



# BDD com Cucumber

Para automação de testes

# BDD: Behavior Driven Development



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

Criado a partir da insatisfação do TDD, pela necessidade de deixar as Regras de negócio mais claras para todo o time.

# TDD X BDD



Desenvolvimento ocorre em ciclos curtos de repetições:

1. Escrever o teste que valide a funcionalidade que desejo construir;
2. Implementar a funcionalidade;
3. A funcionalidade deve passar pelo teste criado.

# TDD X BDD



1. Durante a definição e refinamento das histórias escreva utilizando as palavras padrão (Dado, Quando e Então);
2. O que foi escrito será usado como guia na fase de desenvolvimento e teste.

**Obs.:** Todos os membros da equipe podem e devem participar da escrita!

# Vantagem do BDD



- Escrever histórias que integram regras de negócios com a linguagem de programação;
- Possibilitar que as funcionalidades do sistema sejam descritas em linguagem natural;
- Comunicação entre equipes;
- Documentação dinâmica e sem esforço adicional.

# Exemplo de cenário



```
Dado que insiro o meu cartão no caixa eletrônico
Então deve aparecer uma mensagem na tela solicitando a senha do cartão
Quando informo a senha do cartão
E cliço na opção "Confirmar"
Então deve aparecer a tela do menu principal
Quando seleciono a opção "Saque"
Então deve aparecer a tela com o campo de valor de saque
Quando informo o valor de saque
E cliço na opção "Sacar"
Então 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 -> Então deve aparecer uma mensagem na tela solicitando a senha do cartão
USUARIO -> Quando informo a senha do cartão
USUARIO -> E clico na opção "Confirmar"
SISTEMA -> Então deve aparecer a tela do menu principal
USUARIO -> Quando seleciono a opção "Saque"
SISTEMA -> Então deve aparecer a tela com o campo de valor de saque
USUARIO -> Quando informo o valor de saque
USUARIO -> E clico na opção "Sacar"
SISTEMA -> Então deve aparecer a mensagem "Aguarde 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**.



# Cucumber



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

# Gherkin



Funcionalidade - Cenário - Esquema do Cenário

Dado - Quando - Então - E - Mas

Contexto - Exemplos

# Teste Automatizado sem Cucumber

```
@Test
public void realizarUmaPesquisaNoGithub() {
    page.acessarPaginaInicial("https://github.com/");
    page.deveCarregarPaginaInicial();
    page.pesquisaTermoNoSite("teste");
    page.deveCarregarPaginaComResultados();
}
```

# Teste Automatizado com Cucumber

@teste

Cenário: Realizar uma pesquisa no Github

Dado que acesso o site "https://github.com/"

Entao a pagina inicial deve carregar

Quando pesquiso pelo termo "teste"

Entao deve carregar a pagina com resultados da pesquisa

# Teste Automatizado com Cucumber



```
@Dado("que acesso o site {string}")
public void queAcessoSite(String site) {
    page.acessarPaginaInicial(site);
}

@Entao("a pagina inicial deve carregar")
public void paginaInicialDeveCarregar() {
    page.deveCarregarPaginaInicial();
}

@Quando("pesquiso pelo termo {string}")
public void pesquisoPeloTermo(String termo) {
    page.pesquisoTermoNoSite(termo);
}

@Entao("deve carregar a pagina com resultados da pesquisa")
public void deveCarregarPaginaComResultados() {
    page.deveCarregarPaginaComResultados();
}
```

# Inserindo dados nos Steps

Cenário: Gerar títulos

Dado que gero um titulo com seguintes dados

TIPO	VALOR	DOCUMENTO
PF	10.00	12345678909
PF	12.00	01231234557
PJ	19.00	32165498777

Quando acesso a tela de login

E informo os dados de login

...

# Esquemas: Reutilização de passos

Esquema de Cenário: Gerar titulo <TIPO\_USUARIO>

Dado que gero um titulo para um usuario <TIPO\_USUARIO> com o valor <VALOR>

Quando acesso a tela de login

E informo os dados de login para o usuario <TIPO\_USUARIO>

...

Exemplos:

TIPO_USUARIO	VALOR
PJ	0,01
PF	13,00
PF	999.999,00

# Contexto: evitar repetição de passos

Contexto:

Dado que acesso a tela de login de 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"  
Quando informo o numero da conta  
...

Cenario: Acessar operacao 18

Dado que acesso a operacao "18"  
Quando gero um boleto com o valor padrao  
...



# Automação com BDD



**Classes Runner:** Classe que executa o teste BDD;

**Arquivos de feature:** Arquivo com o BDD que descreve as regras de negócio;

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

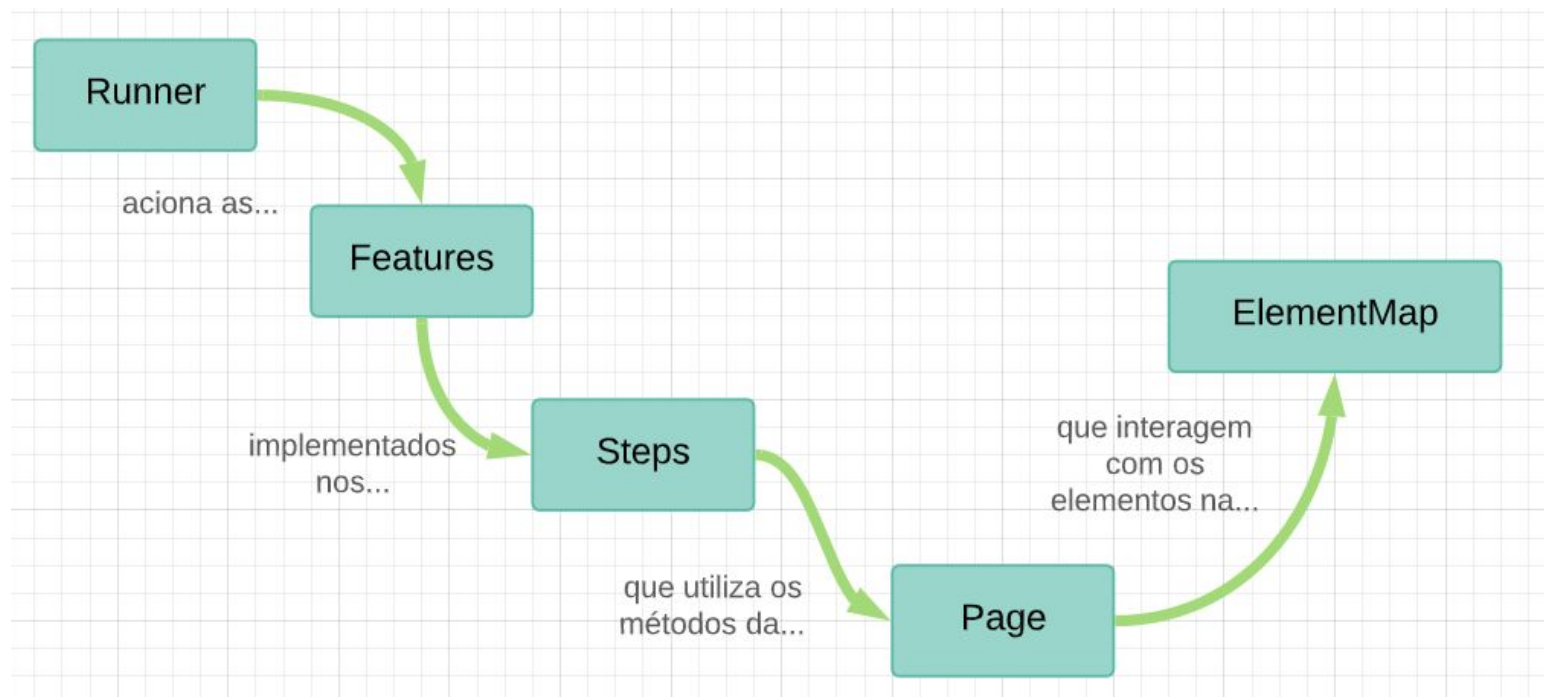
# Automação com BDD



**Classes Pages:** Classes com os métodos que implementam as regras de negócio interagindo com o sistema;

**Classes ElementMap:** Classe opcional que mapeia cada elemento da tela do sistema para fins de reutilização (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