

# **Testes automatizados**

O que são, onde vivem, o que comem?

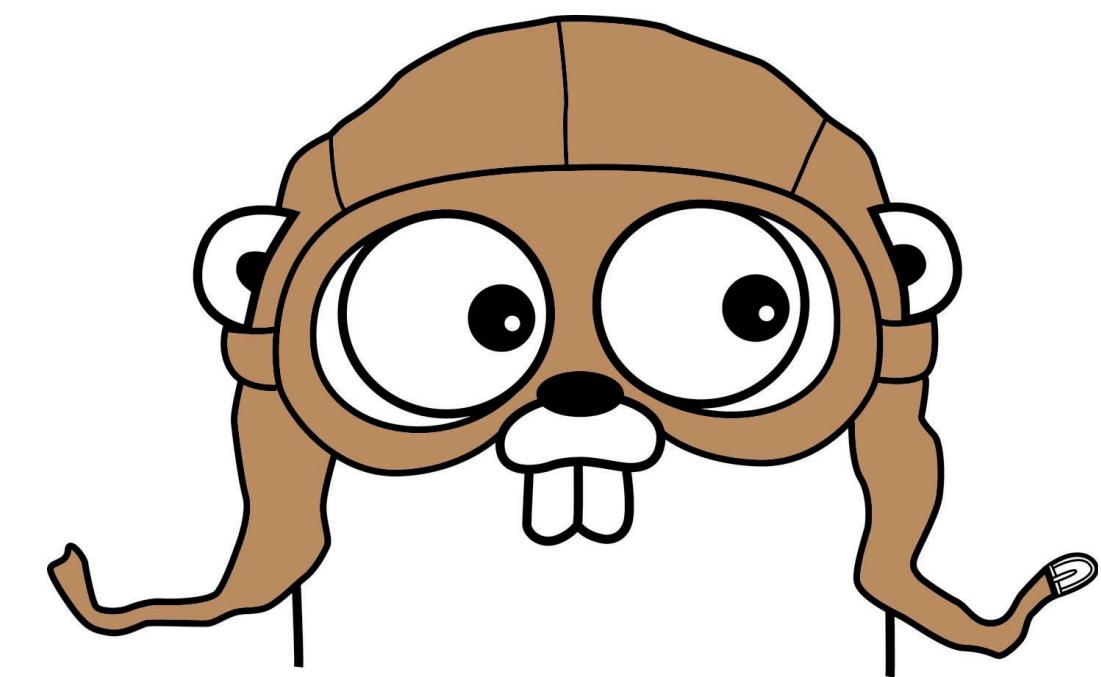
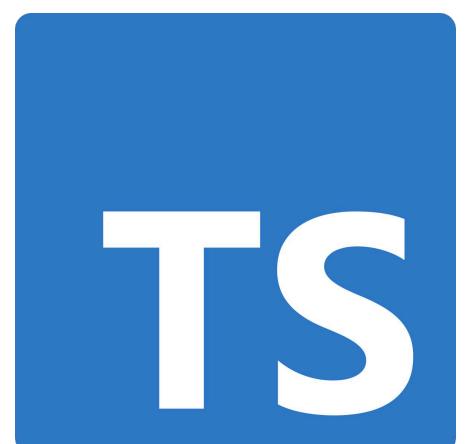
Joel Vítor Torres



jovi

joevtap · he/him

backend software engineer - computer  
information systems student -  
knowledge seeker



# **Testes automatizados**

**Testes au**





**Testes au**

s automatizados

saut dos



**Testes não servem para provar  
que o software funciona!**

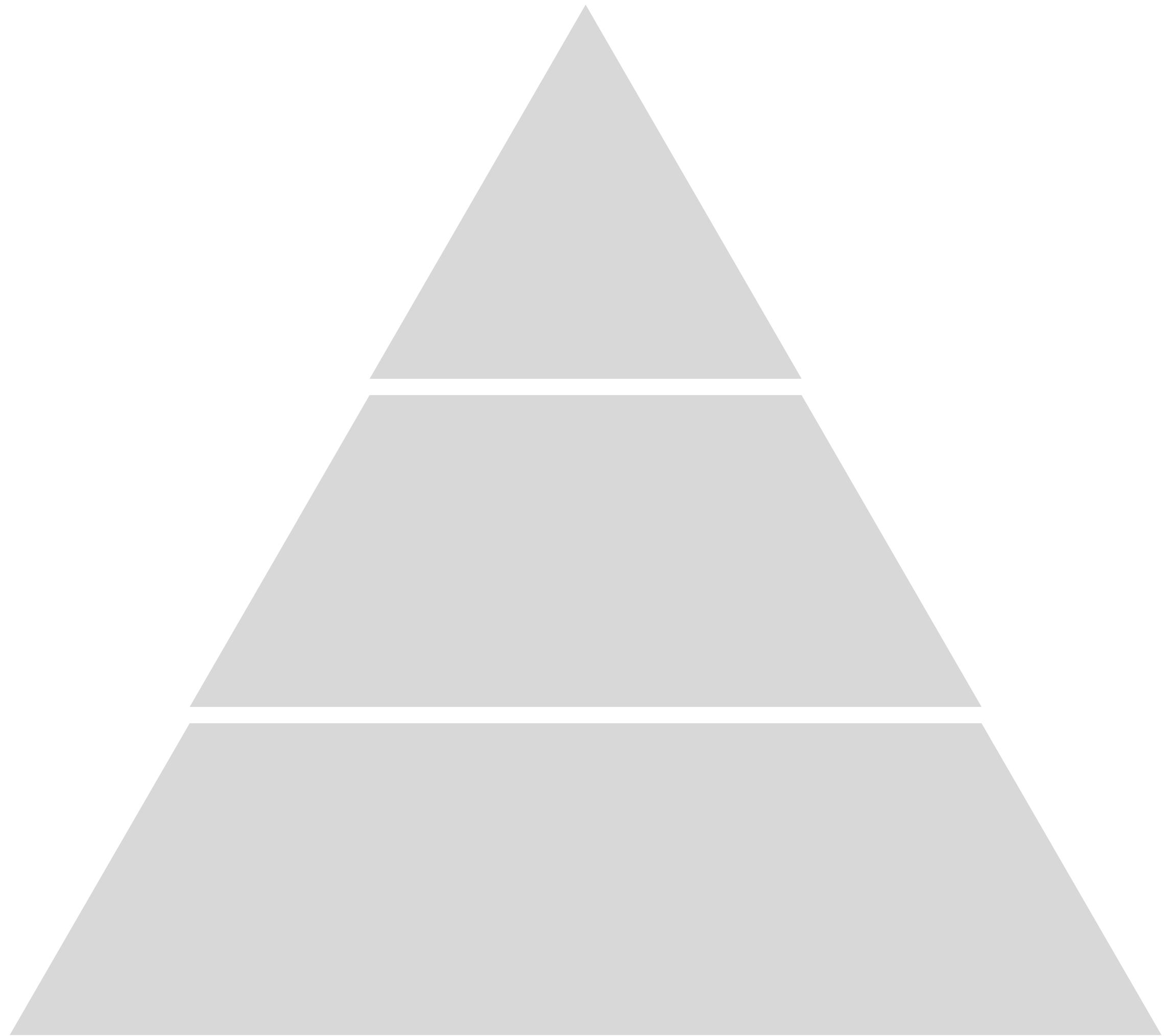
**Os usuários do seu software  
assumem que ele funciona  
corretamente**

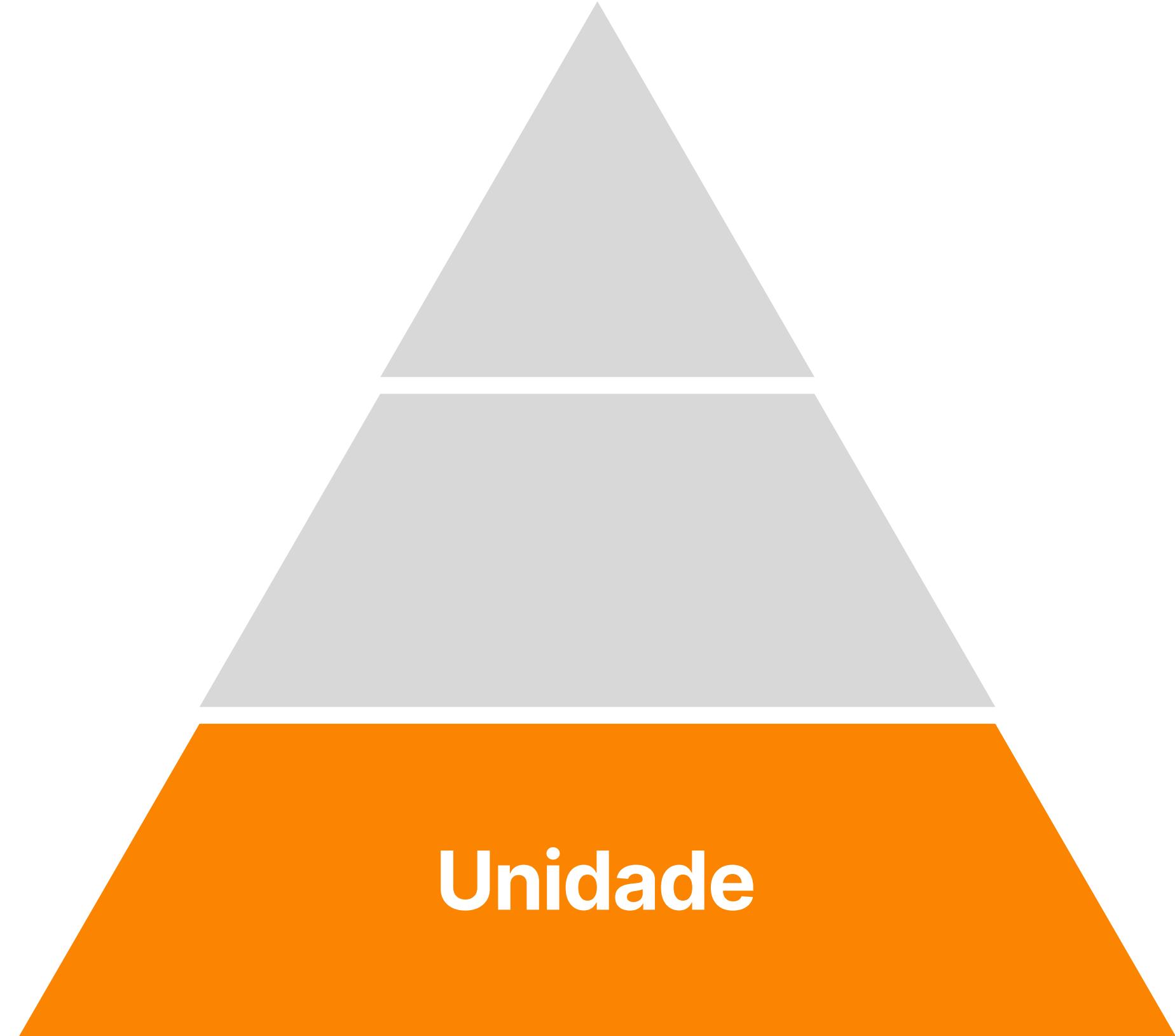
**Não é preciso provar nada!**

**Teste de software é uma disciplina  
de engenharia para garantir  
qualidade**

**A vida do desenvolvedor fica mais  
fácil com testes automatizados,  
porque fica mais fácil garantir a  
qualidade que já esperada**







Testam componentes de um  
sistema **isoladamente**



Testam a **integração** de componentes de um sistema

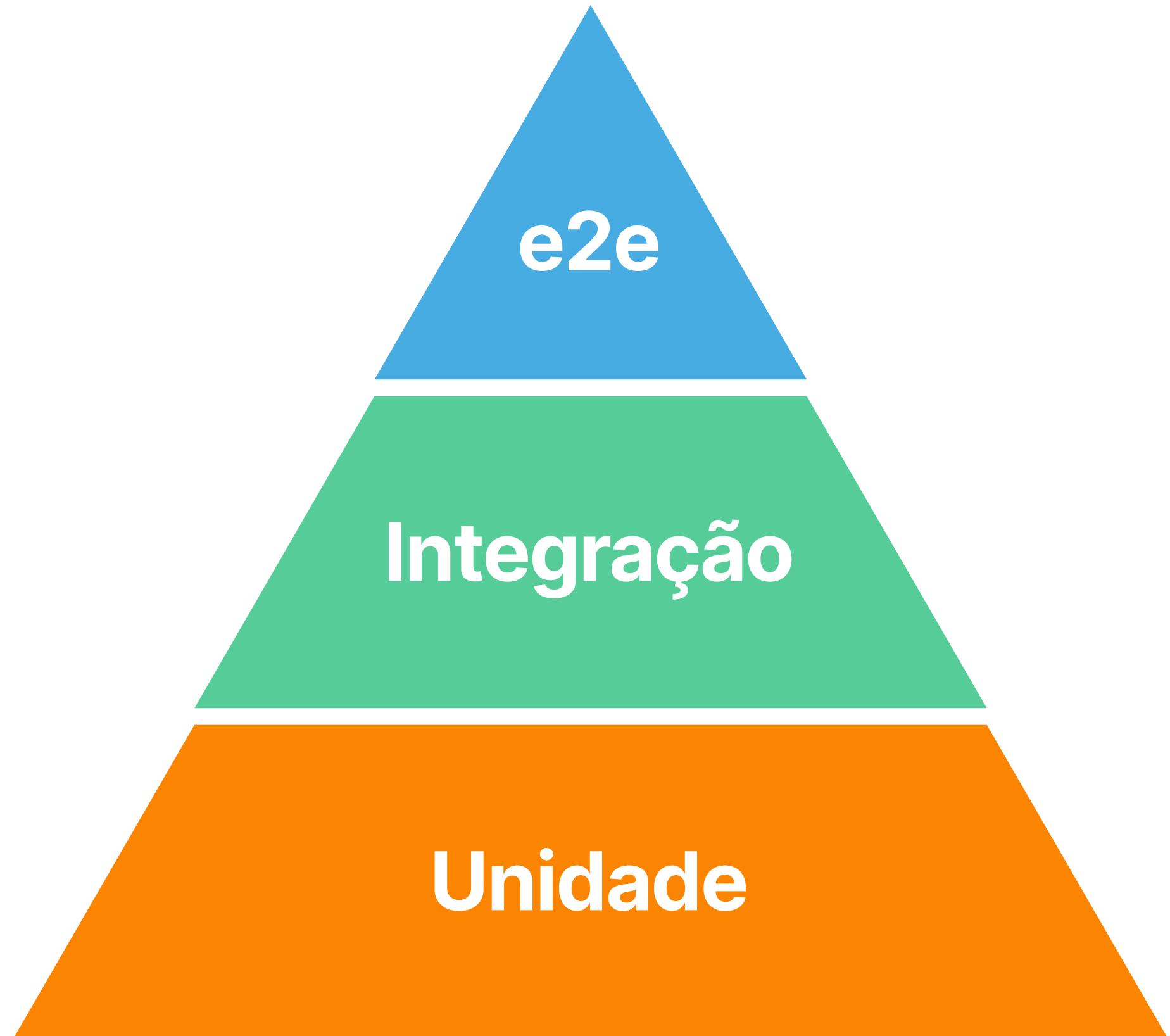
Testam componentes de um sistema **isoladamente**



Testam o **sistema** de ponta a ponta  
(end-to-end)

Testam a **integração** de  
componentes de um sistema

Testam componentes de um  
sistema **isoladamente**



**Pirâmide de Testes**

# Testes de unidade

Testam componentes de um sistema  
**isoladamente**

```
1 const shop = new Wallet();
2 const wallet = new Wallet();
3
4 const [newWallet, newShop] = wallet.transfer(shop, 10000);
```

# Testes de unidade

Testam componentes de um sistema  
**isoladamente**

```
1 const shop = new Wallet();
2 const wallet = new Wallet();
3
4 const [newWallet, newShop] = wallet.transfer(shop, 10000);
```

# Testes de integração

Testam a **integração** de componentes de um sistema

```
1  function processPurchase(userId: string, shopId: string, amount:  
number) {  
2      const userWallet = walletRepo.get(userId);  
3      const shopWallet = walletRepo.get(shopId);  
4  
5      const [newUser, newShop] = userWallet.transfer(shopWallet,  
amount);  
6  
7      walletRepo.save(userId, newUser);  
8      walletRepo.save(shopId, newShop);  
9  }
```

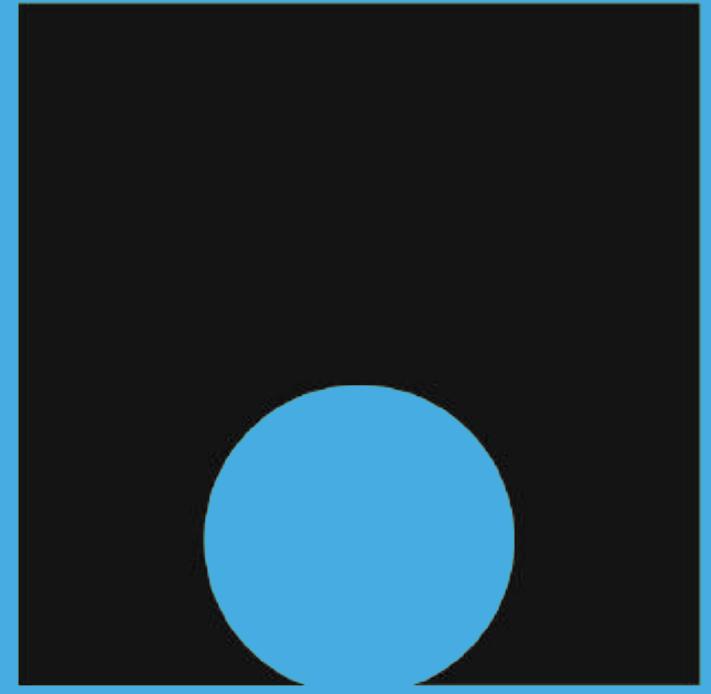
# Testes de integração

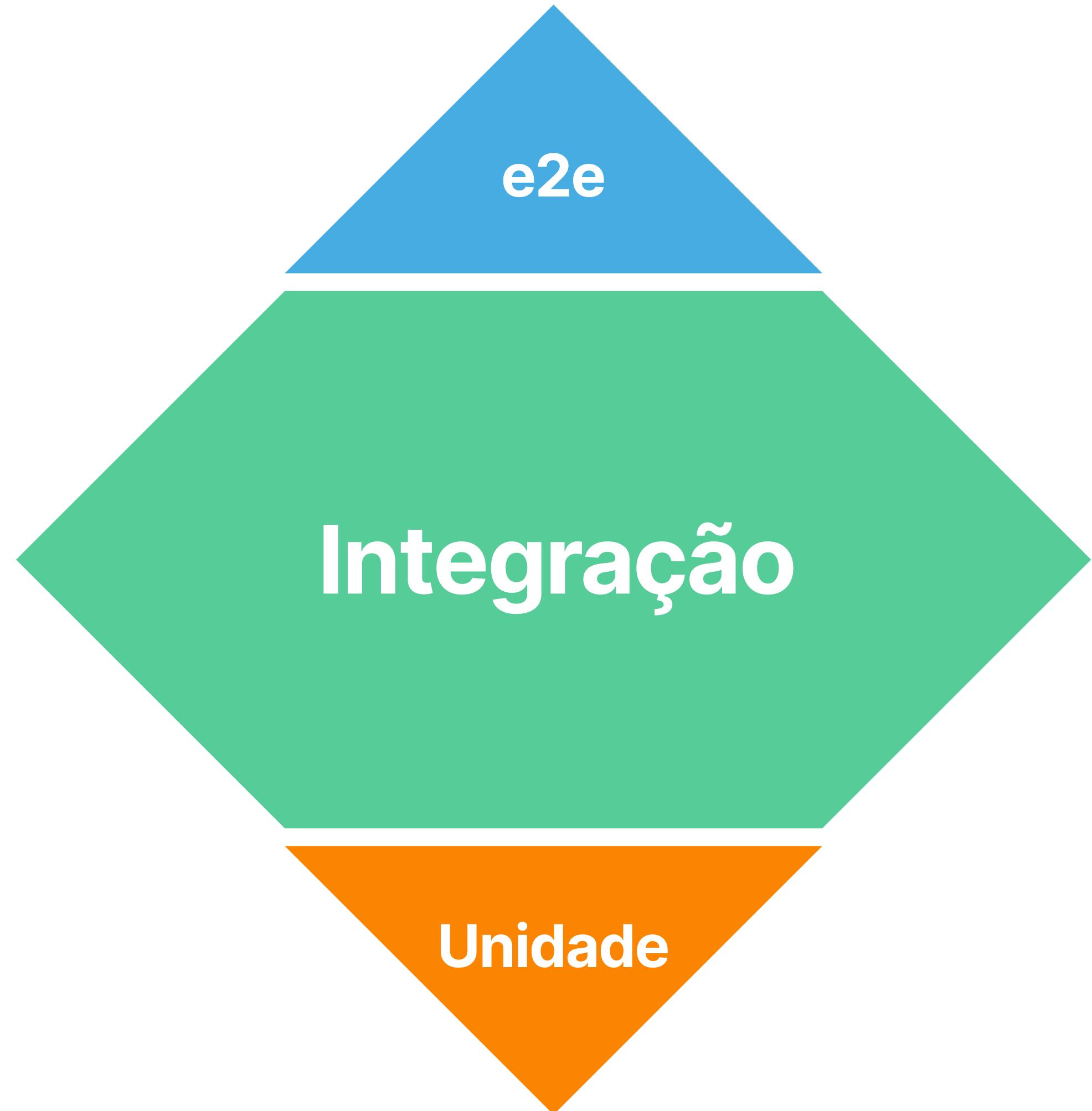
Testam a **integração** de componentes de um sistema

```
1 function processPurchase(userId: string, shopId: string, amount: number) {  
2     const userWallet = walletRepo.get(userId);  
3     const shopWallet = walletRepo.get(shopId);  
4  
5     const [newUser, newShop] = userWallet.transfer(shopWallet,  
amount);  
6  
7     walletRepo.save(userId, newUser);  
8     walletRepo.save(shopId, newShop);  
9 }
```

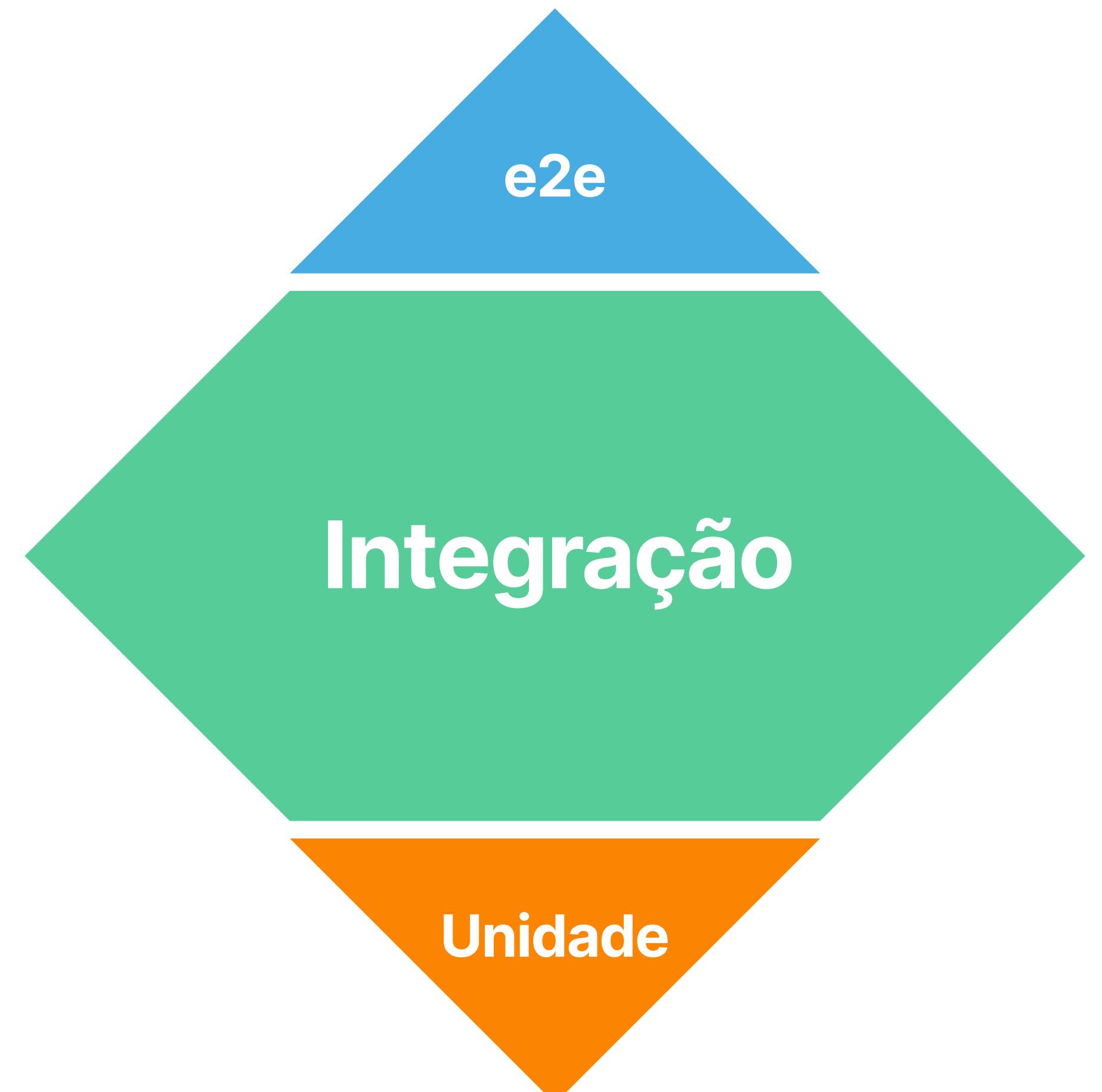
# Testes end-to-end (e2e)

Testam um **sistema como um todo**, de ponta a ponta

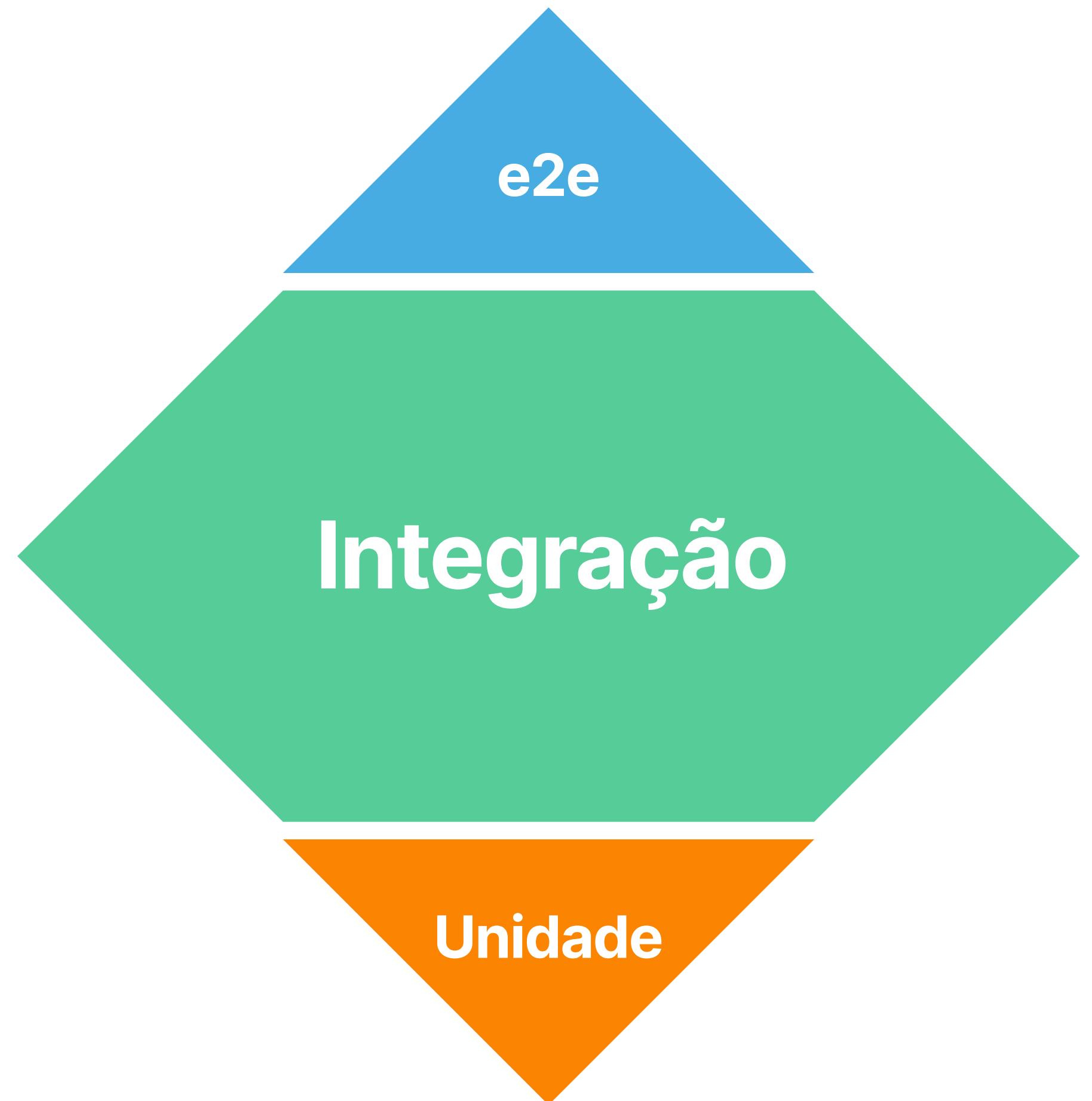




O esforço deve ser  
direcionado a escrever **mais**  
**testes de integração**



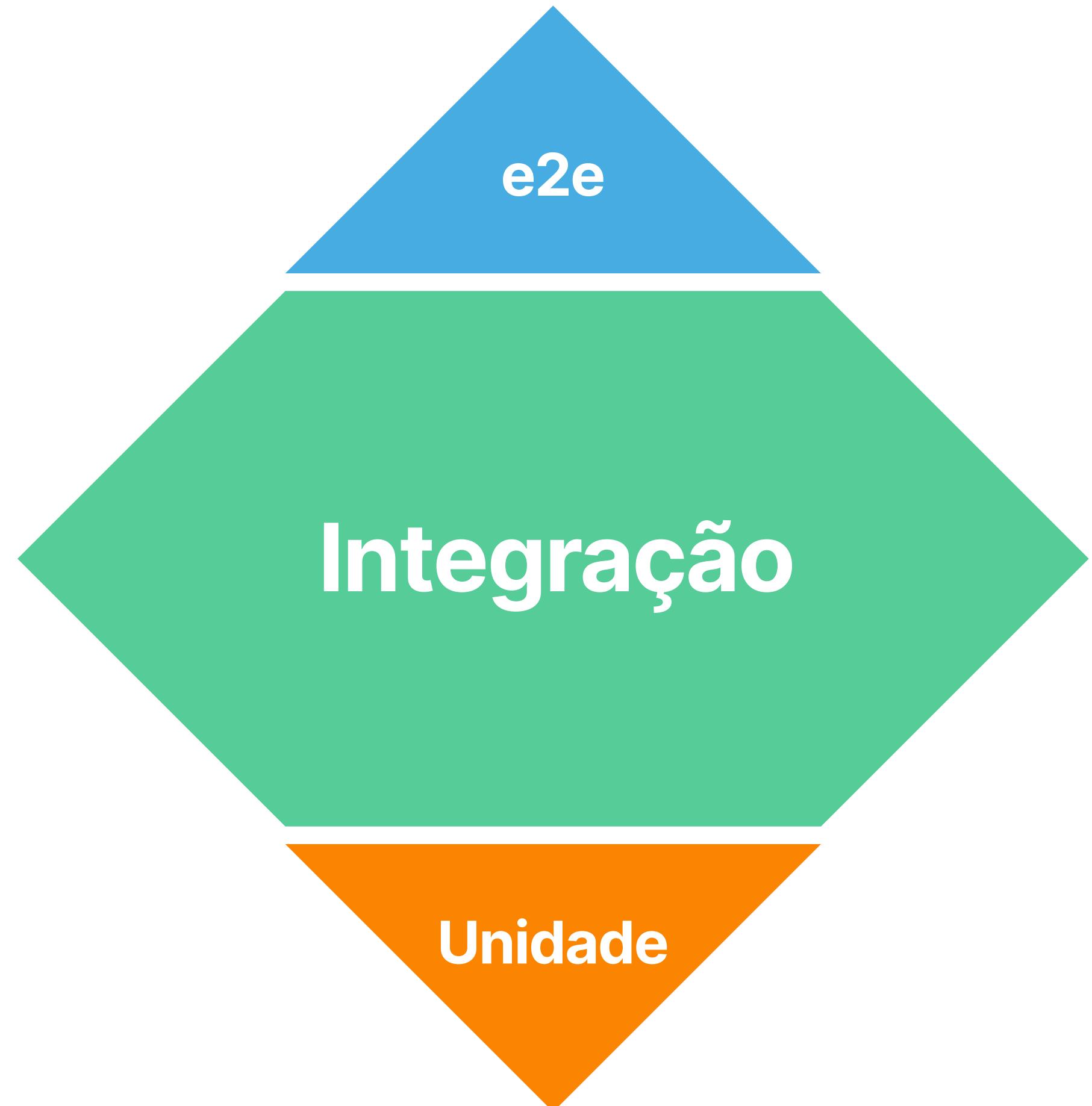
**Diamante de Testes**



Diamante de Testes

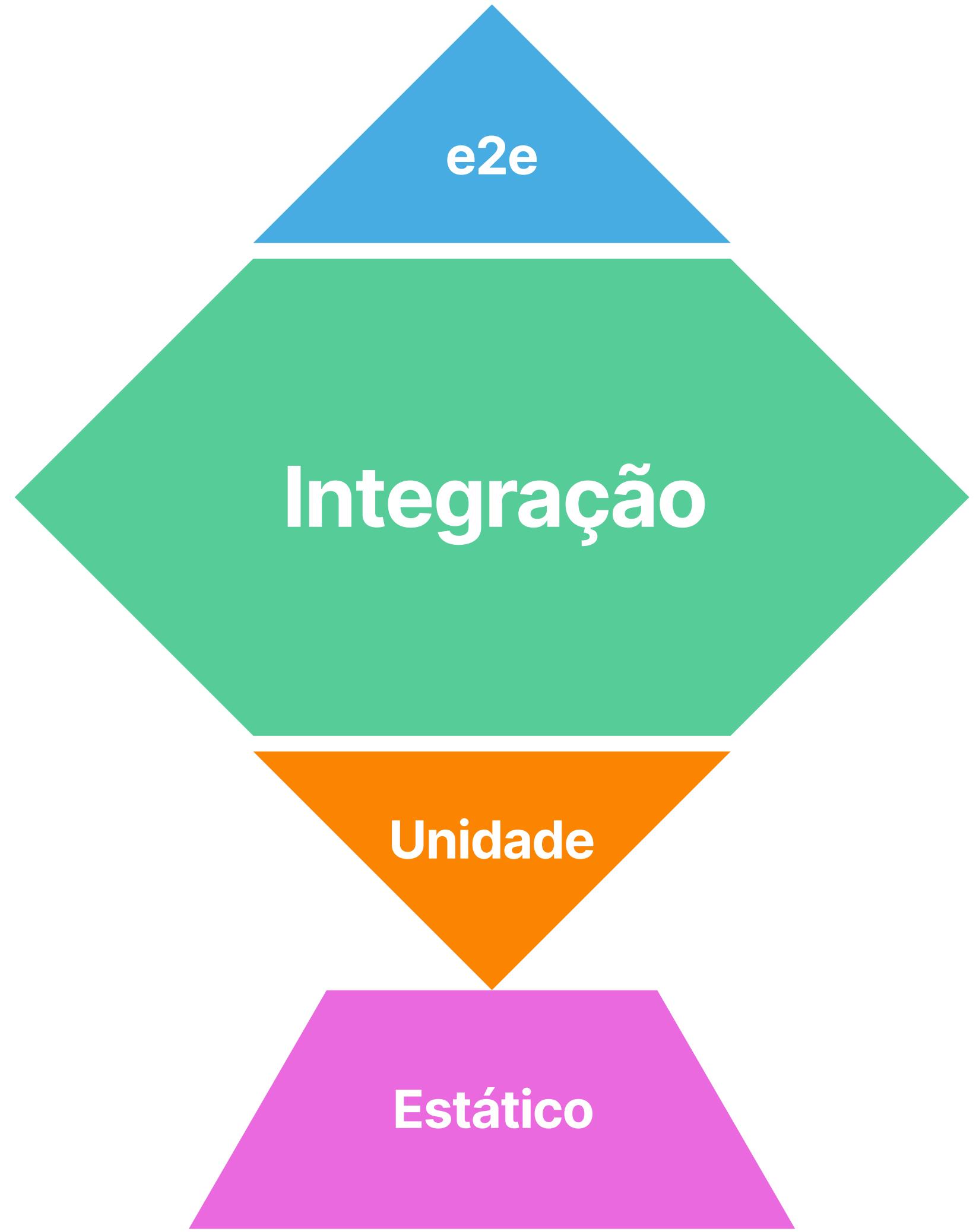


Pirâmide de Testes

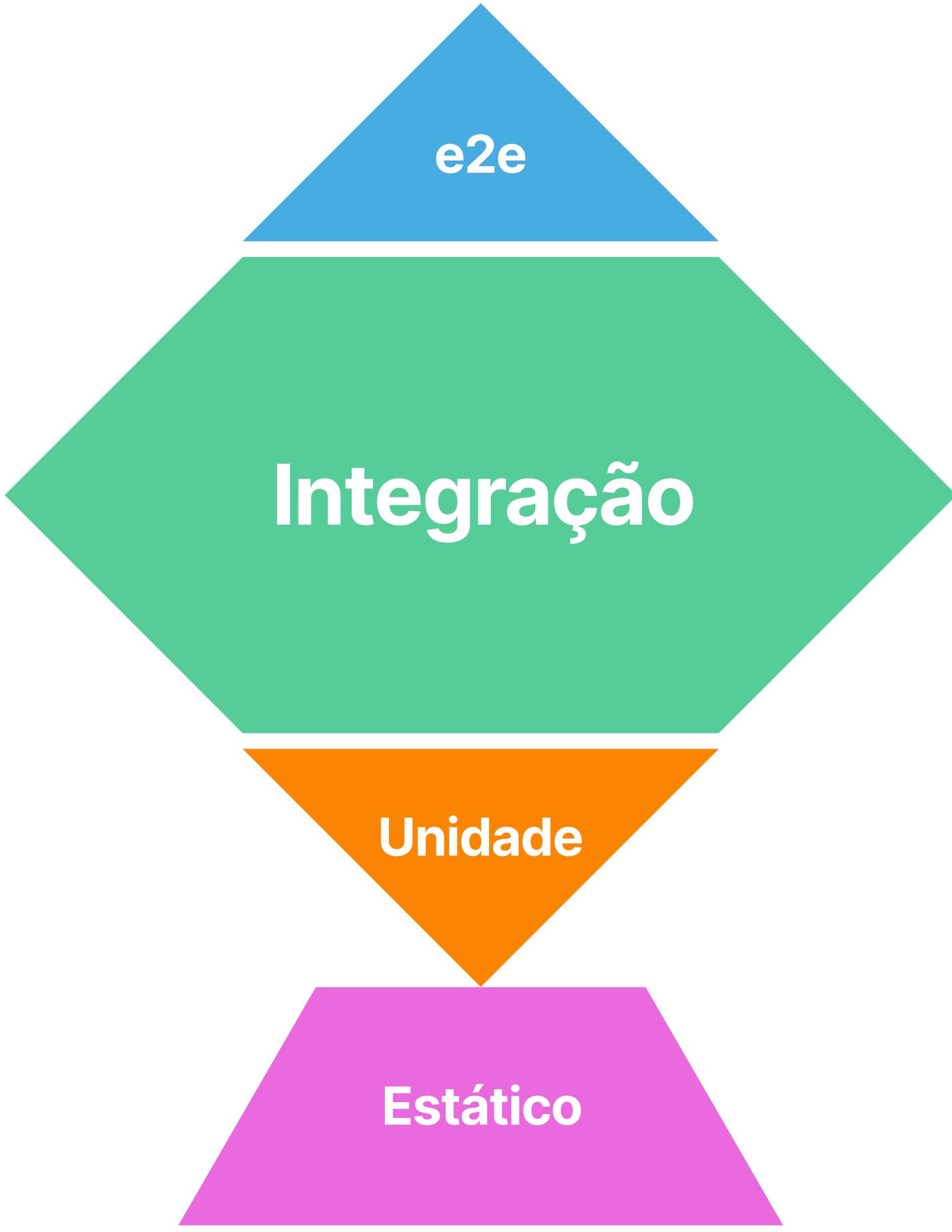


## Diamante de Testes

Testes de integração geram  
**mais confiança** a respeito do  
funcionamento do sistema, ao  
custo de um esforço um  
pouco maior para escrevê-los



Troféu de Testes



## Troféu de Testes

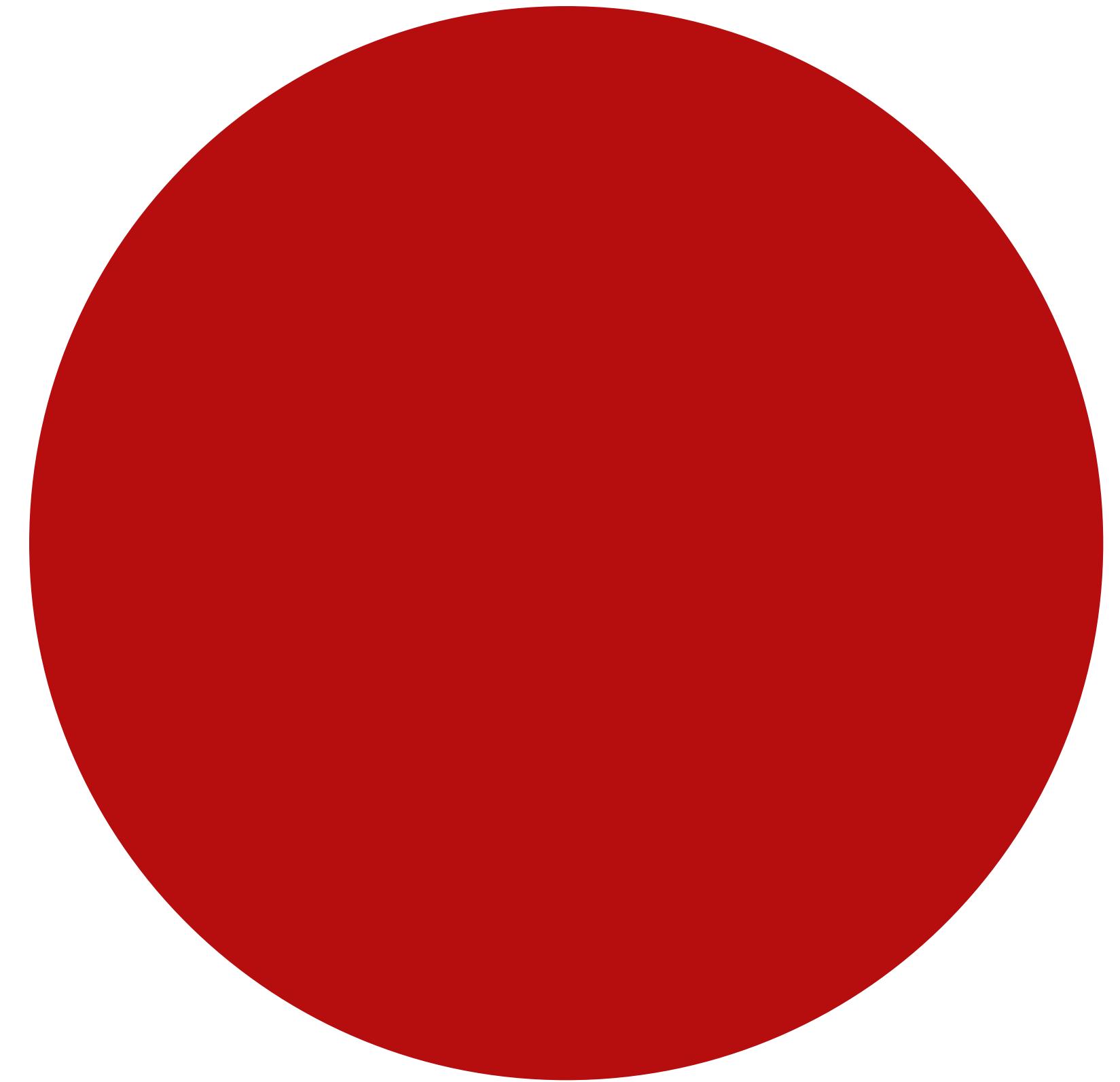
Ferramentas de **análise de código estático** como type checkers e *linters* (**TypeScript**, **ESLint**, por exemplo) encontram problemas no código antes mesmo da compilação!

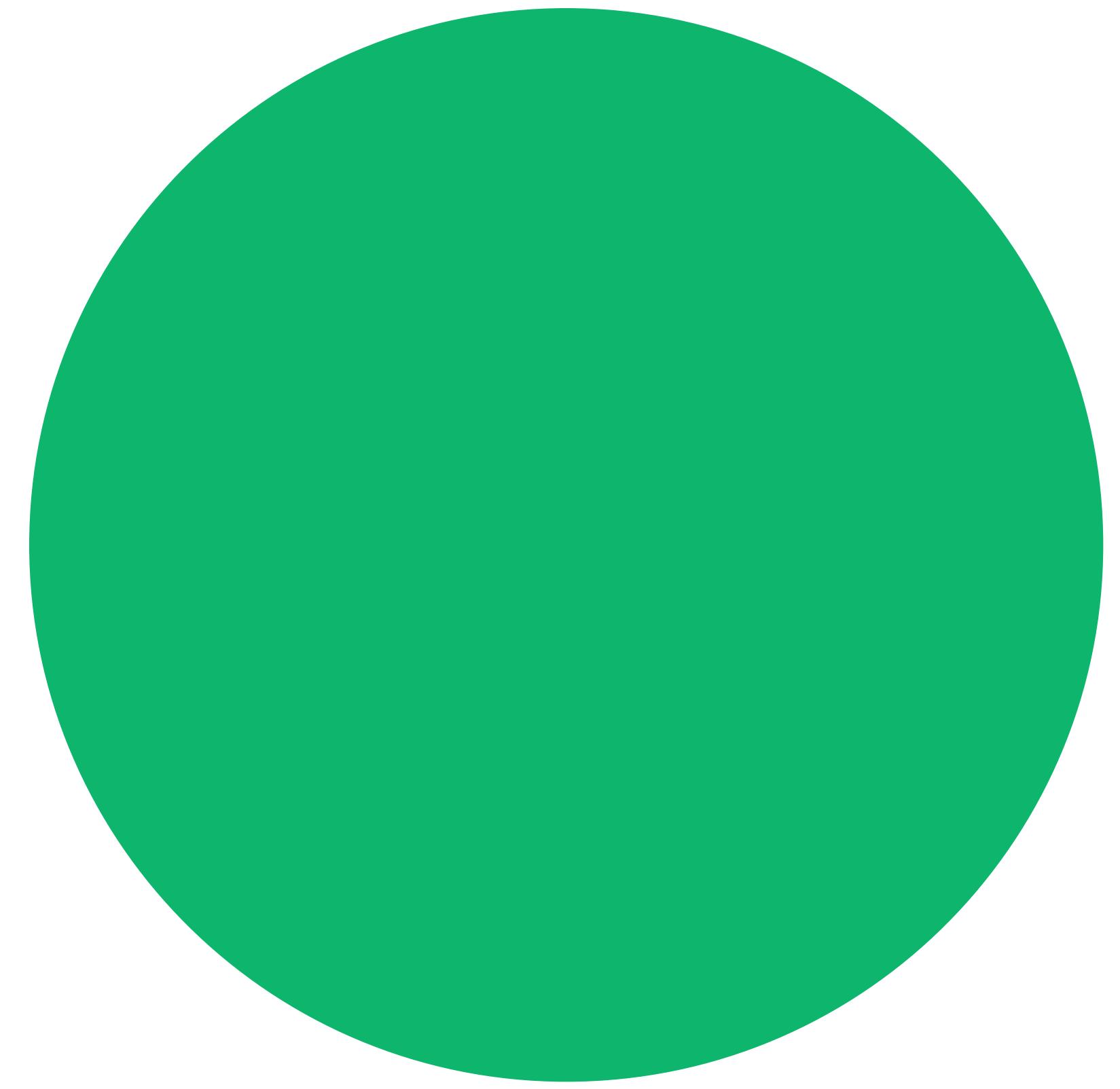
# **Test Driven Development**

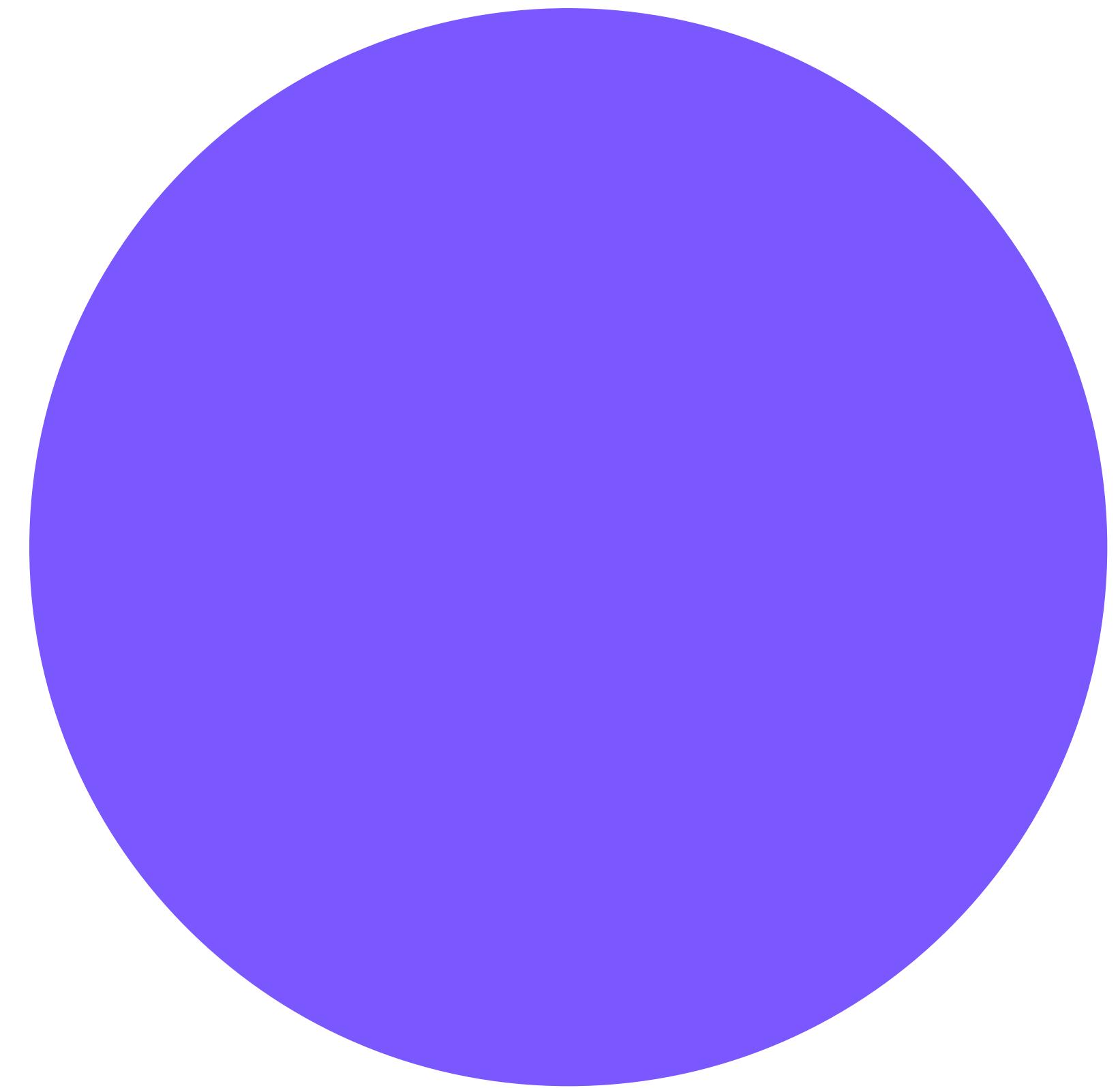
# **Test Driven Development**

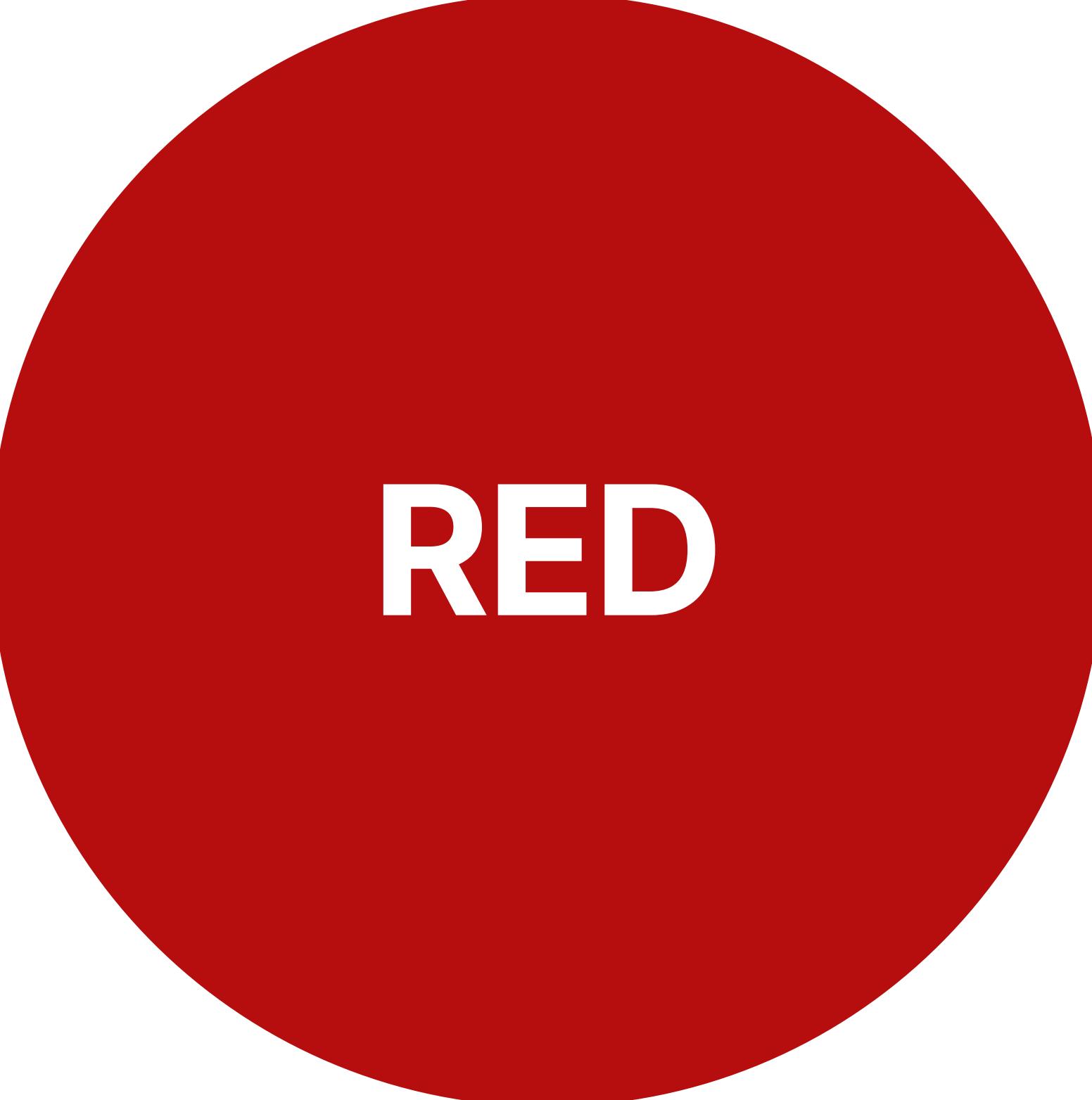
**Abordagem de desenvolvimento  
em que você escreve **testes**  
**antes** do código testado**

**É um processo iterativo e ágil  
que ajuda a pensar no código da  
forma que ele será usado**





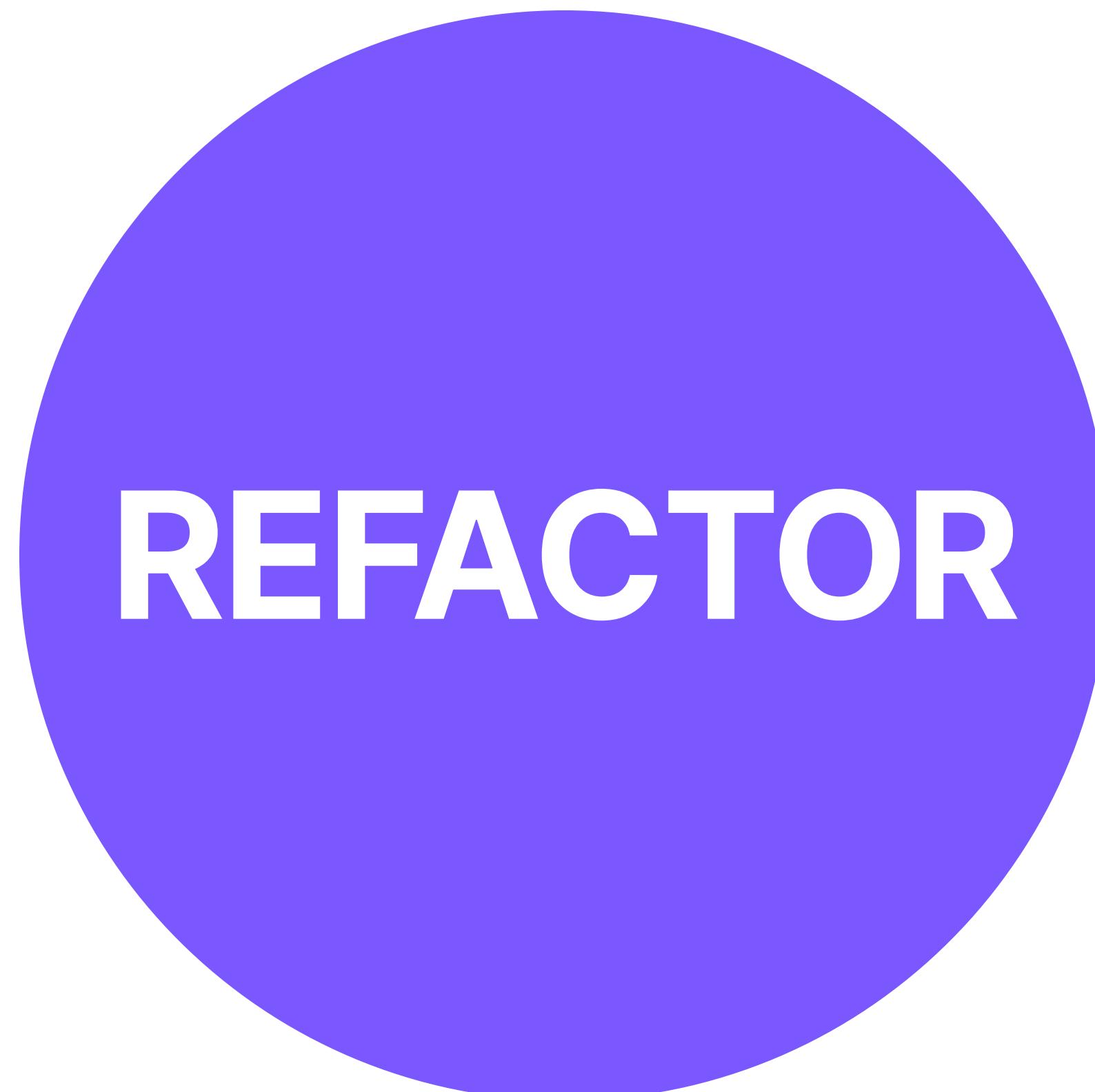




**RED**



**GREEN**



**REFACTOR**

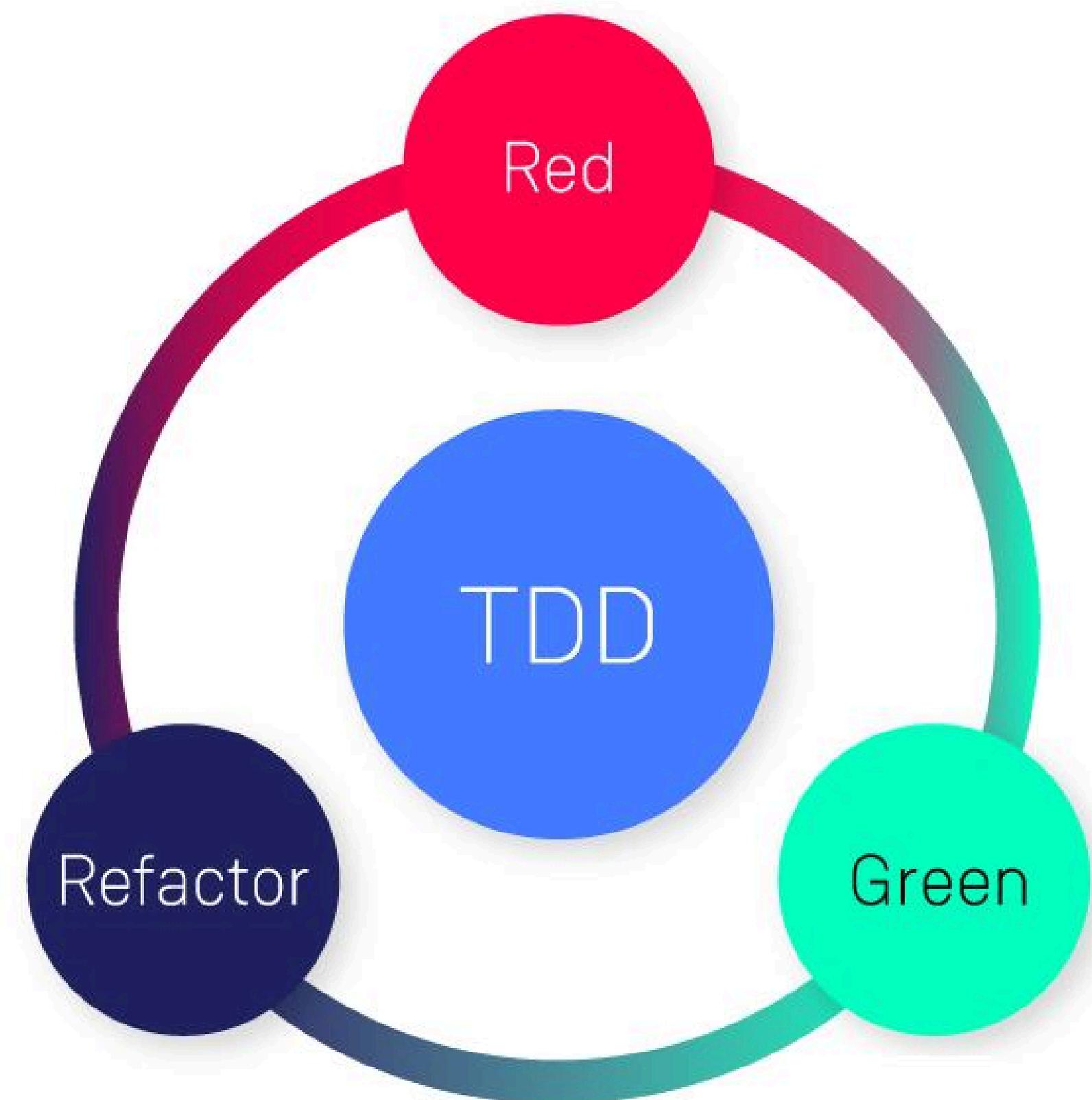
**Escreva o mínimo de código e  
testes que não passam**

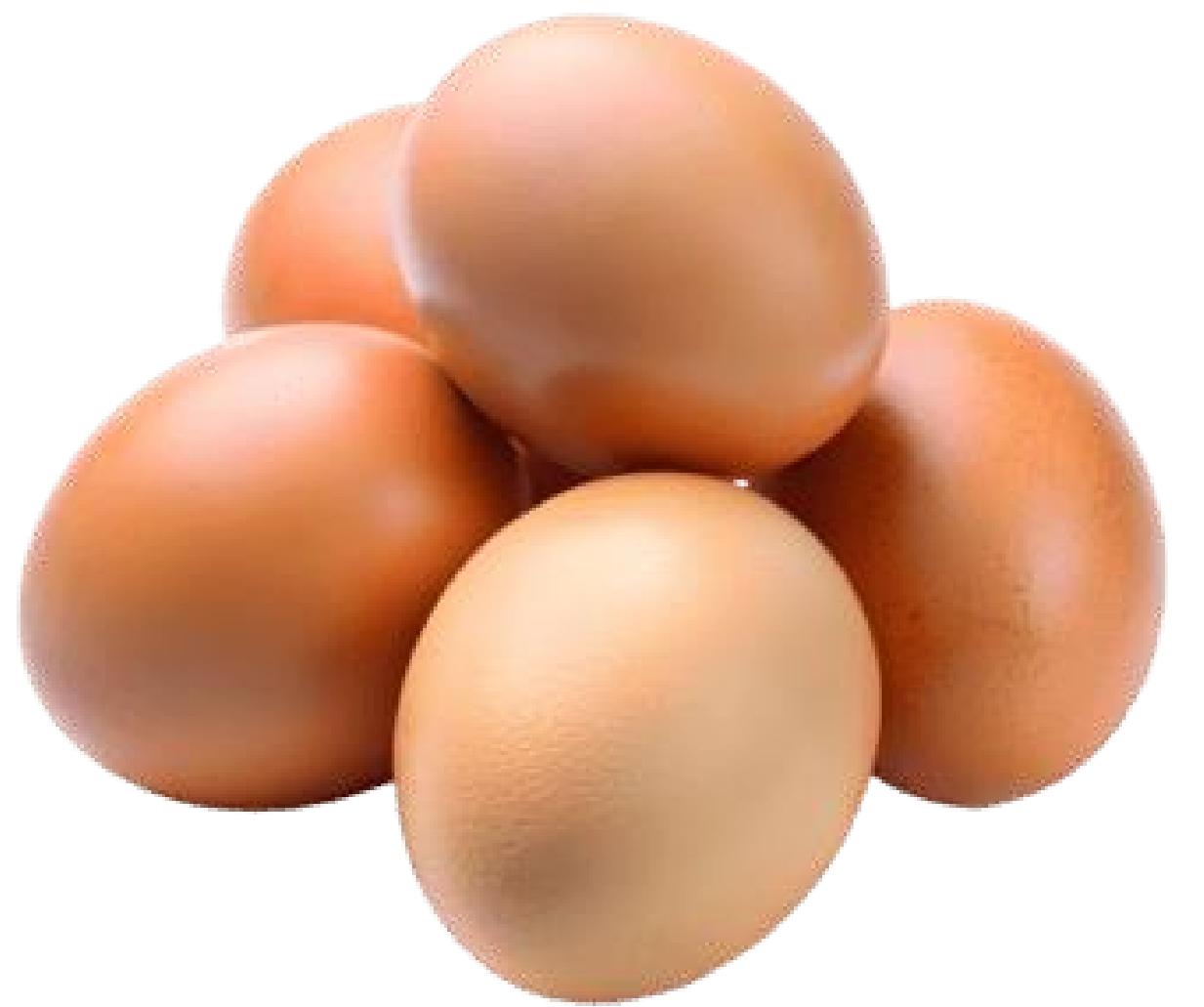
**Escreva o mínimo de código  
para o teste passar**

Refatore.

O código E os testes

**Testes são vivos!**





**Às vezes você precisa escrever  
um pouco de código funcional  
antes dos testes**

**E tá tudo bem...**

**TDD é apenas uma abordagem,  
não uma lei**

**Os seus testes só precisam ser  
vivos e **evoluir** com a base de  
código**

**Teste é documentação!**

# **Behavior Driven Development**

Behavior D

**Testes que descrevem o  
comportamento do seu software**

**Melhor entendimento**

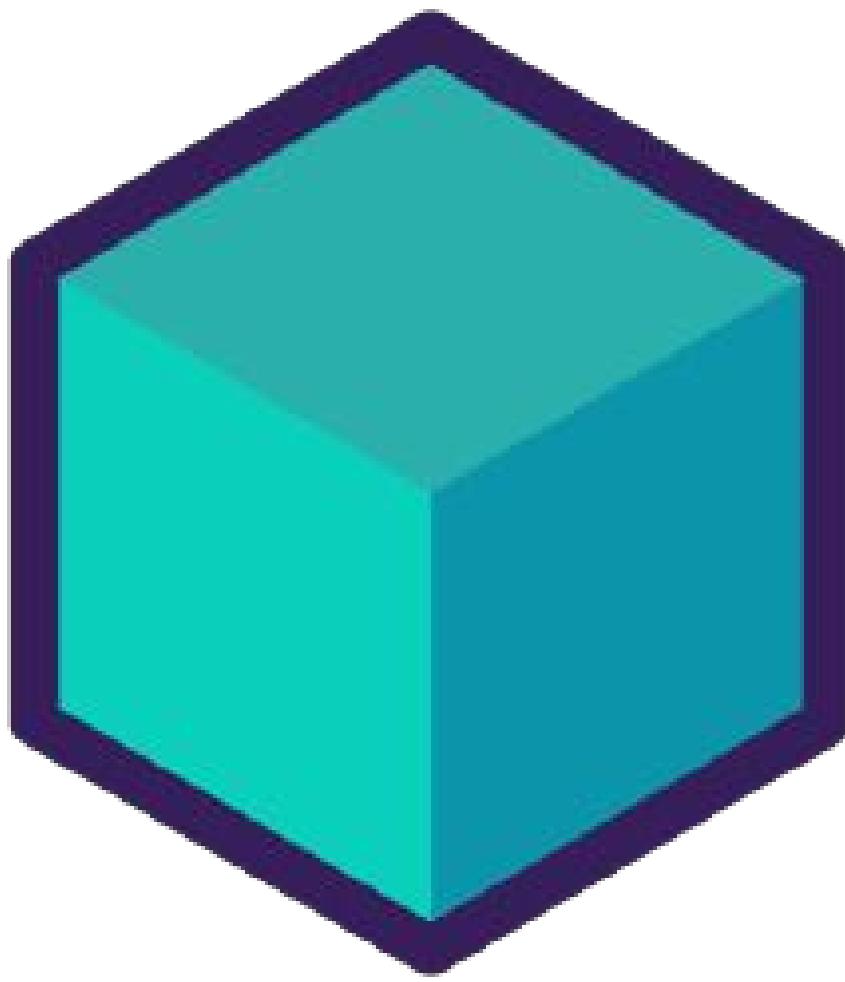
**Línguagem comum**

**Documentação de graça**

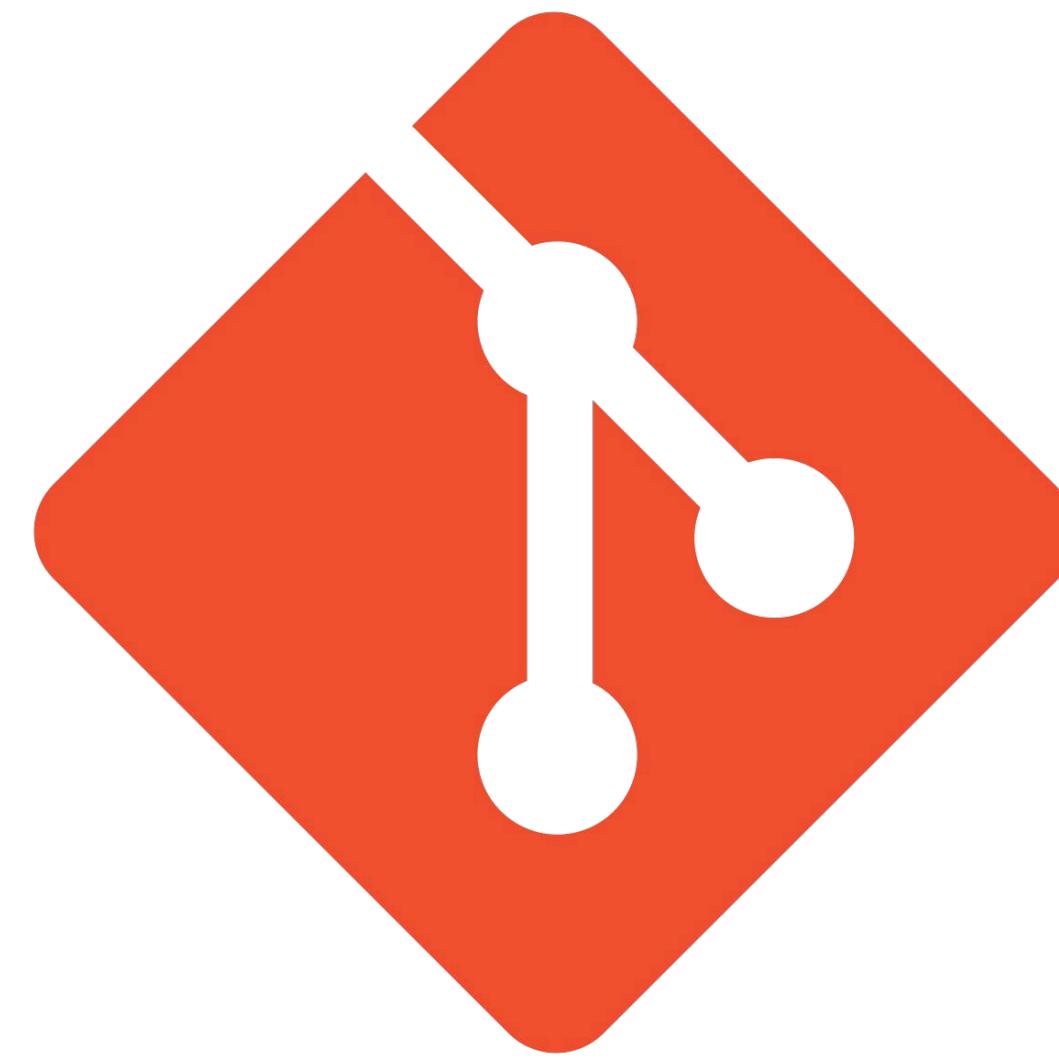
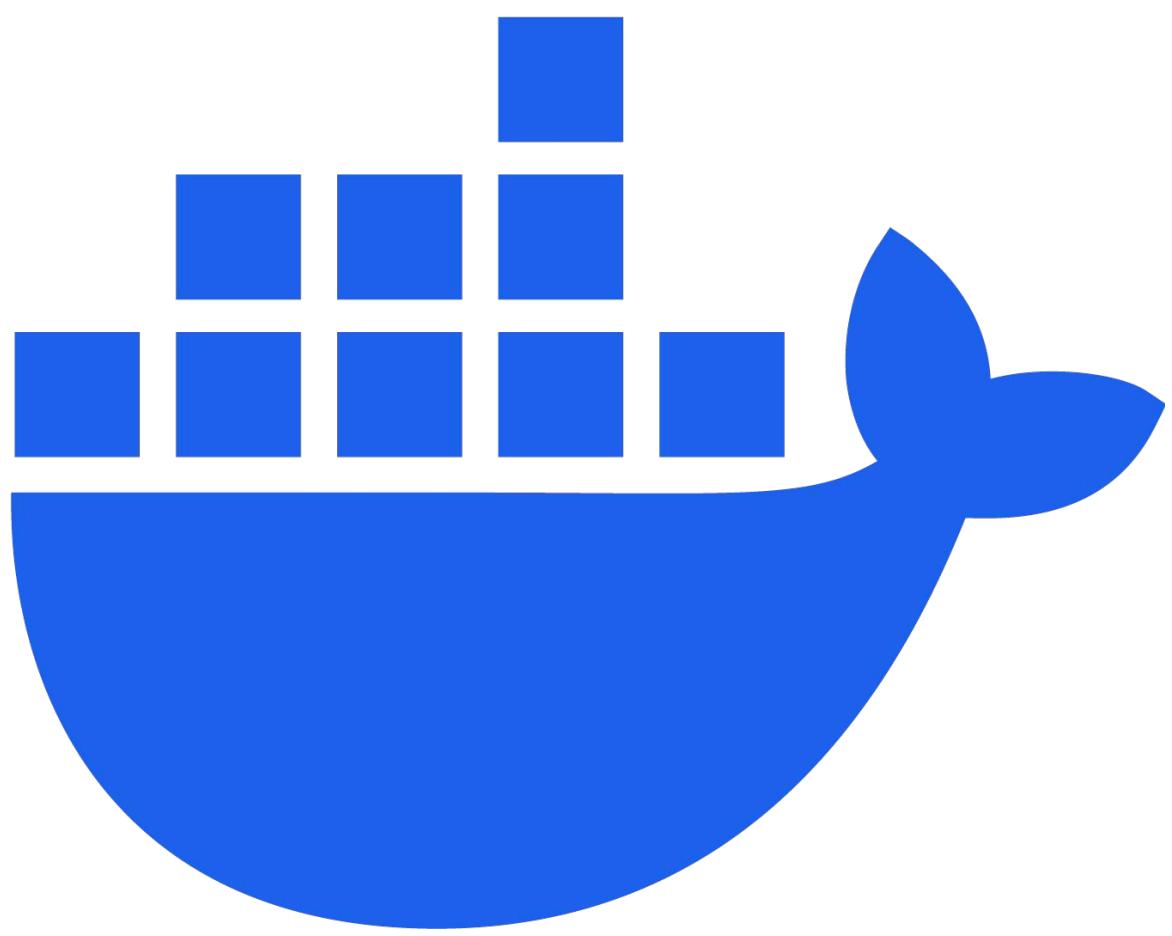
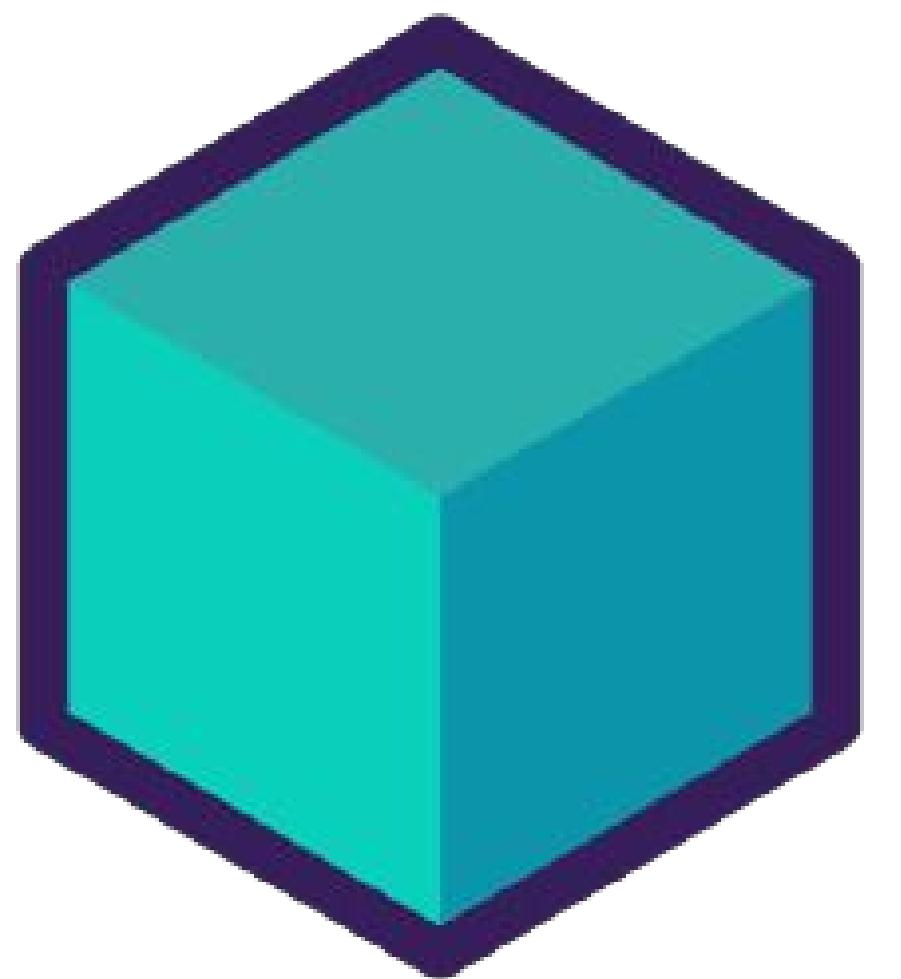
# Práctica



**/joevtap/automated-testing-talk**



**testcontainers**



**Obrigado byronianos!**