

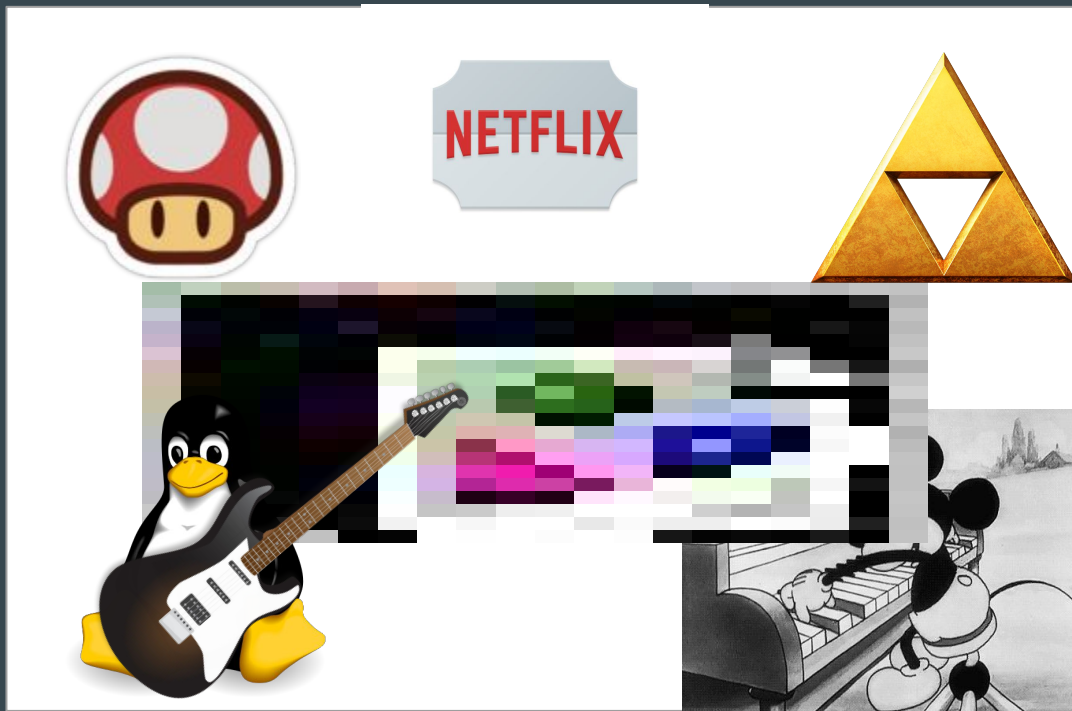
BDD com Cucumber e Watir

cucumber



Carlos Atkinson Ferreira Filho

Quem sou eu?



Mas o que é o BDD?



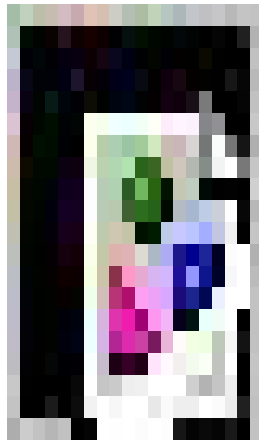
BDD

BDD – Behavior Driven Development (Desenvolvimento Guiado por Comportamento ou Funcionalidade)

“Behaviour-Driven Development is about implementing an application by describing its behavior from the perspective of its stakeholders.” – David Chelimsky, The RSpec Book.

BDD é

- Especificação por exemplo
- Alinhar negócio a tecnologia
- Gerar uma documentação viva (sempre atualizada, e executável)



Por que Ruby?

- Tem o foco na simplicidade e na produtividade
- Sintaxe elegante, leitura natural e de fácil escrita
- Várias bibliotecas para automação disponíveis (Watir, Capybara, Calabash, Selenium, Rautomation, Sikuli entre outras).



Como fazer?



Instale o Ruby e as gems!

Windows -> baixe o ruby installer em <http://rubyinstaller.org/> e siga os passos para a instalação.

Linux(Debian-Like) -> apt-get install ruby

Instale o cucumber , o watir e o rspec -> gem install cucumber watir-webdriver rspec

Watir

É uma biblioteca open-source para automações em navegadores.

Fácil de escrever e fácil de manter.

Suporta Chrome, Firefox, Internet Explorer, Opera.

Olá Mundo com Watir

olawatir.rb

```
1 require 'watir-webdriver'
2
3 #Inicia o watir com o browser desejado
4 browser = Watir::Browser.new :firefox
5 #Abre a página
6 browser.goto("www.google.com.br")
7 #Mapeia a caixa de texto de pesquisa para a variavel 'pesquisa'
8 pesquisa = browser.text_field(:name, 'q')
9 #Digita 'Ola mundo' na caixa de pesquisa
10 pesquisa.set("Ola mundo")
11 #Clica em enter
12 pesquisa.send_keys :enter
13 #Cria uma variavel resultados com uma collection de todos os links retornados na primeira página da pesquisa
14 resultados = browser.links(:css, 'h3 a')
15 #Espera carregar o primeiro link
16 resultados.first.wait_until_present
17 #Clica no primeiro link
18 resultados.first.click
19
```

Crie um novo projeto com o Cucumber

```
carlos@Kaff-VirtualBox:~$ mkdir testecucumber
carlos@Kaff-VirtualBox:~$ cd testecucumber/
carlos@Kaff-VirtualBox:~/testecucumber$ cucumber --init
  create  features
  create  features/step_definitions
  create  features/support
  create  features/support/env.rb
carlos@Kaff-VirtualBox:~/testecucumber$ tree
.
├── features
│   ├── step_definitions
│   └── support
│       └── env.rb
3 directories, 1 file
```

Crie a Especificação!

Crie um novo arquivo na pasta *features* chamado *pesquisa.feature*

```
# language: pt
@google
Funcionalidade: Pesquisa Google

Cenário: Posso pesquisar pelo Google
Dado que estou na página do Google
Quando pesquiso por "Ruby com Cucumber e Watir"
Então posso visualizar o resultado da busca
```

Rode o cucumber!

```
carlos@Kaff-VirtualBox:~/testecucumber$ cucumber
# language: pt
@google
Funcionalidade: Pesquisa Google

  Cenário: Posso pesquisar pelo Google                                # features/pesquisa.feature:5
    Dado que estou na página do Google                               # features/pesquisa.feature:6
    Quando pesquiso por "Ruby com Cucumber e Watir"                 # features/pesquisa.feature:7
    Então posso visualizar o resultado da busca                      # features/pesquisa.feature:8

1 scenario (1 undefined)
3 steps (3 undefined)
0m0.021s

You can implement step definitions for undefined steps with these snippets:

Dado(/^que estou na página do Google$/) do
  pending # Write code here that turns the phrase above into concrete actions
end

Quando(/^pesquise por "([^"]*)"$/) do |arg1|
  pending # Write code here that turns the phrase above into concrete actions
end

Então(/^posso visualizar o resultado da busca$/) do
  pending # Write code here that turns the phrase above into concrete actions
end
```

Implemente os steps no Cucumber!

```
# encoding: utf-8

Dado(/^que estou na página do Google$/) do
  pending
end

Quando(/^pesquisa por "([^"]*)"$/) do |texto|
  pending
end

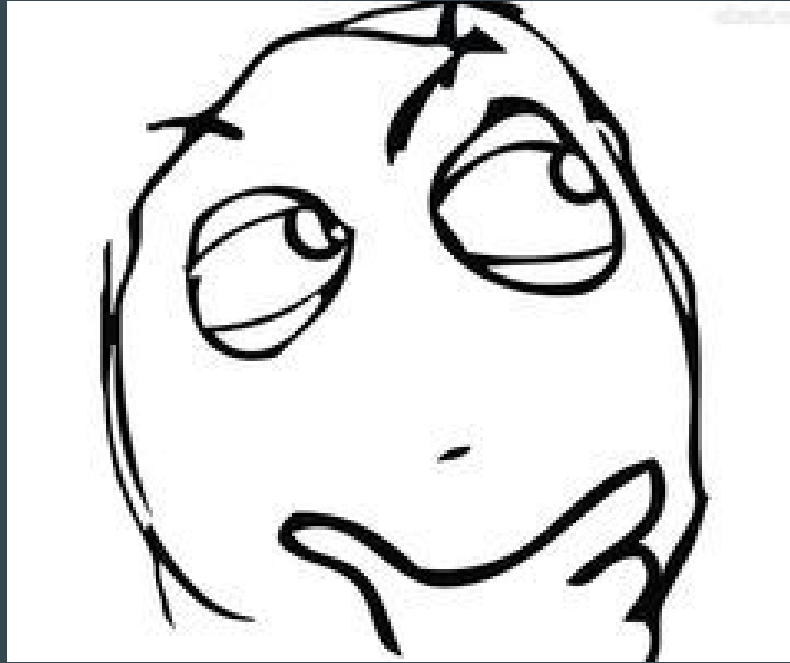
Então(/^posso visualizar o resultado da busca$/) do
  pending
end
```

```
# encoding: utf-8
Dado(/^que estou na página do Google$/) do
  @browser.goto("www.google.com.br")
end

Quando(/^pesquise por "([^"]*)"$/) do |texto|
  pesquisa = @browser.text_field(:name, 'q')
  pesquisa.set(texto)
  pesquisa.send_keys :enter
  sleep 2
end

Então(/^posso visualizar o resultado da busca$/) do
  resultados = @browser.links(:css, 'h3 a')
  resultados.each{|link| puts link.text; expect(link.text).not_to be_empty}
end
```

Dúvidas?



Links

Repositório com o slide e as automações: <https://github.com/catks/devqal-cucumber-watir>

Documentação cucumber:

<https://github.com/cucumber/cucumber/wiki>

Documentação watir:

<http://www.rubydoc.info/gems/watir-webdriver/>

<https://github.com/watir/watir/wiki>

Obrigado!

