

RESUMO DA AULA

Quais tipos de teste existem?

Nesta seção, vamos explorar os três principais tipos de testes automatizados utilizados no desenvolvimento de software: testes de ponta a ponta (e2e), testes de integração e testes de unidade. Embora existam outros tipos de testes além desses, entender esses três será fundamental para a construção de um projeto com cobertura de testes automatizados. O foco principal será nos testes de unidade, mas conheceremos as características e os objetivos de cada tipo de teste para entender como e onde aplicá-los.

Testes de Ponta a Ponta (e2e) Os testes de ponta a ponta, ou end-to-end (e2e), simulam o comportamento do usuário real, automatizando a interação com o sistema completo. Imagine um robô preenchendo formulários, navegando pelos menus, clicando em botões e verificando o comportamento do sistema como um todo. Esse tipo de teste é ideal para validar cenários de uso reais e verificar que todo o fluxo de funcionalidades funciona conforme esperado. Exemplo prático: Testar se um usuário consegue realizar uma compra em um e-commerce desde a escolha do produto, passando pelo checkout, até a confirmação da compra.

Vantagens: Garante que o sistema funciona corretamente em um ambiente real. **Desvantagens:** São mais complexos e demorados de implementar. Custam mais recursos para serem executados. Sua execução consome mais tempo, o que pode impactar o processo de desenvolvimento. Por esses motivos, recomenda-se usar testes de ponta a ponta apenas para cobrir fluxos críticos do sistema, evitando testá-lo inteiramente com e2e.

Testes de Integração Como o próprio nome sugere, os testes de integração são focados em verificar a integração entre diferentes partes do código. Em vez de testar o sistema como um todo, como nos testes de ponta a ponta, os testes de integração verificam se componentes específicos funcionam corretamente em conjunto. Exemplo prático: Testar se a função que salva um usuário em uma tabela no banco de dados e outra que valida o e-mail do usuário interagem corretamente.

Vantagens: Validam fluxos de dados e interações entre componentes, detectando falhas em pontos de conexão no sistema. **Desvantagens:** Embora mais rápidos que os

testes de ponta a ponta, ainda podem ser complexos e demorar mais para serem implementados do que os testes de unidade. Testes de Unidade Os testes de unidade são o tipo mais básico e comum de teste automatizado. Eles verificam funcionalidades isoladas, como funções e métodos específicos. A ideia é testar pequenas partes do sistema de forma independente, garantindo que cada unidade de código funcione como esperado. Exemplo prático: Testar uma função que soma dois números e verificar se, para vários valores diferentes, o resultado está correto. Vantagens: São rápidos de escrever e executar. Cobrem pequenas partes do código, facilitando a manutenção. Detectam falhas em partes específicas do sistema. Desvantagens: Não garantem que a interação entre diferentes partes do código está correta, pois são focados em elementos isolados.

Pirâmide de Testes Uma abordagem comum ao estruturar testes automatizados é a Pirâmide de Testes, que ilustra a quantidade ideal de cada tipo de teste em um projeto saudável:

Testes e2e	Menor quantidade, maior custo e maior tempo de execução.
Testes de Integração	Quantidade intermediária, custo e complexidade moderados.
Testes de Unidade	Maior quantidade, menor custo e menor tempo de execução.

Interpretação da pirâmide: Quanto mais alto na pirâmide, maior o custo e o tempo para implementar e rodar os testes. Os testes de ponta a ponta (e2e) devem estar no topo, pois devem existir em menor quantidade, focados em cobrir fluxos críticos do sistema. Os testes de integração ficam no meio, cobrindo interações entre diferentes partes do sistema. Os testes de unidade devem ser a base da pirâmide, existindo em maior quantidade, com o objetivo de garantir que cada pequena parte do sistema funcione como esperado. Essa estrutura garante que os testes mais rápidos e simples, que são os de unidade, existam em grande quantidade e forneçam uma boa cobertura para o código.

medida que subimos a pirâmide, a quantidade de testes diminui, focando apenas em cobrir casos mais amplos e interações entre componentes.

Resumo Testes de ponta a ponta (e2e): Simulam o comportamento real do usuário. Garantem que o sistema funciona como um todo, mas são demorados e custosos. Testes de integração: Validam a interação entre diferentes partes do sistema, como funções que trabalham em conjunto. Testes de unidade: Focados em pequenos trechos de código.

digos isolados, verificando funcionalidades específicas de maneira rápida e eficiente. Nesta seção, entendemos os conceitos e as características de cada tipo de teste. No próximo tópico, vamos mergulhar nos testes de unidade, que serão o foco principal deste módulo.
