



BDD com Cucumber

Para automação de testes

BDD (Behavior Driven Development)



- Desenvolvimento Orientado a Comportamento
- BDD criado a partir da insatisfação do TDD (Test driven development)
- Necessidade de deixar as Regras de negócio mais claras para todo o time

TDD X BDD



Desenvolvimento em ciclos curtos de repetições:

- Escrever o teste que valide a funcionalidade que desejo construir
- Implementar a funcionalidade
- A funcionalidade deve passar pelo teste criado

TDD X BDD



Técnica de desenvolvimento ágil para obter uma compreensão mais clara dos comportamentos desejados no software:

- Utilizado durante a escrita e refinamento das histórias
- Todos os membros da equipe podem participar da escrita
- O que foi escrito será usado como guia na fase de desenvolvimento e teste

Objetivo do BDD



- Escrever testes que integram **regras de negócios** com a linguagem de programação
- Possibilitar que as funcionalidades do sistema sejam escritas em **linguagem natural**
- Focando no comportamento do Sistema
- Saber mais sobre o que deveria ser desenvolvido e testado

Vantagem do BDD



- **Comunicação entre equipes:** (DEV, Testador e PO falam a mesma linguagem)
- **Documentação dinâmica:** em equipes ágeis, administrar documentações é algo custoso, com BDD a documentação é criada durante o planejamento do produto sem esforço adicional

Exemplo de cenário

Funcionalidade: Sacar no caixa eletrônico

Cenário: Sacar dinheiro com cartão

Dado que insiro o meu cartão no caixa eletrônico

Entao deve aparecer a tela solicitando a senha **do** cartão

Quando informo a senha **do** cartão

E clico confirmo

Entao deve aparecer o menu principal

Quando seleciono a opção "**Saque**"

Entao deve aparecer a tela com o campo de valor de saque

E informo o valor

E clico confirmo

Entao deve aparecer a mensagem "**aguarde separando notas...**"

E o caixa deve expelir o valor solicitado

Interações entre usuário e sistema



```
USUARIO -> Dado que insiro o meu cartão no caixa eletrônico
SISTEMA -> Entao deve aparecer a tela solicitando a senha do cartao
USUARIO -> Quando informo a senha do cartao
USUARIO -> E clico confirmo
SISTEMA -> Entao deve aparecer o menu principal
USUARIO -> Quando seleciono a opcao "Saque"
SISTEMA -> Entao deve aparecer a tela com o campo de valor de saque
USUARIO -> E informo o valor
USUARIO -> E clico confirmo
SISTEMA -> Entao deve aparecer a mensagem "agaurde separando notas..."
SISTEMA -> E o caixa deve expelir o valor solicitado
```


Cucumber



- Criado inicialmente para incrementar testes em **Ruby**
- Utiliza uma linguagem própria chamada **Gherkin** para definir os casos de teste
- Gherkin é o formato para as especificações do cucumber que utiliza a linguagem do **BDD**

Gherkin



Funcionalidade - Contexto - Cenário

Dado - Quando - Então - E

Mas - Esquema do Cenário - Exemplos

Cucumber



- **Dado que (Given):** pré-condições
- **Quando (When):** eventos que devem ocorrer para que o cenário seja executado
- **Então (Then):** expectativas a respeito dos resultados da execução

Inserindo dados nos Steps

Cenário: Gerar títulos

Dado que gero um título com seguintes dados

TIPO	VALOR	DOCUMENTO	
PF	10.00	12345678909	

Esquemas: Reutilização de passos

```
Esquema do Cenário: Gerar titulo <TIPO_USUARIO>  
  Dado que gero um titulo para um usuario <TIPO_USUARIO>  
  ...
```

Exemplos:

TIPO_USUARIO	
PJ	
PF	

Contexto: evitar repetição de passos

Contexto:

Dado que acesso a tela de login do caixa
Quando informo cooperativa, usuario e senha
Entao o usuario deve acessar a tela inicial do caixa com sucesso

Cenario: Acessar operacao 12

Dado que acesso a operacao "12"
...

Cenario: Acessar operacao 18

Dado que acesso a operacao "18"
...

Automação com BDD



Arquivos que vão implementar os métodos que testam as regras de negócio com o BDD:

- Arquivos de feature (funcionalidade)
- Arquivos de definição de passos (Steps)

Automação com BDD



Classes Runner: classes que executam o teste BDD

Arquivos de feature: Arquivo com o BDD que descreve as regras de negocio

Classes Step: Classe que implementam cada passo (step) do BDD

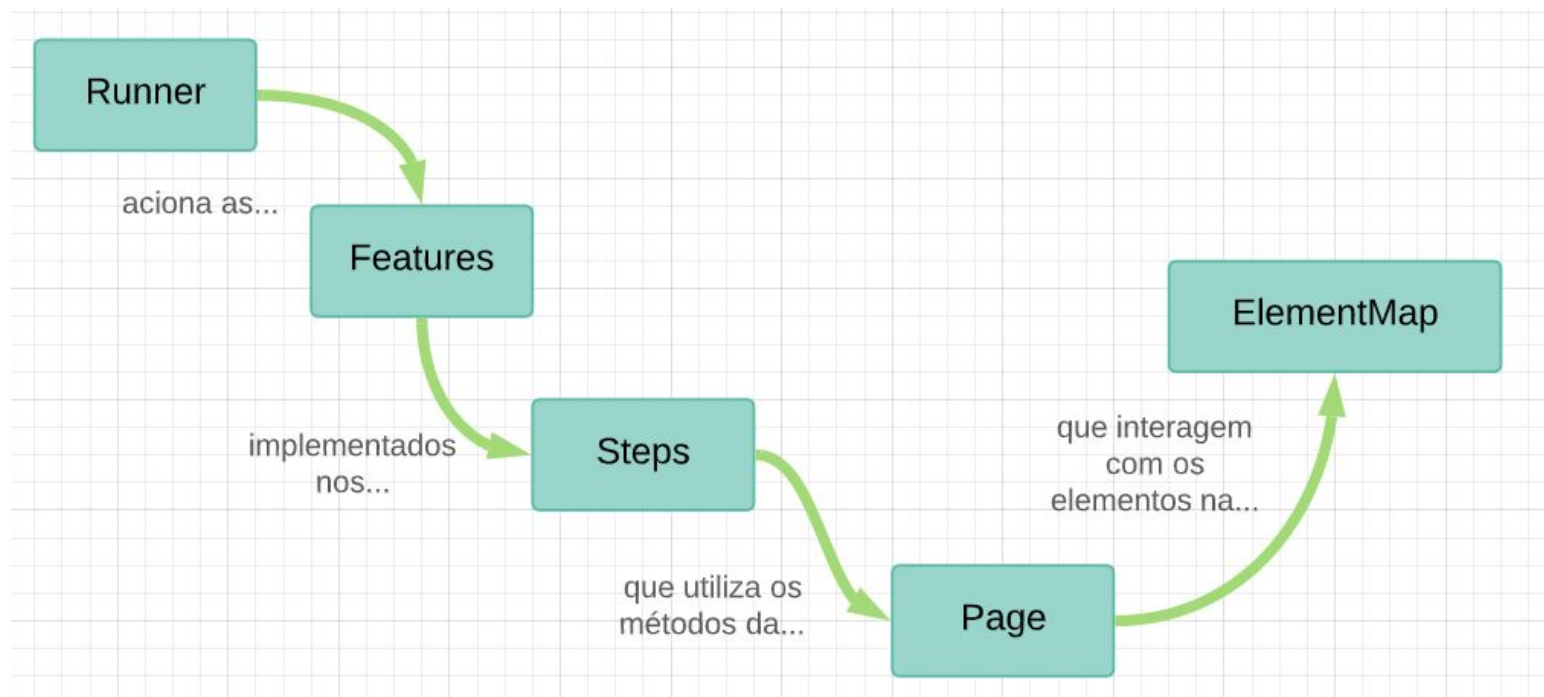
Automação com BDD



Classes Pages: Classes com os metodos que implementam as regras de negocio interagindo com o sistema

Classes ElementMap: Classe opcional que mapeia cada elemento da tela do sistema (é utilizado pela classes pages)

Fluxo de um teste com Cucumber



Referências



<https://cucumber.io/>

<https://dannorth.net/introducing-bdd/>

<https://www.concrete.com.br/2014/12/16/introducao-bdd-e-cucumber/>

06/01/2020