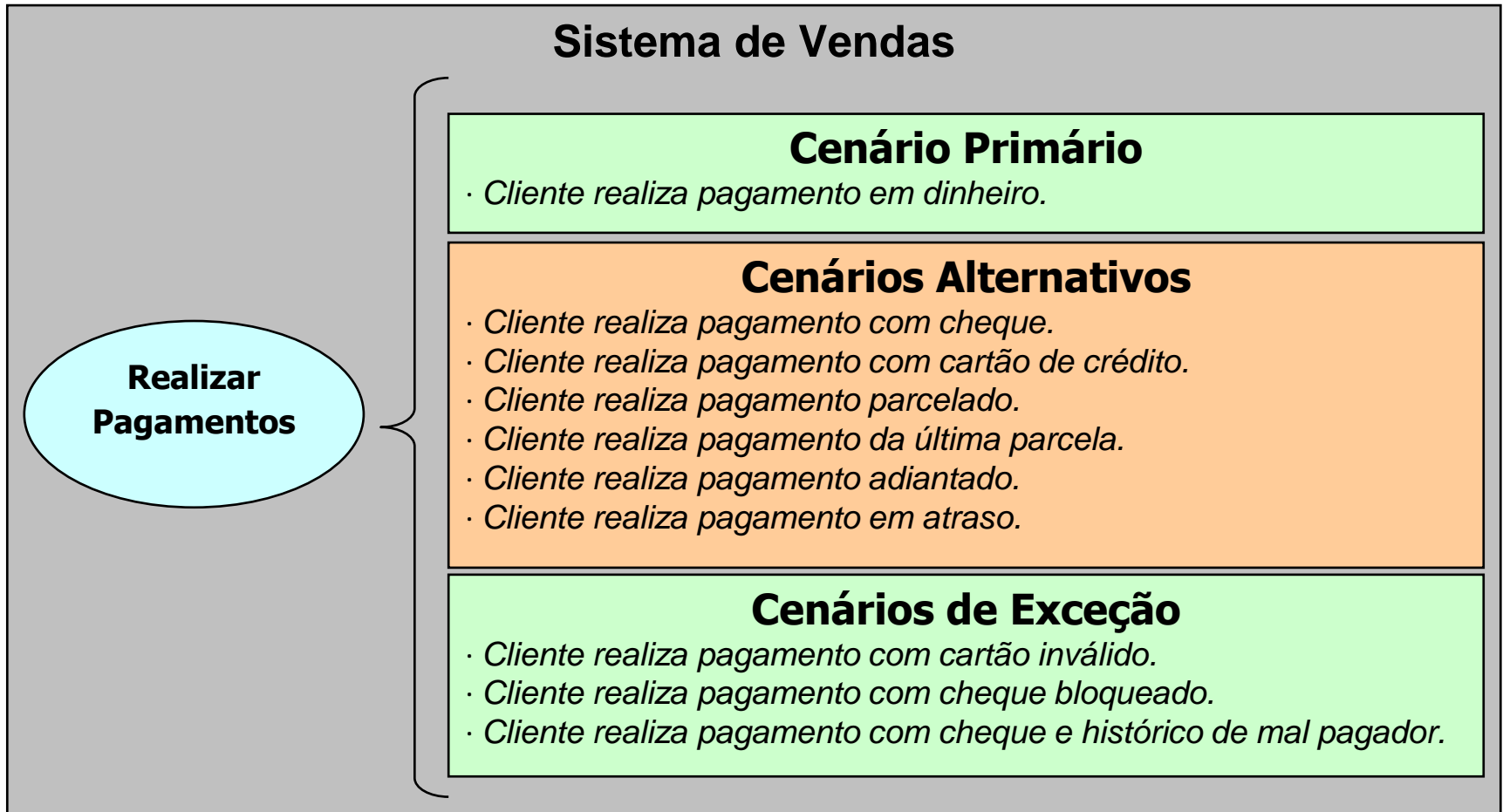
Casos de Testes de Caixa-Branca

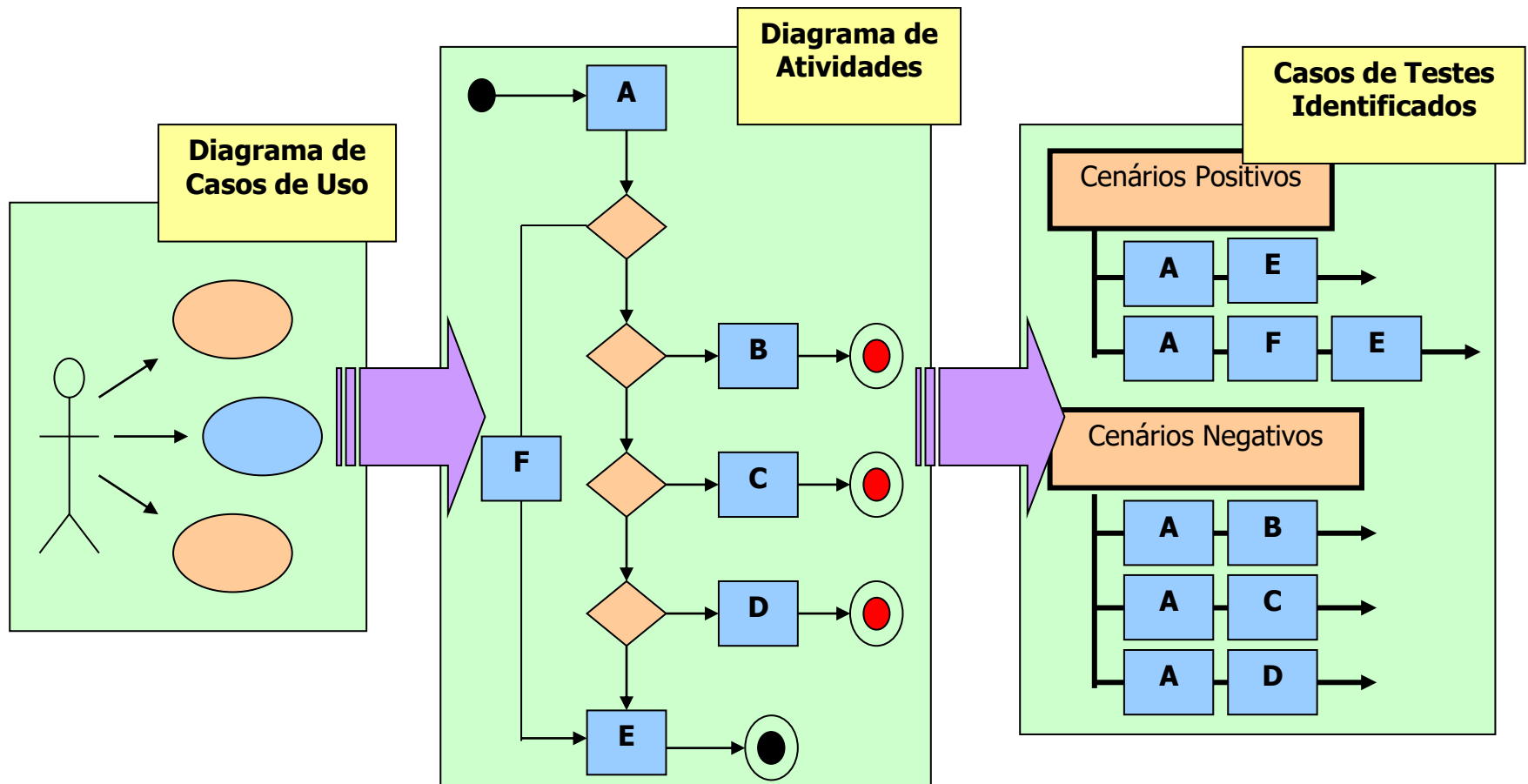


Casos de Testes de Caixa-Preta

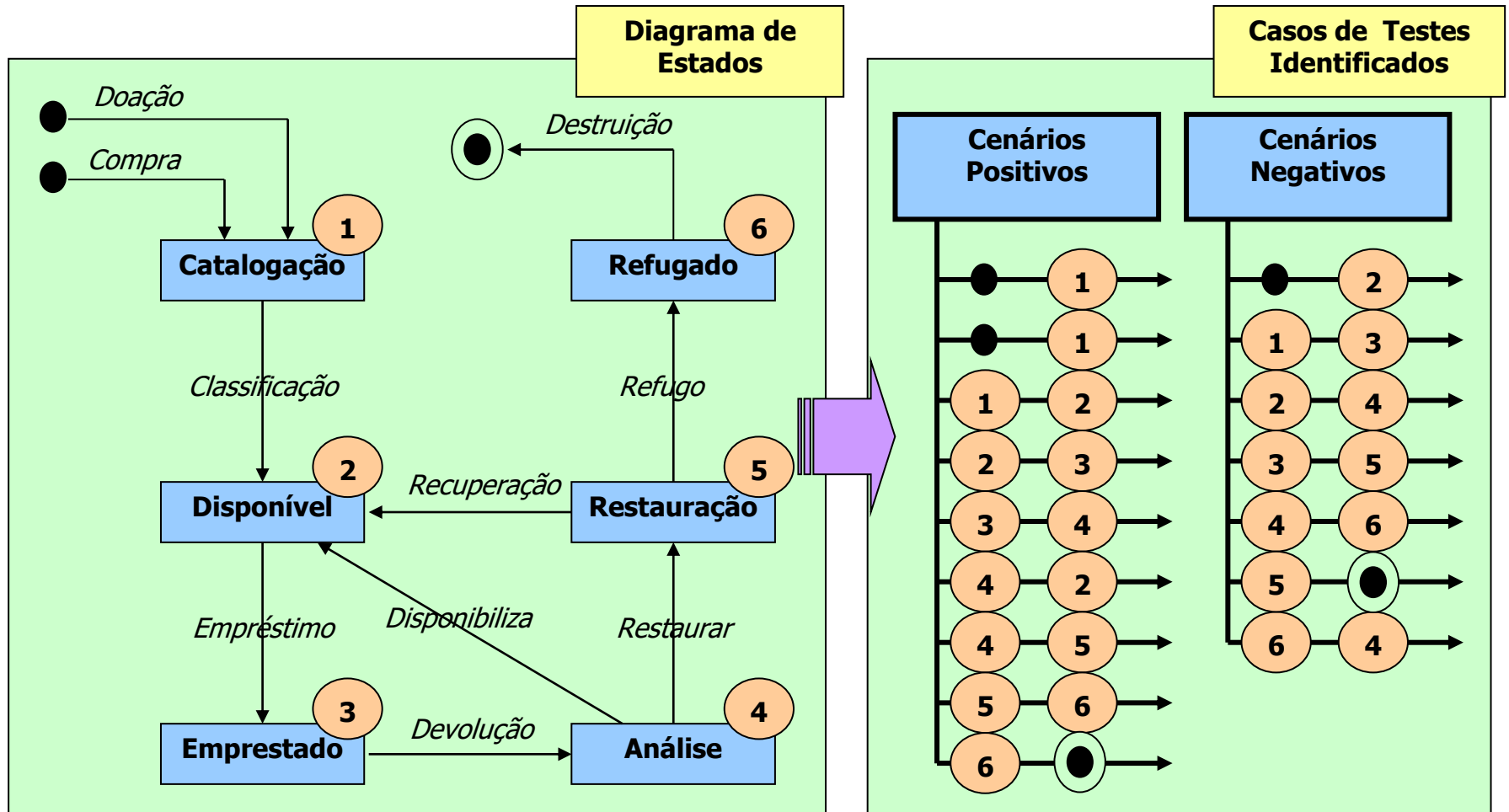


OBTENDO CASOS DE TESTES





Método de Análise de Documentos

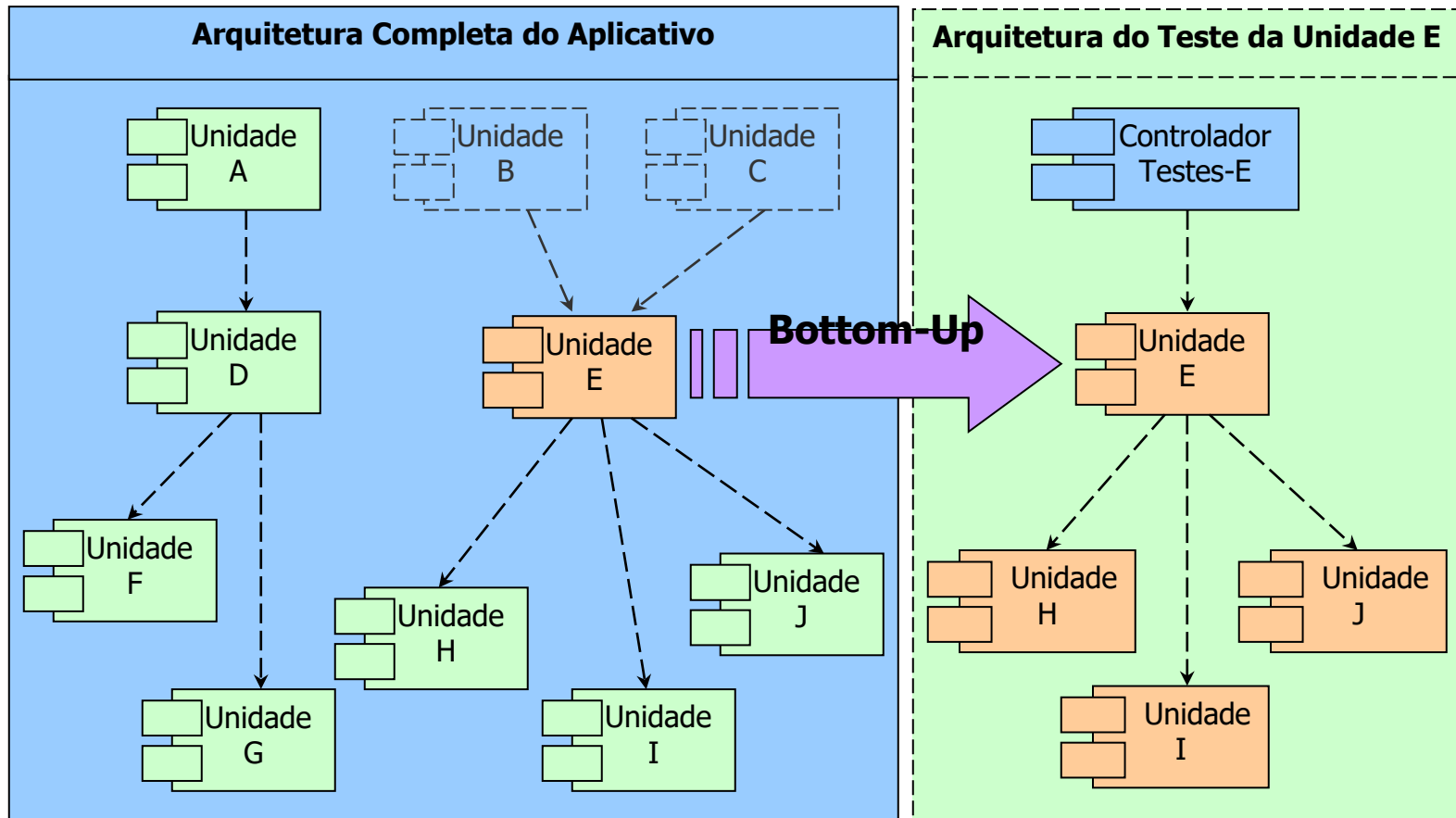


ESTÁGIOS DOS TESTES

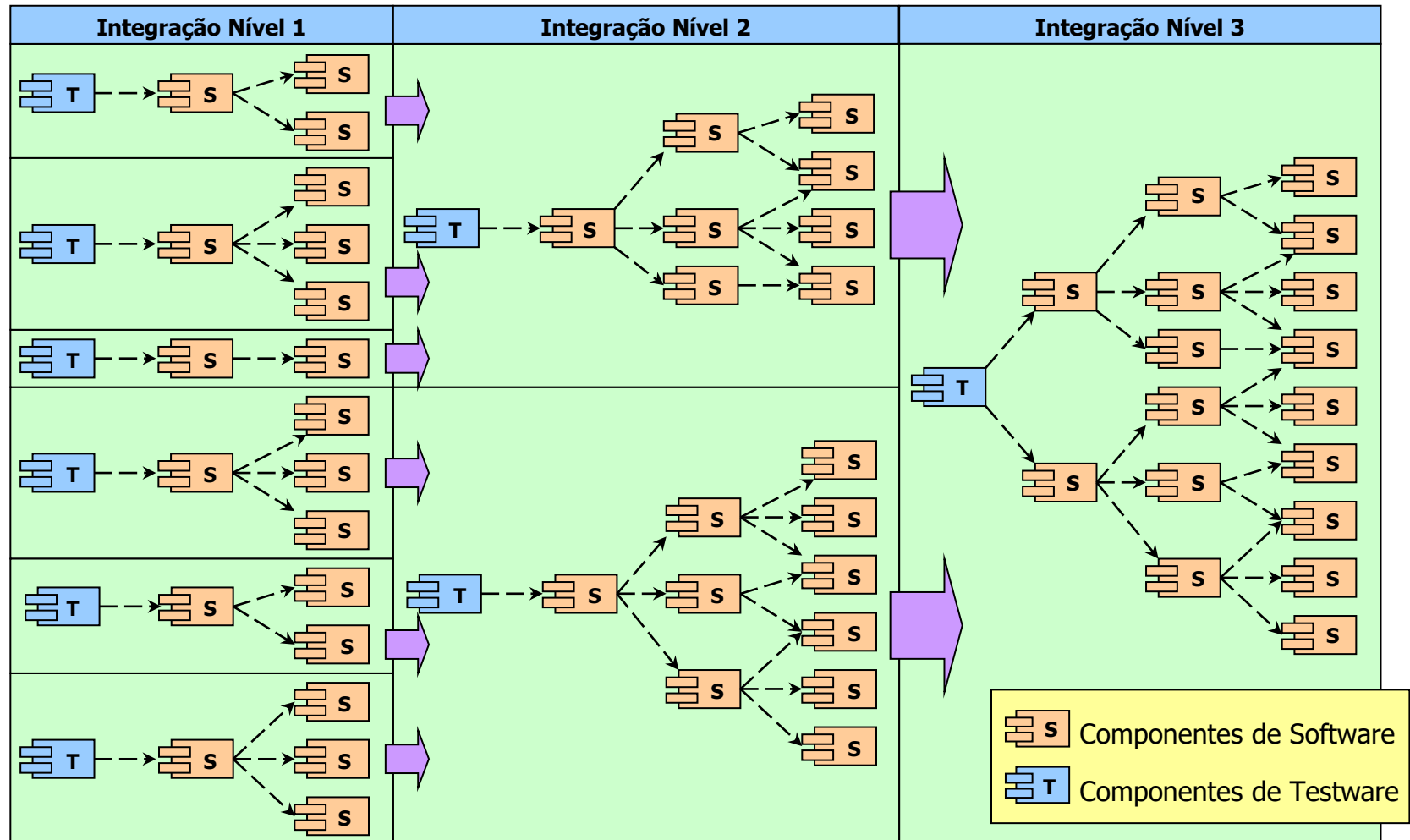
Estágios dos Testes de Software

	Fase da Validação	Categorias de Testes Aplicada	Características da Fase de Validação
Teste de Baixo Nível	Teste de Unidade	<ul style="list-style-type: none"> · Estrutura Interna; · Funcionalidade; · Usabilidade · Segurança; 	<ul style="list-style-type: none"> · Estratégia Caixa-Branca; · Testam partes do software; · Requer conhecimento da estrutura interna; · Executado pelo desenvolvedor ou profissional de teste.
	Teste de Integração	<ul style="list-style-type: none"> · Interfaces; · Dependências entre Componentes; 	<ul style="list-style-type: none"> · Estratégia de Caixa-Branca; · Testam integrações entre partes do software; · Requer conhecimento da arquitetura interna do software; · Executado pelo desenvolvedor ou profissional de teste.
Teste de Alto Nível	Teste de Sistema	<ul style="list-style-type: none"> · Funcionais; · Não Funcionais; <ul style="list-style-type: none"> ○ Performance; ○ Instalação; ○ Recuperação; ○ Carga; 	<ul style="list-style-type: none"> · Estratégia de Caixa-Preta; · Os testes são aplicados no software como um todo; · Não requer conhecimento da estrutura interna do software; · Requer ambiente muito semelhante ao da produção; · Deve ser executado por um grupo de teste independente.
	Teste de Aceitação	<ul style="list-style-type: none"> · Funcional; · Usabilidade; · Segurança; 	<ul style="list-style-type: none"> · Estratégia de Caixa-Preta; · Os testes são aplicados no software como um todo; · Não requer conhecimento da estrutura interna do software; · Requer ambiente muito semelhante ao da produção; · Deve ser executado pelos usuários finais.

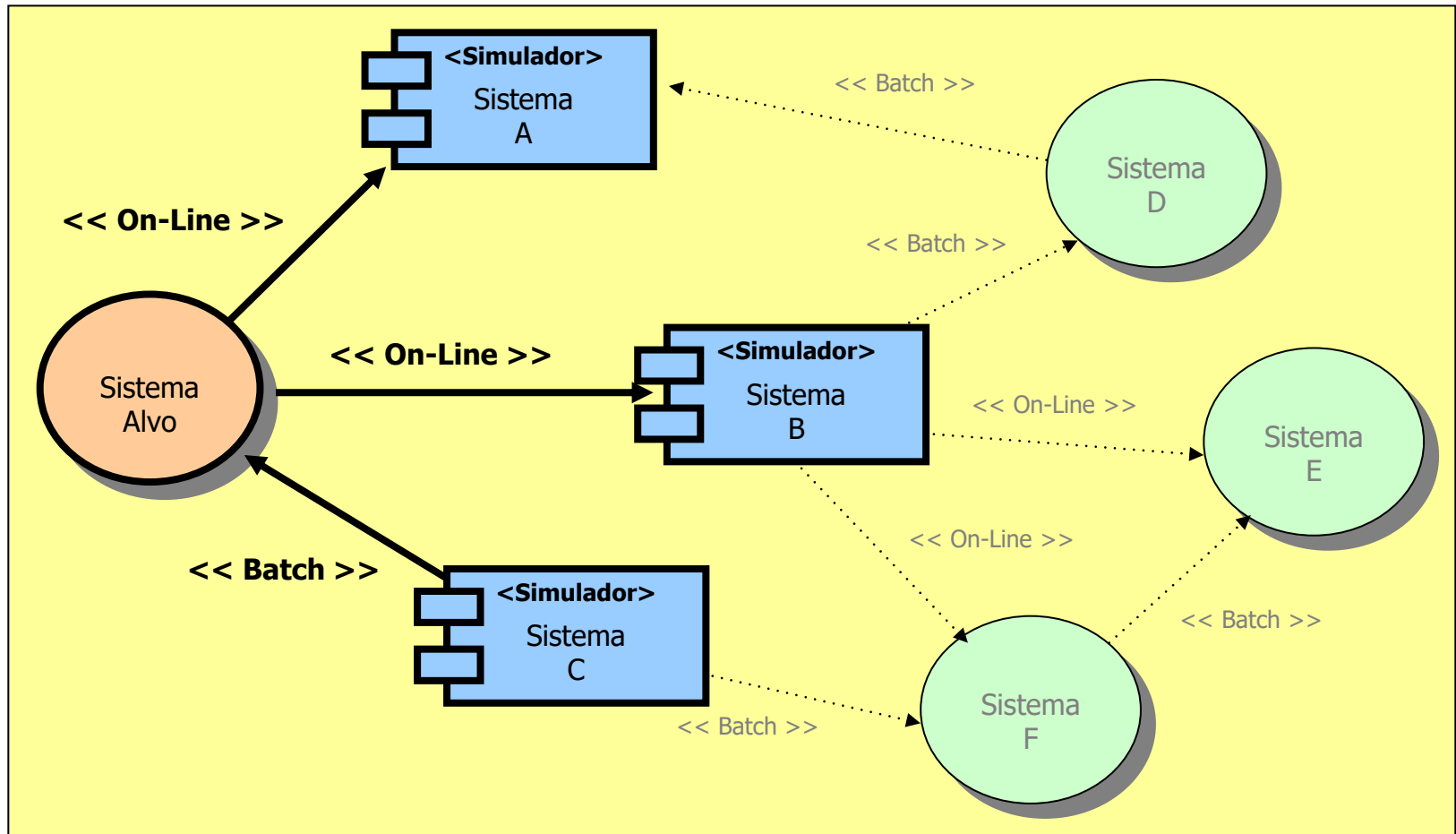
Teste de Unidade



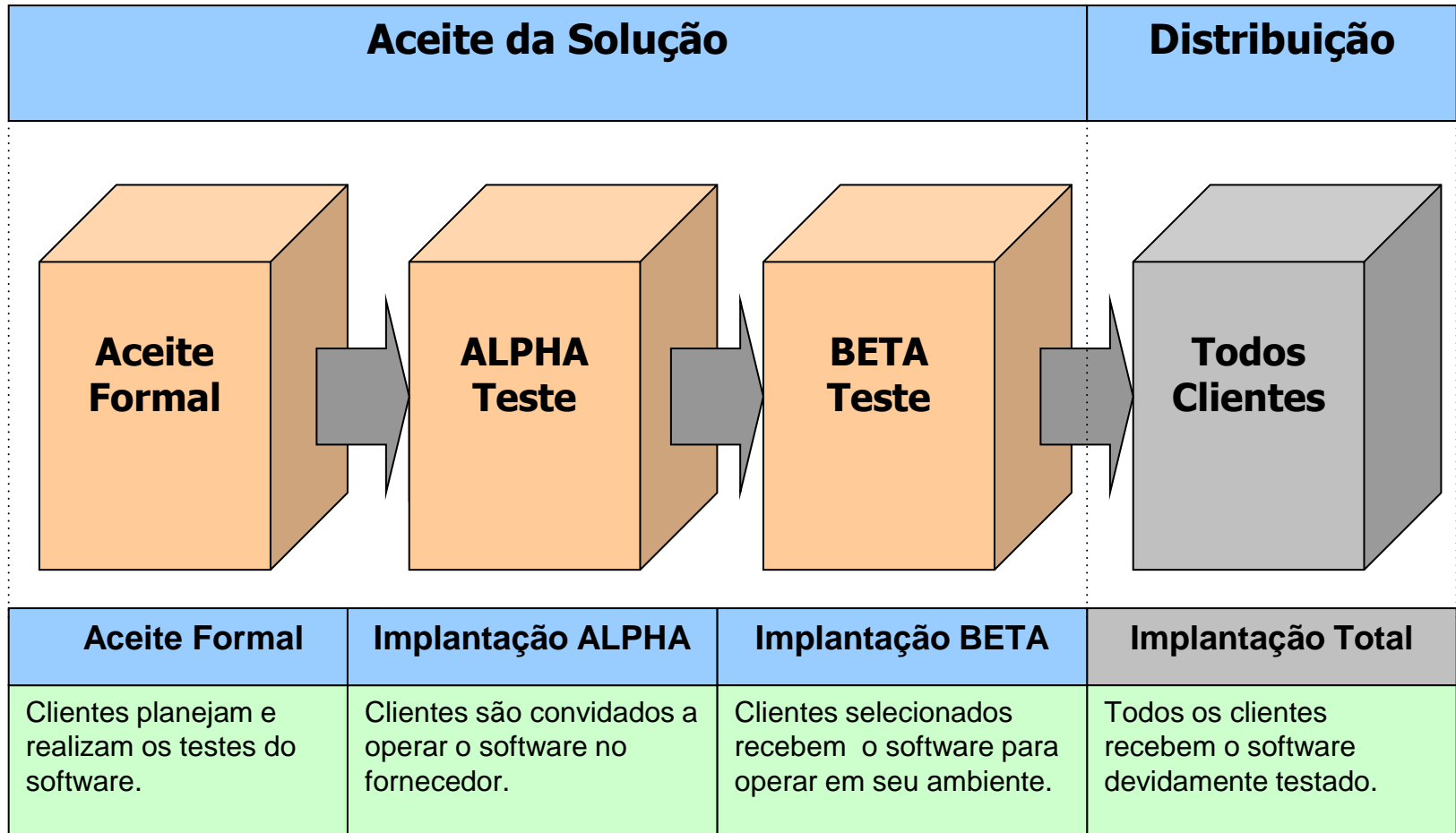
Teste de Integração



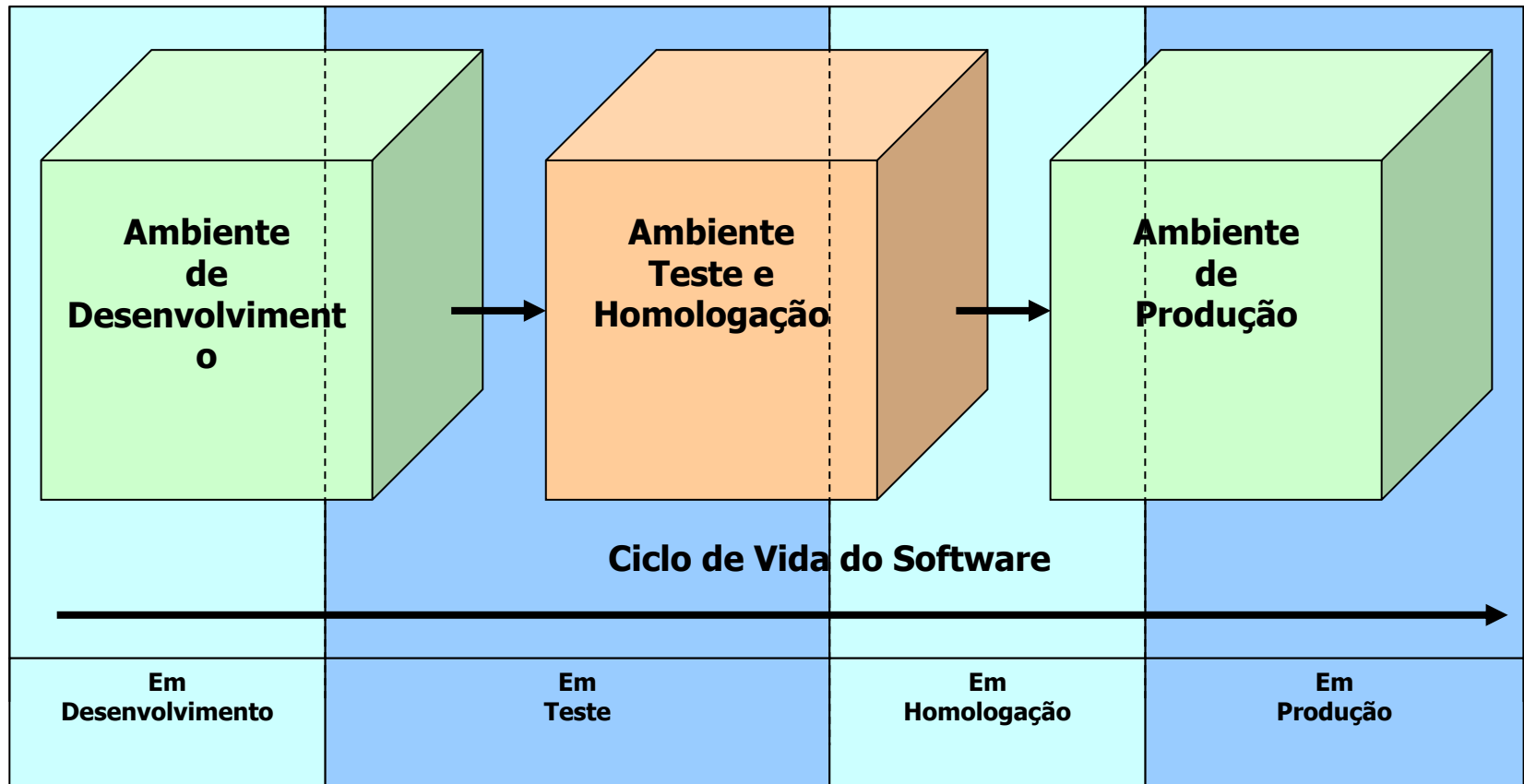
Teste de Sistema



Teste de Aceitação



Testes de Software em vários Ambientes



AUTOMAÇÃO DE TESTES

Automatização de Testes

Existem tarefas que são executadas de formas mais eficiente pelo computador do que pelo ser Humano.

O que é automação de Teste?

- Automação de teste é o uso de **software** para controlar a execução do teste de software;
- Comparação dos resultados esperados com os resultados reais de forma automática.

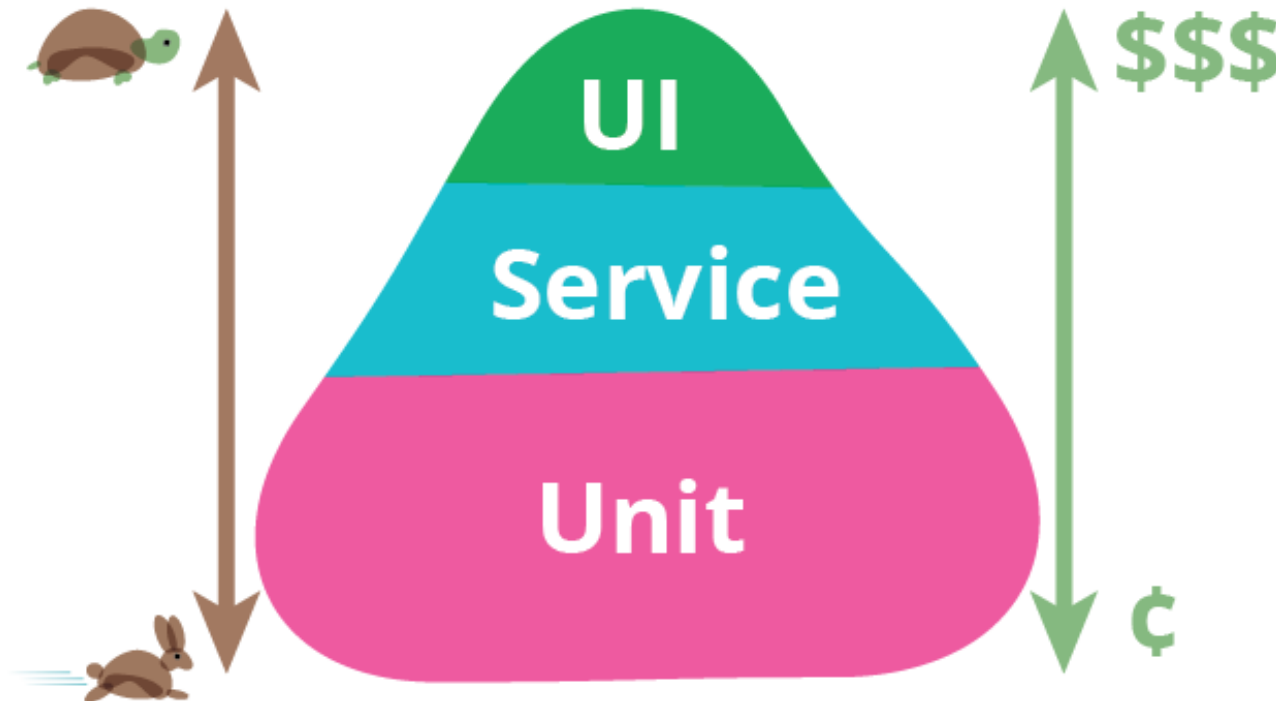


Manual X Automatizado

Testes Manuais	Testes Automatizados
Podem não ser precisos devido a erro humano.	Mais assertividade por ser executado por ferramentas.
Apoio humano analítico	Ferramentas certas e conhecimento
Praticidade quando necessário testar poucas vezes (2, 3x)	Melhor escolha para testes repetitivos e por grandes períodos
Testes de usabilidade	Testes de estresse e carga para experiência de usuário.

Pirâmide de Testes

- A imagem (Martin Fowler) já explica tudo, quanto mais perto do usuário mais caro e lento é o teste, tendo como topo o teste de interface, ou os famosos testes **e2e (end-to-end)**.



Pirâmide de Testes

- A "Pirâmide de Testes" é uma metáfora que nos diz para agrupar testes de software em diferentes granularidades.
- Também dá uma ideia de quantos testes devemos ter em cada um desses grupos.
- Embora o conceito da Pirâmide de Testes já existe há algum tempo, as equipes ainda lutam para colocá-lo em prática de forma adequada.

Créditos

Alexandre Bartie

alexandre_bartie@hotmail.com

Prof. Fabio Teixeira, MSc, PMP, ITIL

teixeif@gmail.com



F a c u l d a d e
IMPACTA
T E C N O L O G I A
