

Testing en Ruby y Rails

RSpec and Fixture Replacement

por Paco Guzmán

Haciendo testing rápido y fiable

Cucumber / Webrat

por Jorge Álvarez

The RSpec Pdf

The
Pragmatic
Programmers

The RSpec Book

Behaviour Driven Development
with RSpec, Cucumber,
and Friends

David Chelimsky
with Dave Astels,
Zach Dennis,
Aslak Hellesøy,
Bryan Helmkamp,
and Dan North

Edited by Jacquelyn Carter

The Facets of Ruby Series

¿Por qué RSpec?

Es más fácil de leer

ActiveSupport::TestCase vs RSpec v1.3.0

```
class AspgemsTest                                # describe Aspgems  
  
def test("something")                             # it "does something descriptive"  
  assert_equal 4, array.length                    # array.length.should == 4  
  
def setup                                           # before(:each) {}  
  
def teardown                                       # after(:each) {}
```

Matchers (RSpec v1.3.0)

rapidito

```
@string.should == 'foo' # @string.should equal('foo'), @string.should eql('foo')
@string.should_not ...
@array.should_not be_empty # calls @array.empty?
@string.should_not be_blank # calls @string.blank?
@hash.should have_key(:foo)
@string.should match(/\d+/) # @string.should =~ /\d+/
@object.should be_an_instance_of(String)
@stack.should have(0).elements
lambda { @stack.pop }.should raise_error(StackUnderflowError)
lambda { @stack.push }.should change(@stack.size).by(1)
```

Rails checklist

Models

Controllers

Helpers

Views

E-mails, Rake tasks, JavaScript, ...

Setting the environment

```
$ gem instal rspec rspec-rails
Successfully installed rspec-1.3.0
Successfully installed rspec-rails-1.3.2
2 gems installed
$ script/generate rspec
Configuring rspec and rspec-rails gems in config/environments/test.rb ...

  exists  lib/tasks
  create  lib/tasks/rspec.rake
  create  script/autospec
  create  script/spec
  create  spec
  create  spec/rcov.opts
  create  spec/spec.opts
  create  spec/spec_helper.rb
$ touch spec/factories.rb
$ gem install factory_girl
Successfully installed factory_girl-1.2.3
1 gem installed
```

spec/spec_helper.rb

```
...  
require 'factory_girl'  
  
Spec::Runner.configure do |config|  
  ...  
end
```


¿Por qué no utilizo fixtures?

Elementos globales

Las fixtures están dispersas

Están alejadas de los test

factory_girl

```
gem install factory_girl --source http://gemcutter.org
```

Creando datos de test

```
Factory.sequence(:login) { |n| "login#{n}" } # Sequence
```

```
Factory.define :user do |u|  
  u.login { Factory.next(:login) } # Using sequence  
  u.password {|a| a.login + "pass"} # Dependent attributes  
  u.password_confirmation {|a| a.login + "pass"}  
  u.email {|a| "#{a.login}@example.com".downcase }  
end
```

```
>>Factory(:user)
```

| | | | | |
|----|--------|--------------------|---------|-------|
| id | login | email | state | admin |
| 1 | login1 | login1@example.com | passive | false |

Herencia

```
Factory.define :admin, :parent => :user do |a|  
  a.admin true  
end
```

```
>>Factory(:admin)
```

| | | | | |
|----|--------|--------------------|---------|-------|
| id | login | email | state | admin |
| 3 | login3 | login3@example.com | passive | true |

Callbacks

```
Factory.define :registered, :parent => :user do |m|  
  m.after_create { |m| m.register! }  
end
```

```
>>Factory(:registered)
```

| id | login | email | activation_code | state | admin |
|----|--------|--------------------|--|---------|-------|
| 7 | login5 | login5@example.com | a6ea204bac5a3122cfa496656196b22e4c24b9b8 | pending | false |

```
Factory.define :member, :parent => :registered do |m|  
  m.after_create { |m| m.activate! }  
end
```

```
>>Factory(:member)
```

| id | login | activated_at | state | admin |
|----|--------|-------------------------|--------|-------|
| 9 | login7 | 2010-01-31 17:00:49 UTC | active | false |

Miss Machinist or ObjectDaddy?

```
require 'factory_girl/syntax/generate'
```

```
User.generate(:name => 'Johnny') # Factory(:user, :name => 'Johnny')
```

```
require 'factory_girl/syntax/make'
```

```
User.make(:name => 'Johnny') # Factory(:user, :name => 'Johnny')
```

Varios

```
Factory.define :user do |u|
  u.activation_code { User.make_activation_code } # Lazy attributes
  u.email {|a| "#{a.login}@example.com".downcase } # Dependant attributes
end
```

```
Factory.define :group do |p|
  # ...
  p.author {|a| a.association(:user, :login => 'Writely') } # Associations
  p.association :author, :factory => :member # o así, si necesitamos un member
end
```

```
>>Factory(:group)
```

| | | | |
|----|-------|---------|---------|
| id | name | state | user_id |
| 2 | name2 | pending | 11 |

```
>>User.find(11)
```

| | | | | |
|----|---------|---------------------|-------------------------|--------|
| id | login | email | activated_at | state |
| 11 | login11 | login11@example.com | 2010-01-31 17:15:37 UTC | active |

Vamos a verlo con ejemplos



<http://mabu.dk/media2/152630/wakecup.jpg>

spork + autospec

```
$ gem install ZenTest autotest-rails
Successfully installed ZenTest-4.2.1
Successfully installed autotest-rails-4.1.0
2 gems installed
```

```
$ gem install spork
Successfully installed spork-0.7.5
1 gem installed
```

```
cd /path/to/project/root
spork --bootstrap
```

Add --drb to spec/spec.opts

spork rspec # in one window

autospec # in other window

Complementos útiles de RSpec

shoulda

email-spec

no-peeping-toms

parallel-specs

rspec-on-rails-matchers

unemcumbered