

Teste de software

Material produzido por: Gustavo Molina

Observações importantes

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>

Visual Studio Code - Extensões



Cucumber (Gherkin) Full Support v2.15.2

Alexander Krechik | 827.281 | ★★★★★ (26)

VSCode Cucumber (Gherkin) Full Language Support + Formatting + Autocomplete

[Desabilitar](#) [Desinstalar](#) ⚙️

Esta extensão foi habilitada globalmente.




Snippets and Syntax Highlight for Gherkin (Cucumber)

Euclidity | 156.045 | ★★★★★ (5)

Code snippets to write scenarios faster + Syntax highlight for .feature files

[Desabilitar](#) [Desinstalar](#) ⚙️

Esta extensão foi habilitada globalmente.



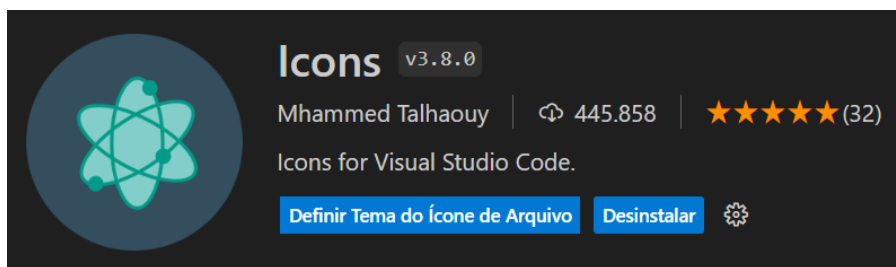
Gherkin Indent v2.0.0

Aravind Kumar | 52.293 | ★★☆☆☆ (12)

Gherkin Indent sets the indentation for Gherkin (cucumberjs) s

[Desabilitar](#) [Desinstalar](#) ⚙️

Esta extensão foi habilitada globalmente.



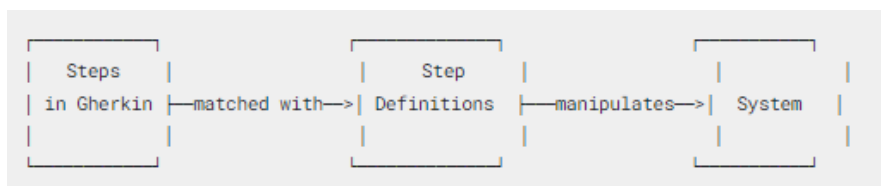
O que é Gherking

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 Development*).

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 → esta 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/>

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 faculdade UniPaulistana
- **DADO** que estou no curso de Análise e Desenvolvimento de Sistemas
- **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 esta ação for gerar a validação você devem implementar o **QUANDO** no início do seu step.

THEN ou ENTÃO

A keyword **ENTÃO** vem na sequência da keyword *QUANDO*, visto que é utilizada para descrever uma validação a ser feita.

Juntando as informações anteriores ...

DADO que sou aluno da faculdade Unipaulistana → contexto de onde o usuário deve se encontrar

QUANDO adiciono meus dados de acesso no portal do aluno → você terá um resultado após completado esta ação

ENTÃO o login é realizado no portal do aluno → resultado da ação

AND ou E

A keyword **E** é utilizada para informar um step importante a ser realizado. Ele nos dá a ideia de junção.

DADO que sou aluno da faculdade UniPaulistana → contexto de onde o usuário deve se encontrar

QUANDO adiciono meus dados de acesso no portal do aluno → você terá um resultado após completado esta ação

E adiciono os dados de acesso corretamente → indica a massa de teste a ser testada

ENTÃO o login é realizado no portal do aluno → resultado da ação

BUT ou MAS

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

DADO que sou aluno da faculdade UniPaulistana

QUANDO adiciono meus dados de acesso no portal do aluno

E adiciono os dados incorretos quando tento logar

ENTÃO o login não é realizado no portal do aluno → teste validando um cenário negativo

MAS uma mensagem de erro é exibida na tela → o mas traz um contra ponto que também é um resultado esperado