

Qualiters Club

Material produzido por: Priscila Caimi

Resumo

Curso	Gherking do básico ao Avançado
Jornada	Aprenda sobre Teste de Software
Trilha	Teste de Software

Pré requisito

☐ Nenhum

Objetivos de Aprendizagem

Aprenda do básico ao avançado sobre Gherking, a linguagem universal de escrita para os seus testes manuais e/ou automatizados.

Você vai aprender

Aula

O que é Gherking

Keywords básica (DADO, QUANDO, ENTÃO, E , MAS)

Keywords intermediárias (Funcionalidade, Cenário, Contexto, Esquema do Cenário)

Keywords avançados: utilizando os steps secundários

Prática

Observações

Links uteis

- Para validar a tradução das keywords: <https://cucumber.io/docs/gherkin/languages/>
- Referência do Gherking: <https://cucumber.io/docs/gherkin/reference/#scenario-outline>

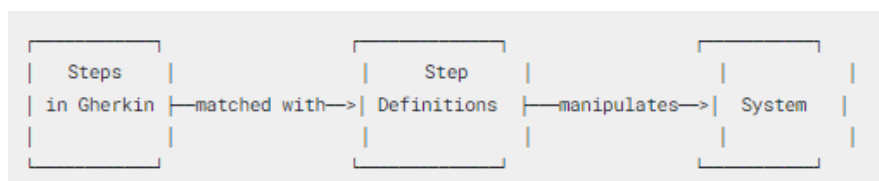
O que é Gherking

Gherking é uma padrão de escrita universal com suporte para muitas linguagens. Muitas vezes é confundido com o BDD, porém o Gherking é uma escrita universal que pode ser utilizado dentro da metodologia BDD (Behavior driven developer).

As **vantagens** de se utilizar o **Gherking** na sua escrita de validação é gerar um documentação clara e com padrão de escrita, possibilidade de reaproveitar a validação para a sua automação e evitar documentações ambíguas.

A escrita do Gherking é baseada em **Steps** onde o início de cada step possui uma **Keyword** padrão que irá determinar a ação ou validação que deve ser realizada.

Para quem já possui conhecimento em automação, o fluxo abaixo mostra como funciona os steps quando implementados dentro do fluxo da automação. O primeiro quadrado **Steps in Gherkin** indica a escrita do step → está escrita irá ter uma conexão com o **Step definitions** declarado no seu código de automação → que irá manipular os dados do seu **System**.



Referência da imagem: <https://cucumber.io/docs/guides/overview/>

Exemplo:

Step In Gherking → When {infor} starts a game

Step Definition → transformando seu step em código de automação

```
When("{maker} starts a game", function(maker) {  
    maker.startGameWithWord({ word: "whale" })  
})
```

System → código do método maker.startGameWithWord({ word: "whale" })

Então a utilização do Gherking irá te ajudar a escrever a validação dos seus cenários manuais de forma clara e intuitiva e no futuro você pode reaproveitar a mesma escrita para escrever seu código de automação.

Keywords básica (DADO, QUANDO, ENTÃO, E , MAS)

Vamos iniciar a nossa escrita pelos comando básico. Estes comandos são simples e você pode já começar a praticar agora mesmo nas validações que você já escreve.

GIVEN ou DADO

A primeira Keyword que vamos aprender é o **DADO**, através dele é indicado o local inicial da sua validação, com ele você irá indicar o contexto inicial do seu sistema.

Alguns exemplos

DADO que sou aluno da plataforma Qualiters Club

DADO que estou no curso Gherking do básico ao Avançado

DADO que moro na cidade X

Em todos os exemplos citados acima foi descrito após a Keyword **DADO** a localidade (contexto) em que você se encontra no momento.

WHEN ou QUANDO

A nossa próxima keyword é o **QUANDO** e aqui ela tem como função indicar a ação que será realizada no sistema e não será qualquer ação, será a ação principal do sistema que irá gerar um resultado (a nossa validação).

Alguns exemplos:

QUANDO adiciono meus dados de acesso → você terá um resultado após completado esta ação

QUANDO eu finalizo as aulas do curso → você terá um resultado após completado esta ação

Então sempre que for realizar uma ação no seu sistema e está ação for gerar a validação você devem implementar o **QUANDO** no inicio do seu step.

THEN ou ENTÃO

A keyword **ENTÃO** ela vem na sequencia da **QUANDO** pois como visto anteriormente o quando tem a funcionalidade a ação e sempre após uma ação você tem a reação e no nosso caso é o resultado que deve ser valido, e para isso utilizamos a keyword **ENTÃO** para descreve que terá uma validação a ser feita.

Vamos juntar os exemplos anteriores para fazer sentido?

***DADO** que sou aluno da plataforma Qualiters Club → contexto de onde o usuário deve se encontrar*

***QUANDO** adiciono meus dados de acesso → você terá um resultado após completado esta ação*

***ENTÃO** o login é realizo na plataforma → resultado da ação*

AND ou E

Keyword **E** é utilizada para informar um step importante de ser realizado, porém ele não é a ação principal. Eu gosto de dizer que ele funciona como um agrupado, para agrupar uma ação da outra.

Vamos de exemplo:

***DADO** que sou aluno da plataforma Qualiters Club → contexto de onde o usuário deve se encontrar*

QUANDO adiciono meus dados de acesso → você terá um resultado após completado esta ação

E adiciono os dados de acesso correto → indicando a massa de teste a ser usada

ENTÃO o login é realizado na plataforma → resultado da ação

Nos exemplo acima vocês podem ver que o **E** foi utilizada para agrupar a ação de adicionar os dados e os dados que foram utilizados no teste.

BUT ou MAS

O **MAS** funciona como um contra ponto positivo no seu teste que teve a sua validação (**ENTÃO**) como negativa.

Vamos exemplificar para fazer sentido, iremos utilizar o mesmo exemplo anterior, porém desta vez utilizando uma validação negativa.

DADO que sou aluno da plataforma Qualiters Club

QUANDO adiciono meus dados de acesso

E adiciono os incorreto para login

ENTÃO o login não é realizado na plataforma → teste validando um cenário negativo

MAS um toast de erro é exibido na tela → o mas trazendo um contra ponto que também é um resultado esperado

Keywords intermediárias (Funcionalidade, Cenário, Contexto, Esquema do Cenário)

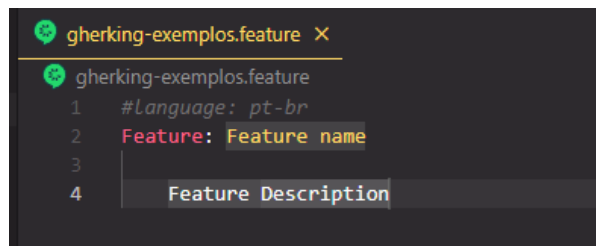
Comandos intermediários nos auxiliam quando nossas validações começam a ficar mais complexo como:

- reaproveitar pedaços de testes que são comuns em vários cenários
- a mesma validação necessita utilizar várias massas de teste
- várias validações partem do mesmo contexto

Vamos começar a estudar uma por uma delas.

Feature ou Funcionalidade

Feature representa a funcionalidade Pai que será testada. Ao utilizar esta keyword você está evoluindo a sua escrita e passa a utilizar uma arquitetura mais complexa, porém você passa a ter uma descrição das validações que serão adicionadas a este arquivo.



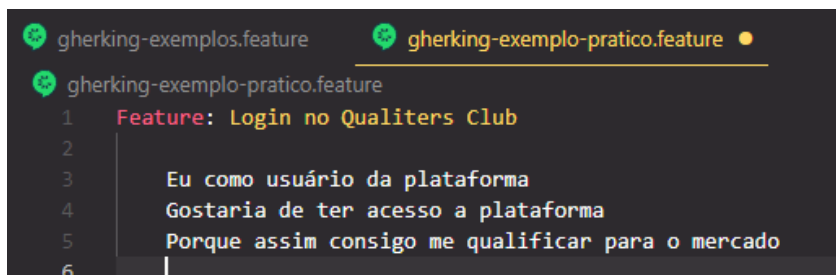
```
gherking-exemplos.feature X
gherking-exemplos.feature
1 #Language: pt-br
2 Feature: Feature name
3
4 Feature Description
```

Referência da imagem: da própria autora

Na imagem acima é ilustrado a forma como é a declaração da Feature, ela segue uma estrutura simples:

- Feature: deve ser adicionado o nome daquela funcionalidade que será testada
- Feature description: uma breve descrição do que será validado

Abaixo irei mostrar um exemplo prático que nós estamos construindo ao longo deste curso.



```
gherking-exemplos.feature gherking-exemplo-pratico.feature ●
gherking-exemplo-pratico.feature
1 Feature: Login no Qualiters Club
2
3 Eu como usuário da plataforma
4 Gostaria de ter acesso a plataforma
5 Porque assim consigo me qualificar para o mercado
6
```

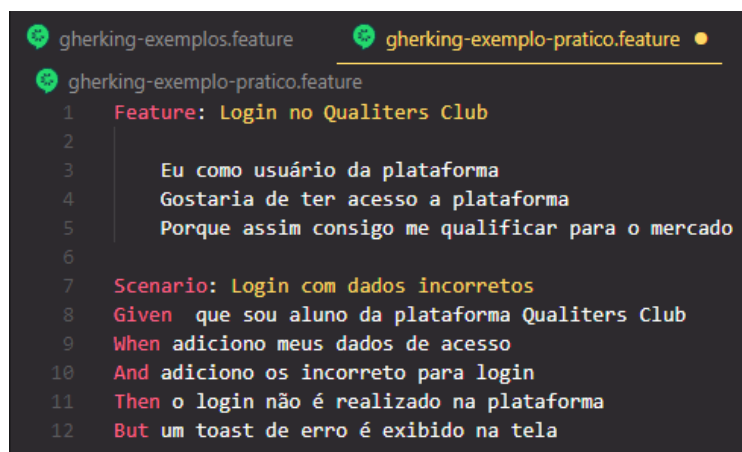
Referência da imagem: da própria autora

No exemplo vocês podem ver que um nome foi dado a funcionalidade e logo abaixo foi apresentado a descrição da funcionalidade. Isso irá indicar o que deve ser validado logo abaixo.

Scenario ou Cenário

Chegamos ao **Cenário** e como diz o nome é onde vai agrupar a nossa validação, da mesma forma da **Feature** aqui nos também precisamos adicionar um nome a nossa validação, e o mais importante o nome tem que ser condizente com a validação que se encontra nele.

Abaixo está indicado o nosso exemplo anterior de login com dados incorretos, porém agora está na com uma estrutura mais complexa.



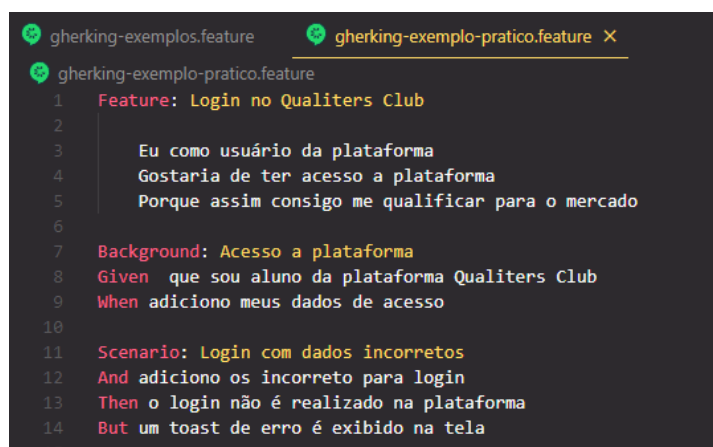
```
gherking-exemplos.feature  gherking-exemplo-pratico.feature ●
gherking-exemplo-pratico.feature
1  Feature: Login no Qualiters Club
2
3      Eu como usuário da plataforma
4      Gostaria de ter acesso a plataforma
5      Porque assim consigo me qualificar para o mercado
6
7  Scenario: Login com dados incorretos
8  Given que sou aluno da plataforma Qualiters Club
9  When adiciono meus dados de acesso
10 And adiciono os incorreto para login
11 Then o login não é realizado na plataforma
12 But um toast de erro é exibido na tela
13
```

Referência da imagem: da própria autora

Background ou Contexto

Agora vamos aumentar um pouco mais o nível de complexidade. Sabe quando você tem várias validações a fazer porém todas partem do mesmo lugar? ou tem vários steps em comum? Sabia que não é uma boa prática ficar duplicando a escrita? Isso faz com que quando for necessário dar manutenção, você tem que reescrever em vários cenários diferentes aumentando assim o risco de erros e escritas desatualizadas.

Para isso, podemos utilizar o contexto desta forma todo o bloco de código que é comum entre os cenários é adicionado a ele. Assim, sempre que tiver contexto o teste irá iniciar nele e após irá para o cenário.



```
gherking-exemplos.feature  gherking-exemplo-pratico.feature X
gherking-exemplo-pratico.feature
1  Feature: Login no Qualiters Club
2
3      Eu como usuário da plataforma
4      Gostaria de ter acesso a plataforma
5      Porque assim consigo me qualificar para o mercado
6
7  Background: Acesso a plataforma
8  Given que sou aluno da plataforma Qualiters Club
9  When adiciono meus dados de acesso
10
11 Scenario: Login com dados incorretos
12 And adiciono os incorreto para login
13 Then o login não é realizado na plataforma
14 But um toast de erro é exibido na tela
15
```

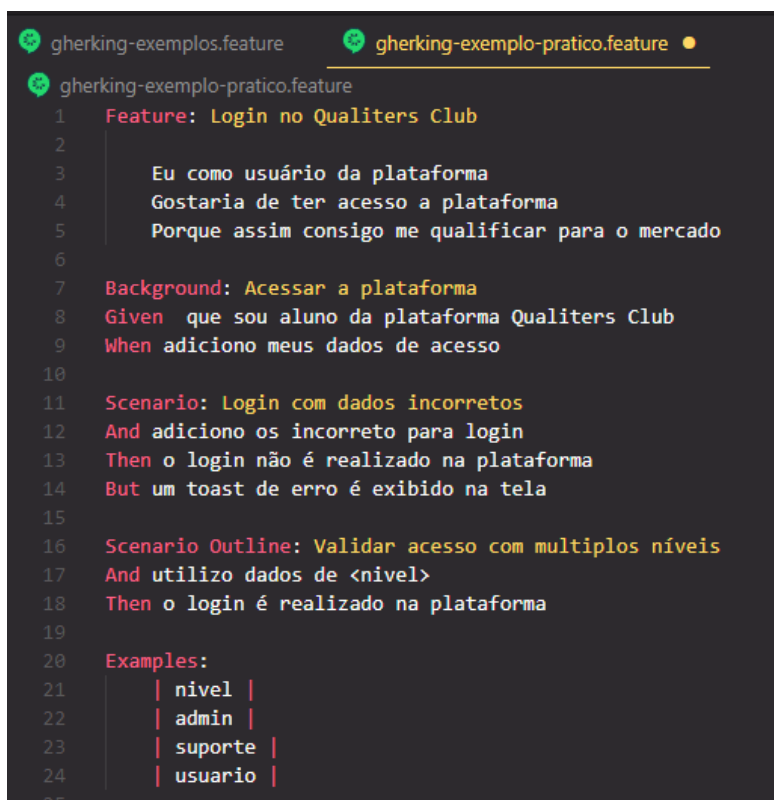
Referência da imagem: da própria autora

No exemplo acima eu mostro a evolução da nossa validação inicial para o nível de **Contexto**. Agora todos os cenários de login desta **Funcionalidade** deverão primeiro passar pelo

Contexto que irá ter os steps iniciais e depois ir para o Cenário que vai ter a validação específica do teste em questão.

Scenario Outline ou Esquema do Cenário

A última evolução da nossa escrita de cenário. Nós utilizamos **Esquema do Cenário** para casos em que utilizamos a mesma validação, porém com diferentes dados para o teste (massa de teste) em vez de escrever um cenário para cada uma destas massas de teste, nós iremos utilizar o **Esquema do Cenário**.



```
gherking-exemplos.feature  gherking-exemplo-pratico.feature ●
gherking-exemplo-pratico.feature
1  Feature: Login no Qualiters Club
2
3      Eu como usuário da plataforma
4      Gostaria de ter acesso a plataforma
5      Porque assim consigo me qualificar para o mercado
6
7  Background: Acessar a plataforma
8  Given que sou aluno da plataforma Qualiters Club
9  When adiciono meus dados de acesso
10
11 Scenario: Login com dados incorretos
12 And adiciono os incorreto para login
13 Then o login não é realizado na plataforma
14 But um toast de erro é exibido na tela
15
16 Scenario Outline: Validar acesso com multiplos níveis
17 And utilizo dados de <nível>
18 Then o login é realizado na plataforma
19
20 Examples:
21 | nível |
22 | admin |
23 | suporte |
24 | usuario |
25
```

Referência da imagem: da própria autora

No exemplo acima mostro como é feita a estrutura do **Esquema do Cenário**, vamos no passo a passo:

- Assim como os outros após o **Esquema do Cenário** deve ter o nome do cenário que será validado
- Uso de **tags**, quando é adicionado a massa de teste que será utilizada no teste ela é usada entre <> e dentro das tags vai o nome da massa, no nosso caso é **nível** que significa que a cada execução um nível diferente de usuário será utilizado no teste
- **Examples**, aqui será adicionado o nome da tag e os dados que será utilizado, cada linha é um dado diferente.
 - No exemplo, a primeira linha está o nome da tag (no nosso caso nível)

- Nas linhas abaixo, cada linha é um dado, primeiro é admin, segundo é suporte e terceiro usuário
- Pode se ter várias tags, não existe número máximo ou mínimo para a escrita.

A execução do teste fica da seguinte forma:

- Executar o contexto
- Iniciar a execução do contexto validando o acesso de usuário com nível admin
- Executar o contexto
- Segunda execução com usuário com nível de suporte
- Executar o contexto
- Terceira execução com usuário de nível usuário

Keywords avançados: utilizando os steps secundários

Para compor a nossa evolução, alguns steps secundários são utilizados para compor, entre eles a utilização de string (""), tabelas (| |), utilização de tags (@) e comentários (#) e uso de asterisco (*).

Utilização de string

Está keyword é importante quando entramos na automação de teste e estamos lidando com **Esquema do Cenário** e precisamos adicionar dados ao nosso teste onde é um texto, ele deve se adicionado desta forma "seu texto aqui".

```
16 Scenario Outline: Validar acesso com multiplos níveis
17 And utilizo dados de <nivel>
18 Then o login é realizado na plataforma
19
20 Examples:
21 | nivel |
22 | "admin" |
23 | "suporte" |
24 | "usuario" |
```

Referência da imagem: da própria autora

Utilização de tabelas

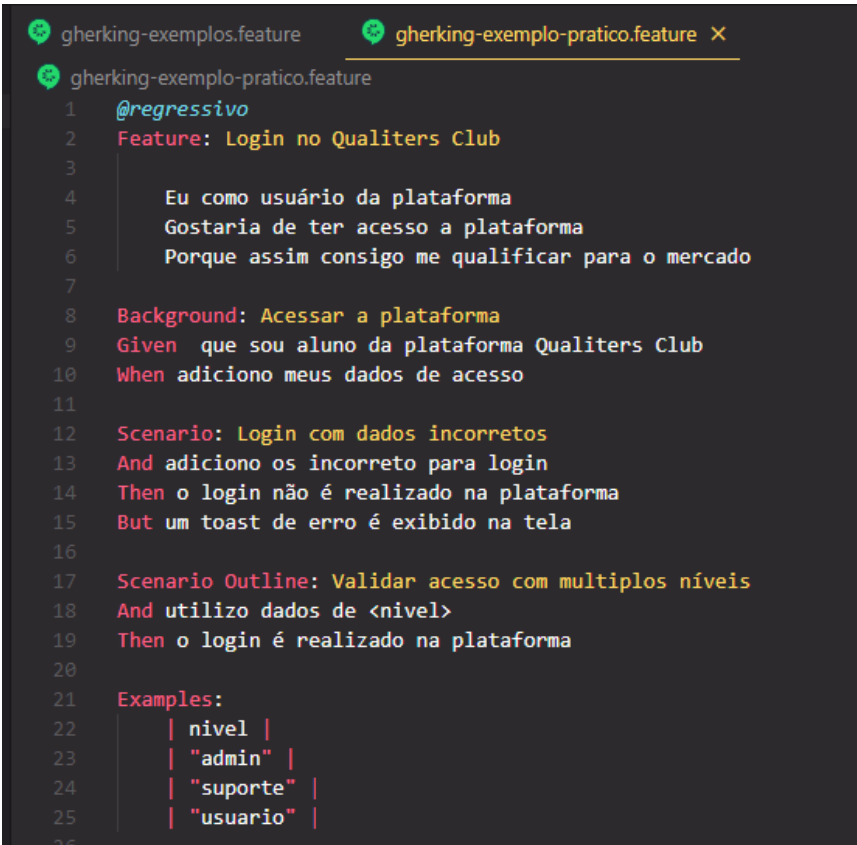
A utilização de tabelas também é realizada quando utilizado **Esquema do Cenário** desta forma além de tabelas os dados, quando for automatizar os testes o seu interpretador de código irá entender que cada coluna e linha é referente a um dado diferente.

Utilização de tags

Outro parâmetro muito utilizado na automação de teste para gerar filtros para execução. Você pode criar tags identificadores para determinadas situações como:

- Prioridade
- Severidade
- Regressivo

Assim ao colocar o @ em cima do teste, sempre quando solicitado para rodar os testes que possuem o @, somente eles serão executados.



```
gherking-exemplos.feature  gherking-exemplo-pratico.feature X
gherking-exemplo-pratico.feature
1  @regressivo
2  Feature: Login no Qualiters Club
3
4      Eu como usuário da plataforma
5      Gostaria de ter acesso a plataforma
6      Porque assim consigo me qualificar para o mercado
7
8  Background: Acessar a plataforma
9  Given que sou aluno da plataforma Qualiters Club
10 When adiciono meus dados de acesso
11
12 Scenario: Login com dados incorretos
13 And adiciono os incorreto para login
14 Then o login não é realizado na plataforma
15 But um toast de erro é exibido na tela
16
17 Scenario Outline: Validar acesso com multiplos níveis
18 And utilizo dados de <nível>
19 Then o login é realizado na plataforma
20
21 Examples:
22 | nível |
23 | "admin" |
24 | "suporte" |
25 | "usuario" |
```

Referência da imagem: da própria autora

Comentar

Você precisa comentar algum dado, ou colocar algum informação que será interpretada como erro? Basta colocar # antes do texto que o interpretador irá ignorar o que está depois dele.

```
gherking-exemplos.feature  gherking-exemplo-pratico.feature X
gherking-exemplo-pratico.feature
1  @regressivo
2  Feature: Login no Qualiters Club
3
4      Eu como usuário da plataforma
5      Gostaria de ter acesso a plataforma
6      Porque assim consigo me qualificar para o mercado
7
8      # adicionando um comentário aqui
9
10 Background: Acessar a plataforma
11 Given que sou aluno da plataforma Qualiters Club
12 When adiciono meus dados de acesso
13
14 Scenario: Login com dados incorretos
15 And adiciono os incorreto para login
16 Then o login não é realizado na plataforma
17 But um toast de erro é exibido na tela
18
19 Scenario Outline: Validar acesso com multiplos niveis
20 And utilizo dados de <nivel>
21 Then o login é realizado na plataforma
22
23 Examples:
24 | nivel |
25 | "admin" |
26 | "suporte" |
27 | "usuario" |
```

Referência da imagem: da própria autora