

# Resumão CTFL

Criador: Thiago Andrade

Linkedin: <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

Email: [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)

## Ajuda

Em algumas ramificações de tópicos existem hiperlink ou comentários para te ajudar a compreender melhor o tópico

<== Existem ramos com ícone de bandeira vermelha, significa que são assuntos, quentes, que provavelmente vai estar na prova

Este Mapa mental não exclui o estudo pelo syllabus, é muito importante estudar toda a apostila do syllabus

### [Syllabus 3.1](#)

Este Mapa mental não exclui o estudo por cursos de CFTL, é muito importante estudar por cursos voltados ao CTFL

#### Cursos

- [Curso CTFL Iterasys](#)
- [Curso CTFL Leonardo Carvalho](#)
- [Curso CTFL Udemey melhor avaliado](#)
- [Curso playlist Pessoni](#)

#### Exemplos de Exame CTFL

- Estude os simulados, na sua prova pode conter algumas questões dos simulados
- [Exemplo de Exame CTFL 01](#)
- [Exemplo de Exame CTFL 02](#)
- [Exemplo de Exame CTFL 03](#)

### [Resumo CTFL](#)

#### Simulados

- [Simulados 01](#)
- [Simulados 02](#)
- Ajuda Técnicas de teste
  - [Caixa preta](#)
  - [Caixa branca](#)

#### Informações

- Informações sobre a prova

- **Pré-requisitos:** nenhum
- **Número de questões:** 40
- **Tipo de questões:** múltipla escolha
- **Tempo de Exame\*:** 60 minutos
- **Pontuação:** 1 ponto por questão
- **Aprovação:** mínimo de 65% de acertos ou 26 pontos
- **Modalidades de Exame:** Acadêmico, Empresarial, Nacional, Especial, Treinamento e Online.
- **Valor do Exame:** ~~R\$ 800,00~~ (consulte os valores promocionais em cada modalidade de exame)

## ○ Informações Importantes

### Informações Importantes

1. **Não é obrigatório** que o candidato passe por um **treinamento** para participar do exame de certificação. Caso o candidato necessite, sugerimos que o faça com um dos **Provedores de Treinamento Certificados**, que possuem treinamentos, específicos para esta certificação, que foram avaliados e aprovados pelo BSTQB.
2. Todas as questões dos exames na língua portuguesa foram criadas e revisadas pela equipe do **GT Exame** do BSTQB.
3. Seguindo padrões internacionais do ISTQB, **não divulgaremos o gabarito**, nem **realizaremos qualquer revisão** na correção dos exames. O processo é eletrônico com múltiplas conferências manuais garantindo que não haja erros de correção.
4. Para auxiliar o candidato sobre sua evolução no exame, o resultado encaminhado conterá os percentuais de acerto em cada capítulo.
5. As **questões do exame são distribuídas** proporcionalmente ao tempo de estudo de cada capítulo, proposto pelo ISTQB no syllabus. Pequenas variações podem ocorrer a cada prova já que a divisão não é exata.

CAPÍTULO	1º	2º	3º	4º	5º	6º
QUESTÕES	7	5	3	12	9	4

## Glossário

- [Glossário padrão de termos utilizados](#)
- Base de teste
  - Todos os documentos dos quais os requisitos de um componente ou sistema podem ser deduzidos
- Carta de teste
  - Declaração dos objetivos do teste e de possíveis ideias sobre como realizar os testes. As cartas de teste são usadas em testes exploratórios.
- caso de uso
  - Sequência de transações em um diálogo entre um ator e um componente ou sistema, com um resultado tangível, onde um ator pode ser um usuário ou qualquer coisa que possa trocar informações com o sistema
- cenário de teste
  - Ver especificação de procedimento de teste
- ciclo de teste
  - Execução do processo de teste contra um único release identificável do objeto de teste
- cobertura

**Criador: Thiago Andrade**

**Linkedin:** <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

**Email:** [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)

- Grau, expresso como uma porcentagem, que indica o quanto um item de cobertura foi exercitado por uma suíte de testes
- Folha de sessão
  - São utilizadas para registrar etapas do teste e registrar descobertas feitas
- Objetivo do teste
  - Software, produto que vai ser testado
- Procedimentos de testes
  - Passo a passo ou sites do caso de teste
- Testware
  - Tudo que você é capaz de produzir
- Time-Box
  - Janela de tempo pré definida
- Caso de teste
  - Título e Descrição
- Roteiro de teste
  - Passo a Passo a passo do caso de teste

#### Normas:

- ISO/IEC/IEEE 29119-1
  - Conceito de teste
- ISO/IEC/IEEE 29119-2
  - Processo de teste
- ISO/IEC/IEEE 29119-3
  - Produtos de trabalho
- ISO / IEC / IEEE 29119-4
  - Técnicas de teste
- ISO20246
  - Processo de Revisão
- ISO25010
  - Características de qualidade

**Criador: Thiago Andrade**

**Linkedin:** <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

**Email:** [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)

Criador: Thiago Maciel

Elogios, reclamações ou colaboração

- Email: [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)
- [Linkedin: Thiago Andrade](#)

## Capítulo 01 Fundamentos do teste

O que é teste

- Processo de teste inclui execução de teste, planejamento de testes, análise, modelagem e implementação dos testes, relatórios de progresso e resultados de testes e avaliação da qualidade de um objeto de teste (Software ou Software Alvo ou SUT).
- Teste dinâmico: Execução de um componente ou sistema
  - Principais níveis de teste: Componente, Integração, Sistema e Aceite (CISA).
- Teste Estático: Teste não envolvem a execução do componente ou sistema (Checagem de documentação, comparar código, verificar padrão).
- Objetivos típicos do teste (São 8 no total):
  - Avaliar os produtos
  - Validar se o sistema em teste está funcionando como o esperado
  - Criar confiança no nível de qualidade do sistema
  - Evitar defeitos
  - Encontrar os defeitos e falhas
  - Fornecer informações suficientes em relação a qualidade
  - Reduzir o nível de risco da qualidade do software
  - Cumprir com os requisitos contratuais do sistema
- Quanto mais cedo for realizado os testes, maior será a cobertura e mais barato será pra corrigir as falhas
- Teste mostra as falhas causadas por defeitos e normalmente é feita pelo testador
- Depuração é a atividade de desenvolvimento que localiza, corrige os defeitos é feita pelo desenvolvedor (Testador também pode participar)

Porque o teste é necessário

- No entanto com o uso de técnicas de teste pode-se reduzir esses problemas de entrega:
  - Ter testadores envolvidos nas revisões de requisitos.

- Ter testadores trabalhando em conjunto com os projetistas do sistema
- Ter testadores trabalhando em estreita colaboração com os desenvolvedores
- Ter testadores para verificar e validar o software.
- Erros, defeitos e falhas
  - Erro (Engano) pode somente achar um erro ou engano a própria pessoa que gerou
  - Defeito ou falha ou Bug: Outra Pessoa encontrou o Defeito ou falha ou Bug
  - Um erro leva a um defeito que pode virar uma falha, mas não vice versa
- Defeitos, causas-raiz e efeitos
  - Ao identificar a causa-raiz pode reduzir a ocorrência de defeitos futuros similares

### Sete princípios de teste

- O teste mostra a presença de defeitos e não a sua ausência
- Testes exaustivos são impossíveis
- O teste inicial economiza tempo e dinheiro
- Defeitos se agrupam
- Cuidado com o paradoxo do pesticida
- O teste depende do contexto
- Ausência (Falácia) de erros é uma ilusão

### Processo de teste

- Não existe um processo universal de teste de software, mas há conjuntos comuns de atividades de teste
  - Processo de teste de uma organização
    - Modelo de ciclo de vida de desenvolvimento de software
    - Metodologia de projetos utilizados
    - Níveis de teste e tipos de testes
    - Riscos de produto e projeto
    - Domínio do negócio, conhecimento do negocio
    - Algumas restrições operacionais.
    - Políticas e praticas
    - Normas internas e externas.

**Criador: Thiago Andrade**

**Linkedin:** <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

**Email:** [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)

- Processo de teste organizacionais
  - Atividade e tarefas de teste
  - Produtos de trabalho de teste
  - Rastreabilidade entre a base de teste e os produtos de trabalho de teste.
- Atividades e tarefas de teste
  - Planejamento do teste
    - Envolve no que testar e a abordagem do teste para atender os objetivos do teste.
  - Monitoramento e controle de teste
    - Monitoramento: Controle de teste envolve a comparação contínua do progresso real com o plano de teste usando qualquer métrica de monitoramento definida pelo plano de teste. Acompanhar uma ação.
    - Controle: Tomada de ações necessárias para atender aos objetivos do plano de teste. Realizar uma ação.
    - Verificar os resultados em relação aos critérios de cobertura
    - Avaliar o nível de qualidade do sistema
    - Determinar se são necessários, mais testes
    - Progresso de teste em relação ao plano é comunicado as partes interessadas.
  - Analise do teste
    - Analisar a base de teste apropriado ao nível de teste (CISA):
      - As especificações dos requisitos
      - modelagem e a implementação de informações.
      - implementação do componente ou sistema
      - Relatórios de análise de risco.
    - Definir e priorizar as condições de teste para cada recurso
  - Modelagem do teste
    - Durante a modelagem de teste, as condições de testes são elaboradas em casos de teste de alto nível.
    - Modelagem de teste é como testar.
    - Projetar o ambiente de teste
  - Implementação do teste

- Implementação “Agora temos tudo pra executar os testes”
- Criar suítes de teste (Conjunto de teste) a partir dos procedimentos de teste (passo a passo)
- Construir ambiente de teste
- Preparar os dados de teste
- Falou em Script, falou em implementação
- Execução do teste
  - Gravar os identificadores e versões do item de teste, objeto de teste, da ferramenta de teste de software
- Conclusão do teste
  - Coletam os dados da atividade de teste já concluídas para consolidar a experiencia, o testware e qualquer outra informação relevante.
  - Todos os defeitos encontrados foram fechados e garantir que esses defeitos não fechados sejam registrados para serem resolvidos
- Produtos de trabalho de teste
  - Produtos de teste são criados como parte do processo de teste
    - Criados através de uma série de ferramenta que gerenciam os testes, defeitos e execução.
  - Produtos de trabalho do planejamento do teste
    - Produtos de trabalho de planejamento do teste geralmente incluem um ou mais planos de teste.
  - Produtos de trabalho de monitoramento e controle de teste
    - Geralmente incluem vários tipos de relatórios, incluindo relatórios de progresso de teste.
  - Produtos de trabalho da análise do teste
    - Condições bem definidas e priorizadas, preferencialmente onde cada uma das quais é bidireccionalmente rastreável
  - Produtos de trabalho da modelagem do teste
    - Resulta em casos de teste e conjuntos de casos de teste para exercer as condições de teste definidas na análise de teste.
  - Produtos de trabalho da implementação do teste
    - Procedimentos de teste e seu sequenciamento (passos ou steps)
    - Suítes de teste (Conjuntos de teste)

**Criador: Thiago Andrade**

**Linkedin:** <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

**Email:** [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)



- Cronograma de execução do teste
- Produtos de trabalho da execução do teste
  - A documentação sobre os quais os itens de teste, objetos de teste, as ferramentas de teste e o testware estavam envolvidos no teste.
- Produtos de trabalho de conclusão de teste
  - Trabalho de conclusão incluem os relatórios de resumo de teste, os itens de ação para melhoria de projetos subsequentes ou iterações.

## Psicologia do teste

- A mentalidade de um testador deve incluir
  - curiosidade
  - pessimismo profissional
  - Olho critico
  - atenção aos detalhes
  - motivação para comunicação
  - relacionamento bons e positivos

## Capítulo 02 Teste durante o ciclo de vida de desenvolvimento de software

### Modelos de ciclo de vida de desenvolvimento de software

- É uma forma de organizar o processo de desenvolvimento de software e como as atividades se relacionam com as outras de forma logica e cronológica.

### Níveis de Teste

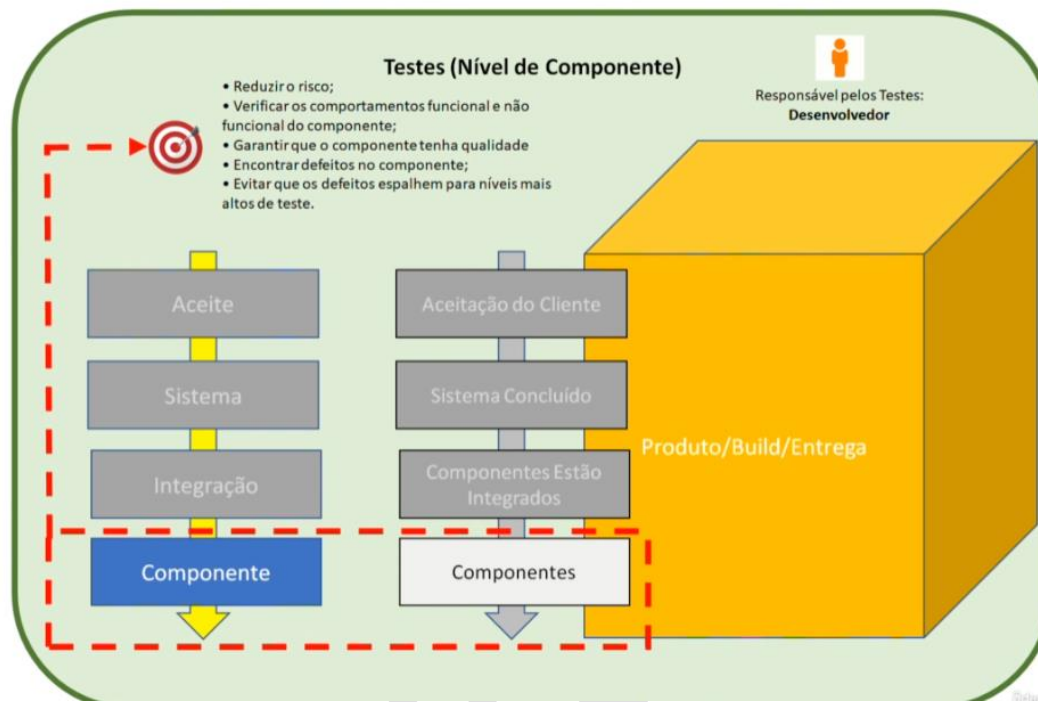
- Os Níveis de teste são grupos de atividades de teste que são organizados e gerenciados juntos
- Cada nível de teste é uma instância do processo de teste
- Nível de teste significa teste dinâmico
- Níveis de teste são caracterizados por:
  - Objetivos específicos
  - Base teste
  - Objeto de teste
  - Defeitos e falhas típicas
  - Abordagens e responsabilidades especificas
- (C) Teste de Componente:

**Criador: Thiago Andrade**

**Linkedin:** <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

**Email:** maciel.thiago@gmail.com

- Componentes que são testáveis separadamente
- Podem exigir objetos simulados (Mock = Driver (Simula a chamada) ou Stubbing (Responder a chamada))
- Teste funcional, características não funcionais e propriedade estruturais.
- Base de teste:
  - Projeto detalhado
  - Código
  - Modelo de dados
  - Especificação de componentes
- Objeto de teste:
  - Componentes, unidade ou módulos
  - Estrutura de código e dados
  - Classes
  - Módulos de comunicação com o banco de dados.
- Defeitos típicos de falha:
  - Funcionalidade incorreta
  - Fluxo de dados
  - Código e lógica incorretos.
- Geralmente é executado pelo desenvolvedor
  - Normalmente codificam e depois fazem o teste
- Resumo



- (I) Teste de Integração (Teste de Serviço ou de API)
  - Teste entre interações entre componentes ou sistemas.
  - Verificar se os componentes funcionais e não funcionais das interfaces
  - Existem 2 níveis diferente:
    - Integração de componentes
      - Integrações entre componentes
      - Geralmente feito por desenvolvedores
    - Integração de sistemas
      - Interações entre sistemas pacotes e micro serviços
      - Geralmente feito por Testadores
  - Base de teste:
    - Software e modelagem do sistema
    - Diagrama de sequencia
    - Especificações de interface e protocolo de comunicação
    - Casos de uso
    - Arquitetura no nível de componente ou sistema
    - Fluxos de trabalho
    - Definições de interface externa

■ Objetivos de teste:

- Subsistemas
- Banco de dados
- Infraestrutura
- Interfaces
- APIs
- Micro serviços

■ Defeitos típicos e falhas componentes e sistemas:

- Dados incorretos, dados ausentes ou codificação incorreta.
- Sequenciamento ou temporização
- Incompatibilidade de interface
- Falhas na comunicação entre componentes
- Falha de comunicação não manipulada ou tratada de forma errada entre componentes
- Suposições incorretas sobre o significado
- Além desses acima o de sistema inclui:
  - Estruturas de mensagens inconsistentes entre sistemas
  - Falha de comunicação entre sistemas
  - Falha no cumprimento dos regulamentos de segurança obrigatórios

■ Devem ser feitos incremental (Pequeno número de componentes adicionais ou sistemas por vez.)

■ Resumo



#### Base de Teste

- Especificações, Arquitetura e Modelagem e especificações
- Diagrama de sequência
- Protocolos de comunicações
- Casos de uso
- Fluxos de trabalho
- Definições de interface externa



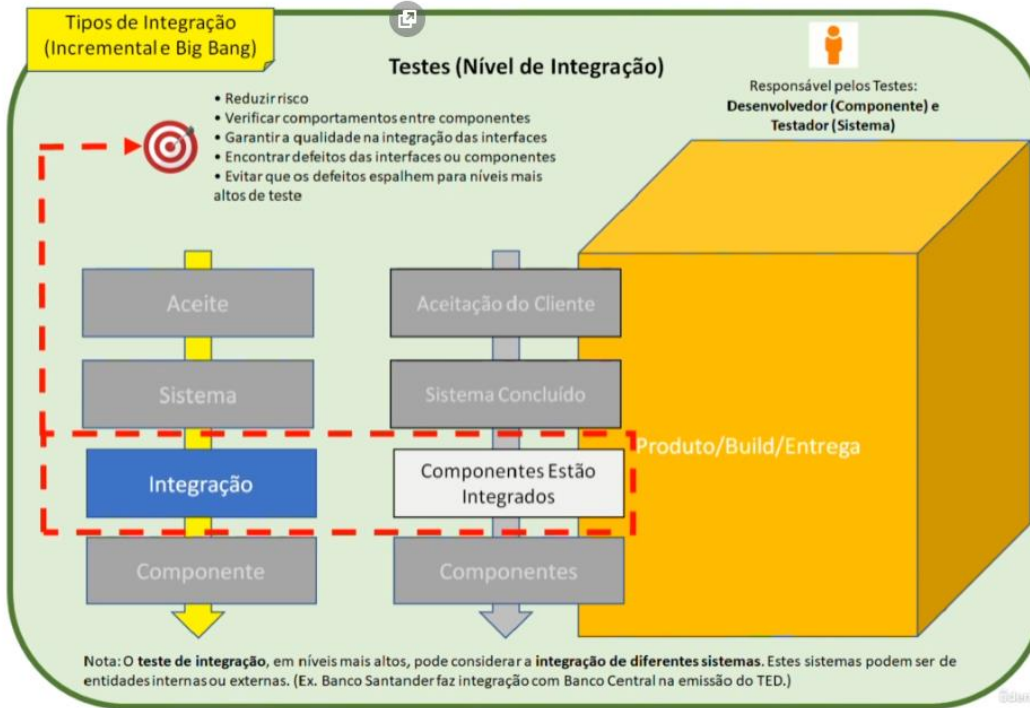
#### Objetos de Teste

- Subsistemas
- Bancos de dados
- Interfaces
- APIs



#### Defeitos Típicos

- Dados incorretos
- Chamadas incorretas de interface
- Interfaces incompatíveis
- Componentes não se comunicam
- Falta de tratamento de erros em componentes



- (S)Teste de sistemas

- Concentra no comportamento e nas capacidades de todo um sistema ou produto

- Execuções de ponta a ponta do sistema e os comportamentos não funcionais

- Reduzir os riscos
- Verificar se os componentes funcionais e não funcionais do sistema estão como projetados e especificados
- Validar se o sistema está completo e funcionara como esperado
- Criar confiança na qualidade
- Encontrar defeitos
- Evitar que os defeitos se espalhem

- Base de teste:

- Especificações de requisitos e sistemas
- Relatório de análise de risco
- Casos de uso
- Épicos e histórias de usuário
- Modelos comportamento do sistema
- Diagrama de estado
- Sistemas e manuais de usuário.

Criador: Thiago Andrade

Linkedin: <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

Email: [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)

## ■ Objetivos do teste:

- Aplicações
- Sistemas de hardware e software
- Sistemas operacionais
- Sistema sob teste (SUT)
- Configuração de sistema e dados

## ■ Defeitos e falhas:

- Cálculos incorretos
- Comportamento funcional e não funcional do sistema incorreto ou inesperado
- Controle e fluxo de dados dentro do sistema
- Falha na execução de tarefas
- Falha no ambiente
- Falha do sistema para funcionar conforme descrito nos manuais do sistema e do usuário.

## ■ Responsabilidade da equipe de teste

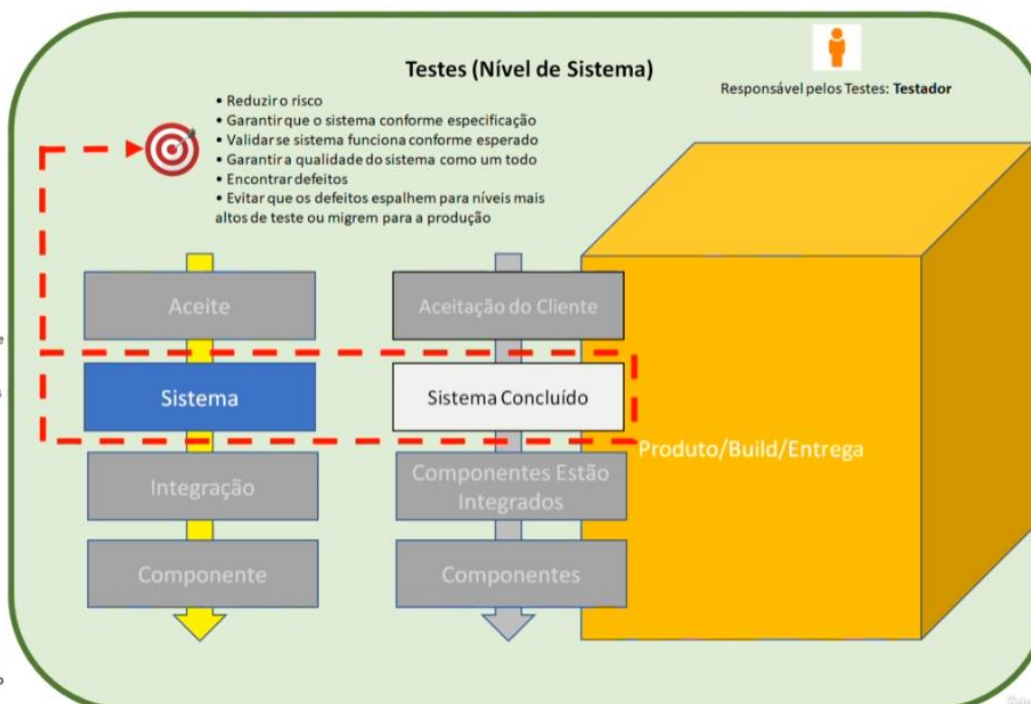
- De preferência de testadores independentes

## ■ Resumo

- Base de Teste**
- Especificações de requisitos funcionais e não funcionais
  - Relatórios de análise de risco
  - Casos de uso
  - Épicas e histórias de usuários
  - Diagramas de estado
  - Sistema e manuais do usuário

- Objetos de Teste**
- Sistemas de hardware e software sob teste (SUT)
  - Sistemas operacionais
  - Configuração do sistema e dados de configuração

- Defeitos Típicos**
- Cálculos incorretos
  - Comportamento inesperado do sistema
  - Controle e/ou fluxos de dados incorretos
  - Falha na execução de tarefas funcionais de ponta a ponta
  - Falha na execução do sistema dentro do ambiente de produção



- (A) Teste de Aceite ou homologação

Criador: Thiago Andrade

Linkedin: <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

Email: [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)

- Se concentra no comportamento e na capacidade de todo um sistema ou produto
  - Validar que o sistema está completo e funcionara como esperado
- Formas comuns de teste de aceite:
  - Aceite do usuário (UAT)
    - Focado em validar a adequação do uso do sistema pelo usuário pretendido em um ambiente operacional real ou simulado.
  - Aceite operacional (OAT)
    - O teste de aceite do sistema pelas operações ou pela equipe de administração de sistemas geralmente é realizado em um ambiente de produção
  - Aceite contratual e regulatório
    - Realizado com base nos critérios de aceite de um contrato para desenvolver softwares específicos.
    - Realizado por testadores ou usuários
  - Alfa e beta
    - Realizado pra software lançado em grande escala, comercial ou prateleira
- Base de teste:
  - Processos de negócios
  - Requisitos de usuário ou de negócios
  - Regulamentos, contratos legais e normas
  - Casos de uso
  - Requisitos de sistemas
  - Documentação do sistema ou usuário
  - Procedimento de instalação
  - Relatórios de análise de teste
  - Procedimentos de backup e restauração, recuperação
  - Requisitos funcionais
  - Documentação de operações
- Objetivos de teste típicos
  - Sistema sob teste (SUT)

**Criador: Thiago Andrade**

**Linkedin: <https://www.linkedin.com/in/thiago-andrade-2a97555b/>**

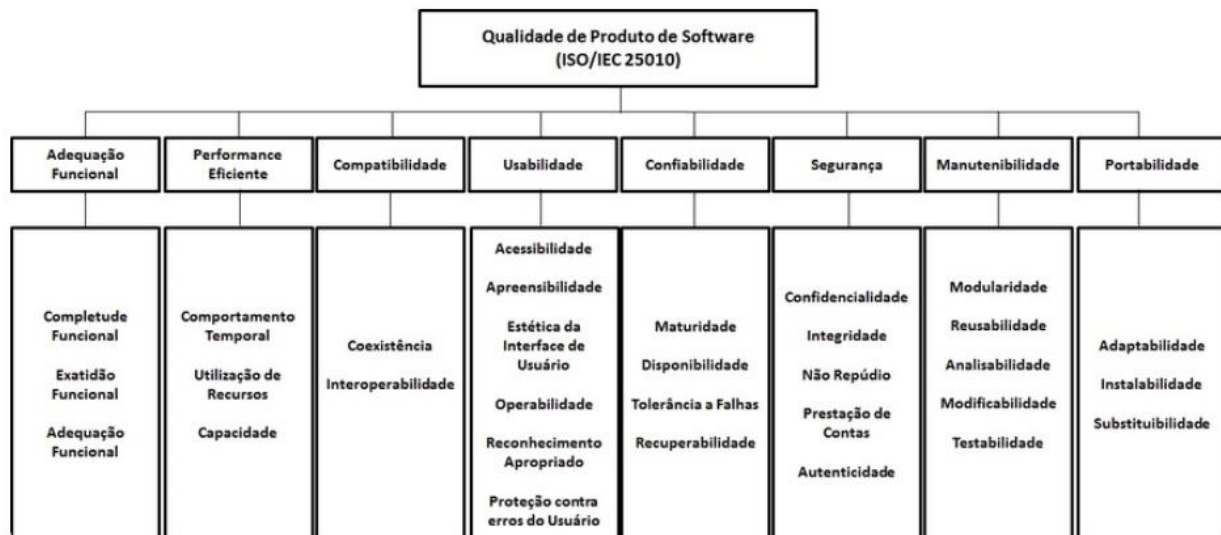
**Email: [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)**

- Configuração do sistema e dados de configuração
- Processos de negócios para um sistema totalmente integrado
- Sistema de recuperação e hot sites (Continuidade de negócios e testes de recuperação de desastres)
- Processos operacionais
- Formulários
- Relatórios
- Dados de produção existentes e convertidos
- Defeitos e típicos e falhas
  - Fluxos de trabalho do sistema não atendem aos requisitos do negocio
  - Regras de negócio não são implementados corretamente
  - Sistema não satisfaz os requisitos contratuais ou regulatórios
  - Falha não funcionais, como vulnerabilidade de segurança, eficiência de performance inadequada sob altas cargas ou operação inadequada em lima plataforma
- Abordagens e responsabilidades
  - Geralmente é responsabilidade dos clientes, usuários de negócios, proprietários de produtos ou operadores de um sistema
  - Teste alfa e beta pode ocorrer no final de cada iteração
- Pode ter teste tipo caixa branca
- Resumo





- Avaliam as características de sistemas e de software, como usabilidade, eficiência de performance e ou segurança.
  - Teste não funcional é o teste de “quão bem” o sistema se comporta
  - Pode e geralmente deve ser realizado em todos os níveis de teste e feito o mais cedo possível.
  - Pode envolver habilidades ou conhecimentos especiais
  - UPS (Usabilidade, performance e segurança) 3 principais testes não funcionais
  - Começa a partir do teste de componente
- Teste caixa branca:
    - Com base na estrutura interna ou na implementação do sistema
    - Pode envolver habilidades ou conhecimentos especiais.
    - Normalmente é feita pelo desenvolvedor
  - Teste Relacionados a mudanças:
    - Quando são feitas mudanças em um sistema deve se testar para confirmar que as alterações ou defeitos foram corrigidos.
  - Exemplos de Tipos e níveis de teste
    - 8 características e 33 subcaracterísticas



- Resumão
  - Tipo de teste é uma forma de testar existem 3 tipos de testes: Teste funcionais, não funcionais e relacionado a mudanças. Na Iso 25010 existem 8 características e 33 subcaracterísticas. Adequação funcional e funcionalidade significam a mesma coisa, são testes relacionados a negócio As outras características são testes técnicos ou não funcionais Os tipos de testes técnicos não funcionais mais comuns são (UPS)

Criador: Thiago Andrade

Linkedin: <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

Email: maciel.thiago@gmail.com

usabilidade, performance e segurança Teste caixa preta, executar o software ou comportamento, tem a divisão funcional ou não funcional Teste caixa branca, vai avaliar a lógica ou código fonte (Teste componente, teste integração de componente) Teste caixa preta e teste de caixa branca se aplica em todos os níveis Teste baseado na experiência, onde o testador tem que ter conhecimento similar, ramo de negócio ou conhecimento naquela tecnologia Níveis de teste são as etapas de teste dinâmico CISA (Componente, integração, sistema e aceite) ou CISIA (Componente, integração de componente, sistema, integração de sistemas e aceite) Existem teste funcional e não funcional em todos os níveis

### Teste de manutenção:

- Após serem implantados em ambientes de produção, o software e os sistemas precisam ser mantidos
- Manutenção é necessária para preservar ou melhorar as características de qualidade não funcionais do componente ou do sistema ao longo de sua vida
- Gatilho para manutenção
  - Motivos para começar o teste de manutenção ocorre alguma mudança:
    - Modificação: Mudança, correção, melhoria em algum componente do sistema, funcionalidade
    - Migração (Hot fixes): Como de uma plataforma para outra, comparar os sistemas pra conhecer as funcionalidades com o novo sistema.
      - Precisa de uma garantia que se acontecer algum problema, tenha uma forma de voltar atrás.
    - Aposentadoria: Quando o aplicativo atinge o fim de sua vida
    - Quando um aplicativo ou sistema é retirado, isso pode exigir o teste de migração ou arquivamento de dados.
    - Garantir antes da modificação, migração ou aposentadoria o procedimento de restauração.

## Capítulo 03 Teste estático

### Noções básicas sobre testes estáticos:

- Em contraste com o teste dinâmico o Teste Estáticos envolve em ler, comparar...
- Teste estático depende do exame manual dos produtos de trabalho, isto é, revisões
- Teste estático depende da avaliação orientada por ferramentas do código ou produtos de trabalho, isto é, análise estática, feita automática.
- Os dois tipos acima avaliam o código do produto sem executar o código.
- Análise estática é importante para sistemas críticos de segurança e frequentemente incorporada aos sistemas automatizados de criação e distribuição.

**Criador: Thiago Andrade**

**Linkedin: <https://www.linkedin.com/in/thiago-andrade-2a97555b/>**

**Email: [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)**

- Produtos de trabalho que podem ser examinados por testes estáticos
  - Análise estática pode ser aplicada em uma estrutura formal (Normalmente código ou modelos) como também aplicada na linguagem natural (verificação de ortografia, gramática...)
- Benefícios do teste estático:
  - Quando aplicado no início permite a detecção antecipada dos defeitos antes da realização dos testes dinâmicos
  - Localizar e corrigir os defeitos rapidamente é quase sempre muito mais barato para a organização
  - Prevenir os defeitos no desenho ou codificação
- Diferenças entre estático e dinâmico:
  - Estático sistema não está em execução! = Dinâmico sistema está em execução
  - Estático encontra-se defeito no dinâmico encontra-se falha
    - Após acionar o defeito que ele vira uma falha
  - Estáticos e dinâmicos se completam
  - Defeitos comuns testes estáticos:
    - Defeitos em requisitos (Ambiguidades, contradições)
    - desenho (Algoritmos ineficientes ou estruturas banco de dados.)
    - codificação (Variáveis com valores indefinidos, variáveis declaradas...)
    - Desvios de padrões (Falta de aderência aos padrões de codificação)
    - especificações incorretas na interface (Unidades de medidas diferentes usadas pelo sistema)
    - vulnerabilidade segurança (sensibilidade a estouro do buffer)
    - Lacunas ou imprecisões na rastreabilidade (Falta de teste para um critério de aceite).
  - Maioria dos defeitos de manutenção só podem ser encontrados por teste estáticos

## Processo de revisão

- As revisões variam de formal para informal
- Informais caracterizam por não seguir um processo definido e não ter um resultado formal documentado
- Formais caracterizam pela participação da equipe, com resultados documentados e procedimentos documentados para revisão. A formalidade depende do modelo de ciclo de vida, maturidade, complexidade do produto...

**Criador: Thiago Andrade**

**Linkedin: <https://www.linkedin.com/in/thiago-andrade-2a97555b/>**

**Email: [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)**

- Processo de revisão compreende as seguintes atividades:

- **Planejar**

- Definir o escopo, que inclui o propósito da revisão, quais documentos ou partes de documentos que devem ser revisados
- Estimativa de esforço
- Identificar as características da revisão
- Selecionar as pessoas para participar da revisão
- Definir os critérios de entrada e saída para os tipos de revisão mais formais
- Verificar se os critérios de entrada foram atendidos

- **Iniciar Revisão:**

- Distribuir o produto de trabalho e outros materiais
- Explicar o escopo, os objetivos, o processo...
- Responder a quaisquer perguntas que os participantes possam ter sobre a revisão

- **Revisão individual:**

- Rever todo ou parte do produto de trabalho
- Observar os possíveis defeitos

- **Analisar e comunicar:**

- Comunicar possíveis defeitos identificados
- Analisar possíveis defeitos, atribuindo responsáveis
- Avaliar e documentar características e qualidades
- Avaliar as conclusões da revisão em relação aos critérios de saída para tomar decisões de revisão

- **Corrigir e reportar:**

- Criar relatórios de defeitos para as descobertas que exigem mudança
- Corrigir os defeitos encontrados
- Comunicar os defeitos a pessoa da equipe
- Registrar o status atualizado dos defeitos
- Métricas de coleta
- Verificar se os critérios de saída foram estabelecidos

**Criador: Thiago Andrade**

**Linkedin:** <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

**Email:** [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)

- Aceitar os produtos de trabalho quando os critérios de saída são atingidos
- Funções e atividades de uma revisão formal:
  - Autor:
    - Criar o produto de trabalho
    - Corrigir os defeitos no produto de trabalho sob revisão
  - Gestor
    - Responsável pelo planejamento da revisão
    - Decidir sobre a execução das revisões
    - Atribuir pessoal, orçamento e tempo
    - Monitorar a rentabilidade continua
  - Facilitador (Moderador):
    - Garantir a execução eficaz das reuniões de revisão
    - Mediar, se necessário entre vários pontos de vista
    - Frequentemente a pessoa sobre quem o sucesso da revisão depende
  - Lider de revisão:
    - Assumir a responsabilidade geral pela revisão
    - Decidir quem será envolvido organizar quando e onde acontecerá a revisão
  - Revisor:
    - Podem ser especialista no tema, pessoas trabalhando no projeto
    - Identificar possíveis defeitos no produto de trabalho
    - Pode representar diferentes perspectivas
  - Redator (Registrador):
    - Coletar possíveis defeitos encontrados durante a atividade de revisão
    - Registrar novos defeitos potenciais, pontos em aberto e decisões da reunião de revisão
  - Em alguns tipos de revisão uma pessoa pode desempenhar mais de uma função
- Tipos de revisão
  - Podem ter mais de um tipo de revisão por produto
  - Podem ser feitas como revisões em pares, ou seja, feitas por colegas em um nível organizacional aproximado ou semelhante

**Criador: Thiago Andrade**

**Linkedin: <https://www.linkedin.com/in/thiago-andrade-2a97555b/>**

**Email: [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)**

■ Quatro tipos mais comuns de revisões:

- Revisão Informal:
  - Não pode envolver uma reunião de revisão
  - Pode ser realizado por um colega do autor
  - Os resultados podem ser documentados
  - Checklist é opcional
  - Muito usado no desenvolvimento ágil
- Acompanhamento:
  - Preparação individual antes da reunião é opcional
  - A reunião normalmente é feita pelo autor do produto
  - Redator é obrigatório
  - Checklist é opcional
  - Pode variar na pratica de bastante informal para muito formal
- Revisão Técnica:
  - Revisor deve ser par técnicos do autor
  - Necessário preparação individual antes da reunião
  - Reunião de revisão é opcional
  - Redator é obrigatório e preferencialmente que não seja o autor
  - Checklist é opcional
  - São produzidos registros de defeitos potenciais e o relatório de revisão
- Inspeção:
  - Utiliza funções claramente definidas
  - Necessário preparação individual antes da reunião
  - Revisores são pares do autor ou especialista
  - São usados critérios de entradas e saídas
  - Redator é obrigatório
  - Reunião de revisão é liderada por um facilitador treinado
  - O autor não pode atuar como líder de revisão, leitor ou redator
  - São produzidos registros de defeitos potenciais e o relatório de revisão

**Criador: Thiago Andrade**

**Linkedin:** <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

**Email:** [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)

- Revisão pode revisar um único documento ou vários documentos
- Aplicando técnicas de revisão
  - Ad Hoc:
    - Os revisores recebem pouca ou nenhuma orientação sobre como essa tarefa deve ser executada
    - Os revisores geralmente leem o produto de trabalho, identificam e documentam os problemas à medida que o encontram.
    - É uma técnica comumente usada que requer pouca preparação
    - Altamente dependente das habilidades do revisor
    - Pode levar a muitos problemas relatados em duplicidades se forem feitos por revisores diferentes
  - Baseada em Checklists:
    - Principal vantagem é uma técnica sistemática, onde os revisores detectam os problemas com base em checklist que são distribuídos no início da revisão
    - Consiste um conjunto de perguntas baseadas em possíveis defeitos, que podem ser derivados da experiência
  - Baseada em Cenários (teste práticos):
    - Revisores recebem orientação estruturadas sobre como ler o produto de trabalho.
    - Fornecem melhores diretrizes sobre como identificar tipos específicos de defeitos do que simples entradas no checklist
  - Baseada em Perspectiva:
    - Semelhante a baseada em papéis
      - Se estiver falando em olhar o software na visão do cliente.
    - Espera-se serem usados checklist
    - É mais eficaz para revisar os requisitos e os produtos de trabalho técnico.
  - Baseada em papéis:
    - Se estiver falando em olhar o software na visão de perfil do usuário final
    - Avaliam o produto de trabalho na perspectiva dos papéis individuais dos stakeholders
    - Pode ser usado checklist
- Fatores de sucesso para revisão



- Fatores organizacionais de sucesso para revisões:
  - Cada revisão tem objetivos claros, definidos durante o seu planejamento e usados como critérios mensuráveis de saída.
  - Todos os checklists abordam os principais riscos e deverão estar atualizados
  - Documentos grandes são escritos e revisados em pequenos pedaços, para que os feedbacks antecipados
  - Os participantes devem ter tempo suficiente para se preparar
- Fatores de sucesso relacionados as pessoas:
  - Os testadores são vistos como revisores valiosos que contribuem para revisão
  - As revisões são realizadas em pequenos trechos
  - Os defeitos encontrados são reconhecidos, apreciados e manipulados objetivamente
  - O treinamento adequado é fornecido

## Capítulo 04 Técnicas de teste

### Categorias de técnicas de teste

- Quantos teste e quais teste tenho que fazer
- A escolha de quais das técnicas de teste usar depende de vários fatores:
  - Complexidade do componente ou do sistema
  - Normas Regulatórias
  - Requisitos contratuais ou do cliente
  - Níveis e tipo de risco
  - Documentação disponível
  - Conhecimento e habilidades do testador
  - Ferramentas disponíveis
  - Tempo e orçamento
  - Modelo de ciclo de vida do desenvolvimento de software
  - Tipos de defeitos esperados no componente ou sistema.
- Algumas são aplicadas em determinadas situações de nível de teste ou em todos os níveis de teste.
- Categorias de técnicas de teste e suas características

- As técnicas de testes são classificadas como:
  - Caixa preta ou comportamentais ou baseada no comportamento
    - São aplicadas a testes funcionais (Negocio) e não funcionais (Técnica)
    - Se concentram nas entradas e saídas do objeto de teste sem referência a sua estrutura interna
  - Caixa branca ou estruturais ou baseada na estrutura
    - São baseadas em análise da arquitetura, do detalhamento do projeto da estrutura interna ou o código do objeto
    - Concentram-se na estrutura e no processamento dentro do objeto de teste
  - Baseada na experiencia
    - Aproveitam o conhecimento dos testadores, desenvolvedores e usuário para projetar, implementar e executar teste
    - São combinadas com caixa preta e caixa branca.
- Características comuns:
  - Caixa preta:
    - As condições de teste, os casos de teste e os dados são derivados de uma base de teste que pode incluir requisitos de software, especificações, casos de uso e história de usuário.
  - Caixa Branca:
    - As condições de teste, os casos de teste e os dados são derivados de uma base de teste que pode incluir código, arquitetura de software, o detalhamento do projeto ou qualquer outra fonte de informação relacionada a estrutura do software
  - Baseada na experiencia:
    - As condições de teste, os casos de teste e os dados são derivados de uma base de teste que pode incluir conhecimento e a experiencia de testadores, desenvolvedores, usuários e stakeholders

## Teste Caixa preta

- [Particionamento de equivalência](#)

- Testar os valores de cada grupo
- Divide os dados em partições (Classes de equivalência), de tal forma que todos os membros de uma determinada partição deve ser processado da mesma maneira.
- Tipos de partições:

**Criador: Thiago Andrade**

**Linkedin: <https://www.linkedin.com/in/thiago-andrade-2a97555b/>**

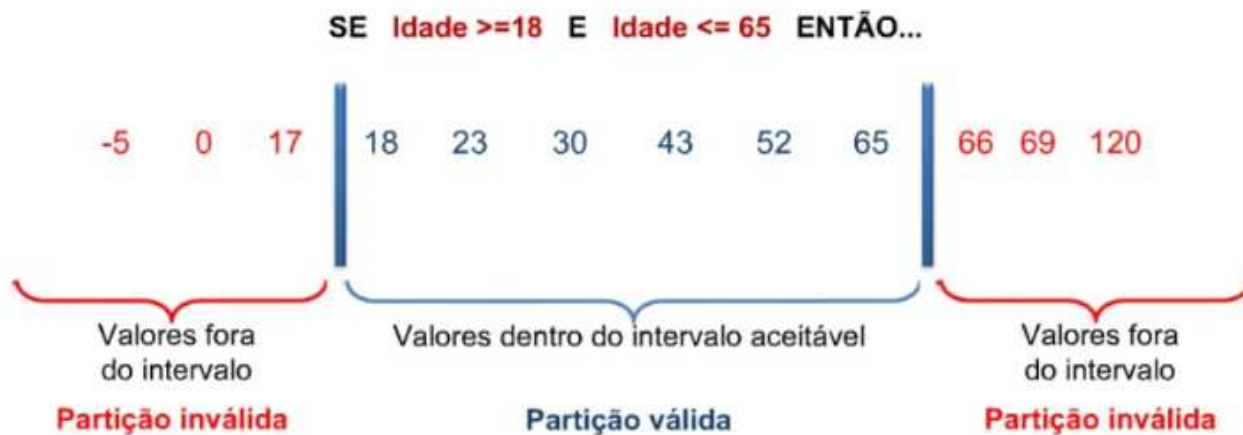
**Email: [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)**

- Validos:
  - Valores aceitos pelo componente do sistema, chamada de partição de equivalência válida
- Inválidos
  - Valores devem ser rejeitados pelo componente do sistema, chamada de partição de equivalência inválida
- Aplicável em todos os níveis de teste.
- Exemplo
  - Exemplo de Particionamento de equivalência
- Exemplo de aplicação das diretrizes:
  - a) Se uma condição de entrada especifica uma **faixa de valores**, uma classe de equivalência válida e duas inválidas são definidas
    - **Entrada:** campo idade deve aceitar valores de **1 a 99** anos.
      - Faixa de valor: 1 a 99
      - Classe válida: números de 1 a 99
      - Classe inválida: números menores que 1
      - Classe inválida: números maiores que 99



- Exemplo de Particionamento de equivalência

## Partição de equivalência



A partição de equivalência pode:

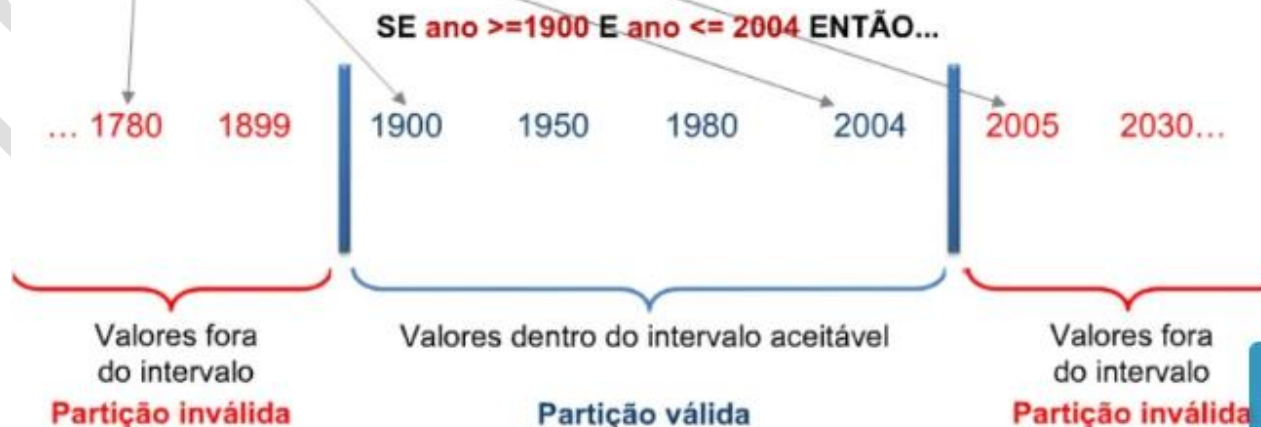
- Ser usada em todos os níveis de teste
- Ser usada para alcançar boa cobertura dos testes
- Ajudar a reduzir o número de testes (um caso de teste para cada partição)

- Exemplo de Particionamento de equivalência

## Partição de equivalência – Resolução do exercício

Um campo de entrada (input field) referente ao ano de aniversário aceita valores de 1900 até 2004. Utilizando a técnica de partição de equivalência, qual opção tem ao menos um valor de cada partição de equivalência?

- a) 1900, 1901, 2000, 2005
- b) 1900, 2004
- X c) 1780, 1900, 2004, 2005
- d) 1899, 1900, 1901, 2003, 2004



- Análise de valor limite

Criador: Thiago Andrade

Linkedin: <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

Email: maciel.thiago@gmail.com

- Valores mínimo e máximo são os valores limites
- Existem 3 partições de equivalência:
  - Invalido (Muito baixo), valido e invalido (Muito alto)
- Pode ser aplicada em todos os níveis de teste
- Exemplo
  - Exemplo Analise de valor limite

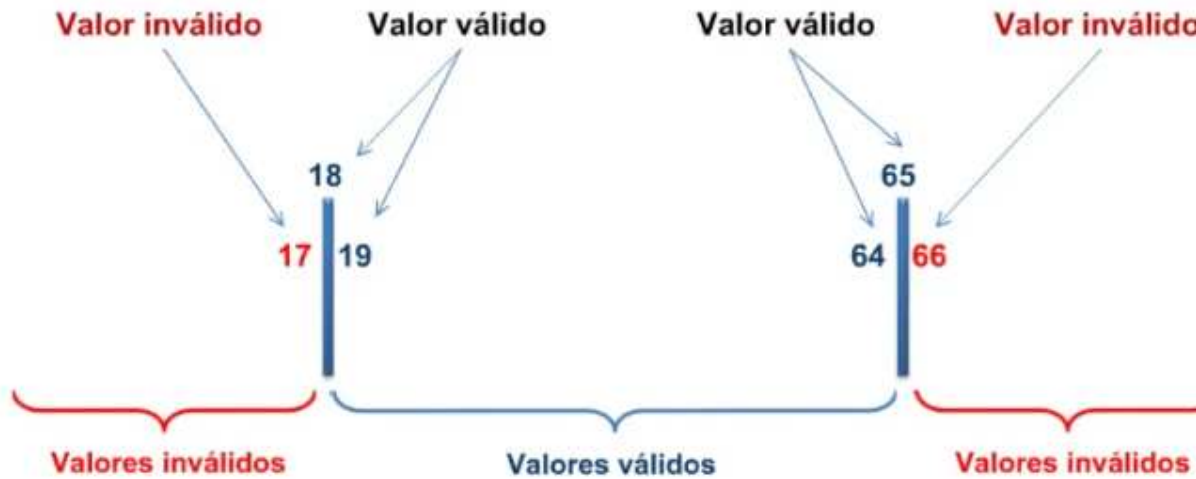


- Exemplo Analise de valor limite

# Exemplo

## Análise de valor limite

SE Idade  $\geq 18$  E Idade  $\leq 65$  ENTÃO...



Utilizando a técnica de análise de valor limite para cada limite sempre teremos 1 limite inválido e 2 válidos. Sendo assim, teremos 3 casos de teste.

- Exemplo Analise de valor limite

Criador: Thiago Andrade

Linkedin: <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

Email: maciel.thiago@gmail.com

# Exemplo

- (CTFL - BSTQB) Um campo de entrada referente a data de nascimento aceita valores de 1860 até 2860. Utilizando a análise de valores limite o teste usaria quais valores.

- a) 0, 1860, 2860, 3000
- b) 1860, 2860
- c) 1859, 1900, 1861, 2859, 2860, 2861
- d) 1859, 1860, 2860, 2861

- Teste de tabela de decisão

- São uteis para testar diferentes condições de combinação levam a resultados diferentes
- Formam linhas da tabela, geralmente com as condições no topo e as ações na parte inferior.
- Tabela completa são todas as condições
  - Resposta elevado as condições (Exemplo ao lado =  $2^3$ )



# Tabela de Decisão

Para que seja definida a **quantidade de regras da tabela**, basta que multipliquemos a quantidade de respostas possíveis de cada condição.

**Exemplo:**

- Condição 1 sim/não: = 2
- Condição 2 sim/não: = 2
- Condição 3 Sim/não: = 2

CONDIÇÕES	Regras 1	Regras 2	Regras 3	Regras 4	Regras 5	Regras 6	Regra 7	Regra 8
Ocupa cargo de chefia?	S	S	S	S	N	N	N	N
Idade maior que 40 anos?	S	S	N	N	S	S	N	N
Mais de 2 anos no cargo?	S	N	S	N	S	N	S	N
AÇÕES								
Exame especial	X	X	X		X	X		
Exame normal				X			X	X

onde: S=sim; N=não; X=ação a ser executada.

– Quantidade de Regras: =  $2 \times 2 \times 2 = 8$

- Tabela reduzida testar cada condição negativamente uma vez e depois o restante teste positivo (Número de ações (Sim ou Não) + 1)

- Exemplo: Regra 2, Regra 3, Regra 5 e Regra 1

• **Exemplo:**

- área de condições;
- área de ações;
- Regras de decisão.

Foi testada negativamente uma vez e o restante positivo

CONDIÇÕES	Regras 1	Regras 2	Regras 3	Regras 4	Regras 5	Regras 6	Regra 7	Regra 8
Ocupa cargo de chefia?	S	S	S	S	N	N	N	N
Idade maior que 40 anos?	S	S	N	N	S	S	N	N
Mais de 2 anos no cargo?	S	N	S	N	S	N	S	N
AÇÕES								
Exame especial	X	X	X		X	X		
Exame normal				X			X	X

onde: S=sim; N=não; X=ação a ser executada.

- Cobertura mínima é ter pelo menos um caso de teste por regra na decisão da tabela

Criador: Thiago Andrade

Linkedin: <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

Email: maciel.thiago@gmail.com



- Cobertura é medida como o número de regras de decisão testadas por pelo menos um caso de teste, dividido pelo número total de regras de decisão
- Teste transição de estado
  - Componentes ou sistemas podem responder de maneira diferente a um evento dependendo das condições atuais ou do histórico anterior, fica conhecido como conceitos de estado.
  - Diagrama de transição de estado mostra os possíveis estados do software, bem como a forma como o software entra e sai e transita entre os estados.
  - Uma transição é iniciada por um evento
  - O evento resulta em transição
  - Imagem de ajuda
    - Transição de estado

## Transição de Estado



- **Estado:** são as condições ou situações durante a vida de um sistema na qual ele satisfaz algumas condições, executa algumas atividades ou espera por eventos.
- **Transição:** é o relacionamento entre dois estados, indicando que o objeto que está no primeiro estado irá passar para o segundo estado mediante a ocorrência de um determinado evento.
- **Evento:** é causa necessária para que haja a transição de estado.
- **Ação:** A ação é basicamente a resposta que o sistema dará ao evento executado.
- **Estado inicial:** estado por onde se começa a leitura de um diagrama de estado.
- **Estado final:** estado que representa o fim.

- Teste caso de uso
  - Os testes podem ser derivados de casos de uso, que são uma maneira específica de projetar interações com itens de software, incorporando requisitos para as funções de software representadas pelos casos de uso.
  - Casos de uso estão associados a atores (Usuário humanos ou componentes do sistema ou sistemas) e assuntos (Componente ou sistemas ao qual o caso de uso é aplicado)

Criador: Thiago Andrade

Linkedin: <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

Email: maciel.thiago@gmail.com

- Podem ser representadas graficamente por fluxos de trabalho, diagramas ou modelos de processos.

## Teste Caixa Branca

- Baseado na estrutura interna do objeto de teste
- Pode ser usado em todos os níveis de teste
- Passam muito pelo conceito de cobertura (Todas as técnicas de caixa branca começam com o nome “Cobertura”)
- Teste e cobertura de instrução (Cobertura de sentença, declaração e instrução)
  - Testa todas as instruções do código
  - Sempre é menor ou igual ao teste de cobertura de decisão
  - Quando tiver 2 ou mais conjuntos, o maior prevalece
- Teste e cobertura de decisão (Desvio)
  - Testa as decisões existentes no código e o código executado com base nos resultados da decisão
- O valor da instrução e teste de decisão
  - Atingir 100% de cobertura de decisão garante 100% de cobertura de instrução (mas não vice versa)
  - Cobertura de Instrução (Cobertura de sentença, declaração e comando)
  - Cobertura de decisão (Cobertura de desvio)
- Resumo teste de cobertura e decisão
  - Ramo de Tópico 1

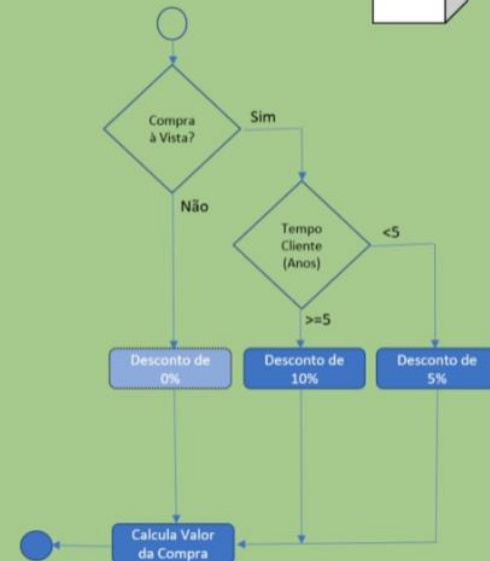
## Técnicas de Teste de Caixa-Branca - Teste de Instruções e Cobertura



Objetivo: passar pelo menos uma vez em cada instrução existente no código para atingir cobertura de 100%.

```
//Tempo em anos
If (compra_a_vista==true)
{
    If (tempo_cliente>=5)
    {
        Desconto=10; // Desconto em percentual
    } else
    {
        Desconto=5;
    }
}

valor_compra = valor_compra * (1-(desconto/100));
```



Você precisará de 2 testes para cobrir todas as instruções. 1. Compra à vista = Verdadeiro e Cliente com mais de 5 anos. 2. Compra à Vista = Verdadeiro e Cliente com Menos de 5 Anos. (100% de cobertura das 5 instruções existentes)

- Cobertura de instrução (CI)  $\leq$  Cobertura de Decisão (CD)
- Complexidade Ciclo matica de MaCabe = Cobertura de decisão

### Teste Baseado em experiencia

- Dependendo da abordagem do testador e experiencia, essas técnicas podem alcançar graus de cobertura e eficácia amplamente variados
- Suposição de erro:
  - É uma técnica usada para prever a ocorrência de erros, defeitos e falhas, com base no conhecimento:
    - Como o aplicativo funcionou no passado
    - Que tipos de erro tendem a ser cometidos
    - Falhas ocorridas em outros aplicativos
  - Uma abordagem metódica para a técnica de suposição de erro é criar uma lista de possíveis erros, defeitos e falhas e modelar os testes que exporão falhas e os defeitos que as causaram
- Teste Exploratório:
  - Testador experiente, planejamento leve que produza carta de teste e fazer o teste em um limite de tempo
  - Os resultados são usados para aprender mais sobre o componente ou sistema e para criar teste para as áreas que podem precisar de mais teste

Criador: Thiago Andrade

Linkedin: <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

Email: [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)

- Teste baseado em sessão, o teste exploratório é conduzido dentro de uma janela de tempo definida, e o testador usa um termo contendo objetivos de teste para orientar o teste.
- É mais útil quando há poucas ou inadequadas especificações ou pressão de tempo significativo nos testes
- Falou em Cartas de teste, falou em teste exploratório
- Teste baseado em checklist:
  - Os testadores modelam, implementam e executam testes para cobrir as condições de teste encontradas em uma lista.
  - Testadores criam uma lista ou expandem uma existente, mas os testadores também podem usar um checklist existente sem modifica-lo
  - Checklist podem ser construídos com base na experiência, conhecimento sobre o que é importante para o usuário ou uma compreensão de por que e como o software falha

## Capítulo 05 Gerenciamento de teste

### Teste independente

- As tarefas de teste podem ser executadas por pessoas em um papel específico ou por pessoas em outros papéis
- Um certo grau de independência geralmente torna o testador mais eficaz em encontrar os defeitos devido as diferenças entre os vieses cognitivos do autor e do testador
- Graus de independência no teste incluem:
  - Sem testadores independentes, a única forma de teste disponível são os desenvolvedores testando seu próprio código.
  - Desenvolvedores independentes ou testadores dentro das equipes de desenvolvimento ou da equipe do projeto, isso poderiam ser desenvolvedores testando os produtos de seus colegas.
  - Equipe de teste independente ou grupo dentro da organização, reportando-se ao gerenciamento de projetos ou ao gerenciamento executivo
  - Testadores independentes da organização empresarial ou da comunidade de usuários, ou com especialização em tipos de testes específicos, como usabilidade, segurança, performance, regulamentação, conformidade ou portabilidade
  - Testadores independentes externos a organização, trabalhando dentro ou fora do escritório
- Benefícios potenciais da independência:
  - Os testadores independentes provavelmente reconhecerão diferentes tipos de falhas em comparação a os desenvolvedores, devido a diferentes históricos, perspectivas técnicas e vieses.

**Criador: Thiago Andrade**

**Linkedin:** <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

**Email:** [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)

- Um testador independente pode verificar, desafiar ou refutar as suposições feitas pelos stakeholders durante a especificação e a implementação do sistema.
- As desvantagens potenciais da independência:
  - Isolamento da equipe de desenvolvimento
  - Os desenvolvedores podem perder o senso de responsabilidade pela qualidade
  - Testadores independentes podem ser vistos como gargalo ou culpados por atrasos na liberação
  - Testadores independentes podem não ter informações importantes
- Tarefas de um gerente de teste e do testador
  - Gerente de teste:
    - Processo de teste e da liderança das atividades bem-sucedidas de teste
    - Pode ser executado por um gerente de teste profissional ou por um gerente de projeto, um gerente de desenvolvimento ou um gerente de qualidade.
    - Em projetos ou organizações maiores, várias equipes de teste podem se reportar a um gerente de teste, um técnico de teste ou um coordenador de teste, cada equipe sendo liderada por um líder de teste ou um testador líder.
    - Tarefas típicas de gerente de teste:
      - Desenvolver ou revisar uma política de teste e uma estratégia de teste para a organização
      - Planejar as atividades de teste
      - Escrever e atualizar os planos de teste
      - Coordenar os planos de teste com gerente de projetos
      - Compartilhar as perspectivas de teste com outras atividades do projeto
      - Iniciar a análise, o projeto, a implementação e a execução dos testes, monitorar o progresso e os resultados obtidos e verificar o status dos critérios de saída
      - Preparar e entregar os relatórios de progresso do teste e resumo com base nas informações coletadas
      - Decidir sobre a implementação dos ambientes de teste
  - Tarefas típicas de testador:
    - Revisar e contribuir para os planos de teste
    - Analisar, revisar e avaliar os requisitos, as histórias de usuários e os critérios de aceite, as especificações e modelos para testabilidades.

**Criador: Thiago Andrade**

**Linkedin:** <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

**Email:** maciel.thiago@gmail.com

- Identificar e documentar as condições de teste e capturar a rastreabilidade entre os casos de teste, as condições de teste e a base de teste
- Projetar, configurar e verificar os ambientes de teste, geralmente coordenando com a administração do sistema e gerenciamento da rede
- Projetar e implementar os casos de teste e procedimento de teste
- Preparar e adquirir os dados de teste
- Criar o cronograma detalhado de execução do teste
- Executar os testes, avaliar os resultados e documentar os desvios de resultados esperados.
- Usar ferramentas apropriadas para facilitar o processo de teste
- Automatizar os testes conforme necessários

### Planejamento e estimativa de testes

- Planejamento é influenciado pela política de teste e a estratégia de teste da organização, pelos ciclos de vida e métodos, pelo escopo dos testes, objetivos, riscos, restrições, criticidade, testabilidade e disponibilidade de recursos
- O planejamento do teste é uma atividade contínua e é executado durante todo o ciclo de vida do produto.
- Planejamento pode ser documentado em um plano de teste principal e em planos de teste separados para cada nível de teste
- As atividades do planejamento do teste podem ser documentados em um plano de teste e incluir:
  - Determinar o escopo, os objetivos e os riscos do teste
  - Definir a abordagem do teste
  - Integrar e coordenar as atividades de teste nas atividades do ciclo de vida do software
  - Tomar decisões sobre o que testar, as pessoas e outros recursos necessários para realizar as várias atividades de teste e como essas atividades serão realizadas
  - Programar as atividades de análise, projeto, implementação, execução e avaliação de teste, em datas específicas
  - Selecionar as métricas para monitoramento e controle de teste
  - Orçar as atividades de teste
  - Determinar o nível de detalhes e a estrutura da documentação de teste
- Tipos comuns de estratégias de teste:
  - Analítica

**Criador: Thiago Andrade**

**Linkedin:** <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

**Email:** [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)

- Teste é baseado em uma análise de algum fator
- Baseada em modelo
  - Teste são projetados com base em algum modelo de algum aspecto necessário do produto, como uma função, um processo empresarial, uma estrutura interna ou uma característica não funcional
- Metódica
  - Depende do uso sistemático de um conjunto predefinido de testes ou condições de teste
- Compatível com processo
  - Envolve análise, projeto e implementação do teste baseado em regras e padrões externos
- Dirigida
  - Orientado principalmente pelo aconselhamento, orientação ou instrução dos stakeholders, especialista no domínio do negócio ou especialista em tecnologia.
- Contra regressão
  - Motivado pelo desejo de evitar a regressão de recursos existentes. Essa estratégia inclui a reutilização do testware existente, da automação extensiva de regressão e de conjunto de teste padrão
- Reativa
  - Teste é reativo ao componente ou sistemas que está sendo testado e aos eventos que ocorrem durante a execução do teste, em vez de serem pré-planejados. Teste exploratório é uma técnica comum empregada em estratégias salvas.
- A adaptação da estratégia baseia-se em decisões tomadas em relação a complexidade e objetivos do projeto, do tipo de produto que está sendo desenvolvido e da análise de risco do produto.
- Critérios de entradas e saídas:
  - Critérios de entrada e saída devem ser definidos para cada nível e tipo de teste e serão diferentes com base nos objetivos do teste
- Critérios de entrada:
  - Disponibilidade dos requisitos testáveis, histórias de usuários e/ou modelos
  - Disponibilidade dos itens de teste que atendam aos critérios de saída para quaisquer níveis de teste anteriores
  - Disponibilidade do ambiente de teste
  - Disponibilidade de ferramentas de teste necessárias

**Criador: Thiago Andrade**

**Linkedin:** <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

**Email:** [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)



- Disponibilidade de dados de teste e outros recursos necessários.
- Critérios de saída:
  - Que os teste planejados forame executados
  - Foi alcançado um nível definido de cobertura
  - O número de defeitos não resolvidos está dentro de um limite acordado
  - O número de defeitos remanescente estimados é suficientemente baixo
- Cronograma de execução do teste
  - Depois que vários casos de teste e procedimentos de teste são produzidos e montados em suítes de teste, os conjuntos de teste podem ser organizados em um cronograma da execução do teste que define a ordem em que devem ser executados
- Fatores que influenciam o esforço do teste
  - Os fatores que influenciam o esforço do teste podem incluir características:
    - Característica do produto:
      - Riscos associados com o produto
      - Qualidade da base de teste
      - Tamanho do produto
      - Complexidade do domínio do produto
      - Requisitos das características de qualidade
      - Nível necessário de detalhes para documentação de teste
      - Requisitos para conformidade legal e regulatória
    - Característica do processo de desenvolvimento:
      - Estabilidade e maturidade da organização
      - O modelo de desenvolvimento em uso
      - A abordagem de teste
      - A ferramenta usada
      - O processo de teste
      - A pressão sobre o tempo de finalização
    - Característica e pessoas:
      - As habilidades e a experiência das pessoas envolvidas, especialmente com projetos e produtos semelhantes

**Criador: Thiago Andrade**

**Linkedin:** <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

**Email:** [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)



- Coesão e liderança de equipe
- Resultados do teste:
  - O número e a gravidade dos efeitos encontrados
  - A quantidade de retrabalho necessário.
- Técnicas de estimativa de teste:
  - Técnica baseada em métricas:
    - Com base nas métricas de projetos anteriores, ou com base em valores típicos
  - Técnica baseada em especialista:
    - Com base na experiências dois responsáveis pelas tarefas de teste ou por especialistas.

## Monitoramento e controle de teste

- Monitoramento (Acompanhar) X Controle (Agir)
- O objetivo é coletar informações e fornecer feedbacks e visibilidades sobre as atividades de teste
- As informações podem ser coletadas manualmente ou automaticamente
- Métricas usadas no teste
  - As métricas podem ser coletadas durante o final das atividades de teste para avaliar:
    - Relação entre o planejado e o orçado em cronograma
    - Qualidade atual do objeto de teste
    - Adequação da abordagem de teste
    - Eficácia das atividades de teste em relação aos objetivos.
  - As métricas de teste comum incluem:
    - Cobertura de teste de requisitos, de histórias de usuários, de critérios de aceite, de risco ou de código
    - Conclusão de tarefas, alocação e uso de recursos e esforços
    - Custo do teste, incluindo o custo comparada ao benefício de encontrar o próximo defeitos ou o custo comparado ao benefício de executar o próximo teste.
- Finalidades, conteúdo e público-alvo para os relatórios de teste
  - O objetivo do relatório de teste é resumir e comunicar informações de atividades de teste, durante e no final de uma atividade de teste

- Relatórios de progresso de teste típicos podem incluir:
  - O status das atividades de teste e o progresso em relação ao plano de teste
  - Fatores impedindo o avanço
  - O teste planejado para o próximo período do relatório
  - A qualidade do objeto de teste
- Quando os critérios de saída são atingidos, o gerente de teste faz o relatório de resumo do teste
- Relatórios típicos de progresso de teste e relatórios:
  - Resumo dos testes realizados
  - Informação sobre o que ocorreu durante um período de teste
  - Desvios do plano, incluindo desvios no cronograma, duração ou esforço das atividades de teste
  - Status do teste e da qualidade do produto com relação aos critérios de saída ou definição de completo
  - Fatores que bloquearam ou continuam bloqueando o progresso
  - Métricas de defeitos, caso de teste, cobertura de teste, progresso da atividade e consumo de recursos
  - Riscos residuais
  - Produtos de teste reutilizáveis produzido

## Gerenciamento de configurações

- Tem como objetivo é estabelecer e manter a integridade do componente ou do sistema, o testware e seus relacionamentos entre si durante o ciclo de vida do projeto e do produto
- Gerenciamento de configurações pode envolver o seguinte:
  - Todos os itens de teste são identificados de forma exclusiva, controlados por versão, rastreados para alterações e relacionados entre si
  - Todos os itens de testware são identificados de forma exclusiva, controlados por versão, rastreados para alterações, relacionados uns aos outros e relacionados as versões dos itens de teste, de forma que a rastreabilidade possa ser mantida durante todo o processo de teste

## Riscos e testes

- Definição de risco
  - $R = P \times I$  (Risco = Probabilidade X Impacto)

- O nível de risco é determinado pela probabilidade do evento e pelo impacto desse evento.
- Riscos de produto e projeto
  - O risco do produto envolve a possibilidade de que um produto de trabalho possa falhar em satisfazer as necessidades legítimas de seus usuários ou stakeholders
  - Riscos do produto incluem:
    - O software pode não executar as funções de acordo com a sua especificação
    - O software pode não executar as funções pretendidas de acordo com as necessidades do usuário, do cliente ou dos stakeholders.
    - Uma arquitetura de sistema pode não suportar adequadamente alguns requisitos não funcionais
    - Um cálculo específico pode ser executado incorretamente em alguma circunstância.
    - Uma estrutura de controle de loop pode ser codificada incorretamente
    - Os tempos de resposta podem ser inadequados para um sistema de processamento de transações de alta performance
    - O feedback da experiência do usuário pode não atender as expectativas do produto
  - Questões organizacionais:
    - Insuficiência de equipe, habilidades e treinamentos
  - Questões políticas:
    - Os testadores podem não comunicar adequadamente suas necessidades ou os resultados do teste
  - Questões técnicas:
    - Os requisitos podem não estar bem definidos
    - Os requisitos podem não ser cumpridos por restrições existentes
    - O ambiente de teste pode não estar pronto no prazo
    - A conversão de dados, o planejamento de migração e o suporte de ferramentas podem atrasar
  - Questões de fornecedores:
    - Questões contratuais podem causar problemas no projeto
  - Os riscos do projeto podem afetar tanto as atividades de desenvolvimento quanto as atividades de teste.

- Os gerentes de projeto são responsáveis por lidar com todos os riscos do projeto, mas não é incomum que os gerentes de teste tenham responsabilidades pelos riscos do projeto relacionados ao teste.
- Teste baseado em risco e qualidade do produto
  - O risco é usado para concentrar o esforço necessário durante o teste
  - Ele é usado para decidir onde e quando começar a testar e identificar áreas que precisam de mais atenção
  - Envolve a análise de risco, que inclui a identificação e a avaliação da probabilidade e impacto de cada risco.
  - Em uma abordagem de teste baseada em risco, os resultados da análise de risco do produto são usados para:
    - Determinar as técnicas de teste a serem empregadas
    - Determinar os níveis e tipos específicos de teste a serem realizados.
    - Determinar a extensão do teste a ser realizado
    - Priorizar o teste na tentativa de encontrar antecipadamente os defeitos críticos
    - Determinar se quaisquer atividades além do teste poderiam ser empregadas para reduzir o risco.
  - As atividades de gerenciamento de risco fornecem uma abordagem disciplinadas para:
    - Analisar (e reavaliar regularmente) o que pode dar errado
    - Determinar quais os riscos são importantes para lidar
    - Implementar ações para atenuar esses riscos
    - Planejar a contingência para lidar com os riscos, caso eles se tornem eventos reais
  - Os testes podem identificar novos riscos, ajudar a determinar quais os riscos devem ser atenuados e reduzir a incerteza sobre os riscos.

## Gerenciamento de defeitos

- A maneira como os defeitos são registrados pode variar, dependendo do contexto do componente ou sistema que está sendo testado, do nível de teste e do modelo de ciclo de vida de desenvolvimento de software
- Para gerenciar todos os defeitos, uma organização deve estabelecer um processo de gerenciamento de defeitos que inclua um fluxo de trabalho e regras sua classificação.
- Os defeitos podem ser relatados durante a codificação, análise estática, revisões, testes dinâmicos ou uso de um produto de software.
- Os defeitos podem ser relatados em qualquer tipo de documentação

- Relatório de defeitos tem os seguintes objetivos:
  - Fornece aos desenvolvedores e a outros, informações sobre qualquer evento adverso ocorrido, para que possam identificar efeitos específicos, isolar o problema com um teste mínimo de reprodução e corrigir o defeito potencial, conforme necessário ou resolver o problema
  - Fornecer aos gerentes de teste um meio de rastrear a qualidade do produto de trabalho e o impacto do teste
  - Fornecer ideias para desenvolvimento e melhoria de processo de teste
- Relatório de defeitos arquivados durante o teste dinâmico inclui:
  - Um identificador
  - Um título e um breve resumo do defeito relatado
  - A data do relatório de defeitos, a organização emissora e autor
  - A identificação do item de teste e ambiente
  - As fases do ciclo de vida de desenvolvimento em que o defeito foi observado
  - Uma descrição do defeito para permitir a reprodução e resolução, incluindo logs, capturas de tela dump de banco de dados ou gravações
  - Resultados esperados e reais
  - Escopo ou grau de impacto do defeito sobre os interesses dos stakeholders
  - Urgência/prioridade para corrigir
  - O estado do relatório de defeitos
  - Conclusões, recomendação e aprovações
  - Questões globais, como outras áreas que podem ser afetadas por uma alteração resultante do defeito
  - Histórico de alterações, como a sequência de ações tomadas pelos membros da equipe do projeto com relação ao defeito para isolar, reparar e confirmar como corrigido
  - Referencias, incluindo o caso de teste que revelou o problema
- Os defeitos encontrados durante os testes estáticos, particularmente revisões, normalmente serão documentados de uma maneira diferente.

## Capítulo 06 Ferramenta de suporte ao teste

### Considerações sobre ferramentas de teste

- As ferramentas de teste podem ser usadas para suportar uma ou mais atividades de teste.

- Ferramentas que ajudam a gerenciar os requisitos, os casos de teste, os procedimentos de teste, os scripts de teste automatizados, os resultados de testes, os dados de teste e defeitos, e para os relatórios e monitoramento da execução do teste
- Ferramentas são usados para investigação e avaliação
- Classificação das ferramentas de teste
  - [ D ] Ferramentas de Desenvolvedor
  - As ferramentas de teste podem ter um ou mais dos seguintes propósitos, dependendo do contexto:
    - Melhorar a eficiência das atividades de teste automatizando tarefas repetitivas ou tarefas que exigem recursos significativos quando feitas manualmente.
    - Melhorar a eficiência das atividades de teste, apoiando as atividades de teste manuais durante todo o processo.
    - Melhorar a qualidade das atividades de teste, permitindo testes mais consistentes e um nível mais alto de reprodutibilidade de defeitos
    - Automatizar as atividades que podem ser executadas manualmente
    - Aumentar a confiabilidade dos testes
  - As ferramentas podem ser classificadas com base em vários critérios, como finalidade, preço, modelo de licenciamento e a tecnologia utilizada.
  - Algumas ferramentas oferecem suporte que é tipicamente mais apropriado para desenvolvedores
  - Ferramentas de suporte para gerenciamento de teste e testware
    - Ferramentas de gerenciamento podem ser aplicadas em qualquer atividade de teste durante todo o ciclo de vida de desenvolvimento de software:
      - Ferramentas de gerenciamento de teste e ferramentas de gerenciamento do ciclo de vida de aplicativo
      - Ferramentas de gerenciamento de requisitos
      - Ferramentas de gerenciamento de defeitos
      - Ferramentas de gerenciamento de configuração
      - Ferramentas de integração contínua [D]
  - Ferramentas de suporte ao teste estático
    - Ferramentas de análise estática [D]
  - Ferramentas de suporte para modelagem e implementação dos testes
    - Ferramentas de projetos de teste auxiliam na criação de produtos de trabalho sustentáveis no projeto e implementação dos testes:

**Criador: Thiago Andrade**

**Linkedin:** <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

**Email:** maciel.thiago@gmail.com

- Ferramentas de teste baseadas em modelo
  - Ferramentas de preparação de dados de teste
- Ferramentas de suporte para execução e registro de teste
  - Existem ferramentas para apoiar e aprimorar as atividades de execução e registro de teste:
    - Ferramentas de execução do teste
    - Ferramentas de cobertura
    - Ferramentas de teste [D]
- Ferramentas de suporte para medição de performance e análise dinâmica
  - As ferramentas de medição de performance e análise dinâmica são essenciais para dar suporte as atividades de performance e de teste de carga, pois essas atividades não podem ser feitas manualmente com eficiência:
    - Ferramentas de teste de performance
    - Ferramentas de análise dinâmica [D]
- Benefícios e riscos da automação de teste
  - Existem benefícios e oportunidades potenciais com o uso de ferramentas nos testes, mas também há riscos
  - Isso é particularmente verdadeiro para as ferramentas de execução do teste (que geralmente são chamados de automação de teste)
  - Benefícios potenciais:
    - Redução no trabalho manual repetitivo, economizando tempo
    - Maior consistência e repetibilidade
    - Avaliação mais objetiva
    - Acesso mais fácil a informação sobre teste
  - Riscos potenciais:
    - O tempo, custo e o esforço para a implantação de uma ferramenta podem ser subestimados
- Considerações especiais para execução de testes e ferramentas de gerenciamento de testes
  - Ferramentas de execução de teste executam objetos de teste usando scripts de teste automatizados
  - Esses tipos de ferramentas geralmente reque um esforço significativo para obter benefícios:

- Abordagem captura de teste:
  - Capturar testes gravando as ações de um testador manual parece atraente, mas essa abordagem não pode ser dimensionada para muitos scripts de teste.
- Abordagem de teste orientada por dados:
  - Essa abordagem separa as entradas de teste e os resultados esperados, geralmente em uma planilha, e usa um script de teste mais genérico que pode ler os dados de entrada e executar o mesmo script de teste com dados diferentes
- Abordagem de teste orientada por palavra-chave:
  - Essa abordagem de teste, um script genérico processa palavra-chave que descrevem as ações a serem executadas, que depois chama scripts de palavra-chave para processar os dados de teste associados.
- As abordagens acima exigem que alguém tenha experiência na linguagem de script.
- Independentemente da técnica de script usada, os resultados esperados para cada teste precisam ser comparados aos resultados reais do teste, dinamicamente (Enquanto o teste está em execução) ou armazenados para comparação posterior
- As ferramentas de teste baseado em modelo (MBT) permitem que uma especificação funcional seja capturada na forma de um modelo, como diagrama de atividades
- As ferramentas de gerenciamento de teste geralmente precisam interagir com outras ferramentas ou planilhas por vários motivos:
  - Para produzir informações úteis em formato que atenda às necessidades da organização
  - Manter rastreabilidade consistente para os requisitos em uma ferramenta de gerenciamento de requisitos
  - Para vincular informação da versão do objeto de teste na ferramenta de gerenciamento de configuração

## Uso eficaz de ferramentas

- As principais considerações para escolha de uma ferramenta ideal para uma organização incluem:
  - Identificação de oportunidade para um processo de teste melhorado suportado por ferramentas
  - Compreensão das tecnologias usadas pelos objetos de teste, a fim de selecionar uma ferramenta que seja compatível com essa tecnologia
  - Avaliação do fornecedor ou suporte para ferramentas não comerciais
  - Identificação de requisitos internos para treinamento e mentoring no uso das ferramentas.

**Criador: Thiago Andrade**

**Linkedin: <https://www.linkedin.com/in/thiago-andrade-2a97555b/>**

**Email: [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)**



- Avaliação das necessidades de treinamento, considerando as habilidades de teste daqueles que trabalharão diretamente com as ferramentas.
- Projeto piloto para introduzir uma ferramenta em uma organização
  - A introdução da ferramenta selecionada em uma organização geralmente começa com um projeto piloto, que tem os seguintes objetivos:
    - Obter conhecimento aprofundado sobre a ferramenta, entendendo seus pontos fortes e fracos
    - Avaliar como a ferramenta se ajusta aos processos e práticas existentes e determinar o que precisar mudar
    - Decidir a forma padrão de usar, gerenciar, armazenar e manter a ferramenta e os recursos de teste
    - Avaliar se os benefícios serão alcançados a um custo razoável
- Fatores de sucesso para ferramentas
  - Fatores de sucesso para avaliação, implementação, implantação e suporte contínuo de ferramentas dentro de uma organização incluem:
    - Estender o uso da ferramenta para o resto da organização de forma incremental
    - Adaptar e melhorar os processos para se adequar ao uso da ferramenta
    - Fornecer treinamento, capacitação e orientação para os usuários das ferramentas
    - Definir as diretrizes para o uso da ferramenta
    - Implementar uma forma de coletar informações de uso real das ferramentas
    - Monitorar o uso e os benefícios de ferramentas
    - Fornecer suporte aos usuários de uma determinada ferramenta
    - Reunir lições aprendidas de todos os usuários
  - Também é importante garantir que a ferramenta seja técnica e organizacionalmente integrada ao ciclo de vida de desenvolvimento de software.

## Tipos de teste

### Teste Componente

- Teste em um único componente do sistema. Ex: Teste de unidade Ramo de Tópico

### Teste de Integração

- Teste de 2 ou mais componentes do sistema (Integração pequena) e Integração de sistemas (Integração grande). Ex: Teste de API Ramo de Tópico

**Criador: Thiago Andrade**

**Linkedin:** <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

**Email:** maciel.thiago@gmail.com

## Teste de Sistema

- Teste do software como um todo. Ex: Teste de sistema camada de usuário, mobile, desktop...Tópico

## Teste de Aceito ou Homologação

- Teste na visão do usuário ou cliente. Ex: Teste Usuário ou cliente que homologa o software.

## Teste de confirmação

- Teste pontual naquele item ou funcionalidade com defeito

## Teste de Regressão

- Teste geral no sistema, normalmente em uma nova implementação ou melhoria.

## Teste Exploratório

- Planejamento (Montar uma carta de teste)
- Time box: Tem um limite de tempo para ser executado (Normalmente 30 a 120 min)
- Testador experiente

## Teste ADhoc

- Não tem planejamento (Testa aí)
- Não existe time box
- Qualquer testador pode testar

## Teste alfa:

- Tem a infraestrutura do fornecedor (Software e hardware)
- Teste mais de logica do programa
- Ambiente controlado

## Teste Beta:

- Software do fornecedor, mas o hardware é do cliente ou usuário.
- Pode ocorrer após o teste alfa ou sem que este seja realizado
- Problemas de incompatibilidade ou portabilidade.

## Teste Caixa preta

- Particionamento de equivalência
  - Testar os valores de cada grupo

**Criador: Thiago Andrade**

**Linkedin:** <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

**Email:** maciel.thiago@gmail.com

- Divide os dados em partições (Classes de equivalência), de tal forma que todos os membros de uma determinada partição deve ser processado da mesma maneira.
- Tipos de partições:
  - Validos:
    - Valores aceitos pelo componente do sistema, chamada de partição de equivalência valida
  - Inválidos
    - Valores devem ser rejeitados pelo componente do sistema, chamada de partição de equivalência invalida
- Aplicável em todos os níveis de teste.
- Exemplo
  - Exemplo de Particionamento de equivalência

## • Exemplo de aplicação das diretrizes:

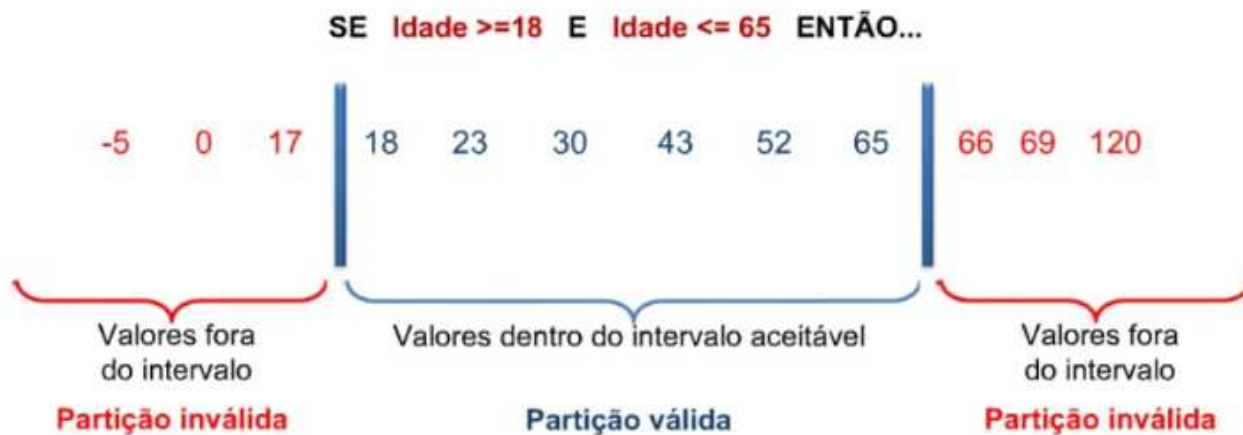
a) Se uma condição de entrada especifica uma **faixa de valores**, uma classe de equivalência válida e duas inválidas são definidas

- **Entrada:** campo idade deve aceitar valores de **1 a 99** anos.
  - Faixa de valor: 1 a 99
  - Classe válida: números de 1 a 99
  - Classe inválida: números menores que 1
  - Classe inválida: números maiores que 99



- Exemplo de Particionamento de equivalência

## Partição de equivalência



A partição de equivalência pode:

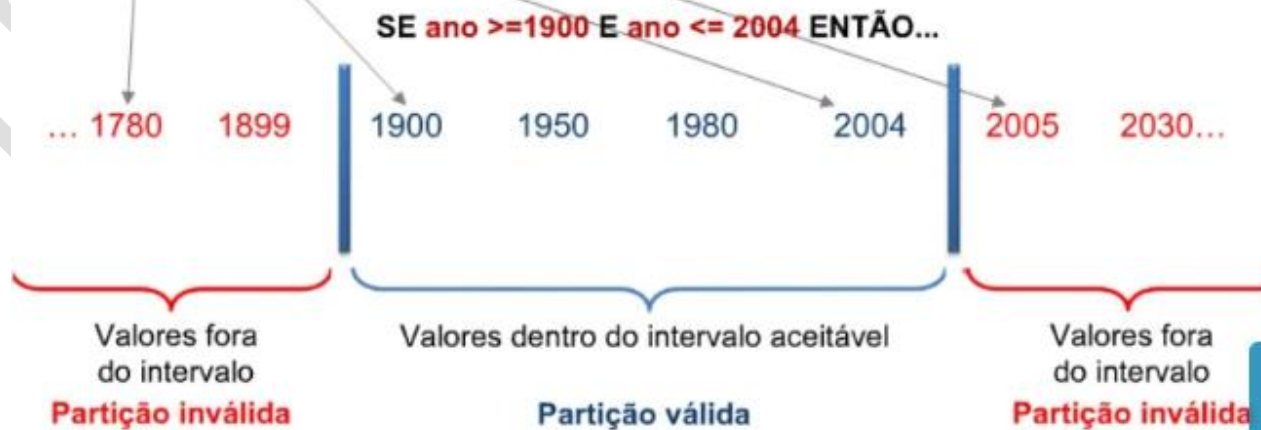
- Ser usada em todos os níveis de teste
- Ser usada para alcançar boa cobertura dos testes
- Ajudar a reduzir o número de testes (um caso de teste para cada partição)

- Exemplo de Particionamento de equivalência

## Partição de equivalência – Resolução do exercício

Um campo de entrada (input field) referente ao ano de aniversário aceita valores de 1900 até 2004. Utilizando a técnica de partição de equivalência, qual opção tem ao menos um valor de cada partição de equivalência?

- a) 1900, 1901, 2000, 2005
- b) 1900, 2004
- X c) 1780, 1900, 2004, 2005
- d) 1899, 1900, 1901, 2003, 2004



- Análise de valor limite

Criador: Thiago Andrade

LinkedIn: <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

Email: maciel.thiago@gmail.com

- Valores mínimo e máximo são os valores limites
- Existem 3 partições de equivalência:
  - Invalido (Muito baixo), valido e invalido (Muito alto)
- Pode ser aplicada em todos os níveis de teste
- Exemplo
  - Exemplo Analise de valor limite

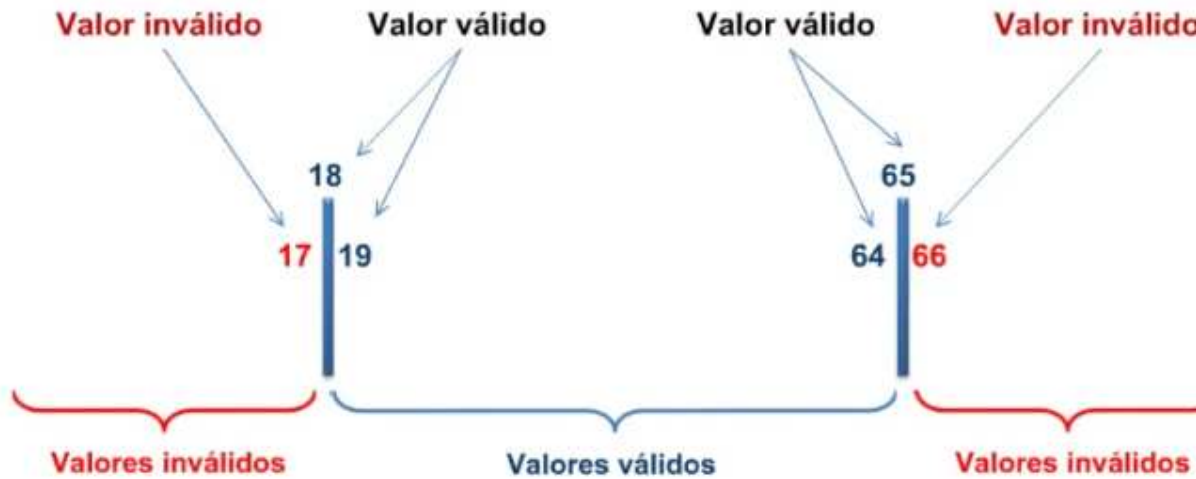


- Exemplo Analise de valor limite

# Exemplo

## Análise de valor limite

SE Idade  $\geq 18$  E Idade  $\leq 65$  ENTÃO...



Utilizando a técnica de análise de valor limite para cada limite sempre teremos 1 limite inválido e 2 válidos. Sendo assim, teremos 3 casos de teste.

- Exemplo Analise de valor limite

Criador: Thiago Andrade

Linkedin: <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

Email: maciel.thiago@gmail.com

# Exemplo

- (CTFL - BSTQB) Um campo de entrada referente a data de nascimento aceita valores de 1860 até 2860. Utilizando a análise de valores limite o teste usaria quais valores.

- a) 0, 1860, 2860, 3000
- b) 1860, 2860
- c) 1859, 1900, 1861, 2859, 2860, 2861
- d) 1859, 1860, 2860, 2861

- Teste de tabela de decisão

- São uteis para testar diferentes condições de combinação levam a resultados diferentes
- Formam linhas da tabela, geralmente com as condições no topo e as ações na parte inferior.
- Tabela completa são todas as condições
  - Resposta elevado as condições (Exemplo ao lado =  $2^3$ )



# Tabela de Decisão

Para que seja definida a **quantidade de regras da tabela**, basta que multipliquemos a quantidade de respostas possíveis de cada condição.

Exemplo:

- Condição 1 sim/não: = 2
- Condição 2 sim/não: = 2
- Condição 3 Sim/não: = 2

CONDIÇÕES	Regras 1	Regras 2	Regras 3	Regras 4	Regras 5	Regras 6	Regra 7	Regra 8
Ocupa cargo de chefia ?	S	S	S	S	N	N	N	N
Idade maior que 40 anos ?	S	S	N	N	S	S	N	N
Mais de 2 anos no cargo ?	S	N	S	N	S	N	S	N
AÇÕES								
Exame especial	X	X	X		X	X		
Exame normal				X			X	X

onde: S=sim; N=não; X=ação a ser executada.

– Quantidade de Regras: =  $2 \times 2 \times 2 = 8$

- Tabela reduzida testar cada condição negativamente uma vez e depois o restante teste positivo (Número de ações (Sim ou Não) + 1)

- Exemplo: Regra 2, Regra 3 e Regra 5

• Exemplo:

- área de condições;
- área de ações;
- Regras de decisão.

Foi testada negativamente uma vez e o restante positivo

CONDIÇÕES	Regras 1	Regras 2	Regras 3	Regras 4	Regras 5	Regras 6	Regra 7	Regra 8
Ocupa cargo de chefia ?	S	S	S	S	N	N	N	N
Idade maior que 40 anos ?	S	S	N	N	S	S	N	N
Mais de 2 anos no cargo ?	S	N	S	N	S	N	S	N
AÇÕES								
Exame especial	X	X	X		X	X		
Exame normal				X			X	X

onde: S=sim; N=não; X=ação a ser executada.

- Cobertura mínima é ter pelo menos um caso de teste por regra na decisão da tabela

Criador: Thiago Andrade

Linkedin: <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

Email: [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)



- Cobertura é medida como o número de regras de decisão testadas por pelo menos um caso de teste, dividido pelo número total de regras de decisão
- Teste transição de estado
  - Componentes ou sistemas podem responder de maneira diferente a um evento dependendo das condições atuais ou do histórico anterior, fica conhecido como conceitos de estado.
  - Diagrama de transição de estado mostra os possíveis estados do software, bem como a forma como o software entra e sai e transita entre os estados.
  - Uma transição é iniciada por um evento
  - O evento resulta em transição
  - Imagem de ajuda
    - Transição de estado

## Transição de Estado



- **Estado:** são as condições ou situações durante a vida de um sistema na qual ele satisfaz algumas condições, executa algumas atividades ou espera por eventos.
- **Transição:** é o relacionamento entre dois estados, indicando que o objeto que está no primeiro estado irá passar para o segundo estado mediante a ocorrência de um determinado evento.
- **Evento:** é causa necessária para que haja a transição de estado.
- **Ação:** A ação é basicamente a resposta que o sistema dará ao evento executado.
- **Estado inicial:** estado por onde se começa a leitura de um diagrama de estado.
- **Estado final:** estado que representa o fim.

- Teste caso de uso
  - Os testes podem ser derivados de casos de uso, que são uma maneira específica de projetar interações com itens de software, incorporando requisitos para as funções de software representadas pelos casos de uso.
  - Casos de uso estão associados a atores (Usuário humanos ou componentes do sistema ou sistemas) e assuntos (Componente ou sistemas ao qual o caso de uso é aplicado)

Criador: Thiago Andrade

Linkedin: <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

Email: [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)

- Podem ser representadas graficamente por fluxos de trabalho, diagramas ou modelos de processos.

## Teste Caixa Branca

- Baseado na estrutura interna do objeto de teste
- Pode ser usado em todos os níveis de teste
- Passam muito pelo conceito de cobertura (Todas as técnicas de caixa branca começam com o nome “Cobertura”)
- Teste e cobertura de instrução (Cobertura de sentença, declaração e instrução)
  - Testa todas as instruções do código
  - Sempre é menor ou igual ao teste de cobertura de decisão
  - Quando tiver 2 ou mais conjuntos, o maior prevalece
- Teste e cobertura de decisão (Desvio)
  - Testa as decisões existentes no código e o código executado com base nos resultados da decisão
- O valor da instrução e teste de decisão
  - Atingir 100% de cobertura de decisão garante 100% de cobertura de instrução (mas não vice versa)
  - Cobertura de Instrução (Cobertura de sentença, declaração e comando)
  - Cobertura de decisão (Cobertura de desvio)
- Resumo teste de cobertura e decisão
  - Ramo de Tópico 1

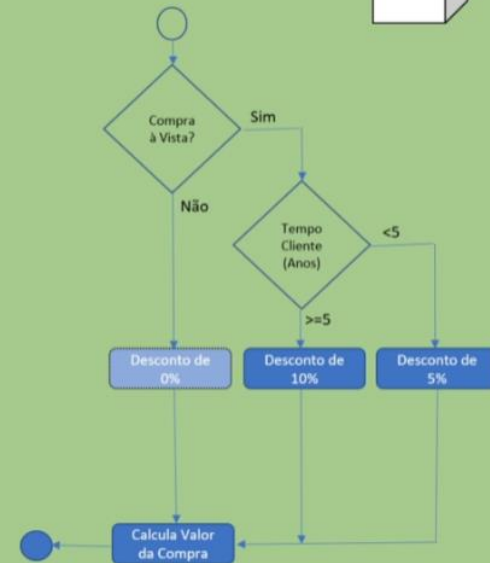
## Técnicas de Teste de Caixa-Branca - Teste de Instruções e Cobertura



Objetivo: passar pelo menos uma vez em cada instrução existente no código para atingir cobertura de 100%.

```
//Tempo em anos
If (compra_a_vista==true)
{
    If (tempo_cliente>=5)
    {
        Desconto=10; // Desconto em percentual
    } else
    {
        Desconto=5;
    }
}

valor_compra = valor_compra * (1-(desconto/100));
```



Você precisará de 2 testes para cobrir todas as instruções. 1. Compra à vista = Verdadeiro e Cliente com mais de 5 anos. 2. Compra à Vista = Verdadeiro e Cliente com Menos de 5 Anos. (100% de cobertura das 5 instruções existentes)

### Teste Baseado em experiência

- Dependendo da abordagem do testador e experiência, essas técnicas podem alcançar graus de cobertura e eficácia amplamente variados
- Suposição de erro:
  - É uma técnica usada para prever a ocorrência de erros, defeitos e falhas, com base no conhecimento:
    - Como o aplicativo funcionou no passado
    - Que tipos de erro tendem a ser cometidos
    - Falhas ocorridas em outros aplicativos
  - Uma abordagem metódica para a técnica de suposição de erro é criar uma lista de possíveis erros, defeitos e falhas e modelar os testes que expõem falhas e os defeitos que as causaram
- Teste Exploratório:
  - Testador experiente, planejamento leve que produza carta de teste e fazer o teste em um limite de tempo
  - Os resultados são usados para aprender mais sobre o componente ou sistema e para criar teste para as áreas que podem precisar de mais teste
  - Teste baseado em sessão, o teste exploratório é conduzido dentro de uma janela de tempo definida, e o testador usa um termo contendo objetivos de teste para orientar o teste.
  - É mais útil quando há poucas ou inadequadas especificações ou pressão de tempo significativo nos testes

Criador: Thiago Andrade

Linkedin: <https://www.linkedin.com/in/thiago-andrade-2a97555b/>

Email: [maciel.thiago@gmail.com](mailto:maciel.thiago@gmail.com)

- Teste baseado em checklist:

- Os testadores modelam, implementam e executam testes para cobrir as condições de teste encontradas em uma lista.
- Testadores criam uma lista ou expandem uma existente, mas os testadores também podem usar um checklist existente sem modifica-lo
- Checklist podem ser construídos com base na experiencia, conhecimento sobre o que é importante para o usuário ou uma compreensão de por que e como o software falha

## Teste Estático

- Em contraste com o teste dinâmico o Teste Estáticos envolve em ler, comparar...

## Teste Funcional

- Os testes funcionais garantem que os requisitos ou especificações sejam devidamente atendidos pela aplicação.

## Teste Não Funcional

- Ele verifica se o comportamento do sistema está de acordo com o requisito ou não

## Iso e normas

(ISO/IEC/IEEE 29119-1) Conceito de teste

(ISO/IEC/IEEE 29119-2) Processo de teste

(ISO/IEC/IEEE 29119-3) Produtos de trabalho do teste, testware.

(ISO / IEC / IEEE 29119-4) Técnicas de teste

(ISO 25010) Características de Qualidade

(ISO20246) Processo de revisão