

Testes com Python #2

Testando unidades





Nome:

Eduardo Mendes

Email:

mendesxeduardo@gmail.com

GIT:

github.com/z4r4tu5tr4

Roteiro

- Um porque das coisas
- Assertivas simples e booleanas (Calculadora)
- Testando instâncias e iteráveis (Mercadinho da Maria)
- Mock, se der tempo

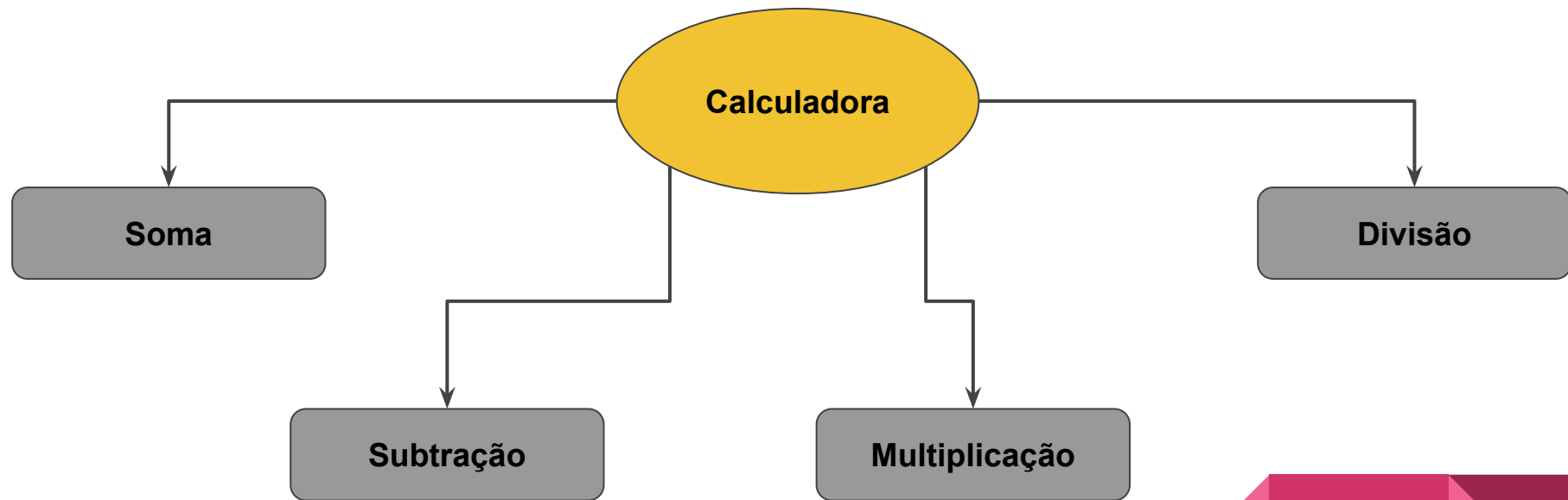


Por que testar? (Ned Batchelder - Pycon 2014)

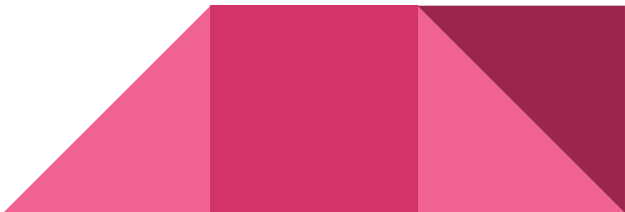
- Saber como seu código funciona
- Economizar tempo
- Melhorar seu código
- Sem medo
- “Debugar é difícil, testar é fácil”



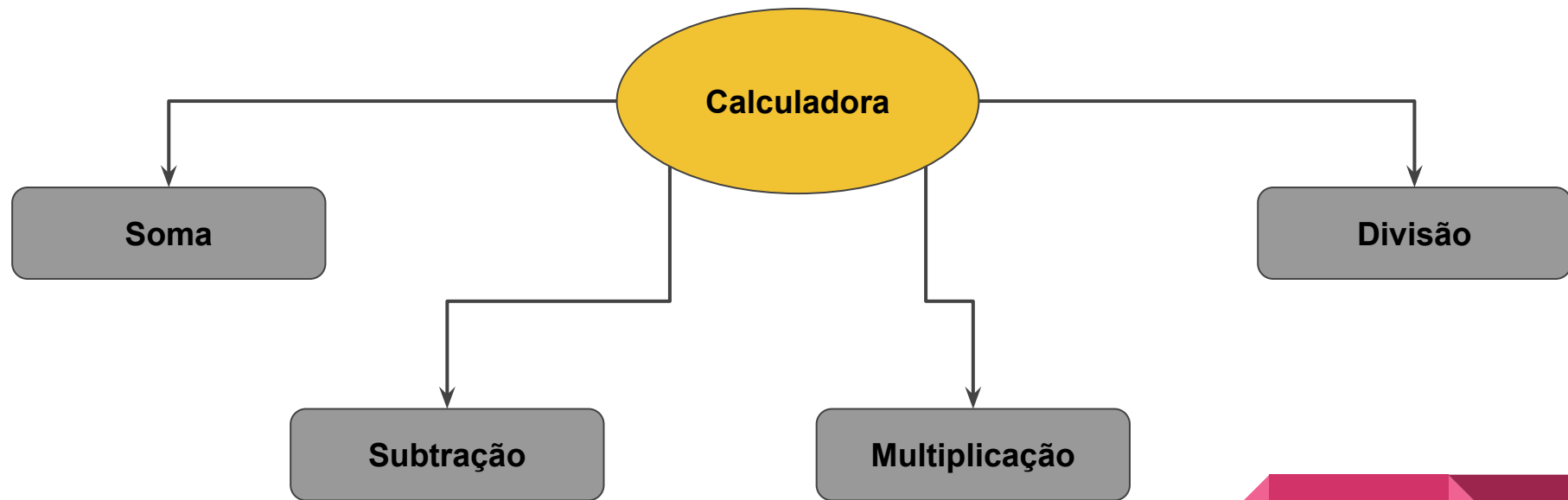
Como funcionam os testes unitários?



Antes de testar

- Testes são a cara da sua aplicação
 - Se alguém quiser entender o que é feito, pode ler os testes
 - Testar todas as unidades é uma coisa importante
 - Se você tem alguns (+2) testes para a mesma coisa, talvez seja melhor refatorar
 - Quando algo quebra, você sabe exatamente o que quebrou
 - Entregas mais confiáveis
 - A medida em que você escreve mais testes, mais coisas tendem a funcionar e código fica mais limpo (para ambos)
 - Menos bugs
- 

Voltando ao exemplo



Unittest

Assert (Bool)	Unittest
==	assertEqual
!=	assertNotEqual
>	assertGreater
>=	assertGreaterEqual
<	assertLess
<=	assertLessEqual

Quem solicitou?

Como eu dou continuidade as contas?



complicando as coisas

