



# RESUMO BDD e Gherkin

BDD - Desenvolvimento baseado em comportamento

Técnica de desenvolvimento do produto, incentiva a focar no **comportamento** do produto (site ou app) e trazer profissionais de diferentes áreas para discutir as funções que serão feitas (ex: um dev, um tester, um chefe) e assim pensar no software e nos testes antes de desenvolver.

Isso será feito pelo modelo de escrita Gherkin (cucumber) listando todos passos da aplicação

## 4. Funcionalidade

- O início do Gherkin deve se dar por uma Funcionalidade, ou seja, a qual funcionalidade aquele cenário(s) pertencem.
  - Funcionalidade: Autenticação
    - Cenário: Realizar login
      - ...
    - Cenário: Realizar cadastro
      - ...

# Contexto

Algumas vezes, para testar uma funcionalidade você precisará repetir uma coleção de steps antes de cada cenário. Vamos supor que em todos os seus cenários você precisará cadastrar um usuário e sabemos que não será muito eficiente ficar duplicando estas steps em todos os cenários, então podemos utilizar o Contexto para executar um conjunto de steps antes de cada cenário.

Exemplo:

Contexto:

Dado que eu esteja deslogado na página inicial

Cenário: Usuário deslogado tenta publicar no blog

Quando eu tentar postar em "Terapia Cara"

Então eu devo ver "Você deverá se logar para publicar"

Cenário: Wilson posta em seu próprio blog

Dado que eu realize o login como Wilson

Quando eu tentar postar em "Terapia Cara"


Então eu devo ver "Seu artigo foi publicado."

Neste exemplo seria como se o Dado fosse a primeira step de cada Cenário, ele será executado antes de cada Cenário.

## cenário

Cenários são uma das principais estruturas do Gherkin. Todo cenário deve iniciar com a palavra chave **Cenário:**, opcionalmente seguido de um título de cenário. Cada funcionalidade pode ter um ou mais cenários e todo cenário consiste em uma ou mais etapa.

Os cenários seguintes tem cada um 3 etapas:

 **Cenário:** Wilson posta em seu blog  
Dado que eu estou logado como Wilson  
Quando eu tento postar "A terapia cara"  
Então eu devo ver "Seu artigo foi publicado."

**Cenário:** Wilson falha ao postar algo no blog de outra pessoa  
Dado que eu estou logado como Wilson  
Quando eu tento postar "Greg esbraveja contra impostos"  
Então eu devo ver "Hey! Este não é o seu blog!"

**Cenário:** Greg posta em blog cliente  
Dado que eu estou logado como Greg  
Quando eu tento postar "Terapia Cara"  
Então eu devo ver "Seu artigo foi publicado."

“Dado”, “Quando” e “Então”. Assim como apresentado no exemplo abaixo:

*Dado que eu tenha X*

*Quando eu faço Y*

*Então eu recebo Z*

Contudo, o cucumber não sabe diferenciar essas 3 palavras em seus steps, porém é fortemente recomendado, para que elas sejam utilizadas da forma correta, pois a ideia é que quem a utilize tenha o mindset do BDD.

E cada uma dessas palavras propõe um mindset diferente.

### **E (And)**

Visa complementar qualquer uma das palavras chaves citadas anteriormente, por exemplo, ao escrever um cenário dessa forma:

*Dado que eu tenha X*

*Dado que eu tenha Y*

*Quando eu faço A*

*Quando eu faço B*

*Então eu recebo C*

Poderia ser implementado dessa forma

*Dado que eu tenha X*

*E que eu tenha Y*

*Quando eu faço A*

*E eu faço B*

*Então eu recebo C*

O Propósito do “E” é tornar o cenário ainda mais legível.

### **Mas (But)**

O “Mas” seria a forma negativa do “Então”, basicamente, quando o resultado esperado for que o usuário NÃO DEVE receber / ter algo, então é recomendado utilizar o “Mas” para também tornar o cenário mais legível.

E por isso cada uma dessas palavras chaves devem ser aplicadas corretamente, pois através delas, é possível melhorar o mindset da utilização do BDD, e consequentemente, aplicar de maneira mais correta o desenvolvimento de software orientado a comportamento para uma maior garantia de qualidade.

## 2. Siga a ordem de palavras-chave rigorosamente:

- **Dado:** a palavra-chave 'dado' indica o local onde o cenário de teste inicia...
- Se a automação for de front-end, o local será uma página (se for mobile, vai ser o mesmo padrão de front-end, mas ao invés de páginas, serão telas)
  - Dado que esteja na home
  - Dado que esteja na página de cadastro
- Se a automação for de back-end, o 'dado' ficará para geração de dados (caso não seja necessário gerar dados para o teste, a automação não utilizará a palavra-chave)
  - **Geração de dados:**
    - **Dado** que possua um usuário com permissões administrativas
  - **Sem geração de dados:**
    - Começará no **quando:** Quando fazer um get para obter dados do cep
- **Quando:** a palavra-chave 'quando' indica as ações que serão tomadas durante aquele cenário de teste, somente com essa palavra-chave deve ser executadas as tarefas do cenário.
  - **Quando** realizar login
  - **Quando** realizar cadastro
  - **Quando** entrar na tela do produto
- **Então:** a palavra-chave 'então' representa a espera de um retorno da aplicação, como uma evidência do que foi feito no 'quando'...
  - **Então** deverá ser exibido o nome do usuário na tela

- **Então** deverá ser exibida a mensagem “Bem vindo!”
  - **Então** o produto deverá ser adicionado ao carrinho
- As próximas duas palavras-chaves podem ser utilizadas logo abaixo de qualquer outra das três citadas a cima:
- **E:** Contêm o mesmo valor da palavra-chave acima, exemplo:
  - **Dado** que esteja na Home Page
  - **E** que esteja logado
- Equivale a escrever:
  - **Dado** que esteja na Home Page
  - **Dado** que esteja logado
- **Mas:** a palavra-chave ‘mas’ representa uma *exception* dentro do cenário testado, exemplo:
  - **Dado** que esteja na página principal
  - **Quando** realizar cadastro
  - **Mas** o e-mail já estiver cadastrado
  - **Então** deverá ser exibido a mensagem “E-mail já cadastrado”
- Padrão de escrita para não errar mais:
  - Dado (que esteja) [que esteja em alguma lugar]
  - Quando (verbo com ar,er,ir) [realizar uma ação]
  - Então (deverá...) [esperar um retorno]

## esquema do cenário

Copiar e colar cenários para usar diferentes valores pode ser muito tedioso e repetitivo:

```
⚙ Cenário: Comer 5 em cada 12
  Dado que tenho 12 pepinos
  Quando eu comer 5 pepinos
  Então eu devo ter 7 pepinos


Cenário: Comer 5 em cada 20
  Dado que tenho 20 pepinos
  Quando eu comer 5 pepinos
  Então eu devo ter 15 pepinos
```

Os *Esquemas do Cenários* nos permitem formular estes exemplos com maior precisão através da utilização de um modelo com espaços reservados

```
⚙ Esquema do Cenário: Comendo
  Dado que tenho <antes> pepinos
  Quando eu comer <come> pepinos
  Então eu devo ter <depois> pepinos

Exemplos:
  | antes | come | depois |
  | 12   | 5    | 7      |
  | 20   | 5    | 15     |
```

As etapas do Esquema do Cenário fornecem um modelo que nunca é executado diretamente. Um Esquema do Cenário é executado uma vez para cada linha na seção de exemplos abaixo dela (exceto para a primeira linha que é o cabeçalho).

O Esquema do Cenário utiliza espaços reservados, que estão contidos  nas etapas de saída do Cenário. Por exemplo:

## GUIA MAIS DETALHADO CUCUMBER

<https://cucumber.io/docs/gherkin/reference/>

SITES COM EXEMPLOS:

<http://shipit.resultadosdigitais.com.br/blog/5-boas-praticas-para-uso-de-cucumber/>