

# FAST

FORMAÇÃO ACELERADA EM  
SOLUÇÕES DE TECHDESIGN



C.E.S.A.R.

## Fundamentos de QA e Qualidade de Software

Todos os direitos reservados.

Este material, ou qualquer parte dele, não pode ser reproduzido,  
divulgado ou usado de forma alguma sem autorização escrita.



# Explorando os Fundamentos dos Testes de Software

Fundamentos de QA e Qualidade de Software

## Aula 2



# Fred Melo

**Trabalho no CESAR há 3 anos como Especialista em QA (Staff QA)**

- Pós graduação em Curso Sequencial em Análise de Testes - Cin / UFPE
- ISTQB - CTFL
- Cin/Motorola (14 anos)
- Cin/Samsung (4 anos)





# Olá! Eu sou Clauber Lima

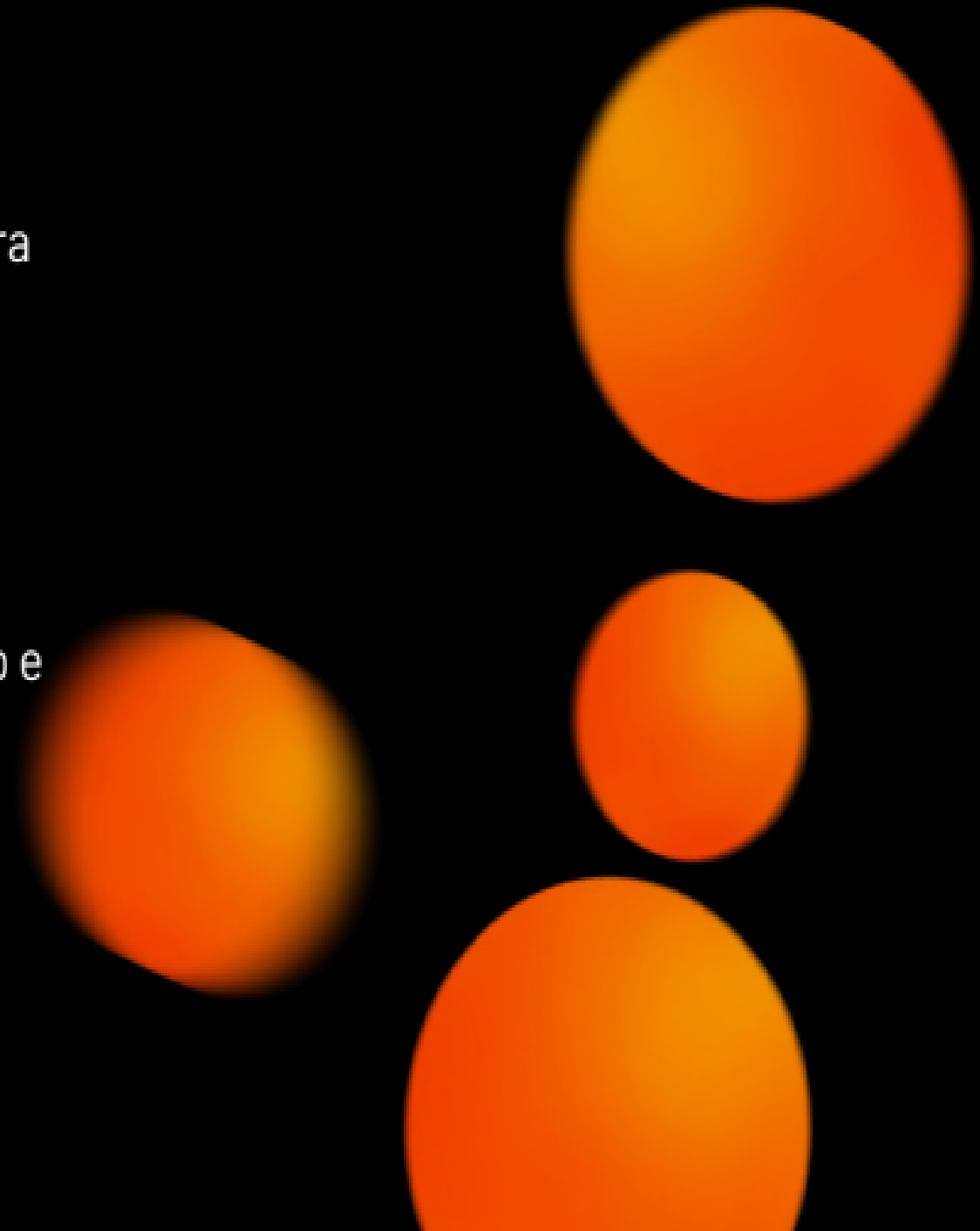
- 15+ anos em QA e no CESAR desde 2020
- Certificações (CTFL, ITIL, CPRE, ICAgile Professional, PS4-RPA-FL)
- Experiência em testes ágeis, automação e qualidade ponta a ponta
- Professor e Mentor
- Motociclista e amante de carros antigos

# O que veremos hoje?

- Conteúdo 01 – Tipos e Níveis de Teste
- Conteúdo 02 – Técnicas de Teste
- Conteúdo 03 – Prática guiada

# Bora revisar?

1. **Qualidade em software:** como ela gera valor e impacta o produto final?
2. **O papel do QA no ciclo de desenvolvimento:** evolução do papel tradicional de tester para QA moderno e colaboração, o que muda?



# Vamos praticar!

📌 Atividade de Fixação – Aula 01

📝 Agora é hora de testar seus conhecimentos!

👉 Acesse o link abaixo e responda ao formulário com perguntas sobre os fundamentos de QA e Qualidade de Software.

🔗 [LINK FORMS](#)

🕒 Tempo: 10 minutos

💡 Responda com base nos tópicos vistos na aula.

Boa sorte!



# Tipos e Níveis de Teste

Fundamentos de QA e Qualidade de Software

## Aula 2





# O que é teste?

## Não é só usar o sistema:

- É um processo que inclui muitas atividades diferentes.
- O processo de teste inclui:
  - Planejamento de teste
  - Análise
  - Modelagem e implementação dos testes
  - Relatórios de progresso
  - Resultados de testes e
  - Avaliação da qualidade de um objeto de teste.



*"O teste de software é uma maneira de avaliar a qualidade do software e reduzir o risco de falha do software em operação"*

# Níveis de testes

## 1 Teste de Unidade / Componente

- Foco em módulos ou componentes isolados.
- Geralmente feito pelo desenvolvedor.
- Exemplo: teste de uma função ou método.

## 2 Teste de Integração

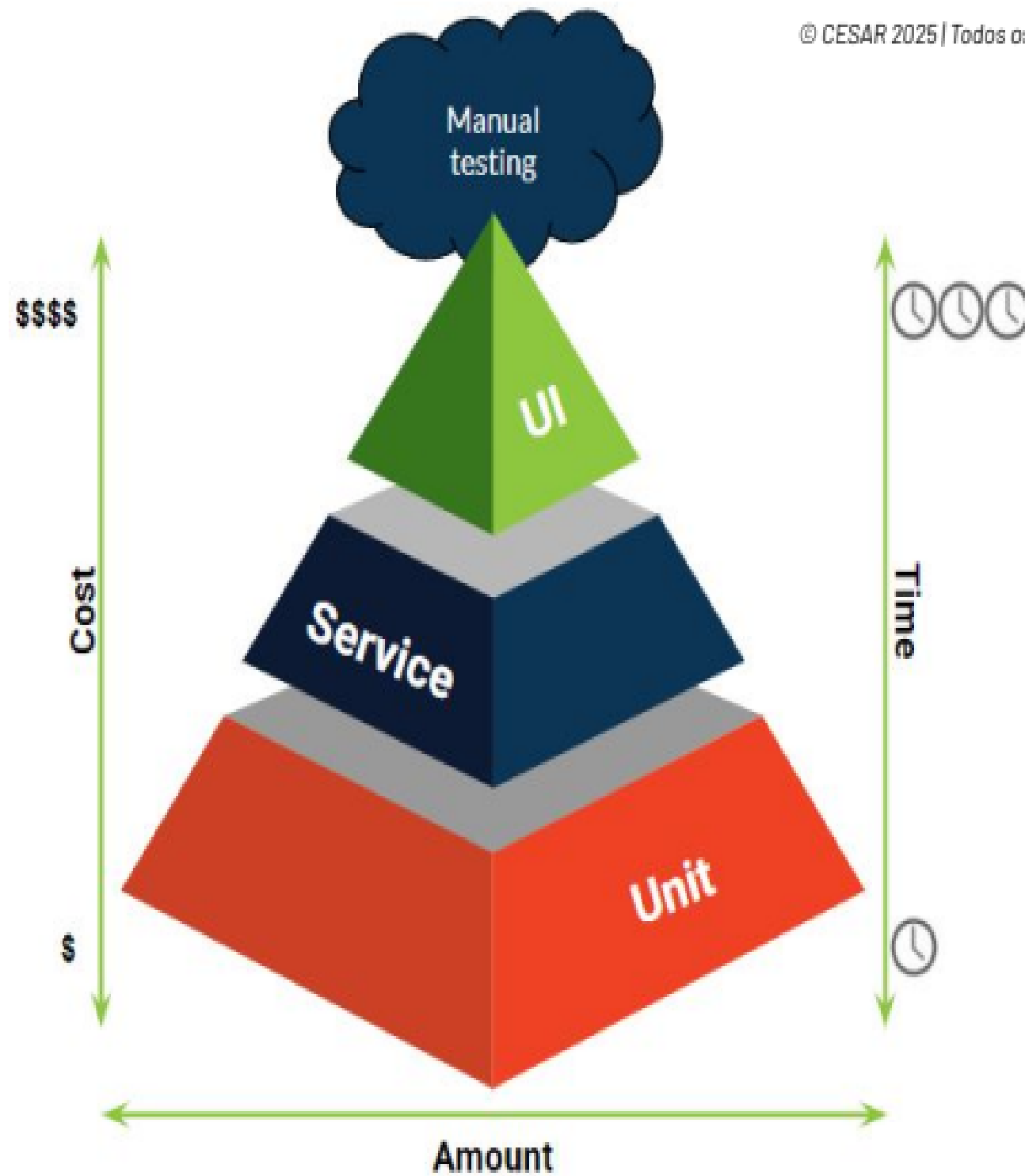
- Foco na interação entre componentes/sistemas.
- Testa se a comunicação entre as partes funcionam corretamente.
- Pode ser incremental (bottom-up, top-down).

## 3 Teste de Sistema

- Testa o sistema como um todo, validando o comportamento conforme os requisitos.
- Inclui testes funcionais e não funcionais.
- Ambiente próximo ao de produção.

## 4 Teste de Aceitação

- Validado pelo cliente ou usuário final.
- Confirma se o sistema atende aos requisitos de negócio.
- Pode incluir Testes de Aceitação do Usuário (UAT), Testes Alfa e Beta.



# Tipos de Testes

## Funcionais

- Foco **no que** o sistema faz.
- Baseados em requisitos funcionais.
- Exemplos: testes de funcionalidades, integração funcional, APIs.



## Não Funcionais

- Foco em **como** o sistema se comporta.
- Avaliam atributos de qualidade como confiabilidade, usabilidade, desempenho, segurança, compatibilidade.



Performance



Reliability



Accountability



Compatibility



Recovery



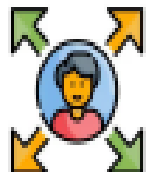
Performance



Portability



Security



Scalability

# Tipos de Testes

## Relacionado à mudança

Testes aplicados após mudanças no sistema em produção:

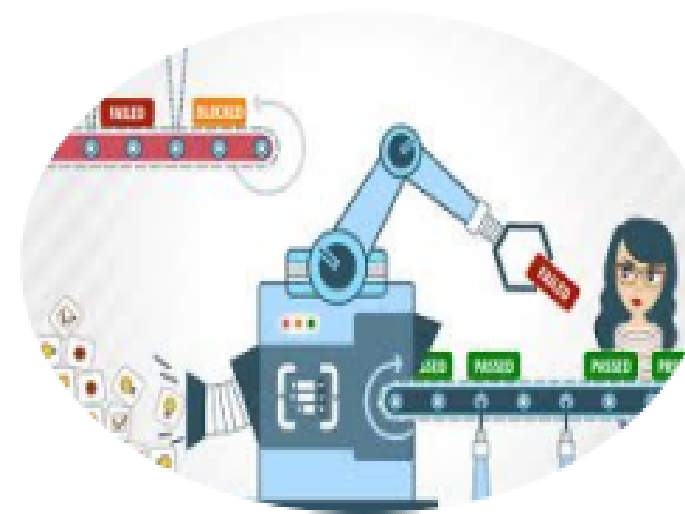
- Correção de bugs
- Atualizações de sistema
- Melhorias

## Manuais

- Testes realizados diretamente pelo QA, sem automação
- Ideal para validar experiências de uso, interfaces e comportamentos subjetivos.

## Automáticos

- testes realizados por scripts e ferramentas especializadas
- Ideal para fluxos repetitivos, grandes volumes, execuções frequentes e múltiplos ambientes.





# Cenários e Casos de Teste

Fundamentos de QA e Qualidade de Software

## Aula 2



# Cenários de Teste

- Descrevem um **contexto** amplo de teste, como "login do usuário" ou "processamento de pagamento".
- Focam no "**o quê**" está sendo testado, definindo o escopo geral da validação.
- Não entram em detalhes sobre as etapas de execução.
- Exemplos:
  - "Verificar a funcionalidade de login do usuário."
  - "Testar o fluxo de compra de um produto."
  - "Validar o cálculo de frete."



# Casos de Teste

- São descrições **detalhadas** de como realizar um teste específico **dentro de um cenário**.
- Especificam as etapas, entradas, resultados esperados e condições para cada teste.
- Focam no "**como**" o teste deve ser executado.
- Um cenário pode ter **múltiplos casos de teste associados**.
- Exemplos:
  - "Login com usuário e senha válidos." (Cenário: Login)
  - "Login com usuário válido e senha inválida." (Cenário: Login)
  - "Login com campos de usuário e senha vazios." (Cenário: Login)
  - "Adicionar um item ao carrinho e finalizar a compra." (Cenário: Processamento de pagamento)
  - "Calcular frete para diferentes endereços." (Cenário: Cálculo de frete)





# Plano de Teste

É um documento detalhado que define os objetivos, escopo, cronograma, estratégias, recursos e riscos de um projeto de testes de software.

Serve como um roteiro que "guia" as atividades de teste, garantindo que todas as etapas, técnicas e estratégias.

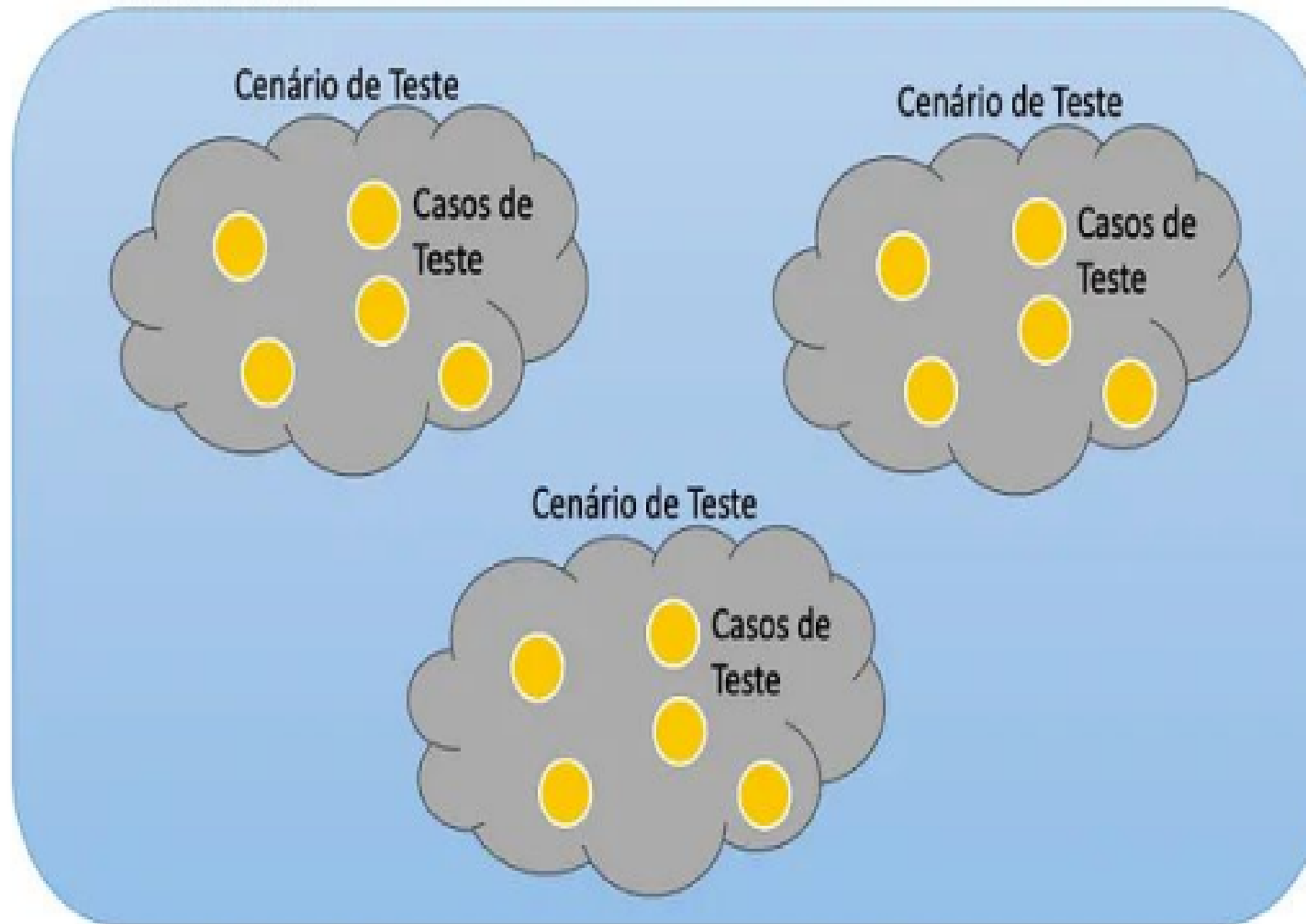


# Detalhes do plano de Teste

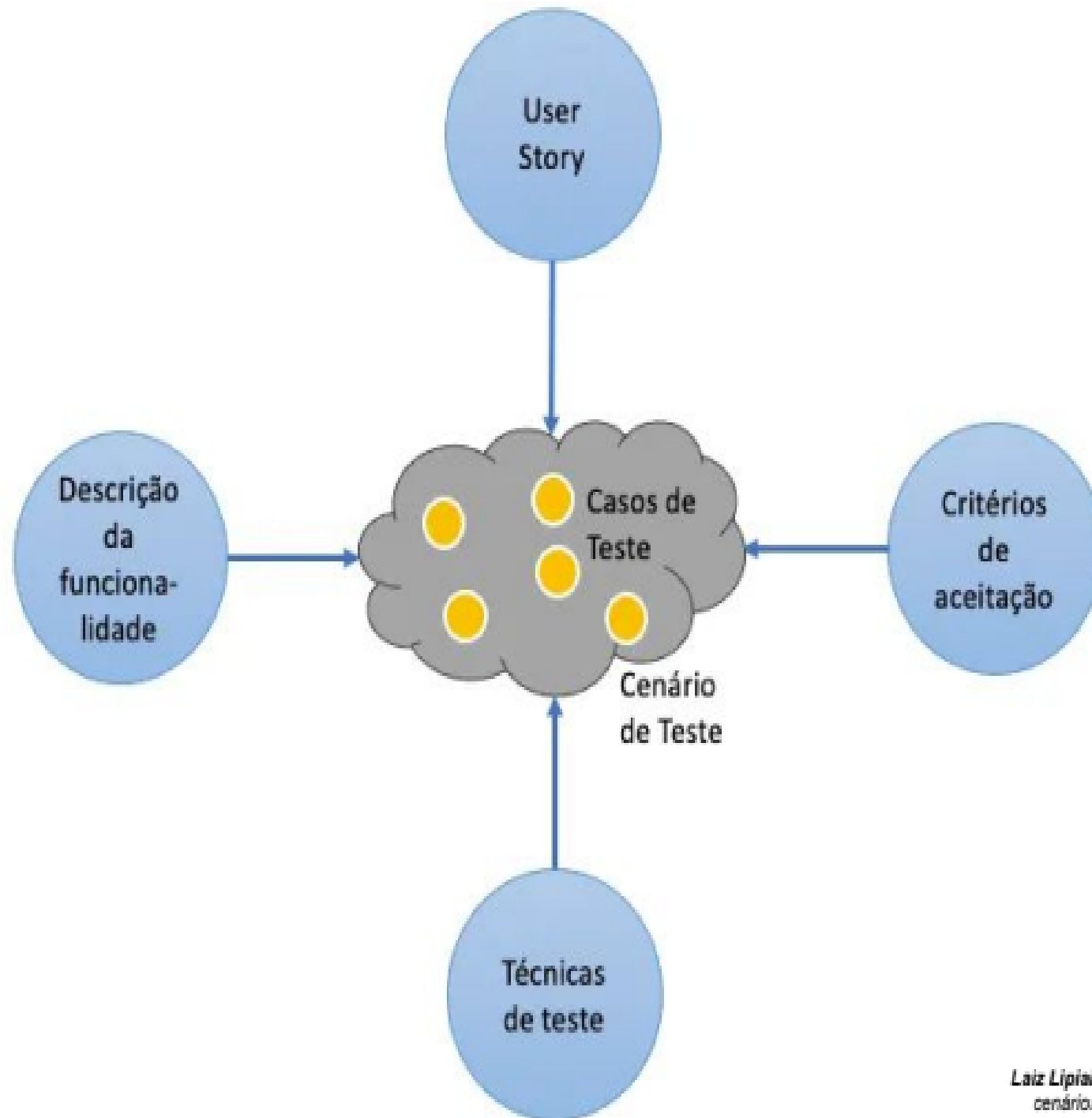
- **Objetivos:** O que se espera alcançar com o processo de teste.
- **Escopo:** Quais partes do software serão testadas e quais não serão (o que está fora do escopo).
- **Abordagem:** Como os testes serão conduzidos (quais tipos de testes serão realizados, como testes funcionais, de performance, segurança, etc.).
- **Cronograma:** O prazo e os marcos para a conclusão dos testes.
- **Recursos:** Os recursos humanos (equipe) e de infraestrutura necessários para os testes.
- **Critérios:** Requisitos de entrada (quando os testes podem começar) e de saída (quando os testes podem ser encerrados).
- **Riscos:** Identificação de possíveis problemas que podem impactar o projeto e como mitigá-los.



## Plano de Teste



PLANO DE TESTE:  
DOCUMENTOS PRODUZIDOS PARA PLANEJAR  
OS TESTES DO PROJETO.



# Cenário

"Testar que feito um depósito em conta, o dinheiro estará disponível no tempo estipulado."

## Caso 1:

- **Pré-condição:** O depósito em conta deve estar funcional
- **Entradas:** Fazer um depósito de R\$ 100
- **Resultado esperado:** A conta deve ter o valor anterior mais os R\$ 100 depositados após 24 horas.

## Caso 2:

- **Pré-condição:** O depósito em conta deve estar funcional
- **Entradas:** Um depósito de R\$ 100 deve ter sido feito há 4 horas
- **Resultado esperado:** O valor de R\$ 100 deve ser exibido na conta como lançamento futuro.

"Testar que feito um depósito em conta, o dinheiro estará disponível no tempo estipulado."

## Caso 1:

**Feature:** Depósito em conta

**Scenario:** Realizar depósito de R\$ 100 com saldo atualizado após 24 horas

**Given** que o depósito em conta está funcional

**When** o usuário realiza um depósito de R\$ 100

**Then** o saldo da conta deve refletir o valor anterior somado a R\$ 100 após 24 horas

## Caso 2:

**Feature:** Depósito em conta

**Scenario:** Exibir depósito recente como lançamento futuro

**Given** que o depósito em conta está funcional

**And** um depósito de R\$ 100 foi realizado há 4 horas

**When** o usuário acessa o extrato da conta

**Then** o valor de R\$ 100 deve ser exibido como lançamento futuro

# INTERVALO!



20 min.



# Técnicas de Teste

Fundamentos de QA e Qualidade de Software

## Aula 2





# Técnicas de Teste

- O objetivo de uma técnica de teste é ajudar a identificar as condições de teste, os casos de teste e os dados de teste.
- Ajudam a aumentar a cobertura de testes, reduzir redundâncias e identificar falhas de forma mais eficiente.
- A escolha de quais técnicas de teste usar depende de vários fatores como tempo, orçamento, riscos, documentação disponível, complexidade, etc.

Caixa Preta

Caixa  
cinza

Caixa Branca

Teste baseado em  
experiência

- **Caixa Preta**

- Testes derivados de requisitos, especificações, casos de uso e histórias de usuários;
- Os casos de teste podem ser usados para detectar lacunas entre os requisitos e as suas implementações, bem como desvios nos requisitos.

- **Caixa Cinza**

- Mistura testes de caixa branca e caixa preta
- Necessita de conhecimento parcial dos componentes internos do sistema.
- O QA pode não ter conhecimento completo do código-fonte de um aplicativo, mas pode ter conhecimento parcial dele e/ou acesso à documentação.

- **Caixa Branca**

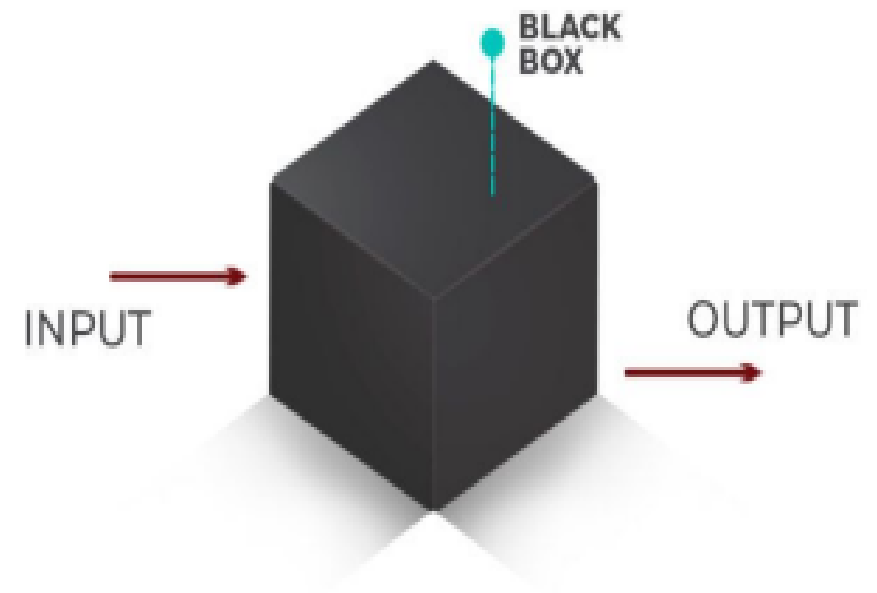
- Teste derivados de código, arquitetura de software, detalhamento do projeto ou qualquer outra fonte de informação relacionada à estrutura do software.

- **Baseado em experiência**

- Teste derivados de uma base de teste que pode incluir conhecimento e a experiência de testadores, desenvolvedores, usuários e stakeholders.
- Esse conhecimento e experiência inclui o uso esperado do software, seu ambiente, possíveis defeitos e a distribuição desses defeitos.

- **Caixa Preta**

- Particionamento de equivalência
- Análise de valor limite
- Teste de tabela de decisão
- Teste de transição de estado



# Técnicas de Teste – Caixa preta

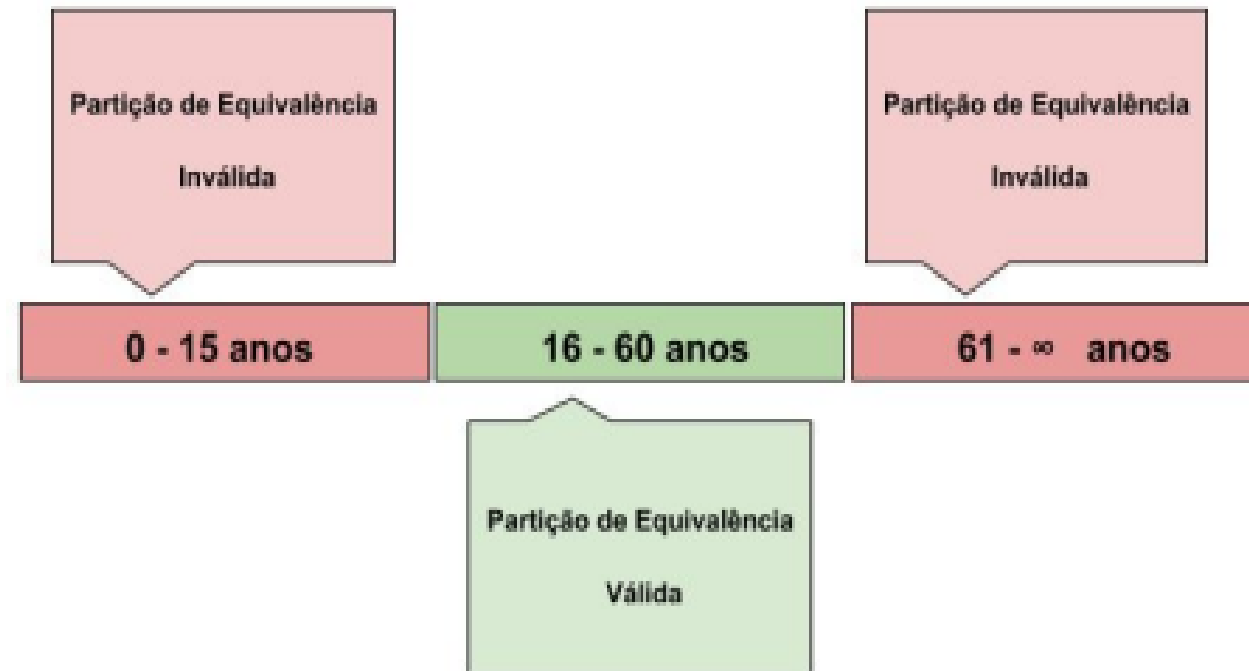
## Particionamento de equivalência

- Técnica de caixa-preta usada para reduzir a quantidade de testes.
- Divide os dados de entrada em **classes (ou faixas)** que se comportam da mesma forma.
- Pressupõe que **testar um valor de cada classe é suficiente** para representar todos os valores daquela faixa.
- Muito útil para formulários, entradas numéricas e validações de regras.

### Exemplo prático:

Para um campo que aceita idades de 15 a 60 anos:

- **Classe válida:** 18 a 60 → ex: testar com 30
- **Classes inválidas:**
  - Menor que 18 → ex: testar com 10
  - Maior que 60 → ex: testar com 70







# Técnicas de Teste – Caixa preta

## Valor Limite

- **Técnica de caixa-preta** focada nos **limites das faixas válidas e inválidas**.
- Baseia-se na ideia de que os erros acontecem com mais frequência **nas extremidades dos intervalos**.
- Testa os **valores mínimos e máximos válidos**, e também os **valores logo fora do intervalo**.
- Complementa bem o particionamento de equivalência
- Ideal para campos numéricos, tamanhos, limites de caracteres etc.

## Exemplo prático:

Para um campo que aceita idades de **18 a 60 anos**:

- **Limites válidos:** 18 e 60
- **Valores adjacentes (inválidos):** 17 (abaixo), 61 (acima) Valores a testar:
  - 17  (inválido)
  - 18  (válido)
  - 60  (válido)
  - 61  (inválido)

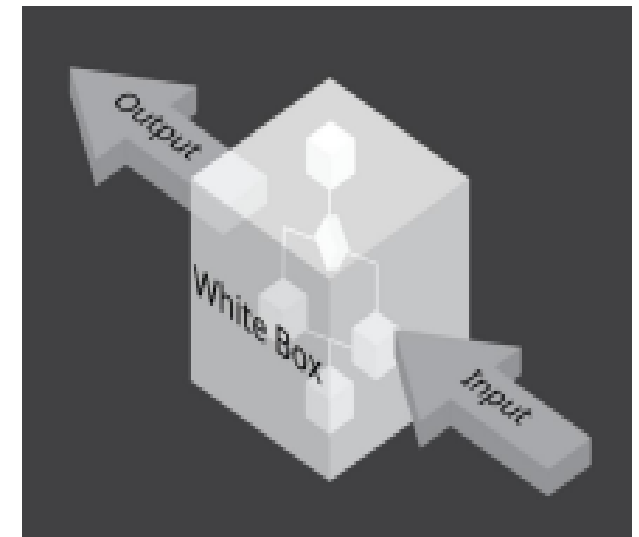
# Técnicas de Teste – Caixa Cinza

- **Teste de matriz:** Concentra-se nas variáveis dentro de um programa, enumerando-as, avaliando os riscos que representam e testando se são usadas correta e eficientemente.
- **Teste de regressão:** O código pode ser modificado para adicionar funcionalidade ou corrigir problemas de segurança. O teste de regressão verifica se um aplicativo ainda passa nos testes após ser modificado.
- **Teste de padrão:** Analisa o passado de um aplicativo para identificar tendências que causaram defeitos no passado e que podem causar defeitos no futuro.



# Técnicas de Teste – Caixa Branca

- **Teste de caminho de código**  
Garante que todos os caminhos de execução possíveis em uma função ou método sejam testados.
- **Teste de condição**  
Avalia se todas as condições lógicas em um bloco de código são testadas com valores verdadeiros e falsos.
- **Teste de ciclo**  
Garante que todas as estruturas de repetição em um ciclo sejam testadas em diferentes situações, como ciclos vazios, ciclos com um único elemento e ciclos com múltiplos elementos.





# Técnicas de Baseados em Experiência

- Suposição de erro
- Teste exploratório
- Teste baseado em checklist



# Em resumo...



## Níveis de Teste

- Teste de Unidade / Unitário
- Teste de Componentes
- Teste de Integração
- Teste de Sistema
- Teste de Regressão
- Teste de Aceitação

## Técnicas de Teste

- Teste Caixa-Branca / *White-Box* / Estrutural
- Teste Caixa-Preta / *Black-Box* / Funcional

## Tipos de Teste

- Teste de Usabilidade
- Teste de Desempenho / *Performance*
- Teste de Carga
- Teste de Estresse / Esforço
- Teste de Segurança

# Vamos praticar!

Você está testando uma tela de cadastro de usuário, que possui os seguintes campos obrigatórios:

- **Nome:** mínimo 3 caracteres
- **E-mail:** deve ter formato válido (ex: usuario@dominio.com)
- **Idade:** deve estar entre 18 e 60 anos
- **Senha:** mínimo 6 caracteres

O sistema **não permite** o cadastro se qualquer campo estiver inválido.

## Cadastro de Usuário

Nome:

E-mail:

Idade:

Senha:

Cadastrar

# Vamos praticar!

## Atividade:

1. Descreva um cenário funcional simples para o cadastro com dados válidos.
2. Escreva 2 casos de teste cobrindo:
  - Um caso com particionamento de equivalência
  - Um caso com valor limite
3. Identifique o tipo e o nível de teste que você aplicaria nos seus exemplos.

## Cadastro de Usuário

Nome:

E-mail:

Idade:

Senha:

Cadastrar