

# TESTE DE SOFTWARE

## TIPOS DE TESTE DE SOFTWARE

Prof. Flávio Belizário da Silva Mota  
Universidade do Vale do Sapucaí - UNIVAS  
Sistemas de Informação

# TESTE DE SOFTWARE

- Vamos recordar que testes tem como objetivo causar **falhas** em um programa
- Depois do teste, é preciso encontrar e corrigir os defeitos
- Para isso existem diferentes tipos de teste que são classificados de diversas maneiras, dependendo do **objetivo**, do **nível** de aplicação, do **grau de automação** e da **abordagem** adotada

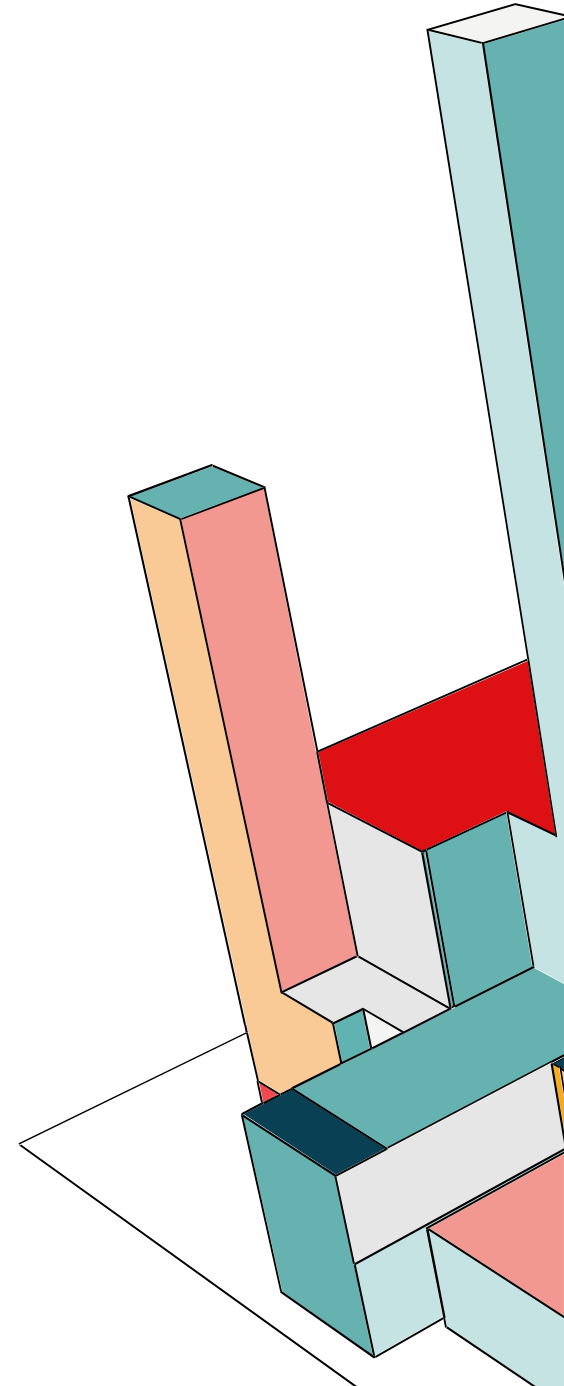


# TIPOS DE TESTES DE SOFTWARE

Os testes de software se dividem em diferentes categorias:

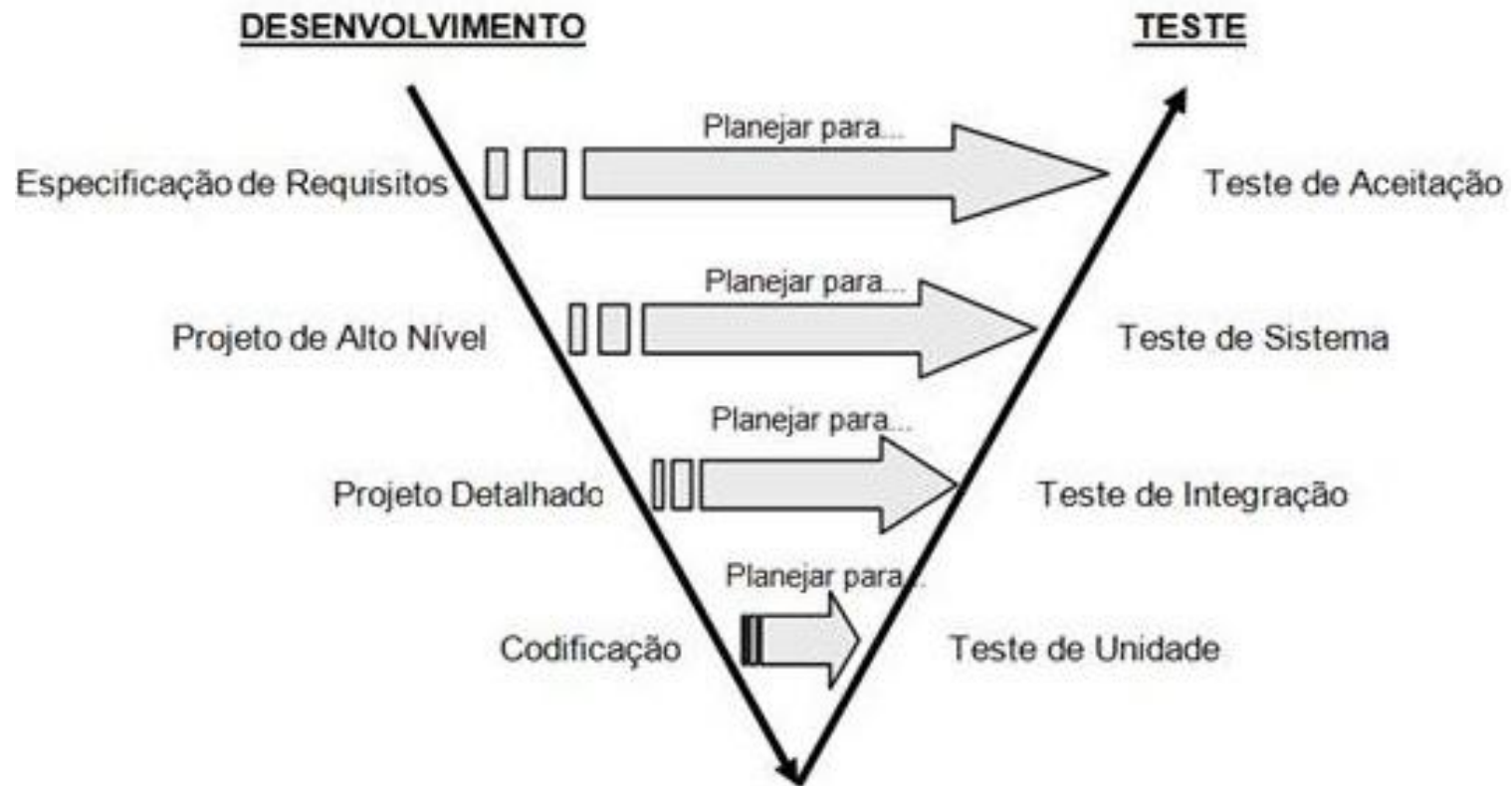
- Níveis (Unidade, Integração, Sistema, Aceitação)
- Objetivo (Funcional, Não Funcional)
- Automação (Manuais, Automatizados)
- Abordagem (Caixa Branca, Caixa Preta, Caixa Cinza)

Cada tipo atende a uma necessidade específica e é aplicado **em momentos distintos do ciclo de vida do software.**



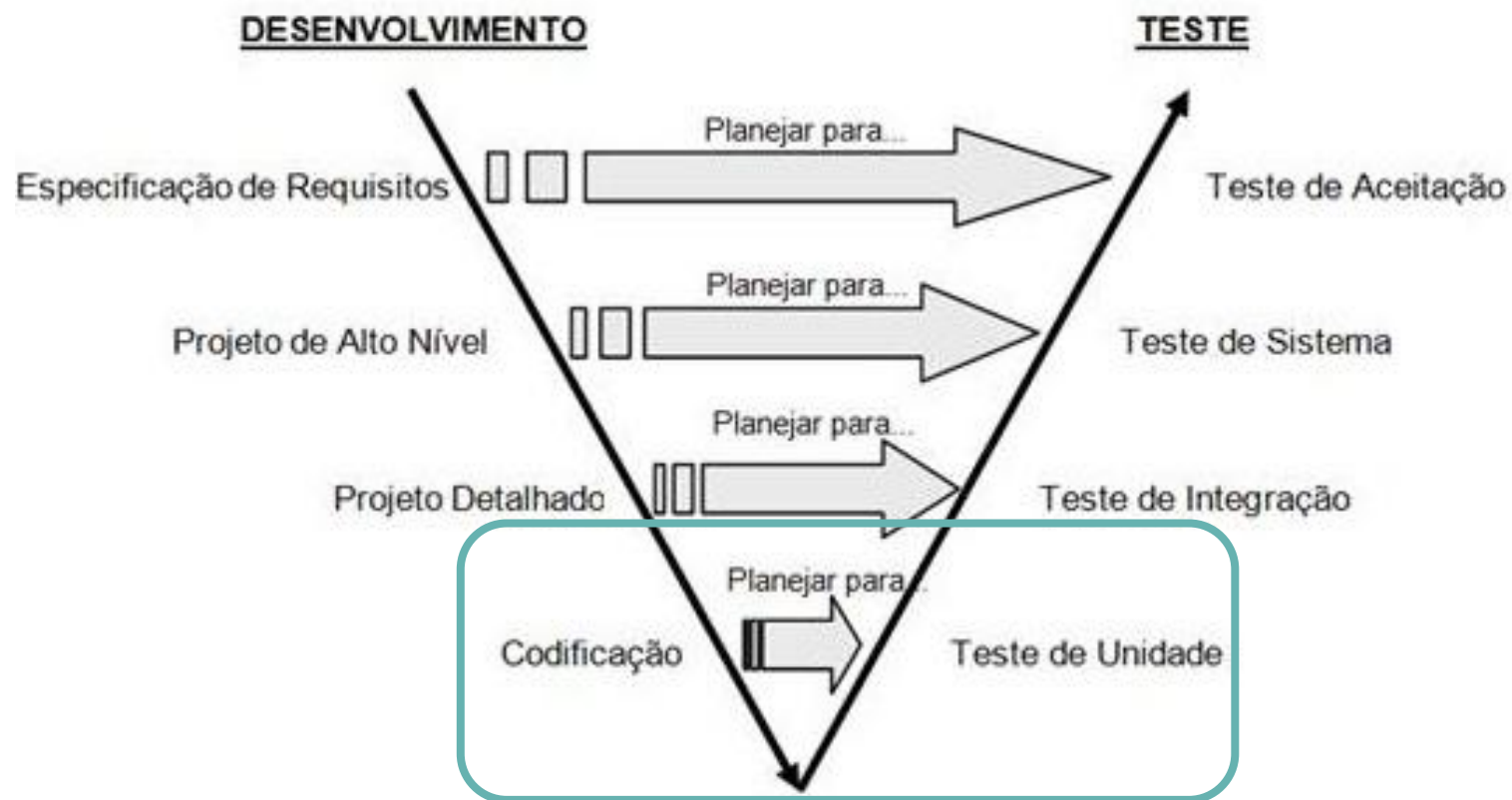
# TESTES POR NÍVEL

Os níveis de teste indicam a profundidade e o momento da verificação dentro do processo de desenvolvimento.



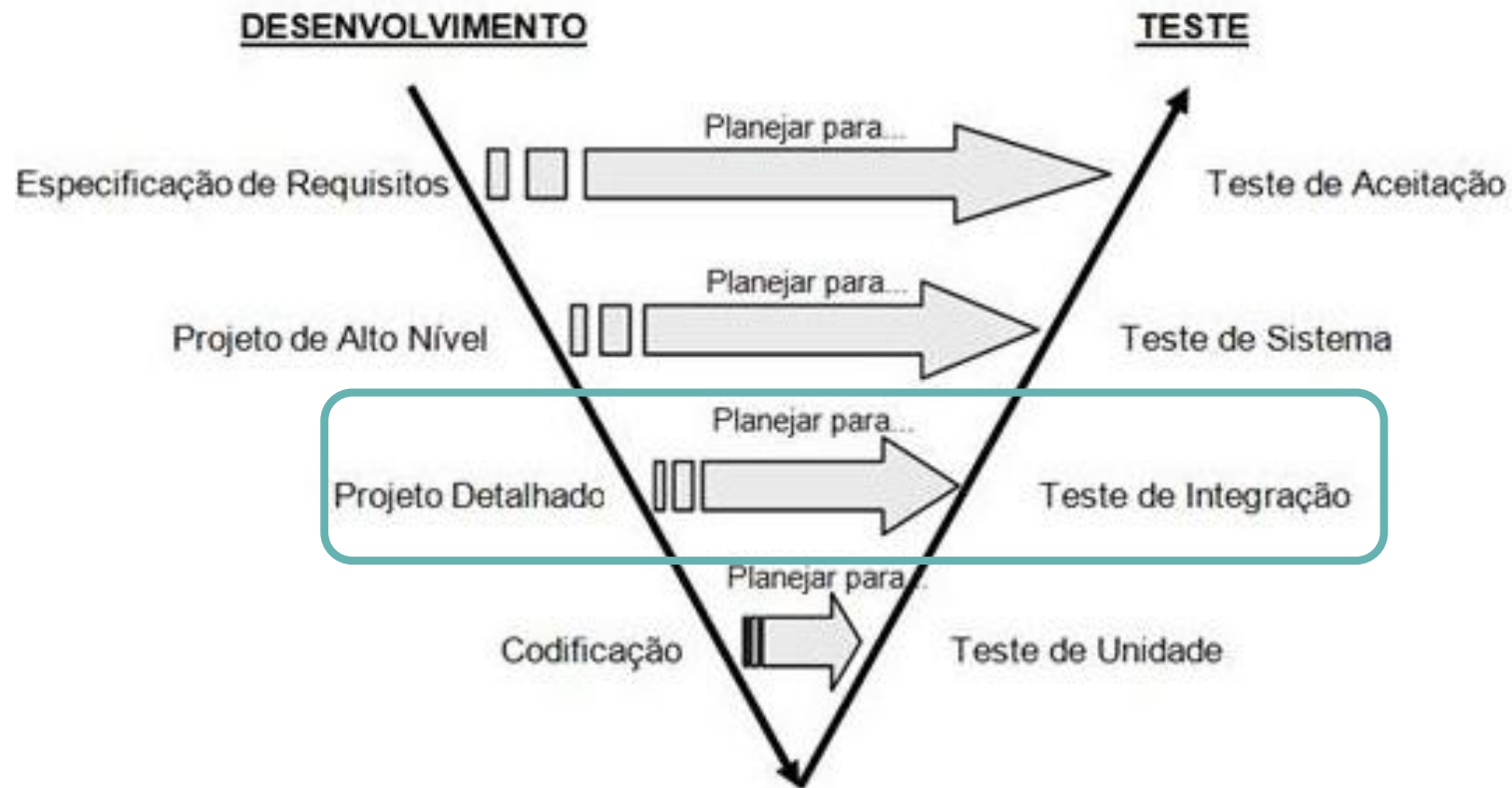
# TESTE DE UNIDADE

- Testa a menor unidade do programa
- Quem realiza é o dev
- Usa a descrição do projeto no nível da unidade como guia para os testes
- Os erros estarão nos limites destas unidades
- Pode ser conduzido em paralelo para diversas unidades



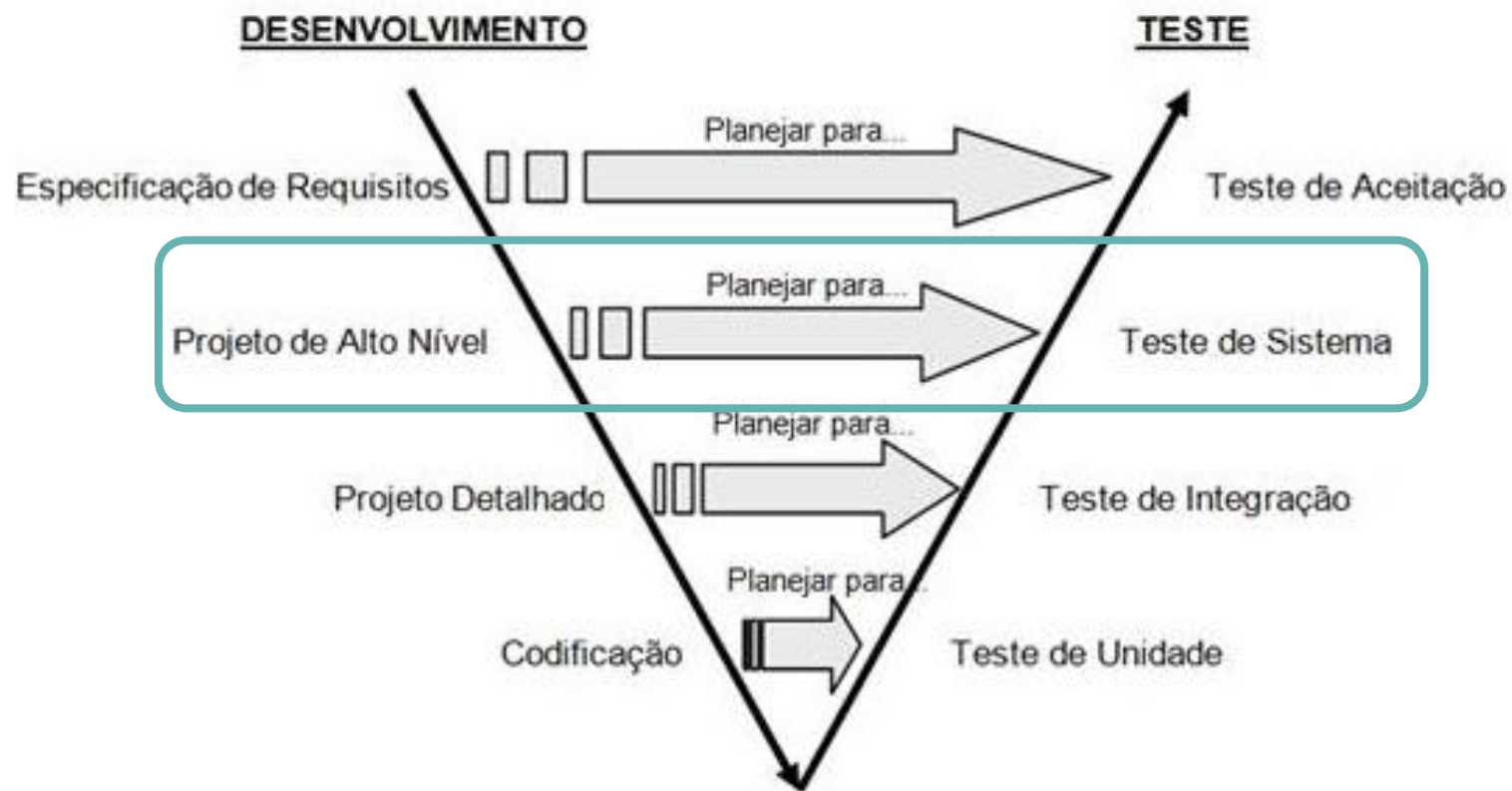
# TESTE DE INTEGRAÇÃO

- Testa a integração entre módulos (diferentes partes do sistema, diferentes sistemas)
- Quem realiza é o dev/tester
- Não incremental: testa todos os componentes após o desenvolvimento (big-bang)
- Incremental: pequenos testes são conduzidos conforme os componentes são desenvolvidos



# TESTE DE SISTEMA

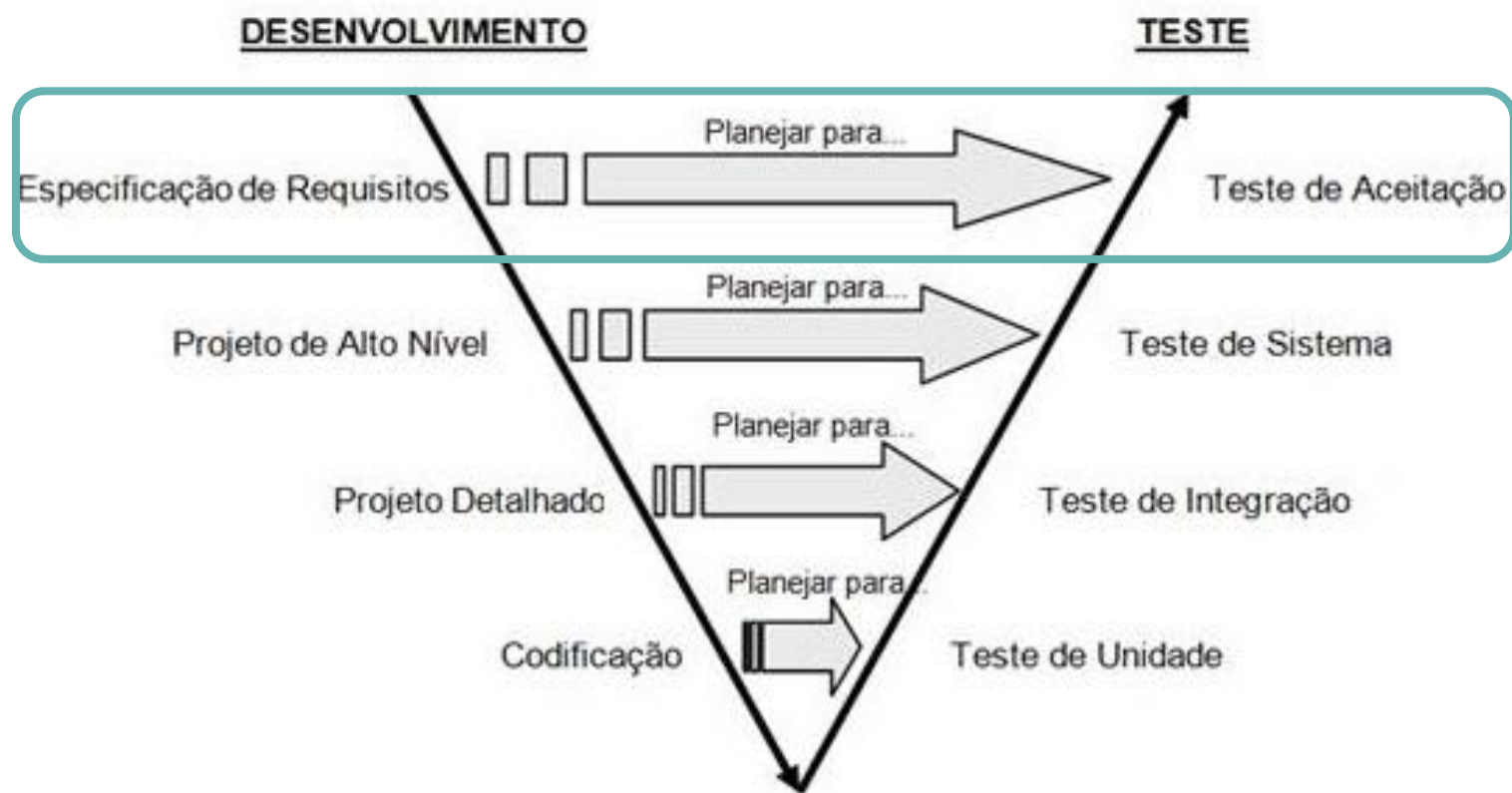
- Testa o sistema como um todo, simulando o ambiente de produção para verificar se atende aos requisitos especificados antes de sua implantação.
- Quem realiza é o time de QA
- Envolve testar:
  - Recuperação: tolerância a falhas
  - Segurança: mecanismos de proteção
  - Estresse: leva o sistema ao limite dos recursos (quantidade, frequência, volume)
  - Desempenho: verifica como o sistema se sai quanto todos os componentes estão integrados





# TESTE DE ACEITAÇÃO

- Testa se o sistema funciona de acordo com as necessidades do usuário.
- Quem realiza é o usuário final/cliente
- Testes alfa e beta:
  - Alfa: na presença do desenvolvedor
  - Beta: no ambiente real (ainda que não para todos)







# TESTES POR NÍVEL

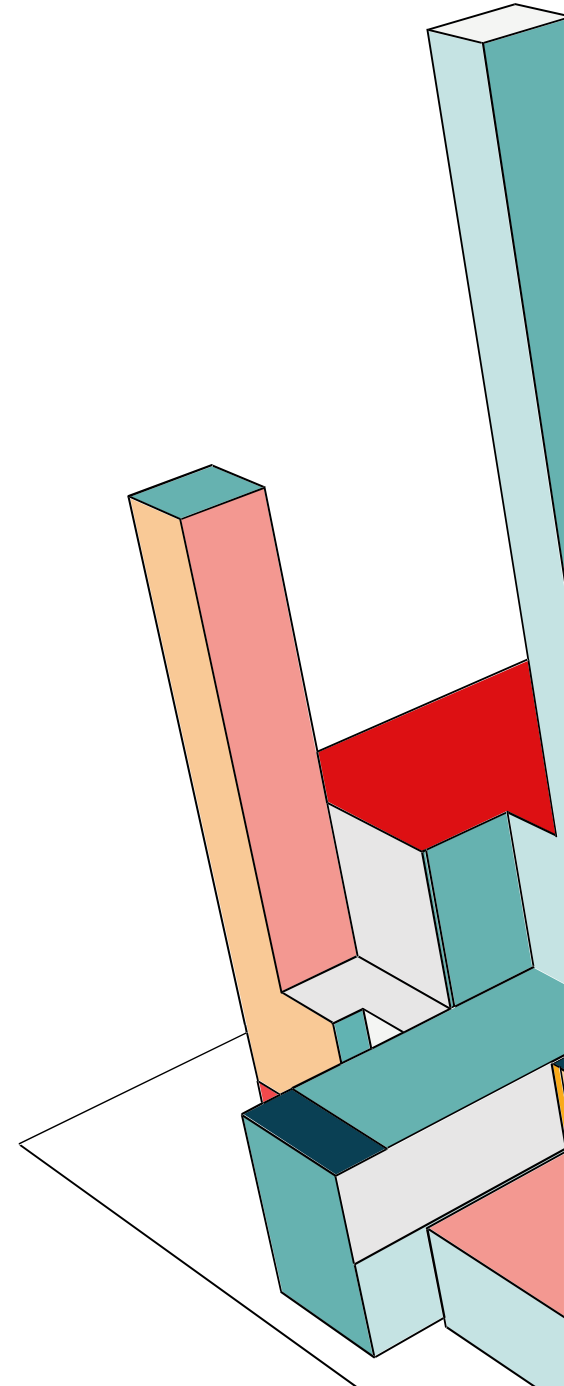
É importante observar que cada nível é mais abrangente que o anterior.

Nível	O que testa	Quem realiza	Exemplo
Unidade	Pequenas funções/métodos	Desenvolvedor	Testar cálculo de desconto
Integração	Interação entre módulos	Desenvolvedor/testador	Verificar comunicação entre login e banco
Sistema	O sistema como um todo	Equipe de QA	Validar cadastro completo de cliente
Aceitação	Validação final com o cliente	Usuário final/cliente	Verificar se o sistema atende aos requisitos

# TESTES QUANTO AO OBJETIVO

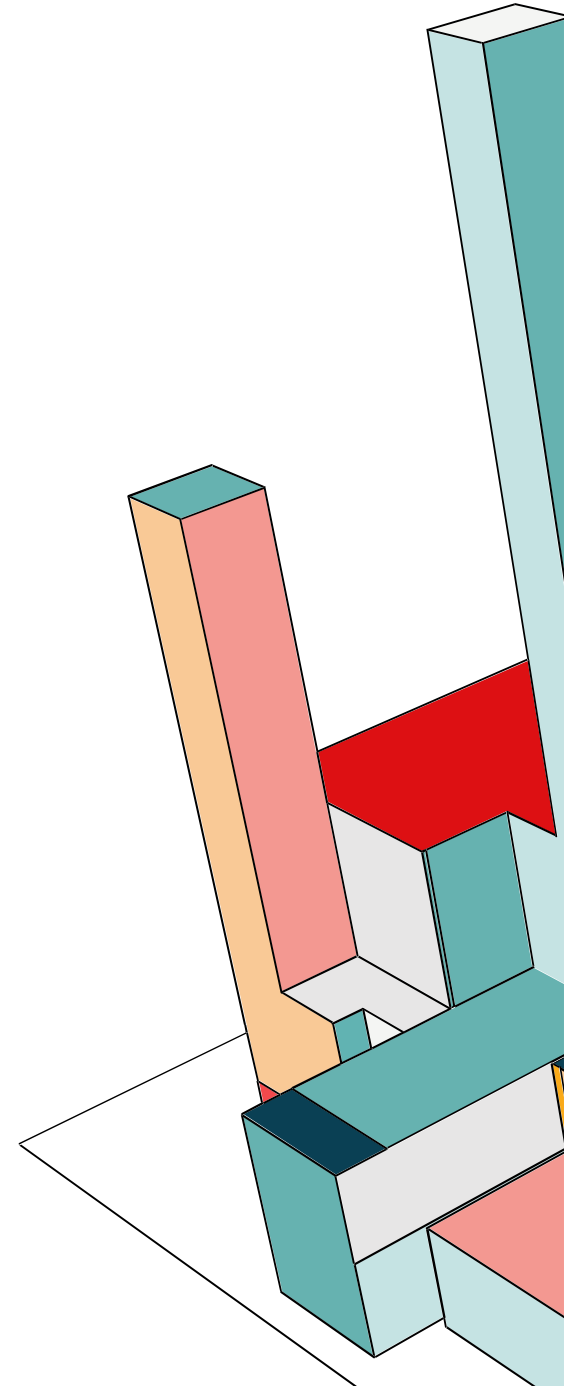
Os testes também podem ser classificados conforme o **objetivo** da verificação.

- Testes funcionais: avaliam recursos e comportamentos específicos do software
- Testes não funcionais: avaliam a performance e a eficácia do software sob condições diversas



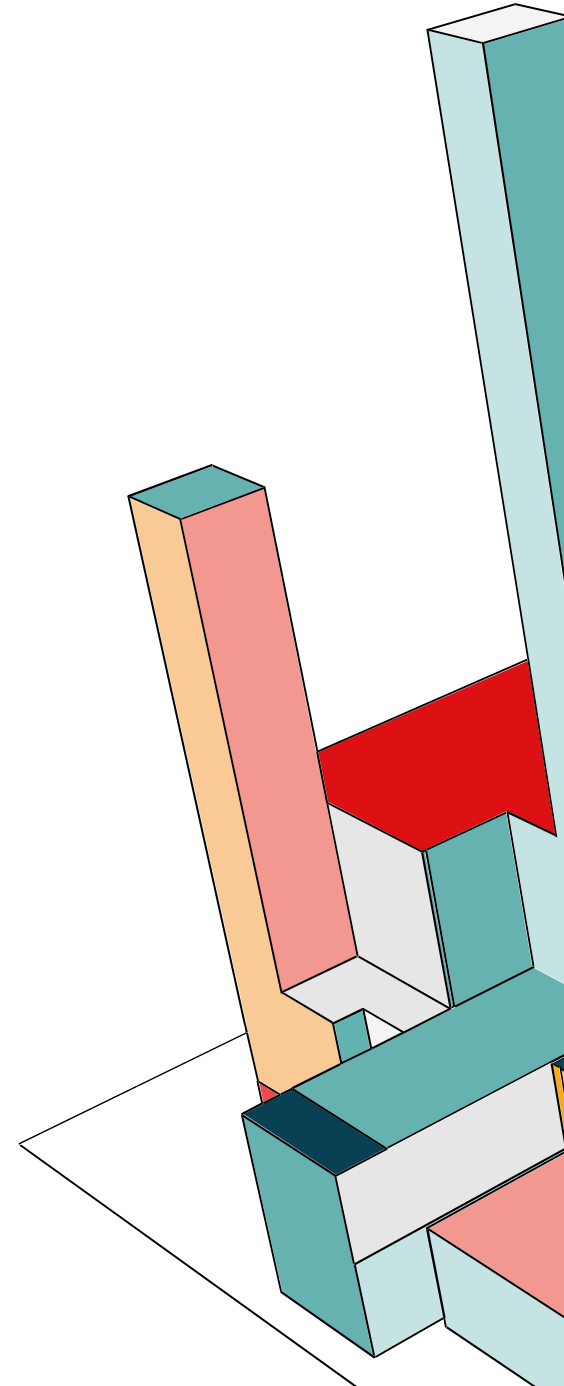
# TESTES FUNCIONAIS

- Verificam a conformidade do software com os requisitos funcionais especificados.
- Utiliza casos de teste baseados em especificações para validar funcionalidades específicas.
- Não verificam os estados intermediários do sistema ao executar uma ação.
- Exemplos comuns: testes de aceitação, testes de sistema.



# TESTES NÃO FUNCIONAIS

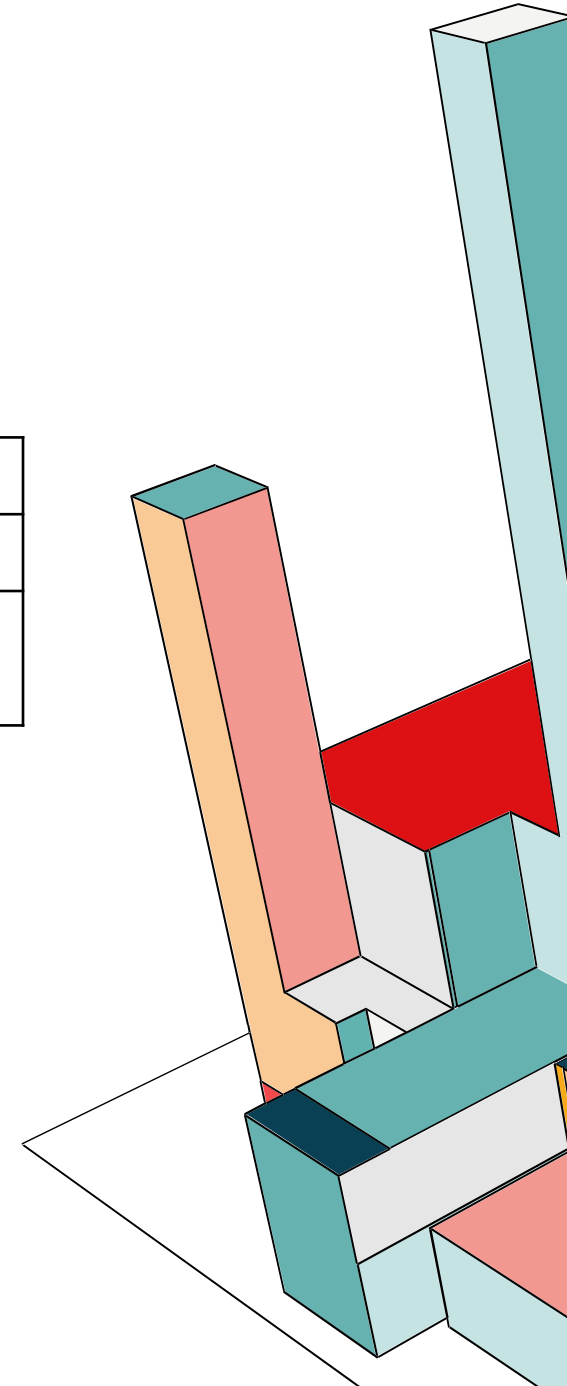
- Avaliam atributos de qualidade do software como desempenho, segurança, usabilidade e compatibilidade.
- Essenciais para garantir a satisfação do usuário, a estabilidade e a segurança operacional do software.
- Exemplos comuns: teste de carga, teste de segurança, teste de usabilidade e teste de compatibilidade.



# TESTES QUANTO AO OBJETIVO

Tipo	Foco	Exemplo
Funcionais	O que o software faz	Testar se o login aceita senha incorreta
Não Funcionais	Como o software se comporta	Medir o tempo de resposta do login

- **Testes funcionais:** garantem que o sistema realiza as operações esperadas.
- **Testes não funcionais:** verificam desempenho, usabilidade, segurança e confiabilidade.



# TESTES MANUAIS E AUTOMATIZADOS



Outra forma importante de classificar os testes é quanto à forma de execução

- Os **testes manuais** são realizados por pessoas, verificando manualmente o comportamento do software.
- Os **testes automatizados** são realizados por scripts e ferramentas de automação.



## TESTES MANUAIS

- Ideal para cenários com requisitos visuais, como layouts de interface de usuário e experiência do usuário.
- Benefícios como flexibilidade e insights humanos diretos.
- Limitações como consumo de tempo e inconsistência nos resultados.





## TESTES AUTOMATIZADOS

- Ideal para tarefas repetitivas e extensas que seriam impraticáveis ou muito demoradas para serem feitas manualmente.
- Benefícios como velocidade, precisão e eficiência.
- Limitações como custo inicial alto e necessidade de manutenção dos scripts de teste.

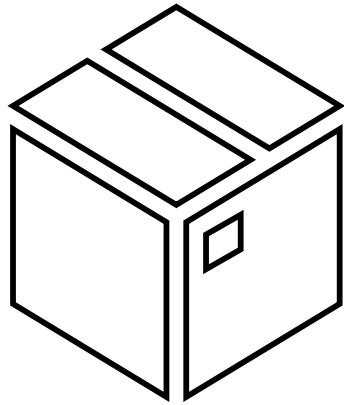
# TESTES MANUAIS E AUTOMATIZADOS

- Os **testes manuais** são úteis para avaliar experiência do usuário e aspectos subjetivos.
- Os **automatizados** aceleram o processo e garantem repetibilidade.

Tipo	Características	Quando usar	Ferramentas comuns
Manuais	Executados por pessoas, seguindo roteiros	Testes exploratórios, usabilidade, aceitação	Planilhas, roteiros, checklist
Automatizados	Executados por scripts e ferramentas	Testes repetitivos, regressão, performance	Selenium, JUnit, PyTest, Cypress...

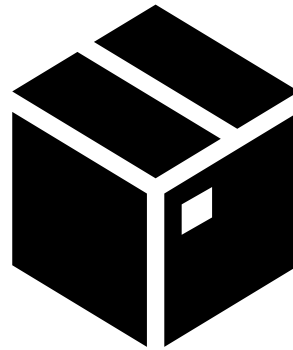
# ABORDAGEM DE TESTES

As **abordagens** indicam o quanto o testador conhece sobre o funcionamento interno do software. São três:

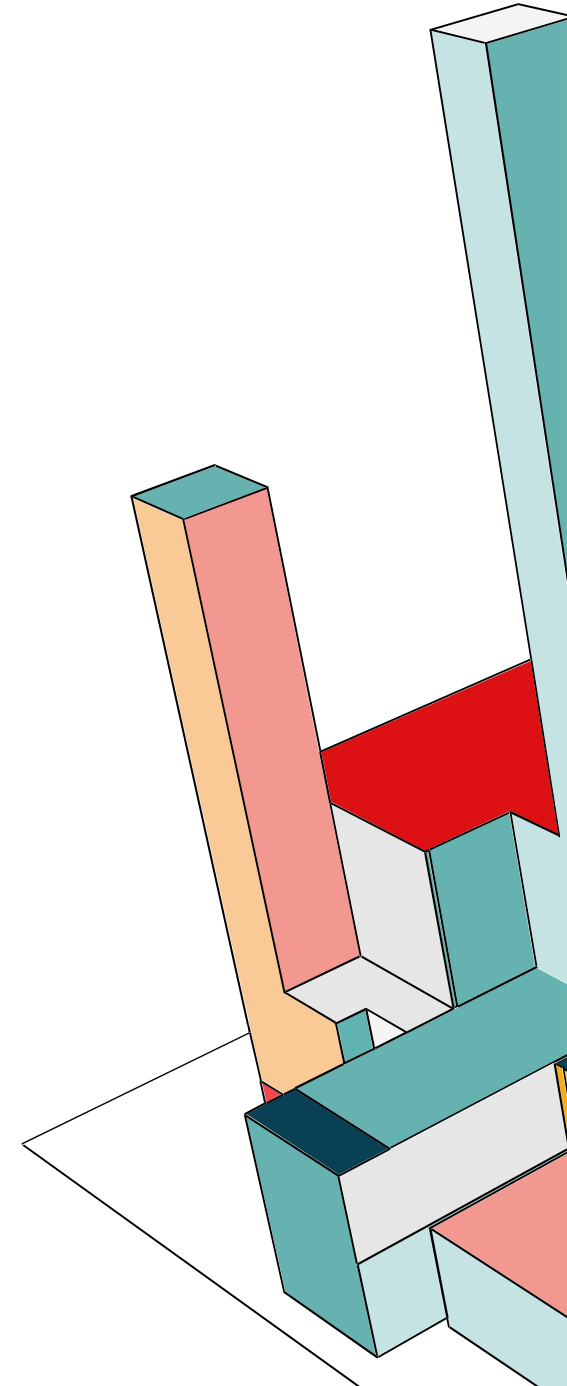


**Caixa Branca**

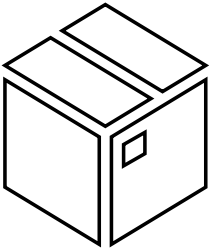
**Caixa Preta**



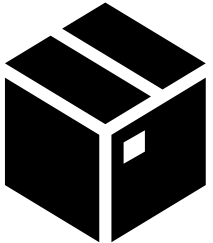
**Caixa Cinza**



# ABORDAGEM DE TESTES



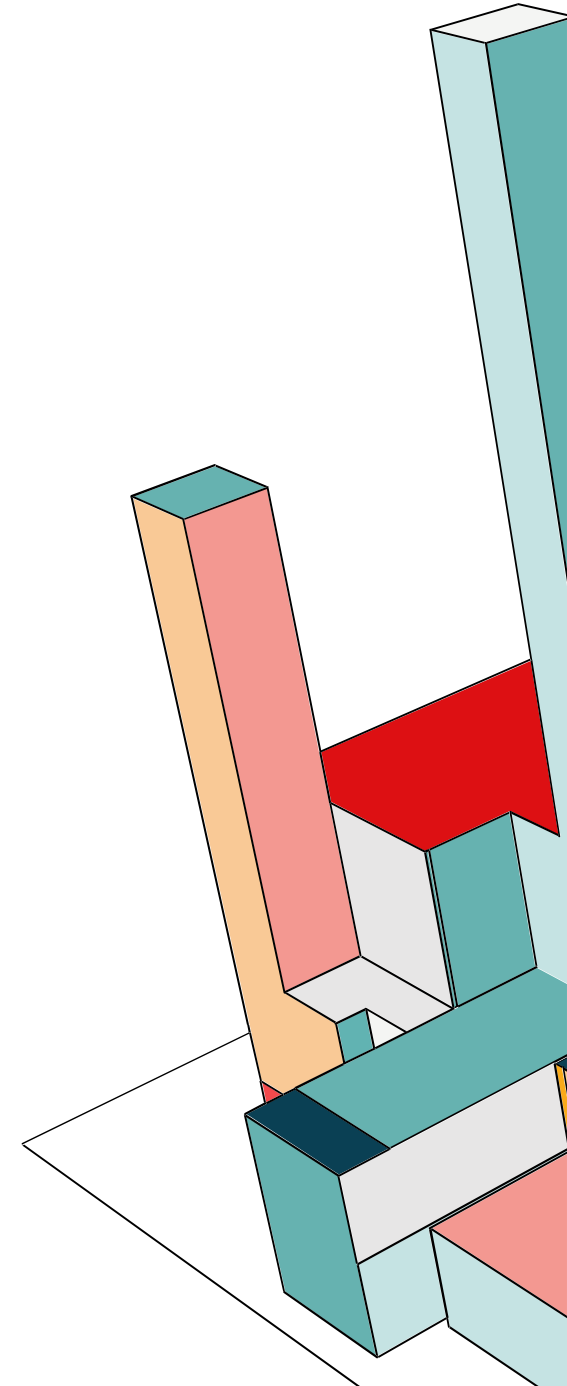
- **Teste de caixa branca:** envolve testes com base no conhecimento da estrutura interna, lógica e funções do software em teste.



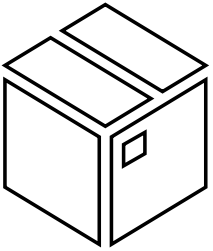
- **Teste de caixa-preta:** nesse tipo de teste, o testador não tem nenhuma informação sobre o funcionamento interno do sistema de software.



- **Testes de caixa cinza:** uma combinação de testes de caixa branca e caixa preta. O teste é realizado com conhecimento parcial do código interno e estruturas, combinado com o teste de funcionalidades externas.



# ABORDAGEM DE TESTES



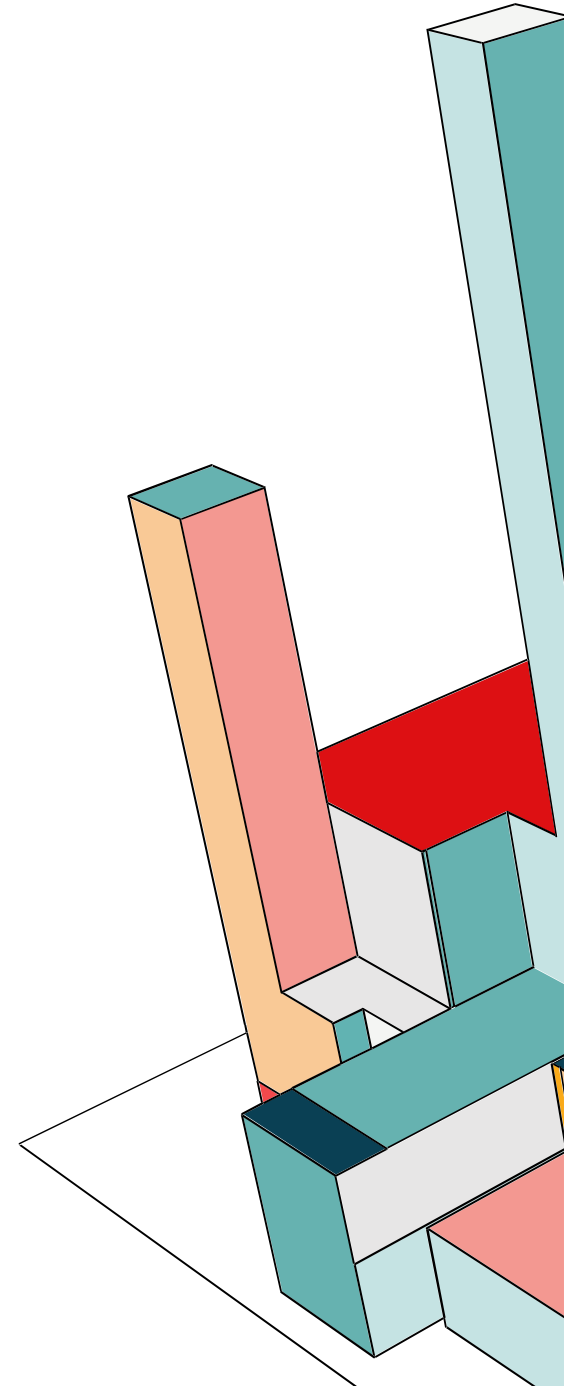
- **Teste de caixa branca:** Utilizados principalmente durante as fases de desenvolvimento e unidade de teste para validar o caminho lógico, estruturas de dados e fluxo de controle dentro do programa.



- **Teste de caixa-preta:** Mais comumente utilizados em fases de teste de aceitação, teste de sistema e teste de regressão para validar a funcionalidade e o comportamento externo do software.

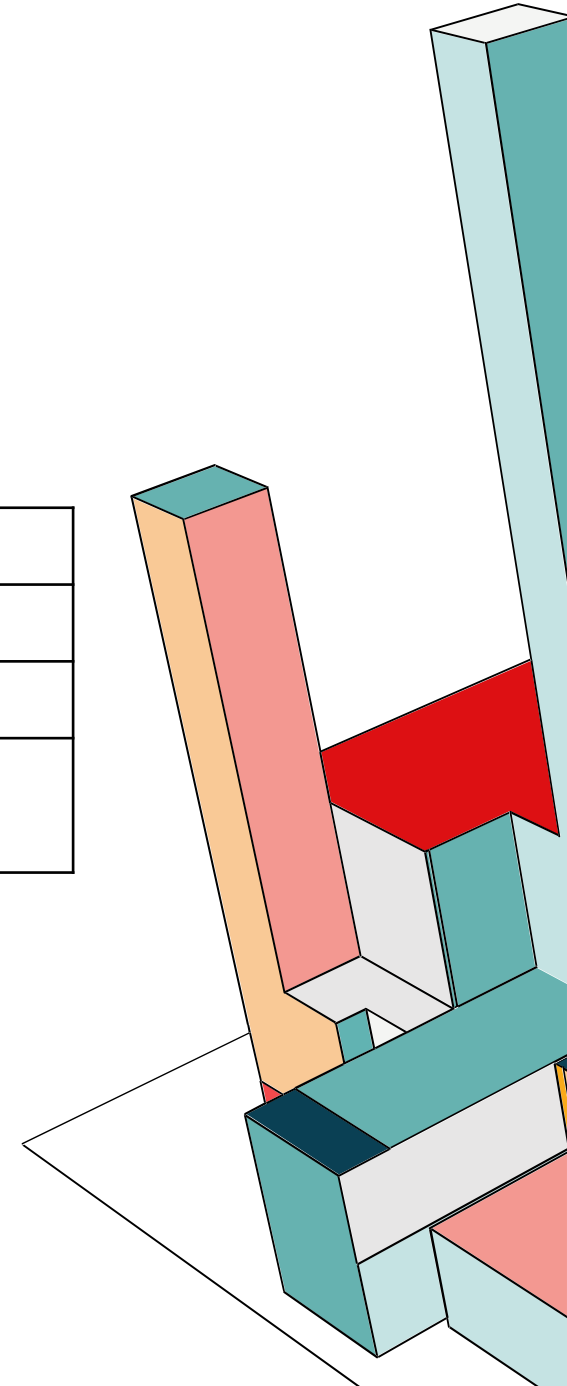


- **Testes de caixa cinza:** Aplicáveis durante a integração de sistemas ou em uma fase de teste de interface onde é necessário conhecer o suficiente sobre o código para testar as interfaces e fluxos de dados entre os módulos.



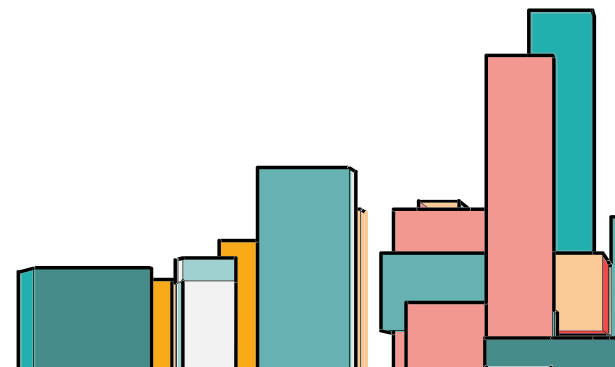
# ABORDAGEM DE TESTES

Tipo	Conhecimento do código	Exemplo
Caixa Branca	Total	Testar lógica e fluxos internos
Caixa Preta	Nenhum	Testar entradas e saídas
Caixa Cinza	Parcial	Testar APIs com conhecimento limitado do sistema



# ATIVIDADE

- Quais testes seriam feitos para:
- Cadastro de Usuário
  - Um sistema web permite o cadastro de novos usuários. O formulário pede nome, e-mail, senha e confirmação de senha. O backend deve verificar se o e-mail já está em uso, se a senha atende aos critérios mínimos e salvar os dados no banco.





# ATIVIDADE

- Quais testes seriam feitos para:
- **Login e Autenticação**
  - Um aplicativo mobile exige login com e-mail e senha. Caso o usuário erre a senha três vezes, o acesso deve ser bloqueado por 5 minutos. Também há a opção de login com Google.



# ATIVIDADE

- Quais testes seriam feitos para:
- Cálculo de Frete
  - Um e-commerce calcula automaticamente o valor do frete conforme o CEP e o peso do produto. O cálculo deve usar uma API externa de transportadora.

