

Live de Python #83

BDD - Desenvolvimento guiado por comportamento

Roteiro

- Uma revisão sobre Testes de unidade
- Uma revisão sobre TDD
- Ah.... o BDD
- Linguagem ubíqua (DDD)
- Gherkin
 - Ações
 - “Negócios”
- Behave

Obrigado!

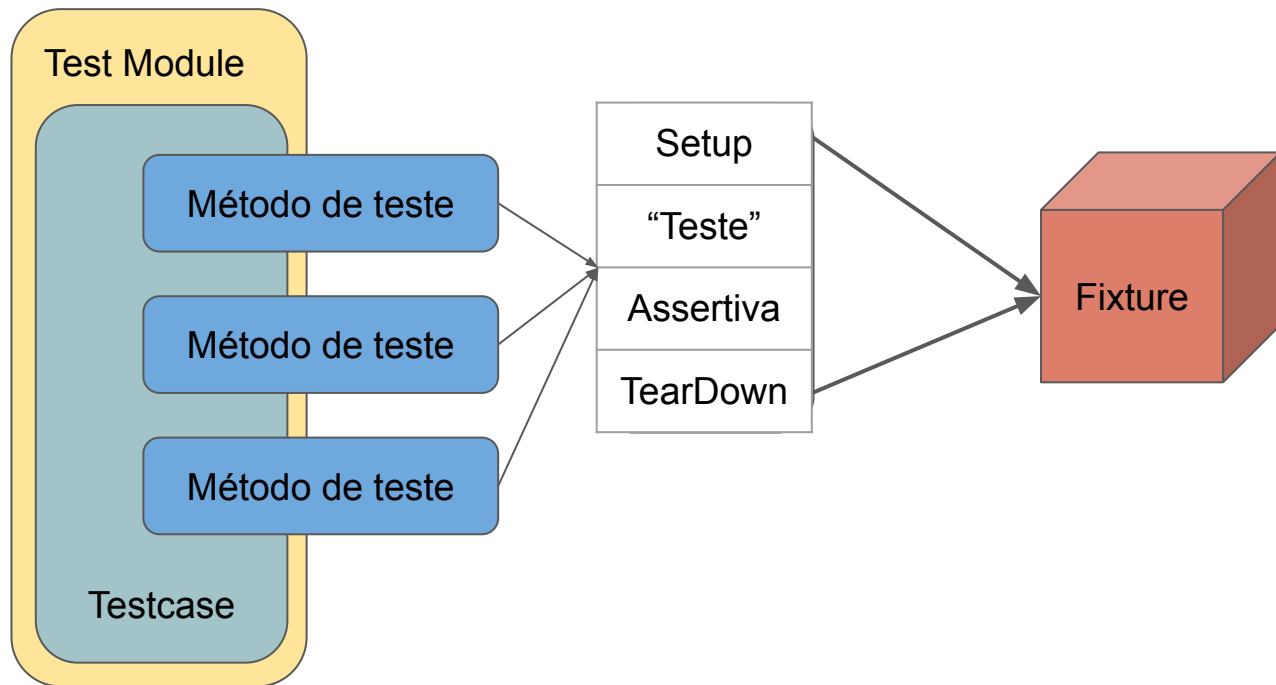
apoia.se/livedepython

```
~/g/apoiadores >>> python apoiadores.py
```

Humberto Rocha	Vicente Marcal	Maria Boladona	Pedro Alves
Thiago Araujo	Eliabe Silva	João Lugão	Sérgio Passos
Magno Malkut	David Reis	Dayham Soares	Fabiano Teichmann
Vergil Valverde	Edimar Fardim	Regis Santos	Wander Silva
Jonatas Oliveira	Fernando Furtado	Rennan Almeida	Renato Santos
Fábio Serrão	Jonatas Simões	Jucélio Silva	Wellington Camargo
Júlia Kastrup	Johnny Tardin	Paulo Tadei	Elias Soares
Jean Vetorello	Rodrigo Vaccari	Fabiano Silos	Tiago Cordeiro
Willian Gl	Andre Machado	William Oliveira	Gleison Oliveira
Bruno Guizi	Pablo Henrique	Nilo Pereira	Renan Moura



Testes de unidade

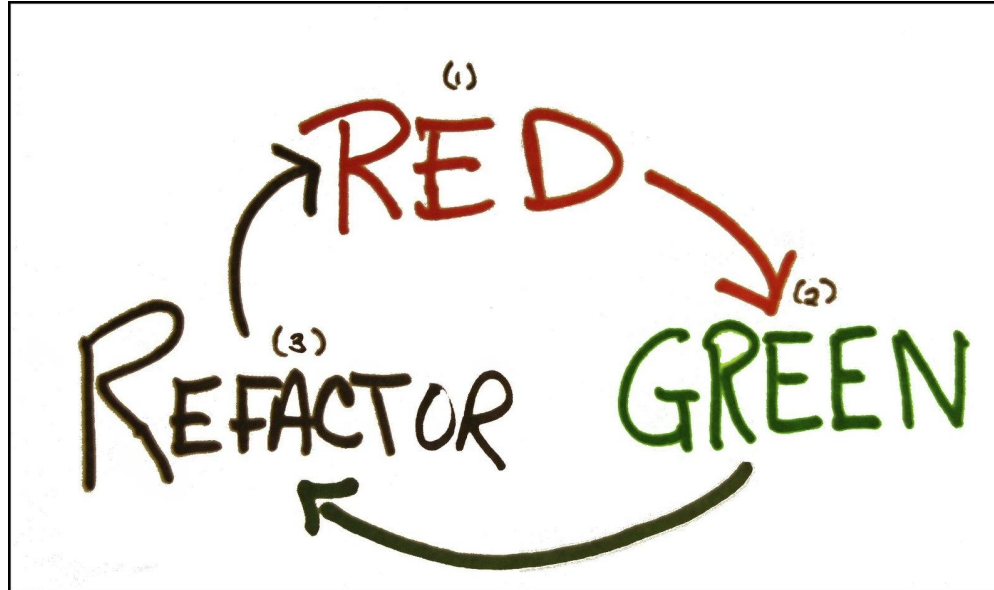


Testes de unidade

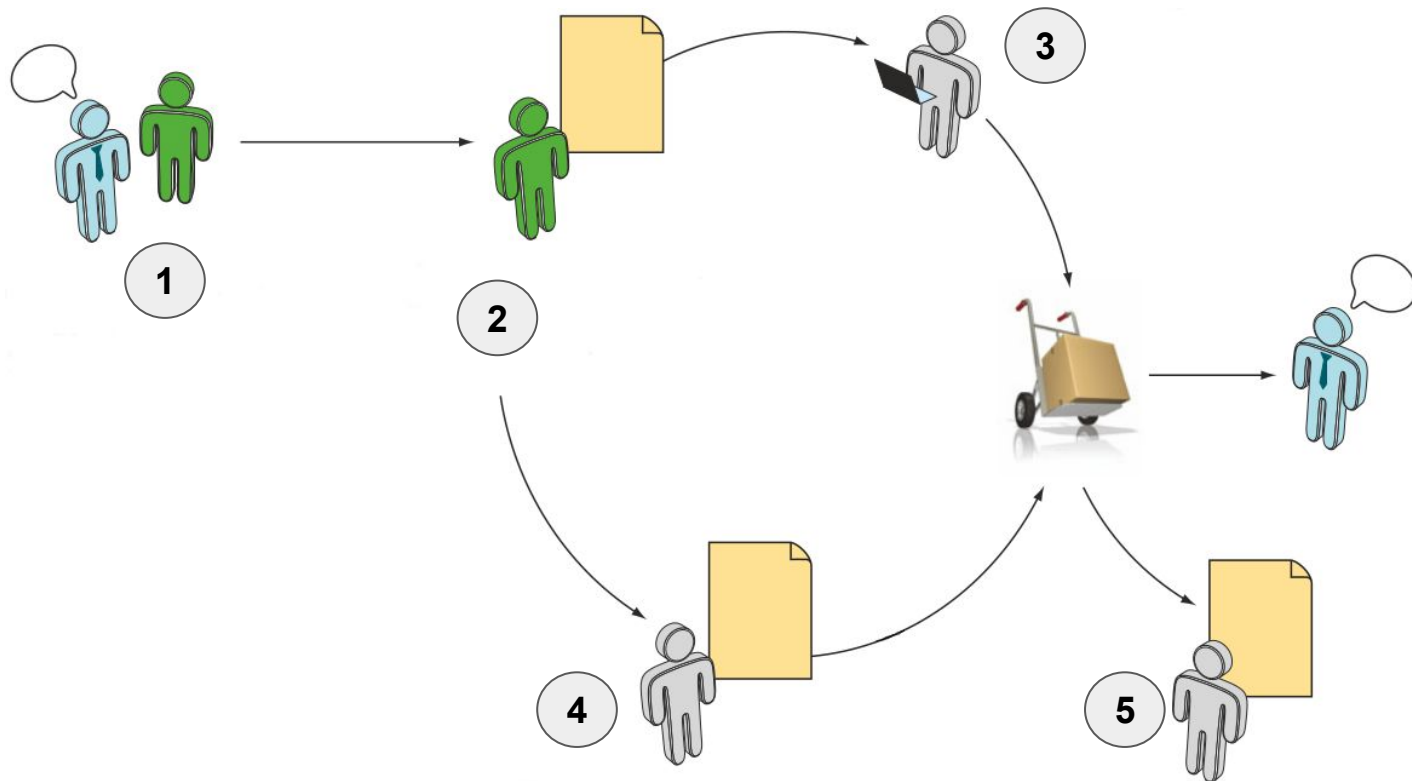
```
1  class TestNinjas(TestCase):  
2      def test_ninja_deve_correr_quando_encontrar_lutador_mais_forte(self):  
3          ...  
4  
5      def test_ninja_deve_lutar_quando_encontrar_lutador_mais_fraco(self):  
6          ...  
7  
8      def test_ninja_deve_ganhar_mais_xp_quando_vencer(self):  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

TDD

Desenvolvimento “dirigido” (orientado) a testes [Test Driven Development], é uma metodologia criada por Kent Beck.



Ciclo de desenvolvimento



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

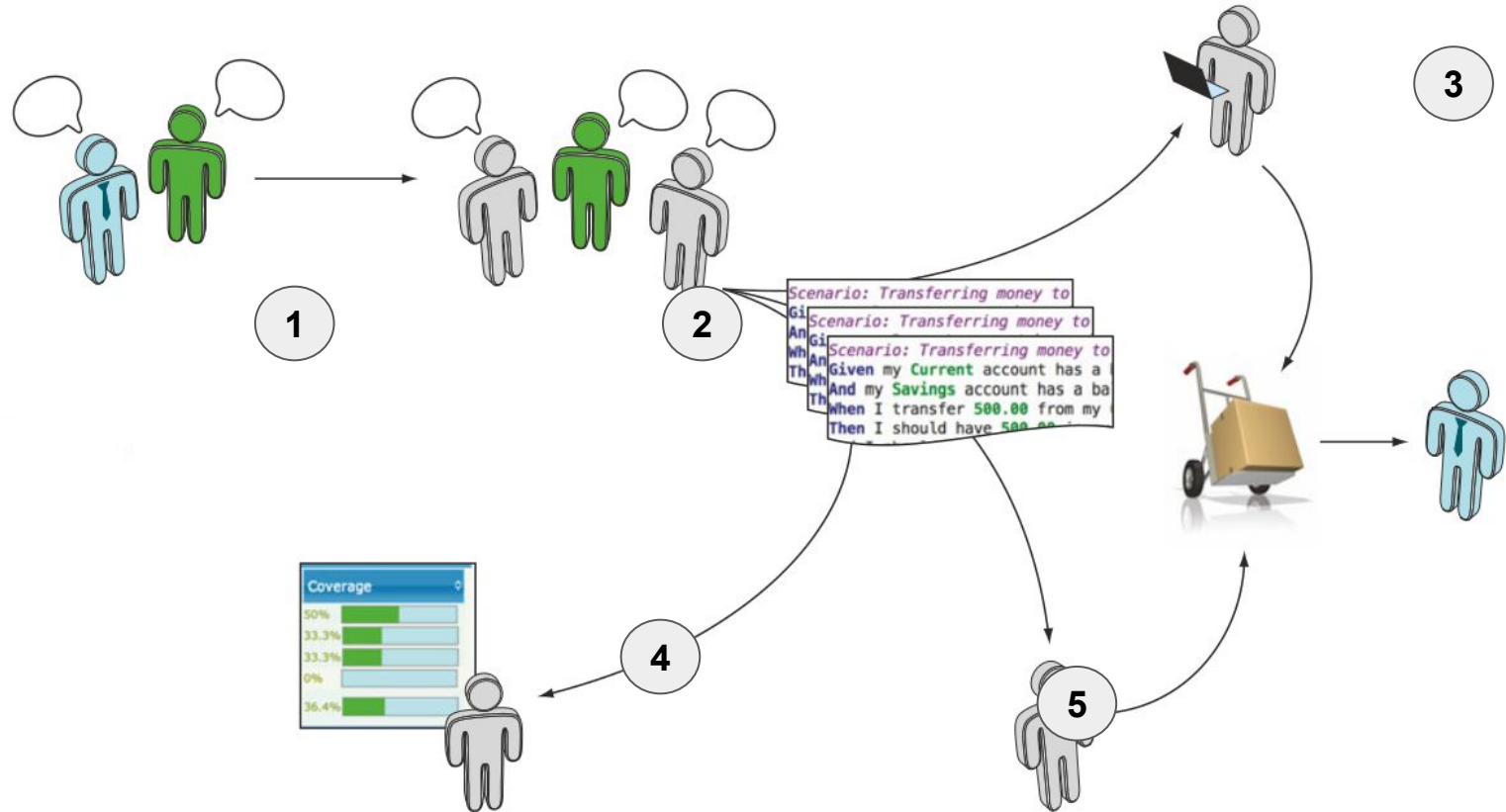
The image shows the front cover of the book 'Specification by Example' by Gojko Adzic. The cover has a dark red background. At the top, the author's name 'GOJKO ADZIC' is printed in a small, white, sans-serif font. Below the name is a square inset containing a portrait of a man with dark hair, wearing a bright yellow turban and a green and white patterned garment. The title 'SPECIFICATION BY EXAMPLE' is written in large, bold, yellow, sans-serif capital letters. Below the title, the subtitle 'How successful teams deliver the *right* software' is printed in a smaller, white, sans-serif font. At the bottom right, the Manning logo (two small squares) and the word 'MANNING' are visible in white.

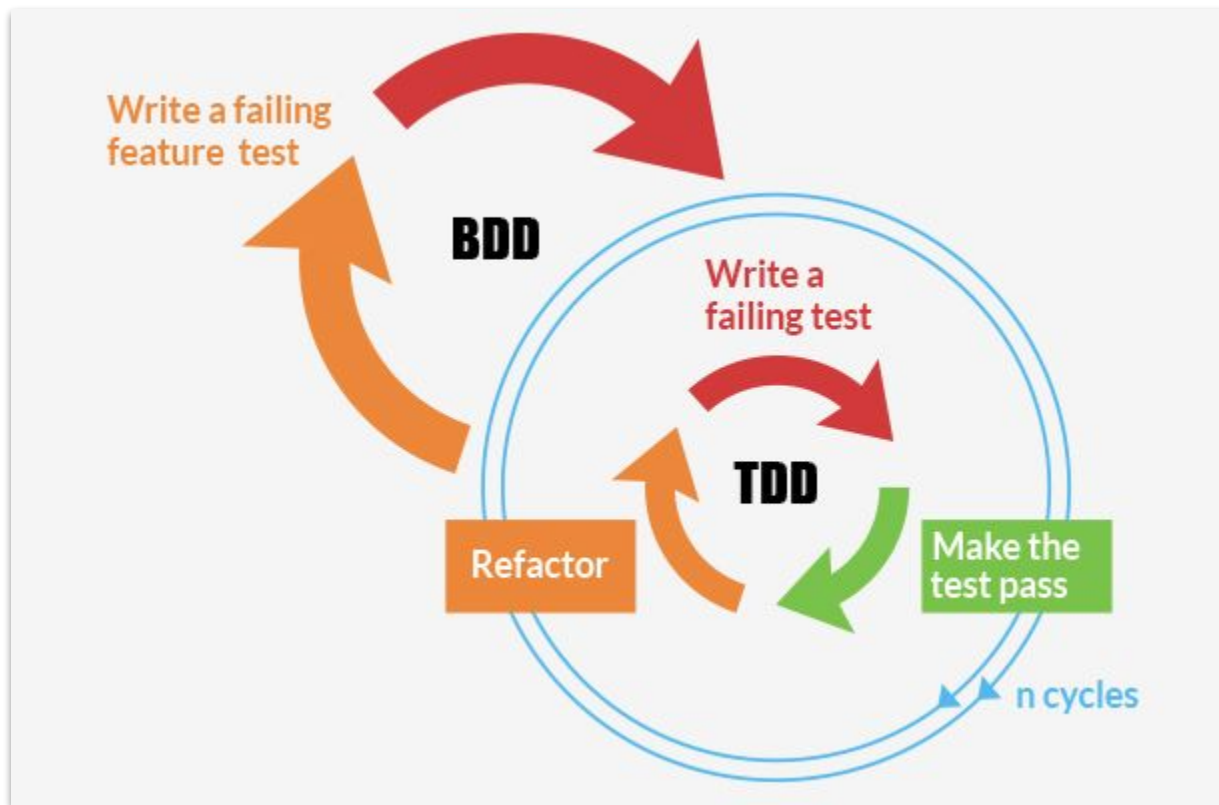
SPECIFICATION BY EXAMPLE

How successful teams deliver the *right* software

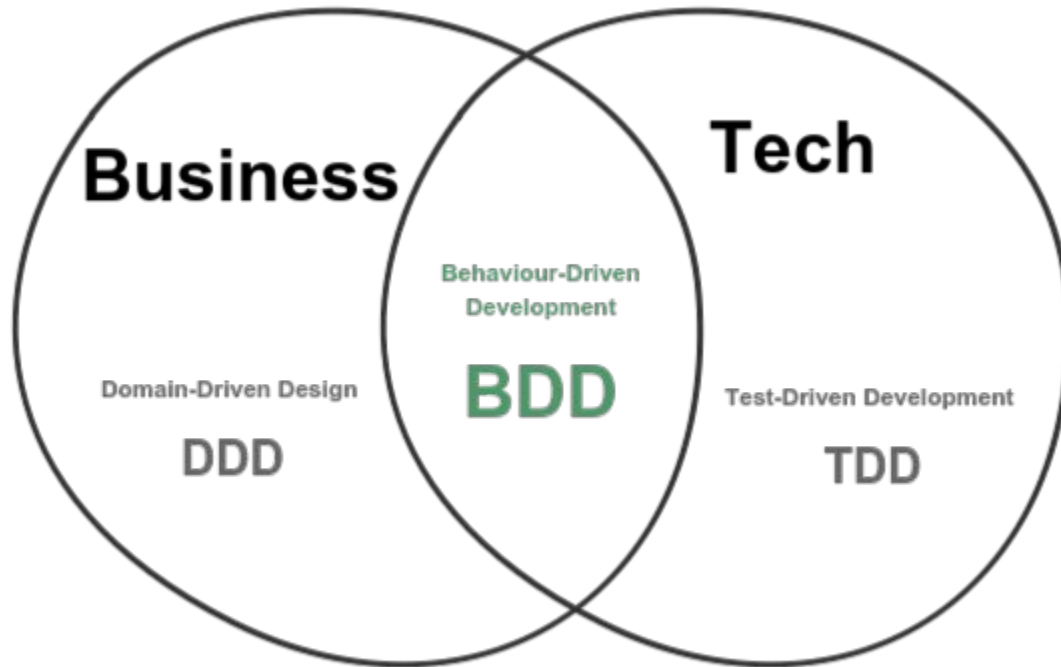
 MANNING

Processo de desenvolvimento (com BDD)

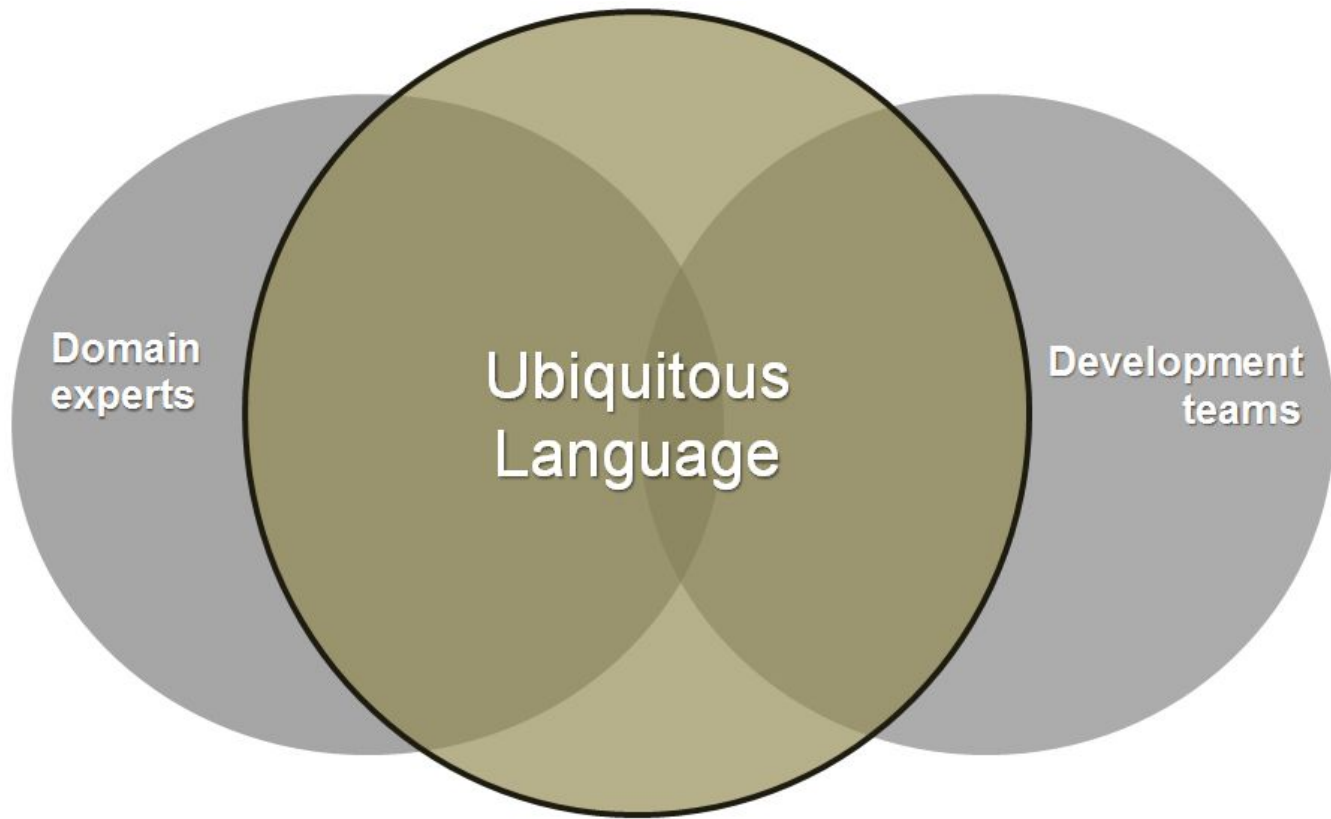




Ah... o BDD



Um modelo possível



Linguagem ubíqua

Gherkin

- “É uma linguagem de domínio específico de negócios criada especialmente para descrições de comportamento”
- Tem como propósito servir de documentação e roteiro para testes automatizados

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

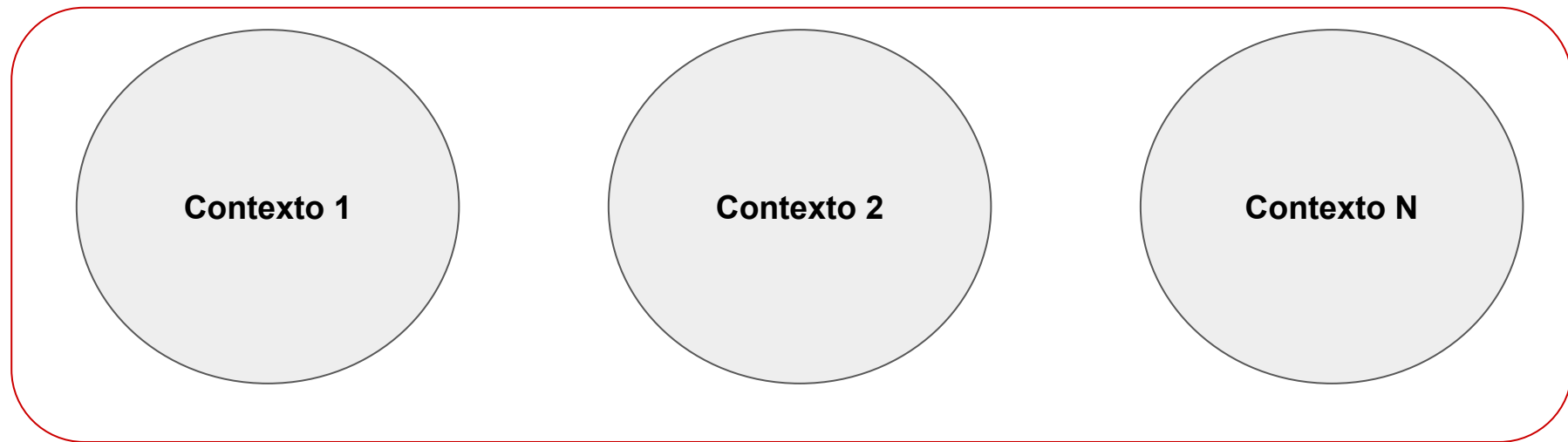
14 **Quando** atacado pelo Chuck Norris

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

As palavras mágicas

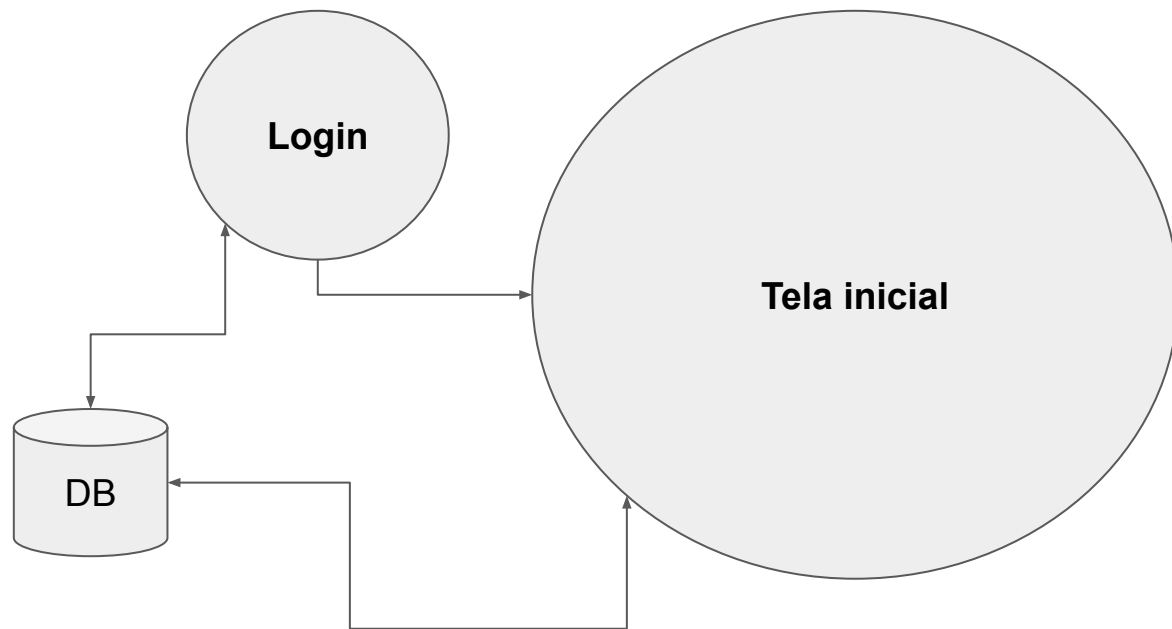
- Dado [Given]:
 - Tem a função de avaliar as condições existam, ou levar até elas. *como um fixture*
 - EX: Dado que o usuário “Eduardo” esteja logado
- Quando [When]:
 - Executa alguma ação (chama uma função, invoca um método, clica em um botão)
 - Podem ser um conjunto de ações
 - EX: Quando acessar a página inicial
- Então [Then]:
 - Faz a assertiva da ação passada
 - EX: Então a mensagem “Olá Eduardo” deve estar na página

Exemplificando

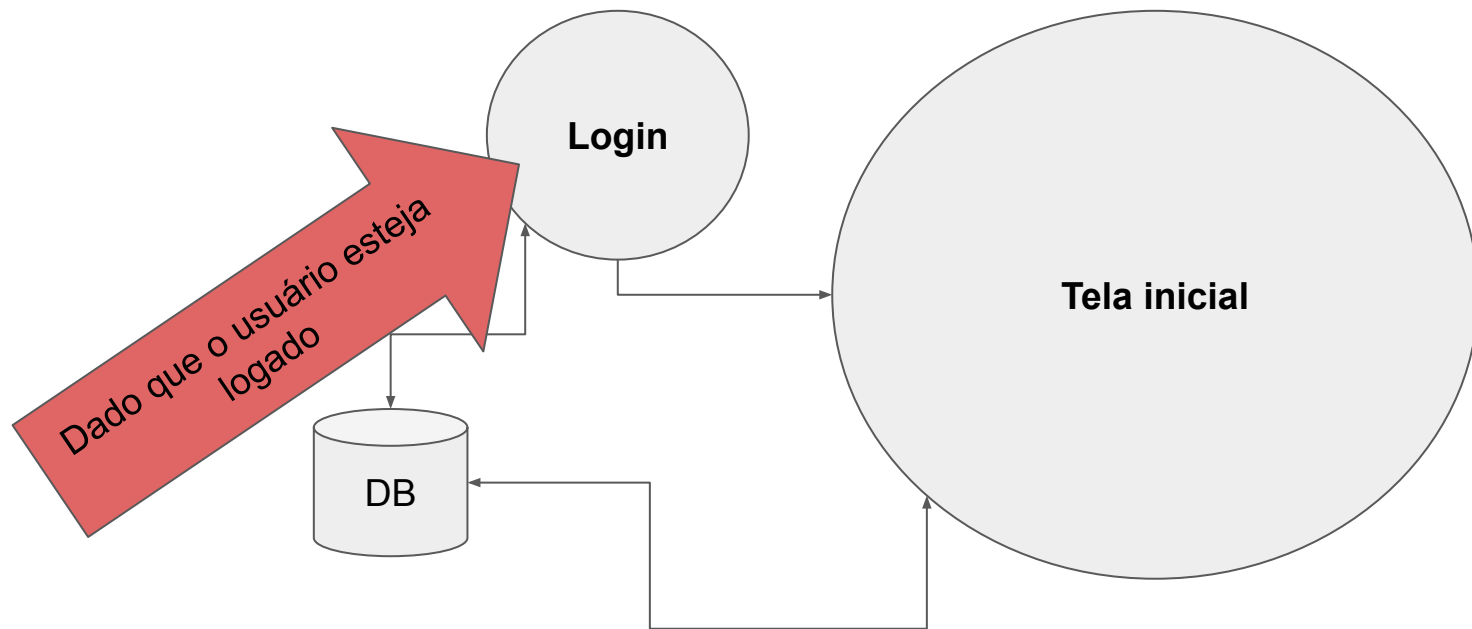


Nosso Software

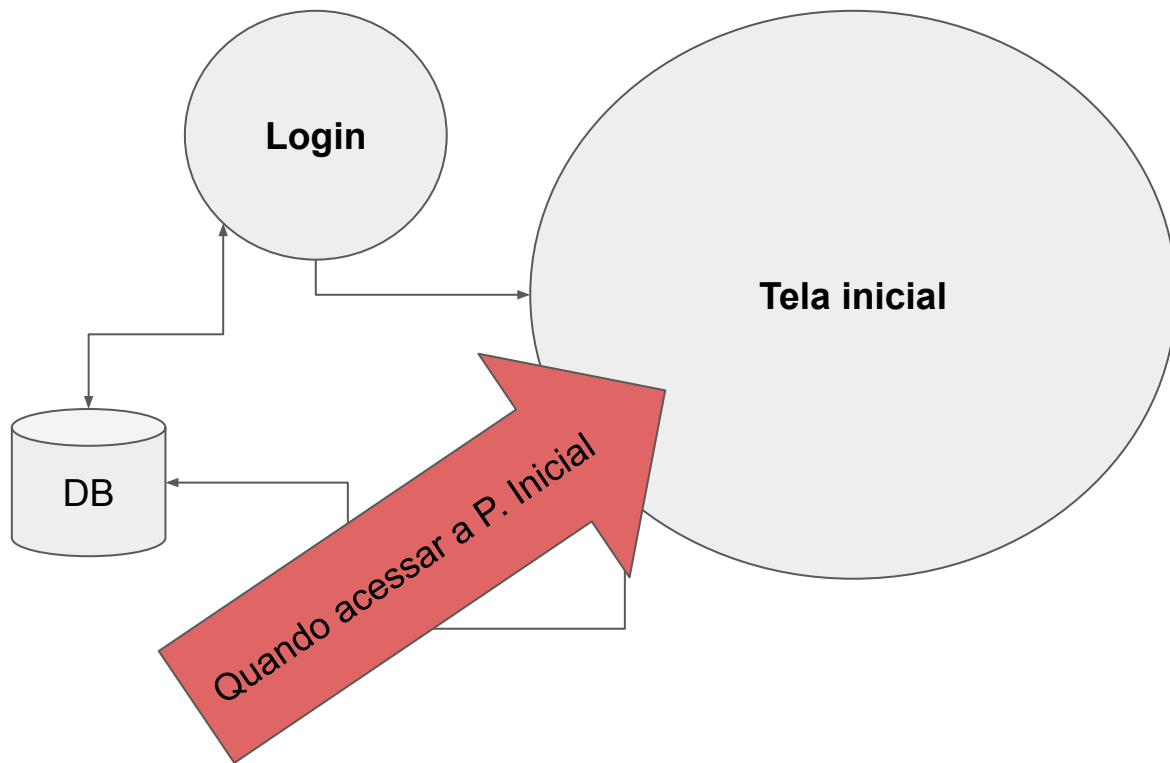
Exemplificando



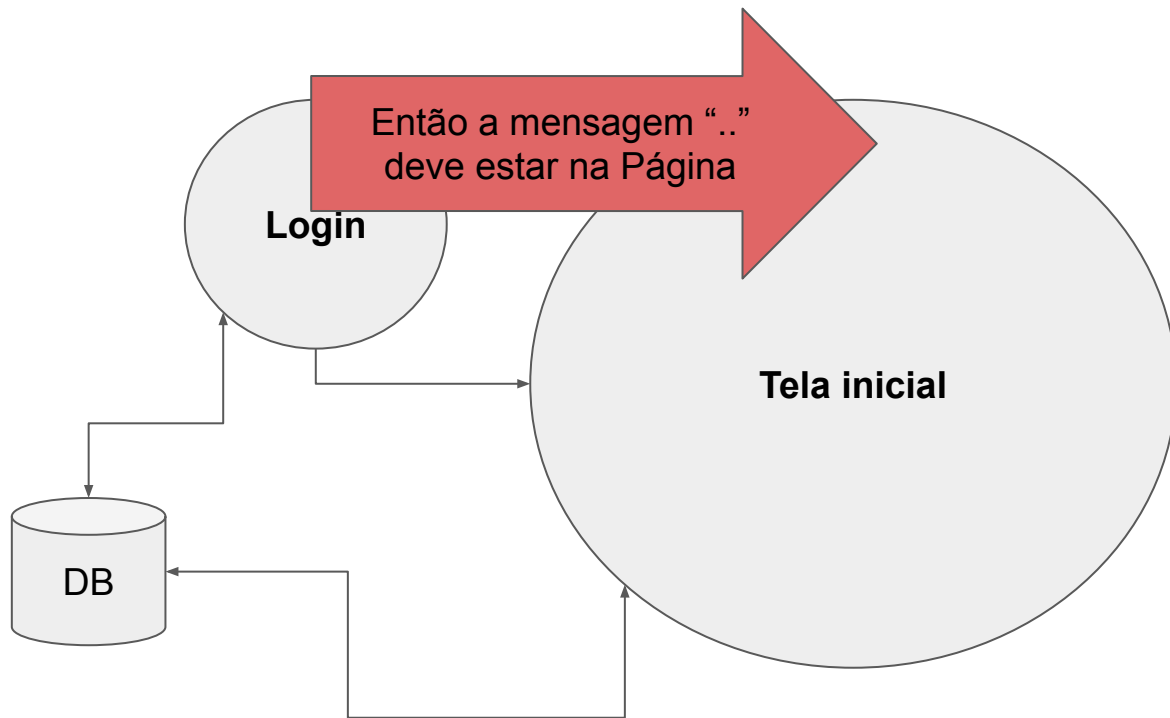
Exemplificando



Exemplificando



Exemplificando



Behave (Gherkin Files)



Behave é um framework de BDD baseado em arquivos de features e segue os padrões do gherkin

```
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
```

Estrutura de diretórios do behave

