

Para conhecimento e entendimento do funcionamento das técnicas de BDD(Behavior-Driven-Development) foi elaborado um código utilizando Cucumber e Gherkin, de forma simples e direta, apenas com cenários de exemplo. Esse código inicial foi gerado em JavaScript. Segue código abaixo:

```
const assert = require('assert');
const { Given, When, Then } = require('cucumber');
const isItFriday = today => {
  if (today === "Friday") {
    return "TGIF";
  } else {
    return "Nope";
  }
}

Given('today is {string}', givenDay => {
  this.today = givenDay;
});

When('I ask whether it\'s Friday yet', () => {
  this.actualAnswer = isItFriday(this.today);
});

Then('I should be told {string}', expectedAnswer => {
  assert.equal(this.actualAnswer, expectedAnswer);
});
```

Utilizando os seguintes cenários, escritos utilizando Gherkin:

Feature: Is it Friday yet?

Everybody wants to know when it's Friday

Scenario Outline: Today is or is not Friday

Given today is "<day>"

When I ask whether it's Friday yet

Then I should be told "<answer>"

Examples:

| day | answer |
|----------------|--------|
| Friday | TGIF |
| Sunday | Nope |
| anything else! | Nope |

Dado o cenário em que esses códigos serão utilizados posteriormente para a realização do Trabalho de Conclusão de Curso, foi levantada a necessidade de alteração da linguagem do código para Python.

Sendo feito isso, segue código abaixo:

```
from behave import given, when, then

def is_it_friday(today):
    if today == "Friday":
        return "TGIF"
    else:
        return "Nope"

@given('today is {day}')
def step_given_day(context, day):
    context.today = day
    print(f'Given: today is "{day}"') # Adicionando instrução de
impressão

@when("I ask whether it's Friday yet")
def step_ask_if_friday(context):
    context.actualAnswer = is_it_friday(context.today)
    print(f'When: I ask whether it\'s Friday yet, actualAnswer is
"{context.actualAnswer}"') # Adicionando instrução de impressão

@then('I should be told {answer}')
def step_should_be_told(context, answer):
    assert context.actualAnswer == answer
    print(f'Then: I should be told "{answer}", actualAnswer is
"{context.actualAnswer}"') # Adicionando instrução de impressão
```

Para que o código funcionasse corretamente, foi necessário utilizar o Behave e fazer uma pequena alteração no arquivo .feature com os cenários para testes.

```
Feature: Is it Friday yet?
```

```
    Everybody wants to know when it's Friday
```

```
    Scenario Outline: Today is or is not Friday
```

```
        Given today is <day>
```

```
        When I ask whether it's Friday yet
```

```
Then I should be told <answer>
```

Examples:

| | |
|----------------|--------|
| day | answer |
| Friday | TGIF |
| Sunday | Nope |
| anything else! | Nope |

Para executar o código em JavaScript, basta utilizar o comando `npm test`.

Para o código em Python, basta utilizar o comando `behave`.

Repositório do GitHub para código em Javascript: [GitHub - sabrinachimidt/cucumber](https://github.com/sabrinachimidt/cucumber)

Repositório do GitHub para código em Python: [GitHub - sabrinachimidt/cucumber_python](https://github.com/sabrinachimidt/cucumber_python)