



MÓDULO 01: TESTE FUNCIONAL

Yury Alencar

O que é teste?

VERIFICAR "SE ALGO FUNCIONA COMO PLANEJADO"



TESTE MANUAL

Tedioso e trabalhoso

Propenso a erros

Muitas vezes incompleto (nem todos os cenários são cobertos)



TESTE AUTOMATIZADO

**Esforço inicial (codificação),
redução de esforço a longo prazo**

Previsível e consistente

**Cobertura de código e cenários
alta/completa pode ser alcançada**

O que são testes unitários?

UNIDADE

Um **bloco de construção** do seu aplicativo

Idealmente, o **menor bloco** de construção possível

Exemplo: A função, a classe, o componente...

Aplicação

UNIDADE

UNIDADE

UNIDADE

UNIDADE

Aplicativo = Combinação de todas as unidades

Se todas as unidades foram testadas, o aplicativo geral deve funcionar

As alterações são sempre testadas em todas as unidades para evitar bugs

Apoiado por **testes de integração**

Por que testes unitários?



Evitar quantidades infinitas de testes manuais



Permite cobrir (perto de) 100% do seu código e cenários

As alterações de código são testadas em todos os cenários (quase) instantaneamente



Escreva um código mais limpo e melhor (Código testável)

Testes unitários, de integração e de ponta a ponta (E2E)



Teste unitário

Testar as unidades individuais de um aplicativo

Cada unidade (bloco de construção) é testada de forma independente

Se todas as unidades funcionarem, o aplicativo geral funcionará



Teste de integração

Testar a combinação de unidades

Verificar se as unidades (blocos de construção) funcionam juntos

Mesmo que todas as unidades funcionem sozinhas, a combinação pode falhar



Teste de ponta a ponta (E2E)

Testar fluxos inteiros e recursos de aplicação

Testar as funcionalidades do ponto de vista do usuário final

Usuários reais usam seu aplicativo e seus recursos, não unidades individuais

Testes unitários, Integração e de Ponta a Ponta (E2E)

Você deve combinar todos os tipos de testes



Teste unitário

Identifique rapidamente alterações e *Breaking Changes*

Ignora fluxos e interferências reais do usuário



Teste de integração

Teste processos e combinação de unidades

Identificar a raiz exata de um erro pode ser complicado

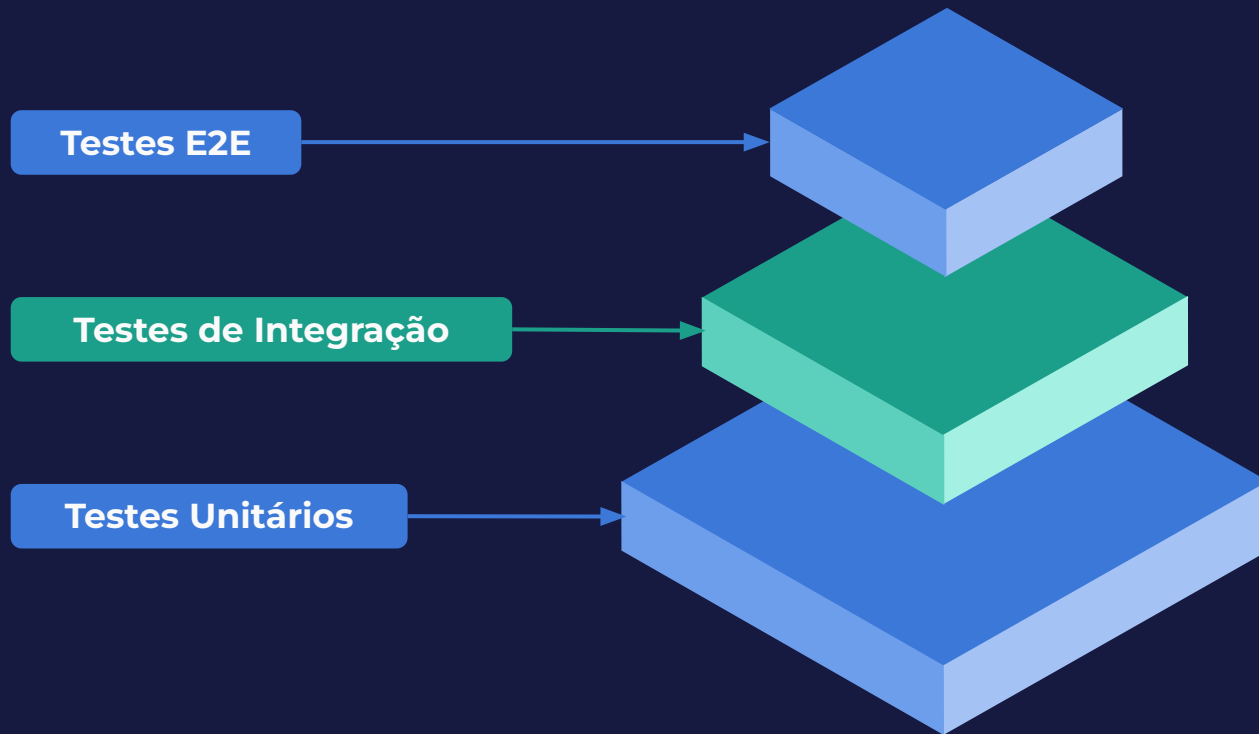


Teste de ponta a ponta (E2E)

Teste fluxos e comportamentos realistas do usuário

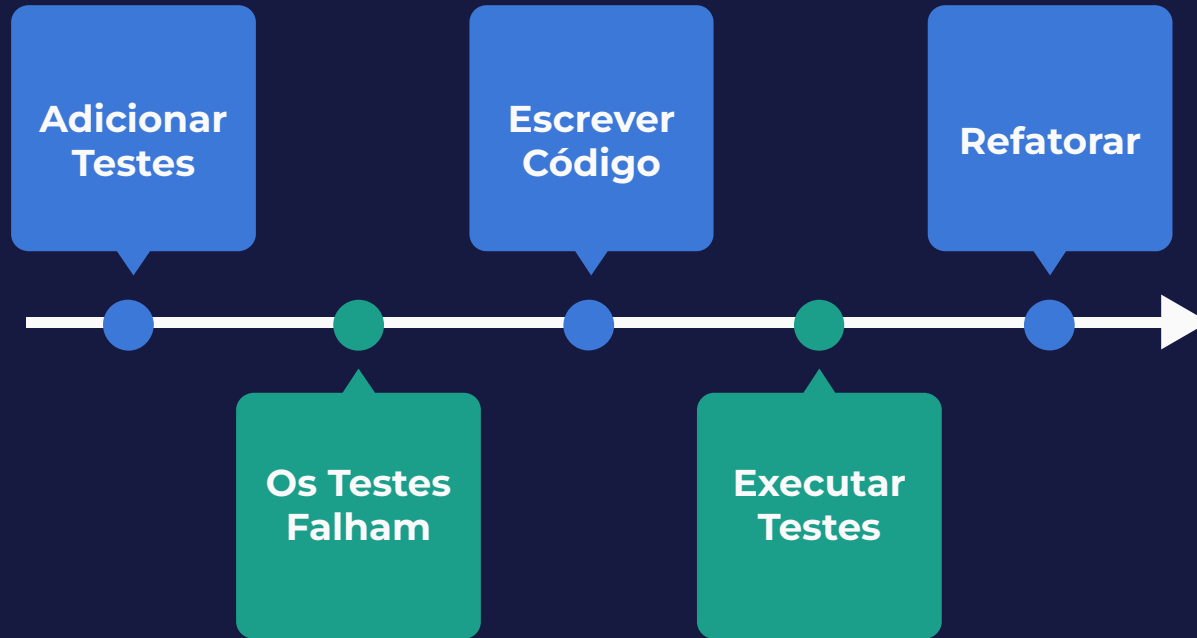
Cobrir todos os comportamentos possíveis pode ser um desafio

A pirâmide de testes



Test-Driven Development (TDD)

TDD - Desenvolvimento Orientado a Testes



Sobre este curso

Visão Geral

Configuração do projeto e
ferramentas necessárias

Fundamentos de testes

Escrevendo bons testes

Conceitos de teste
avançados

Mocks & Spies

Um pouco mais sobre
Mocks & Spies

Testes e o DOM