

Testes Funcionais

Watir Webdriver + Ruby

Ruby

- ◆ Dinâmica
- ◆ Interpretada
- ◆ OO

Java vs Ruby

```
public int retornaSum(int num1, int num2) {  
    return num1 + num2;  
}
```

```
def retorna_sum(num1, num2)  
    num1 + num2  
end
```

```
public void forExample() {  
    String[] nomes = {"Meddle", "The Wall", "DSOT", "Pulse"};  
  
    for (int i = 0; i < nomes.length; i++) {  
        System.out.println(nomes[i]);  
    }  
}
```

```
def for_example  
    array = ["Meddle", "The Wall", "DSOT", "Pulse"]  
  
    array.length.times do |x|  
        puts array[x]  
    end  
  
    #mesmo output  
    array.length.times { |x| puts array[x] }  
  
end
```

```
public void forEachExample() {  
    String[] nomes = {"Meddle", "The Wall", "DSOT", "Pulse"};  
    for(String nome: nomes) {  
        System.out.println(nome);  
    }  
}
```

```
def for_each_example  
    array = ["Meddle", "The Wall", "DSOT", "Pulse"]  
  
    array.each { |x| puts x }  
  
end
```



```
public void javaHash() {  
    HashMap<String,String> album = new HashMap<String, String>();  
    album.put("nome", "Meddle");  
  
    System.out.println(album.get("nome"));  
}
```

```
def ruby_hash  
    album = {:nome => "Meddle", :banda => "Pink Floyd"}  
    #album = {nome: "Meddle", banda: "Pink Floyd"}  
  
    puts hash[:nome]  
  
end
```

```
public void arrayBool() {  
    Boolean[] arrayBool= {false,false,true,false};  
    Boolean resultado = false;  
  
    for (int i = 0; i < arrayBool.length; i++) {  
        if (arrayBool[i])  
            resultado = true;  
    }  
  
    System.out.println(resultado);  
}
```

```
def arrayBool  
    arrayBool = [false, false, true, false]  
  
    puts arrayBool.any?  
  
    #puts arrayBool.all?  
  
end
```


Ruby Class

```
class Pessoa
  attr_accessor :name, :last_name

  def initialize(name, last_name)
    self.name = name
    self.last_name = last_name
  end

  def full_name
    self.name + " " + self.last_name
  end
end

p = Pessoa.new("rafael", "lima")
puts p.full_name
```


Basic Watir

```
require 'watir-webdriver'
```

```
browser = Watir::Browser.new :firefox
```

```
browser.goto "http://demo.moodle.net/login/index.php"
```

```
browser.text_field(id: 'username').set 'admin'
```

```
browser.text_field(id: 'password').set 'sandbox'
```

```
browser.button(id: 'loginbtn').click
```

```
puts browser.p(id: 'frontpagesettings').exists? && browser.span(xpath: "//span[text()='Site  
administration']").exists?
```


Page Object Pattern

- ◆ Cada página da aplicação terá uma Page Object
- ◆ Elimina duplicação
- ◆ Fácil de manter
- ◆ Robusto

Instalação

- ◆ instalar Rbenv ou RVM
- ◆ Ruby
 - ◆ windows
 - ◆ unix
- ◆ instalar o bundler (link)
- ◆ montar estrutura de arquivos igual ao "page_object_tree.txt"
- ◆ executar "bundle install" no root do projeto
- ◆ executar "bundle exec cucumber" no root do projeto

TestGen

Gem que cria monta o projeto automaticamente

- ◆ Repositório
- ◆ Gem install testgen
- ◆ Executar "testgen project <project_name> --
pageobject-driver=watir --with-lib"

Referência

- ♦ Códigos e apresentação
 - ♦ https://github.com/rafalima/watir_webdriver
- ♦ Watir-webdriver
 - ♦ <http://watirwebdriver.com/>
 - ♦ <http://watir.com/>
- ♦ Page Object
 - ♦ <https://github.com/watir/watir-webdriver/wiki/Page-Objects>
 - ♦ <https://github.com/cheezy/page-object>

Contato

- ◆ email pessoal: rafalima07@gmail.com
- ◆ email profissional: rlima@thoughtworks.com