MOTIVACIÓN

4 DE JUNIO 1996: LANZAMIENTO DEL **ARIANE 5**

- Comenzó a construirse en 1984 por la Agencia Espacial Europea
- Realizaba su vuelo inaugural
- No tripulado

4 DE JUNIO 1996: LANZAMIENTO DEL **ARIANE 5**

- Explota a segundos de comenzar el vuelo
- Perdidas millonarias

¿POR QUÉ?

¿POR QUÉ?

Se intentó convertir un float de 64 bits a un integer de 16 bits

TESTING Y RSPEC

meretamal@uc.cl
@meretamal

¿QUÉ ES TESTING?

¿QUÉ SON TESTS UNITARIOS?

¿QUÉ ES Y CÓMO USO RSPEC?

¿CÓMO USO RSPEC EN MI PROYECTO?

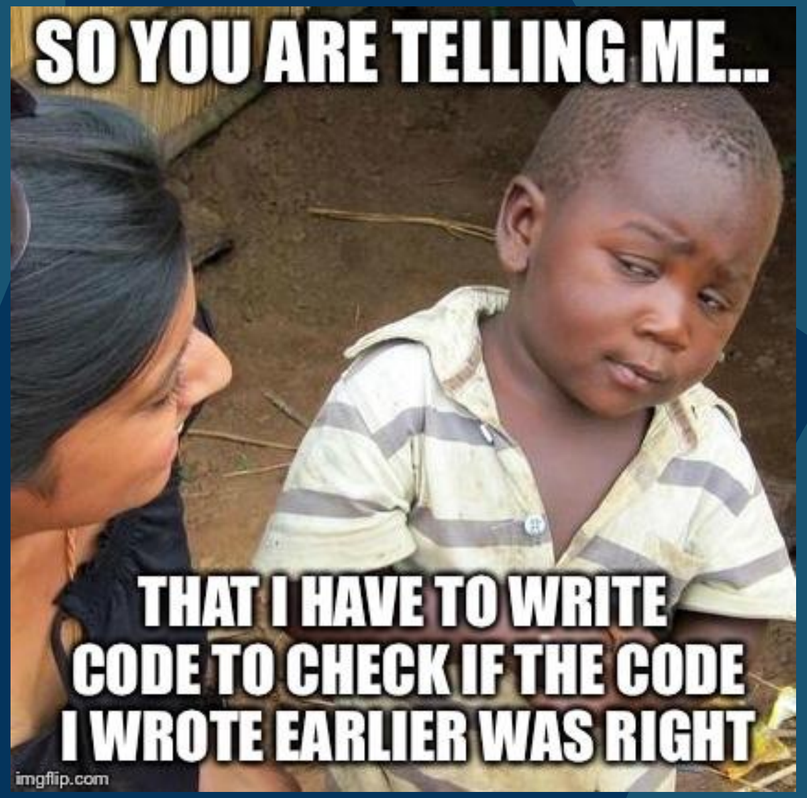
¿QUÉ ES TESTING?

TESTING DE SOFTWARE

El Testing de Software es la realización de pruebas sobre él mismo, con el fin de obtener información acerca de su calidad.

TESTING DE SOFTWARE

El Testing de Software es la realización de pruebas sobre él mismo, con el fin de obtener información acerca de su calidad.



¿POR QUÉ?

NOS PERMITE
IDENTIFICAR ERRORES

¿POR QUÉ?

NOS PERMITE
IDENTIFICAR ERRORES

NOR PERMITE EVADIR
COSTOS ADICIONALES

¿POR QUÉ?

NOS PERMITE
IDENTIFICAR ERRORES

NOR PERMITE EVADIR
COSTOS ADICIONALES

¿POR QUÉ?

MANTIENE LA
CONFIANZA DEL
CLIENTE

¿POR QUÉ?

NOS PERMITE IDENTIFICAR ERRORES

NOS PERMITE EVADIR COSTOS ADICIONALES

MANTIENE LA CONFIANZA DEL CLIENTE

NOS MANTIENE EN EL MERCADO

¿QUÉ SON TESTS UNITARIOS?

- Es una forma de comprobar el correcto funcionamiento de una unidad de código.
- Esto sirve para asegurar que cada unidad funcione correctamente y eficientemente por separado.

¿QUÉ ES Y CÓMO USO RSPEC?

RSPEC

RSpec es una herramienta que me permite crear pruebas unitarias sobre código escrito en Ruby

¿CÓMO LO USAMOS?

```ruby
# human.rb

class Human
  def greet
    "Hello world!"
  end
end
```

```ruby
# human.rb

class Human
  def greet
    "Hello world!"
  end
end
```

```ruby
# spec/human_spec.rb

require_relative '../human.rb'

RSpec.describe Human do




end
```

```ruby
# human.rb

class Human
  def greet
    "Hello world!"
  end
end
```

```ruby
# spec/human_spec.rb

require_relative '../human.rb'

RSpec.describe Human do
  describe '#greet' do


  end
end
```

```ruby
# human.rb

class Human
  def greet
    "Hello world!"
  end
end
```

```ruby
# spec/human_spec.rb

require_relative '../human.rb'

RSpec.describe Human do
  describe '#greet' do
    it "says hello" do

    end
  end
end
```

```ruby
# human.rb

class Human
  def greet
    "Hello world!"
  end
end
```

```ruby
# spec/human_spec.rb

require_relative '../human.rb'

RSpec.describe Human do
  describe '#greet' do
    it "says hello" do
      expect(Human.new.greet).to eq("Hello, world!")
    end
  end
end
```

```ruby
# human.rb

class Human
  def greet
    "Hello world!"
  end
end
```

```ruby
# spec/human_spec.rb

require_relative '../human.rb'

RSpec.describe Human do
  describe '#greet' do

    let(:human) do
      human = Human.new
    end

    it "says hello" do
      expect(human.greet).to eq("Hello, world!")
    end
  end
end
```

MATCHERS

Equality/Identity Matchers

Matchers to test for object or value equality.

Matcher	Description	Example
eq	Passes when actual == expected	expect(actual).to eq expected
eql	Passes when actual.eql?(expected)	expect(actual).to eql expected
be	Passes when actual.equal?(expected)	expect(actual).to be expected
equal	Also passes when actual.equal?(expected)	expect(actual).to equal expected

Comparison Matchers

Matchers for comparing to values.

Matcher	Description	Example
>	Passes when actual > expected	expect(actual).to be > expected
>=	Passes when actual >= expected	expect(actual).to be >= expected
<	Passes when actual < expected	expect(actual).to be < expected
<=	Passes when actual <= expected	expect(actual).to be <= expected
be_between inclusive	Passes when actual is <= min and >= max	expect(actual).to be_between(min, max).inclusive
be_between exclusive	Passes when actual is < min and > max	expect(actual).to be_between(min, max).exclusive
match	Passes when actual matches a regular expression	expect(actual).to match(/regex/)

Class/Type Matchers

Matchers for testing the type or class of objects.

Matcher	Description	Example
be_instance_of	Passes when actual is an instance of the expected class.	expect(actual).to be_instance_of(Expected)
be_kind_of	Passes when actual is an instance of the expected class or any of its parent classes.	expect(actual).to be_kind_of(Expected)
respond_to	Passes when actual responds to the specified method.	expect(actual).to respond_to(expected)

True/False/Nil Matchers

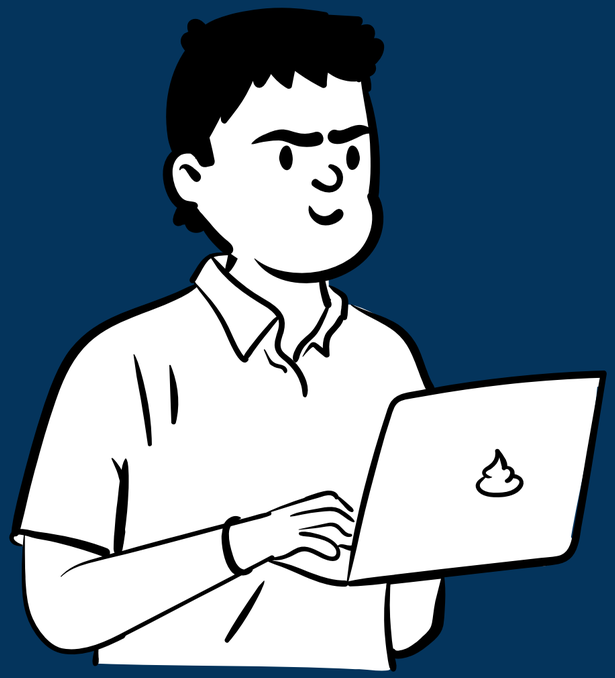
Matchers for testing whether a value is true, false or nil.

Matcher	Description	Example
be true	Passes when actual == true	expect(actual).to be true
be false	Passes when actual == false	expect(actual).to be false
be_truthy	Passes when actual is not false or nil	expect(actual).to be_truthy
be_falsey	Passes when actual is false or nil	expect(actual).to be_falsey
be_nil	Passes when actual is nil	expect(actual).to be_nil

Error Matchers

Matchers for testing, when a block of code raises an error.

Matcher	Description	Example
raise_error(ErrorClass)	Passes when the block raises an error of type ErrorClass.	expect {block}.to raise_error(ErrorClass)
raise_error("error message")	Passes when the block raise an error with the message “error message”.	expect {block}.to raise_error(“error message”)
raise_error(ErrorClass, "error message")	Passes when the block raises an error of type ErrorClass with the message “error message”	expect {block}.to raise_error(ErrorClass,“error message”)



VEAMOS UN EJEMPLO

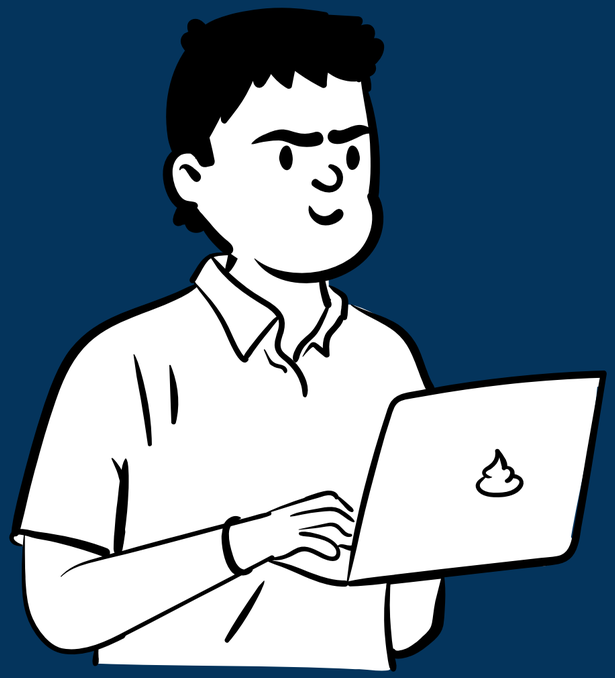
¿CÓMO USO RSPEC EN MI PROYECTO?

gem 'rspec-rails'

- Nos permite integrar RSpec en nuestros proyectos de Ruby on Rails
- Se integra con los generadores de Rails
- Nos da *matchers* específicos para testar una app de Rails

TEST DE REQUESTS

Nos permiten validar la respuesta de nuestro servidor frente a distintos requests de HTTP



VEAMOS UN EJEMPLO

Más material

https://www.rubyguides.com/2018/07/rspec-tutorial/
https://www.tutorialspoint.com/rspec/index.htm
https://www.betterspecs.org/
https://relishapp.com/rspec/rspec-rails/docs


HAPPY TESTING!!