

Coma seus  
vegetais<sub>(v3)</sub>





# Olar bebês!

Sou Eduardo Mendes (@dunossauro)

Apto fazedor de lambdas, pythonista, apaixonado por software **livre** e ciência.

A green circle containing two dark blue double quote marks ("").

“

5 perguntas pra gente se entender

1. Quem desenvolve software  
profissionalmente?

The slide is decorated with various hand-drawn illustrations of fruits and leaves. In the top left, there are blueberries. Next to them is an orange slice. To the right is a green leaf. Further right is a slice of watermelon. In the top right corner is a lime. On the left side, there is a yellow lemon. Below the lemon is a green leaf. At the bottom left is a strawberry. In the bottom center is a banana. To the right of the banana is a green leaf. Further right is a lime slice. At the bottom right is a cherry. In the top center, there is a green circle containing the text '44'.

44

5 perguntas pra gente se entender

## II. Quem trabalha com metodologias ágeis?

The slide is decorated with various hand-drawn illustrations of fruits and vegetables. At the top left is a blue bubble. Next to it is a slice of orange. To the right is a green vegetable, possibly a cucumber. Further right is a slice of watermelon. At the top right is a green pear. In the center, above the text, is a green circle containing two black quotation marks. Below this is the text "5 perguntas pra gente se entender". The main title "III. Quem trabalha com testers na equipe?" is in the center. Surrounding the text are more illustrations: a yellow lemon on the left, a green leaf on the left, a yellow slice of orange on the right, a green leaf on the right, a strawberry on the bottom left, a banana on the bottom left, a green leaf on the bottom center, a green vegetable on the bottom center, a cherry on the bottom right, and an orange on the bottom right.

“

5 perguntas pra gente se entender

### III. Quem trabalha com testers na equipe?



5 perguntas pra gente se entender

## IV. Quem entrega software com qualidade?

The background of the slide is decorated with various hand-drawn illustrations of fruits and vegetables. These include a blueberry cluster in the top left, an orange slice in the top center, a green cucumber in the top right, a watermelon slice in the top right, a lemon in the middle left, a green leaf in the middle left, a yellow lemon slice in the middle right, a strawberry in the bottom left, a banana in the bottom left, a green leaf in the bottom center, a green cucumber in the bottom center, a cherry in the bottom right, and an orange in the bottom right.

“

5 perguntas pra gente se entender

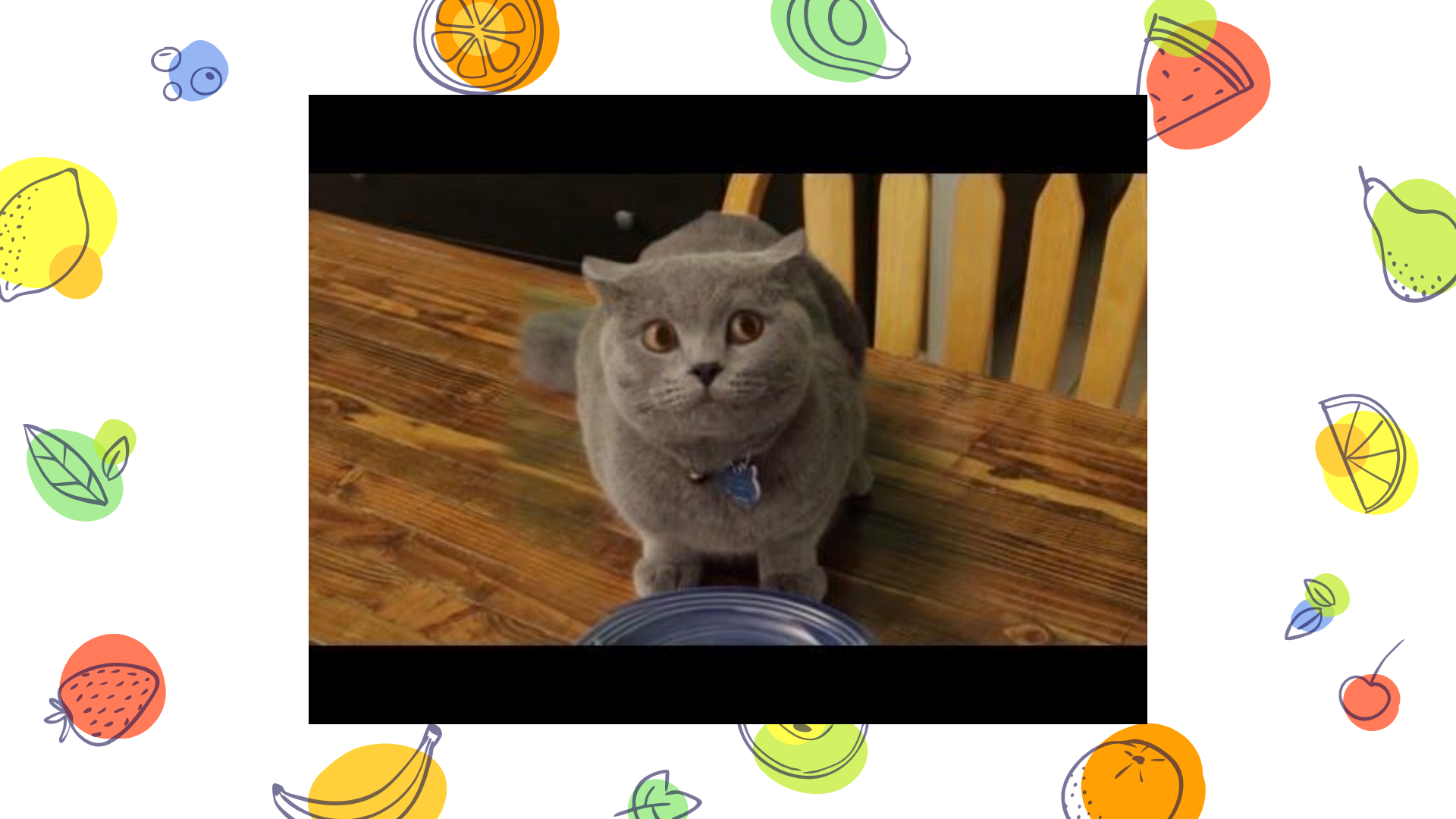
V. Quem desenvolve software  
para usuários?



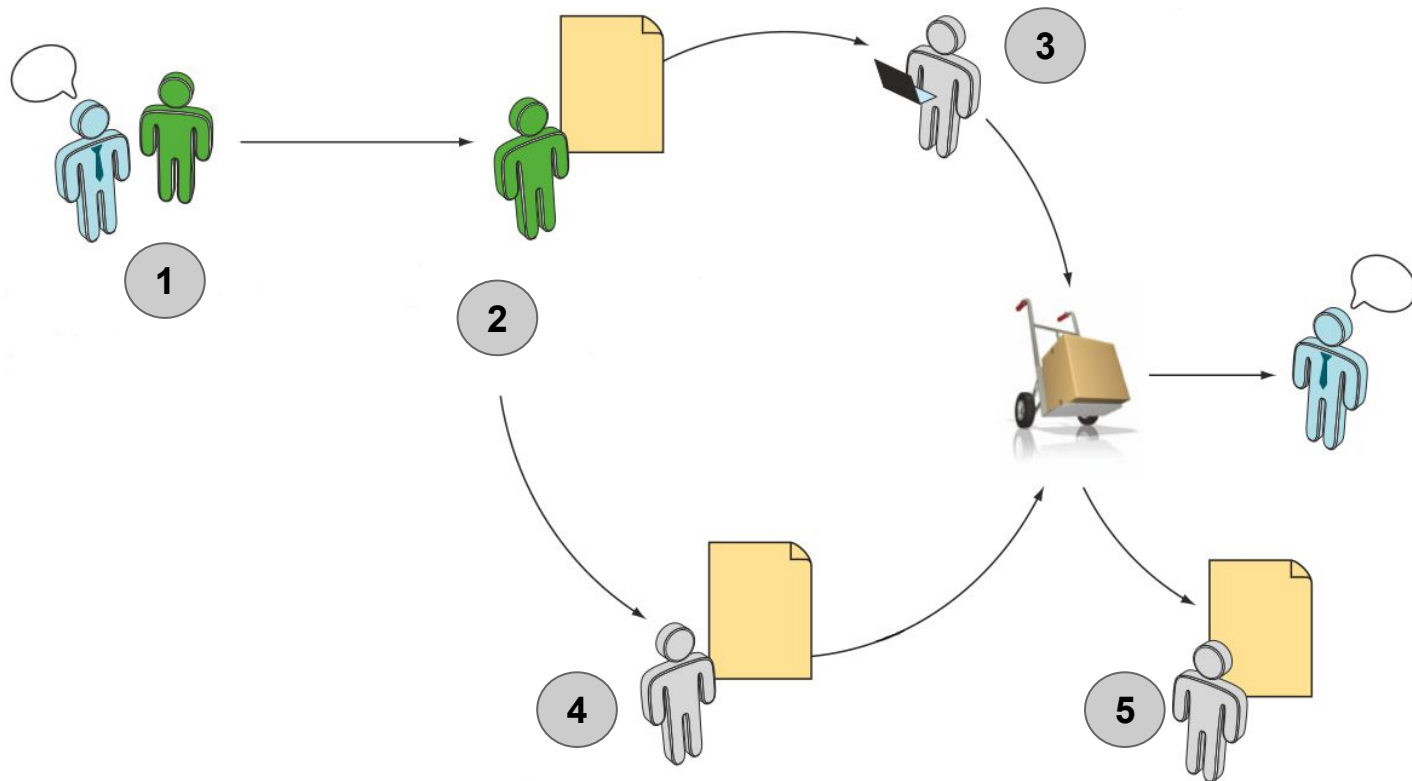
# SIM

Eu vim falar sobre qualidade de  
software

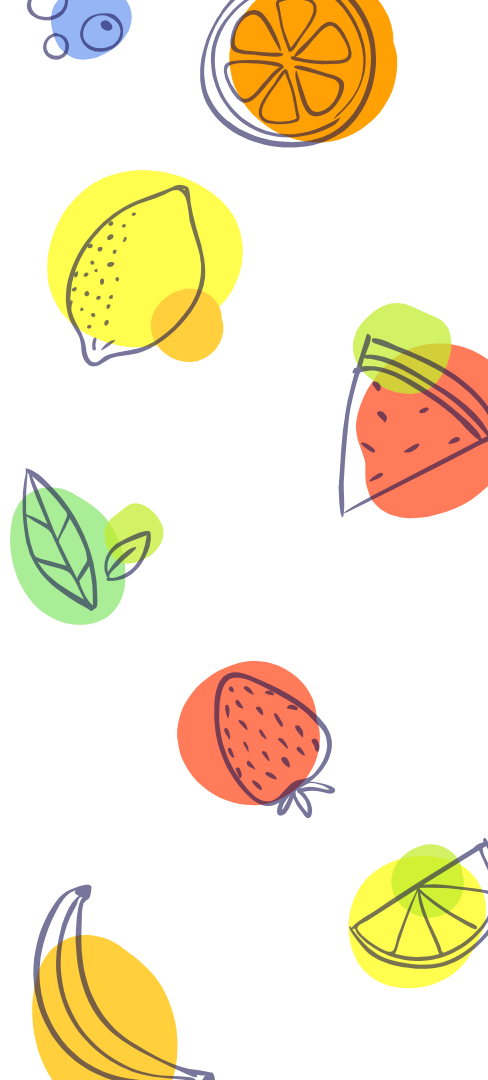
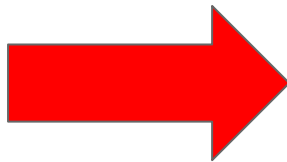




# Processo de desenvolvimento (Tradicional)



Sim, eu não queria colocar esse slide



# Testes automatizados / CI (P)

- × Testes unitários
  - Testar todas as menores parcelas de código possíveis
- × TDD
  - Escrever o teste para falhar
  - Fazer o teste passar
  - Refatorar o código
- × CI
  - Garantir que toda modificação possível no código não faça nenhuma outra parte do mesmo falhar



# Testes automatizados / CI (N)

## × Testes unitários

- Testes unitários geralmente testam funções/métodos/classes
- Testam as regras de negócio?

## × TDD

- Garante que o que foi feito está funcionando, mas garante a integração de tudo que foi feito?

## × CI

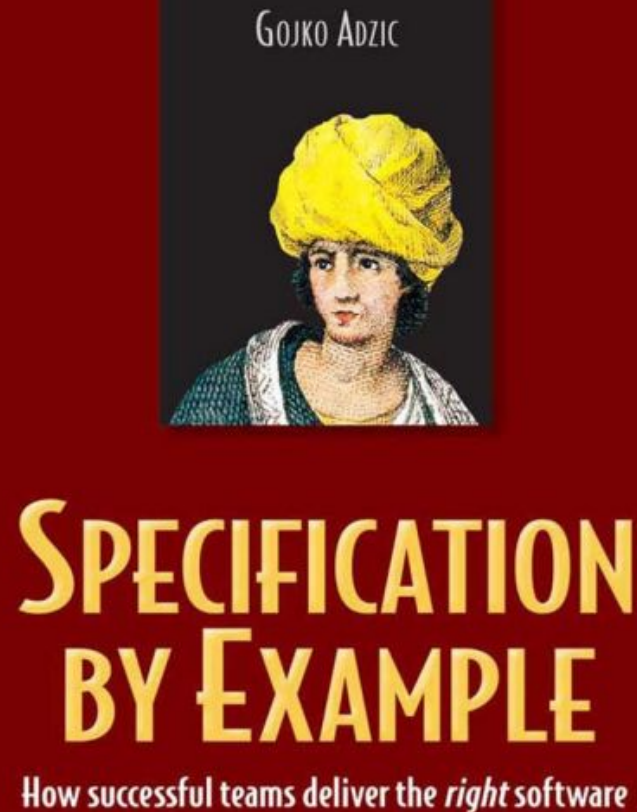
- A cada commit você garante que a entrega tem mesmo de ser aquela?
- A documentação é atualizada a cada commit?



Então a ideia aqui é falar sobre  
testes como o centro do  
desenvolvimento

Não testes como testes, mas como  
assertivas de critério de aceite. ou  
seja documentação executável.

- ✗ Testes que agregam valor
- ✗ Documentação viva
- ✗ Noção de progresso
- ✗ Testes automatizados
- ✗ Funcionalidades funcionais





Mas como? (Testes unitários, como eu fazia)

```
1 class TestNinjas(TestCase):  
2     @mock.path('Samurai')  
3     def test_briga_ninja(self, inimigo):  
4         ...  
5  
6     def test_ninja(self):  
7         ...  
8  
9     def test_inimigo(self):  
10        ...  
11
```

Mas como? (visando a regra de negócio) - ATDD

```
1 class TestNinjas(TestCase):
2     def test_ninja_deve_correr_quando_encontrar_lutador_mais_forte(self):
3         ...
4
5     def test_ninja_deve_lutar_quando_encontrar_lutador_mais_fraco(self):
6         ...
7
8     def test_ninja_deve_ganhar_mais_xp_quando_vencer(self):
9         ...
10
```

1 **Funcionalidade:** Lutar ou correr

2 A fim de aumentar a taxa de sobrevivência ninja,  
3 Como um comandante ninja  
4 Eu quero que meus ninjas decidam se querem combater  
5 seu oponente com base em seus níveis de habilidade  
6

7 **Cenário:** Oponente fraco

8 **Dado** um ninja com terceiro nível de faixa-preta

9 **Quando** atacado por um samurai

10 **Então** o ninja deve enfrentar o adversário  
11

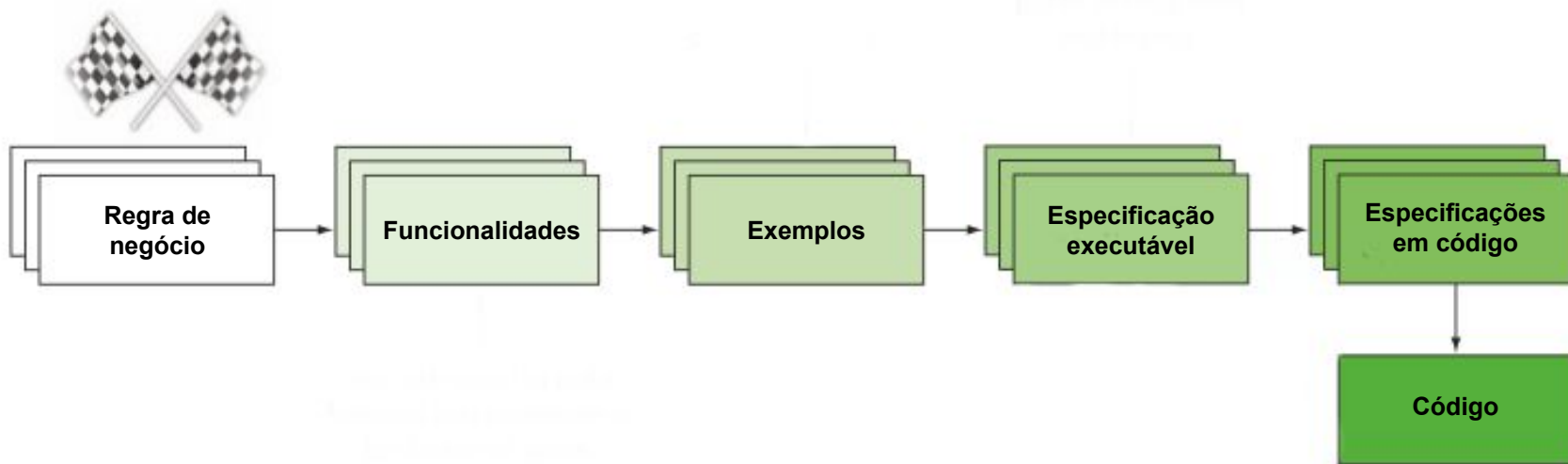
12 **Cenário:** Oponente forte

13 **Dado** um ninja com terceiro nível de faixa-preta

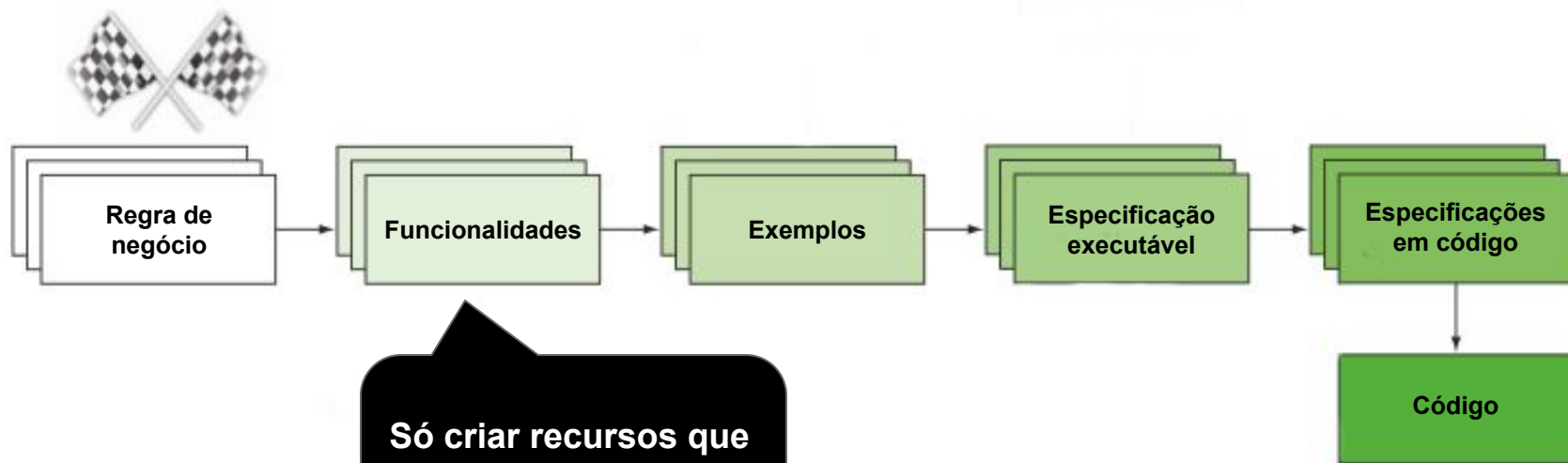
14 **Quando** atacado pelo Chuck Norris

15 **Então** o ninja deve correr por sua vida

Mas como? Uma visão geral do modelo

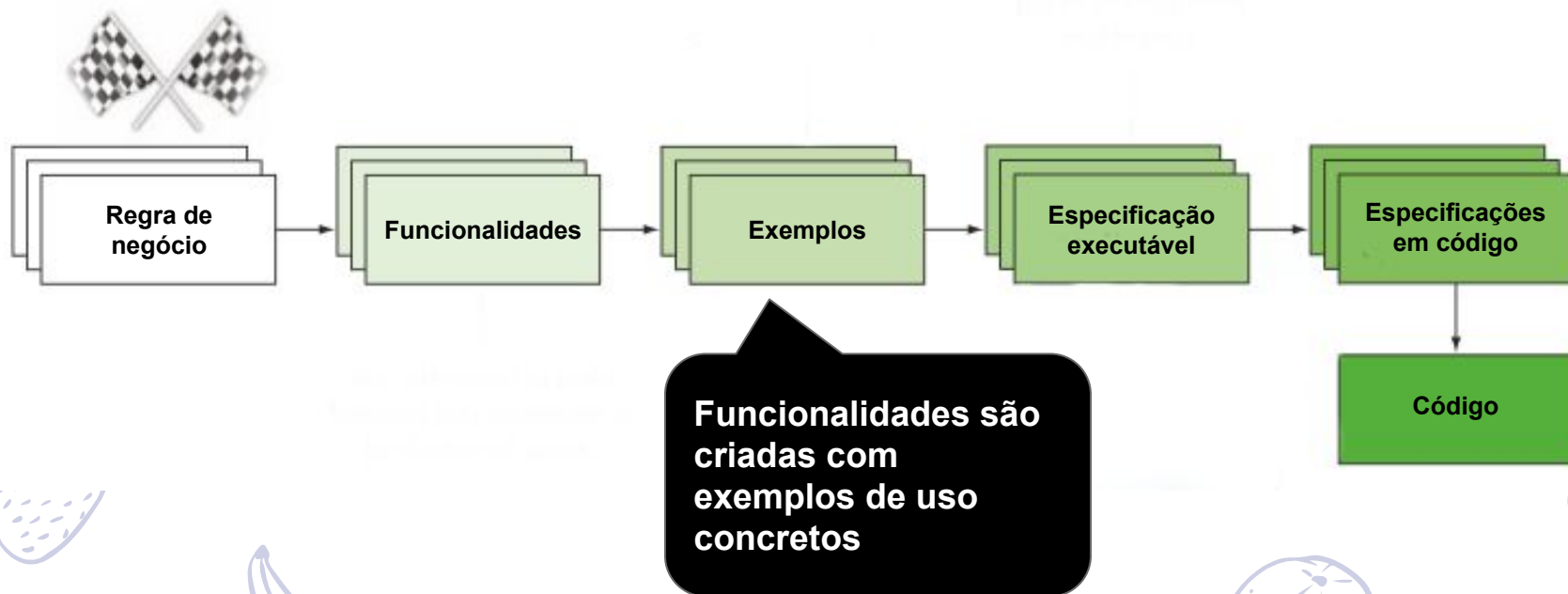


Mas como? Uma visão geral do modelo

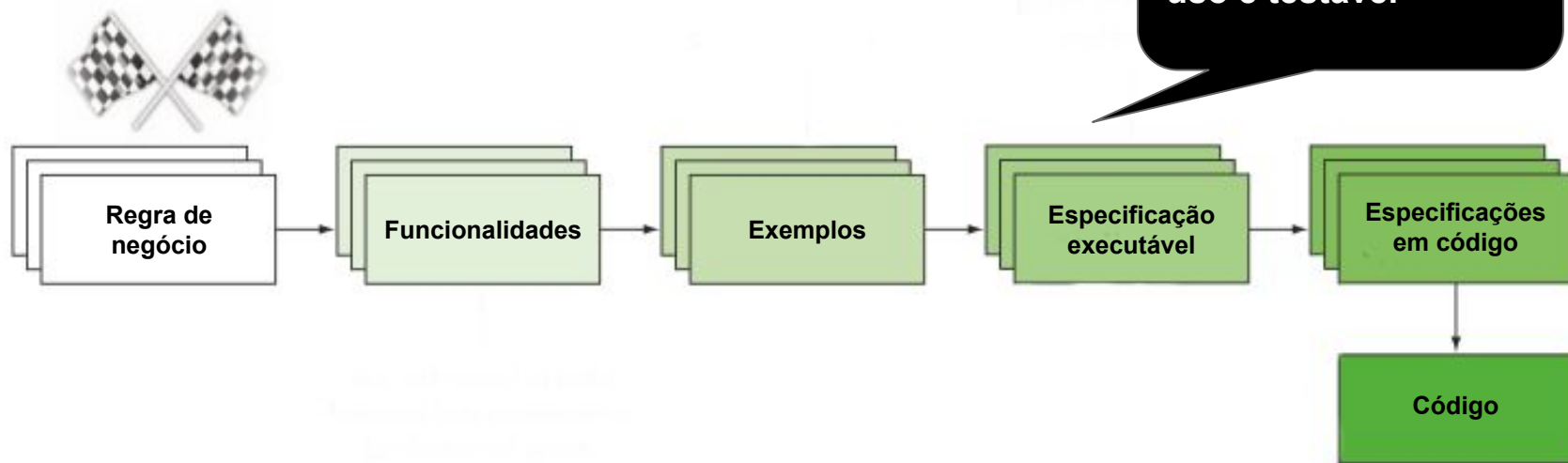


**Só criar recursos que  
contribuam para  
regra de negócio**

Mas como? Uma visão geral do modelo

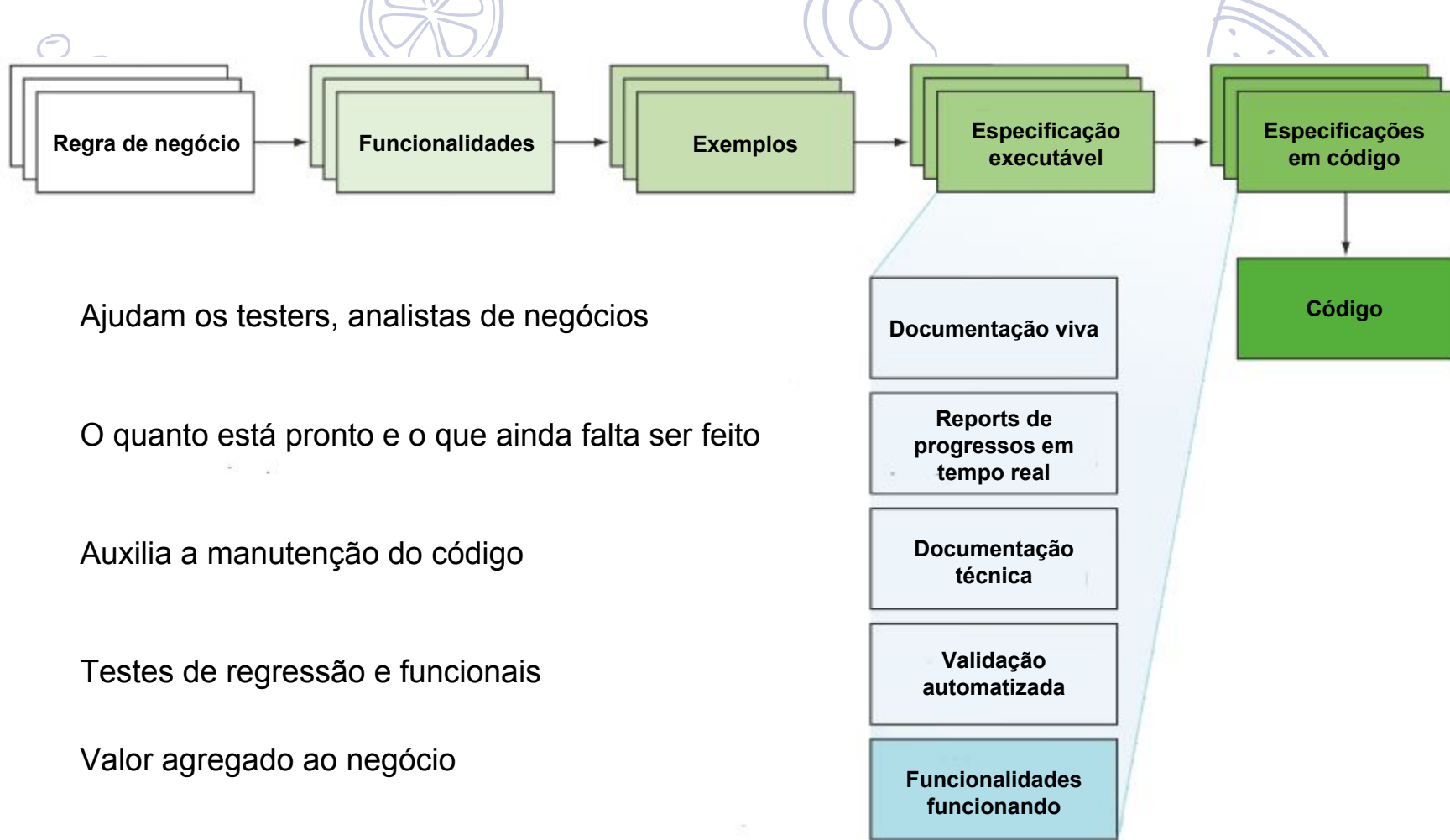


Mas como? Uma visão geral do modelo

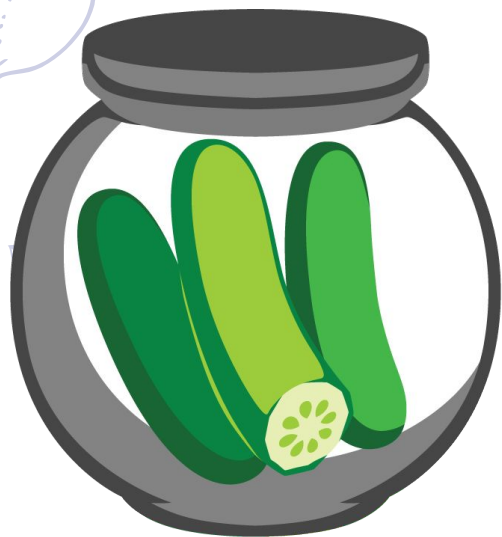


```
1 from behave import given, then, when
2 from personagens import Ninja, Samurai, Chuck
3 import combate
4
5 @given('um ninja com terceiro nível de faixa-preta')
6 def nija_faixa_preta(context):
7     context.ninja = Ninja(faixa='Preta', nível=3)
8
9 @when('atacado por um samurai')
10 def ataque(context):
11     context.adversário = Samurai()
12
13 @then('o ninja deve enfrentar o adversário')
14 def lutar(context):
15     resultado = combate(context.ninja, context.adversário)
16     assert resultado.vencedor == Ninja
```





Documentação viva



# Pickles

## Living Documentation

# Noção de progresso

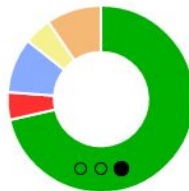
Project	Number	Date
Damian - project with Cucumber	10	11 gru 2016, 00:16

Branch	release/1.0
Browser	Firefox
Platform	Windows

## Features Statistics

The following graphs show passing and failing statistics for features

Steps



Feature	Steps						Scenarios			Features	
	Passed	Failed	Skipped	Pending	Undefined	Total	Passed	Failed	Total	Duration	Status
1st feature	10	0	0	0	0	10	1	0	1	1m 39s 263ms	Passed
Second feature	5	1	2	1	2	11	1	1	2	092ms	Failed
2	15	1	2	1	2	21	2	1	3	1m 39s 355ms	
	71.43%	4.76%	9.52%	4.76%	9.52%		66.67%	33.33%			50.00%

# Noção de progresso

## Feature Report

Feature	Steps						Scenarios			Features	
	Passed	Failed	Skipped	Pending	Undefined	Total	Passed	Failed	Total	Duration	Status
Second feature	5	1	2	1	2	11	1	1	2	092ms	Failed

### Feature Second feature

As an Account Holder I want to withdraw cash from an ATM,  
so that I can get money when the bank is closed

#### @checkout

##### Scenario Outline Account may not have sufficient funds ▼

Account holder withdraws more cash

##### Hooks ▼

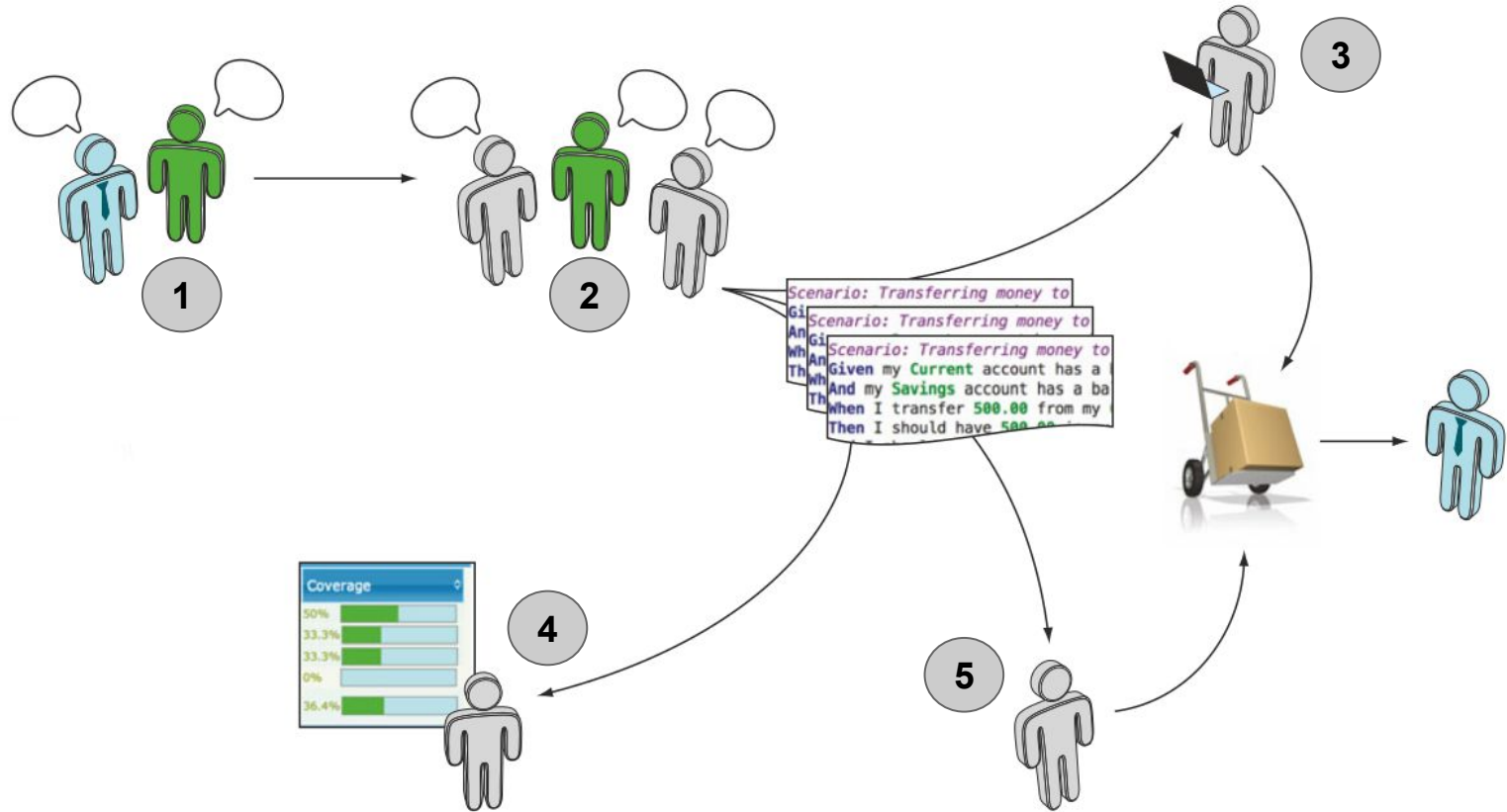
Before MachineFactory.findCachMachine()	010ms
Before MachineFactory.wait()	001ms

##### Steps ▼

Given the account balance is 100	000ms
And the card is valid	000ms
And the machine contains 100	000ms
When the Account Holder requests 20	000ms
Then the ATM should dispense 20	000ms
And the account balance should be 90	001ms

Error message

# Processo de desenvolvimento (com BDD)



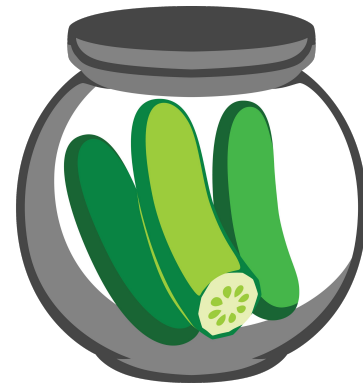
Nome da palestra



**Cucumber**



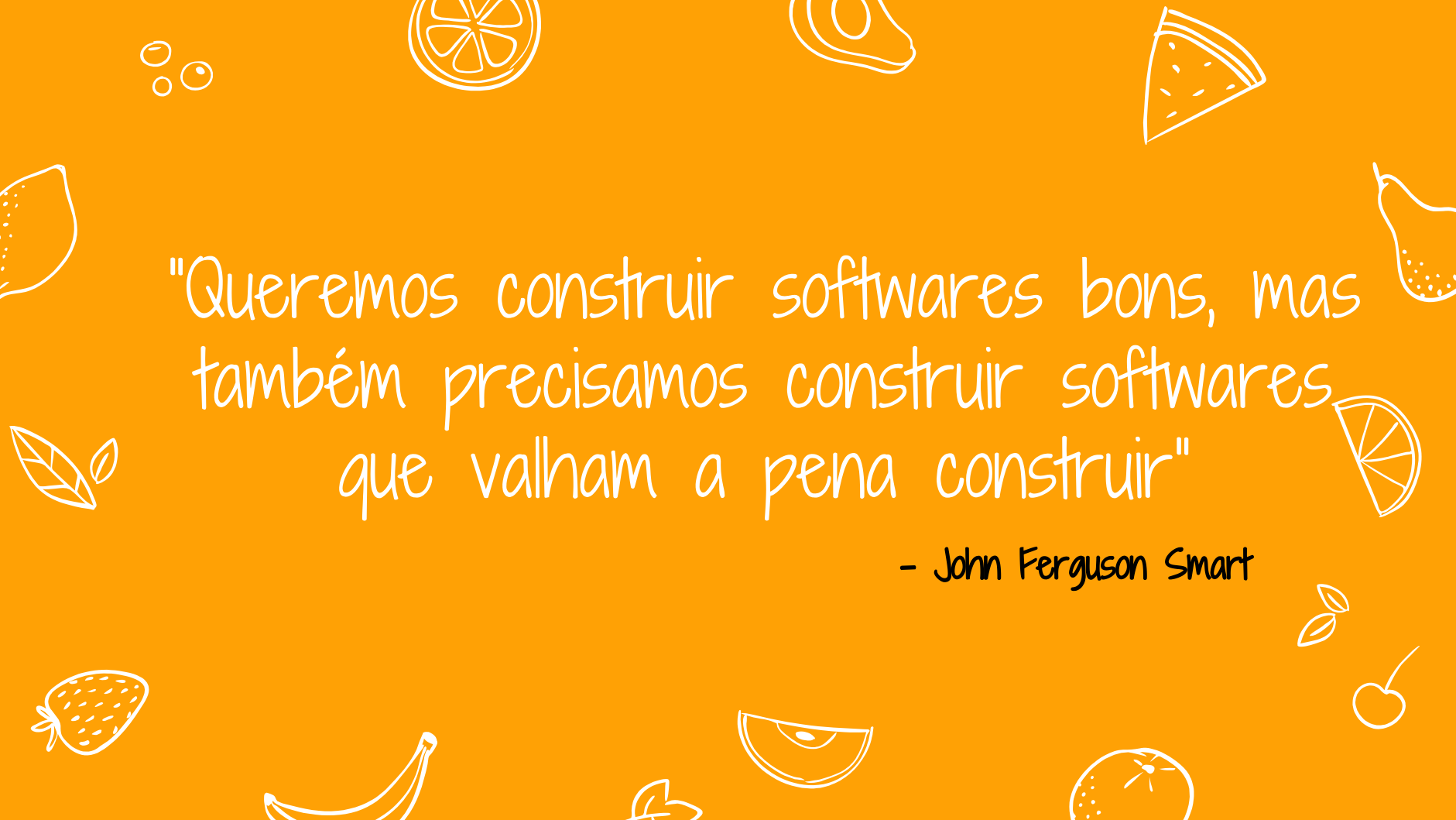
**Radish**



**pickles**

**Lettuce**

**Gherkin**



"Queremos construir softwares bons, mas também precisamos construir softwares que valham a pena construir"

- John Ferguson Smart

- John Ferguson Smart

The slide is decorated with various hand-drawn fruit illustrations in a whimsical style. At the top, there's a blue cluster of grapes, an orange slice, a green kiwi, and a slice of watermelon. On the right side, there's a whole green kiwi and a slice of yellow citrus. The bottom features a strawberry, a banana, a green kiwi, a whole orange, and a cherry. A small green leaf is also visible near the bottom center.

# XOXO

Perguntas?

Você pode me achar em

@dunossauro

mendesxeducardo@gmail.com

**[youtube.com/c/eduardomendes](https://youtube.com/c/eduardomendes)**