

Gherkin

Lenguaje para escribir las historias narrativas de casos de uso (en php es Behat).

Behat es Cucumber para phph*

Cada `feature` (característica) se define en un archivo, una característica normalmente consiste en una serie de escenarios, el texto entre Característica y "Esquema del escenario" permite definir libremente ya que este texto no será usado para ninguna funcionalidad de Cucumber, es meramente descriptivo.

Este código nos generará un esqueleto (dependiendo del lenguaje en el que vayamos a producir los test y el adaptador que estemos usando), el cual pasará los test con los distintos casos que le hayamos definido. Luego será nuestra tarea la de definir los casos de prueba que comprobarán las diferentes características.

Se pueden diferenciar claramente tres grupos que cumplen funciones distintas:

Feature: Aquí es donde indicamos el nombre de la característica a probar.

Scenario: Esta "keyword" describe los distintos escenarios que van a probar las diferentes posibilidades.

Given (Dado) : El propósito de los "Dado" es el de poner al sistema en el estado deseado para pasar los test deseados, antes de que el usuario (o un sistema externo), interactúe con el sistema.

When (Cuando): En este caso el propósito de los "Cuando" es el describir la acción que realiza el usuario, la cual vamos a probar.

Then (Entonces): El propósito de los "Entonces" es observar los resultados, estas observaciones han de ser realizadas desde el punto de vista de negocio y el valor para el usuario. Estas observaciones también deben de estar en algún sitio visible para el usuario, no en una base de datos o capas inferiores del sistema.

And, But (Y, Pero): Estas "keywords" están destinadas a aumentar la legibilidad y pueden ser usadas en vez de definir repetidas veces cualquiera de las anteriores.

Background (Antecedentes): Datos compartidos entre los escenario.

Ejemplo ##@

```
# language: es
Característica: adición
  Para evitar hacer errores tontos
  Como un ser humano
  Quiero saber la suma de los números

  Esquema del escenario: Sumar dos números
    Dado que he introducido <entrada_1> en la calculadora
    Y que he introducido <entrada_2> en la calculadora
    Cuando oprimo el <botón>
    Entonces el resultado debe ser <resultado> en la pantalla

  Ejemplos:
    | entrada_1 | entrada_2 | botón | resultado |
    | 20        | 30        | add   | 50        |
    | 2         | 5         | add   | 7         |
    | 0         | 40        | add   | 40        |
```

Comparando lexico entre idiomas

Un ejemplo en ingles

```
# language: us
[Feature|Business Need|Ability]: Internal operations
  In order to stay secret
  As a secret organization
```

We need to be able to erase past agents' memory

Background:

Given there is agent A

And there is agent B

Scenario: Erasing agent memory

Given there is agent J

And there is agent K

When I erase agent K's memory

Then there should be agent J

But there should not be agent K

[Scenario Outline|Scenario Template]: Erasing other agents' memory

Given there is agent <agent1>

And there is agent <agent2>

When I erase agent <agent2>'s memory

Then there should be agent <agent1>

But there should not be agent <agent2>

[Examples|Scenarios]:

agent1	agent2
--------	--------

D	M	
---	---	--

El mismo ejemplo en español

language: es

Característica: Internal operations

In order to stay secret

As a secret organization

We need to be able to erase past agents' memory

Antecedentes:

[Dadas|Datos|Dada|Dado] there is agent A

Y there is agent B

Escenario: Erasing agent memory

[Dadas|Datos|Dada|Dado] there is agent J

Y there is agent K

Cuando I erase agent K's memory

Entonces there should be agent J

Pero there should not be agent K

Esquema del escenario: Erasing other agents' memory

[Dadas|Datos|Dada|Dado] there is agent <agent1>

Y there is agent <agent2>

Cuando I erase agent <agent2>'s memory

Entonces there should be agent <agent1>

Pero there should not be agent <agent2>

Ejemplos:

agent1	agent2
--------	--------

D	M	
---	---	--

Instalando Behat

Agregar al `composer.json`, yo lo agregue en `require-dev`, las siguiente entras:

```
"behat/symfony2-extension": "*",
"behat/mink-extension": "*",
"behat/mink-browserkit-driver": "*"

```

Crear un archivo `behat.yml` en el root del proyecto

```
//behat.yml
default:
  suites:
    default:
      contexts:
        - FeatureContext:
            session: '@session'

      miapp:
        type: symfony_bundle
        bundle: AppBundle

  extensions:
    Behat\Symfony2Extension: ~

    Behat\Symfony2Extension: ~
    Behat\MinkExtension:
      default_session: 'symfony2'
      sessions:
        symfony2:
          symfony2: ~
```

Inicializar con:

```
bin/behat --init --suite=miapp
```