

```
josh@debian: ~/practica3
josh@debian:~$ mkdir practica3
josh@debian:~$ cd practica3
josh@debian:~/practica3$ ls
josh@debian:~/practica3$ touch Gemfile
josh@debian:~/practica3$
```

```
GNU nano 7.2 Gemfile *
source 'https://rubygems.org/'

gem 'rspec'
```


```
josh@debian: ~/practica3
josh@debian:~/practica3$ su
Contraseña:
root@debian:/home/josh/practica3# bundle install
Don't run Bundler as root. Bundler can ask for sudo if it is needed, and
installing your bundle as root will break this application for all non-root
users on this machine.
Fetching gem metadata from https://rubygems.org/...
Resolving dependencies...
Using bundler 2.3.7
Fetching diff-lcs 1.5.1
Fetching rspec-support 3.13.1
Installing rspec-support 3.13.1
Installing diff-lcs 1.5.1
Fetching rspec-core 3.13.0
Fetching rspec-expectations 3.13.2
Installing rspec-expectations 3.13.2
Fetching rspec-mocks 3.13.1
Installing rspec-core 3.13.0
Installing rspec-mocks 3.13.1
Fetching rspec 3.13.0
Installing rspec 3.13.0
Bundle complete! 1 Gemfile dependency, 7 gems now installed.
Use `bundle info [gemname]` to see where a bundled gem is installed.
root@debian:/home/josh/practica3#
```

end

```

root@debian:/home/josh/practica3# bundle exec rspec --init
  create  .rspec
  create  spec/spec_helper.rb
root@debian:/home/josh/practica3# ls
Gemfile  Gemfile.lock  spec
root@debian:/home/josh/practica3# mkdir lib
root@debian:/home/josh/practica3# ls
Gemfile  Gemfile.lock  lib  spec
root@debian:/home/josh/practica3# cd lib
root@debian:/home/josh/practica3/lib# touch ruby.rb
root@debian:/home/josh/practica3/lib# █

```



```

ruby.rb  x
ruby.rb
1  def intro array
2    #escribir el código
3  end
4  def nombre nom
5    #escribir el código
6  end
7  def datos? array
8    #escribir el código
9  end

```

```

root@debian:/home/josh/practica3# bundle exec rspec --init
  create  .rspec
  create  spec/spec_helper.rb
root@debian:/home/josh/practica3# ls
Gemfile  Gemfile.lock  spec
root@debian:/home/josh/practica3# mkdir lib
root@debian:/home/josh/practica3# ls
Gemfile  Gemfile.lock  lib  spec
root@debian:/home/josh/practica3# cd lib
root@debian:/home/josh/practica3/lib# touch ruby.rb
root@debian:/home/josh/practica3/lib# cd ..
root@debian:/home/josh/practica3# ls
Gemfile  Gemfile.lock  lib  spec
root@debian:/home/josh/practica3# cd spec
root@debian:/home/josh/practica3/spec# touch rspec_1.rb
root@debian:/home/josh/practica3/spec# █

```

```

require "ruby.rb"
describe "#intro" do
  it "Debe estar definida" do
    expect{intro([1,2,3,4])}.not_to raise_error
  end
  it "retorna la suma del array" do
    expect(intro ([2,6,10,5,2])).to eq(25)
    expect(intro ([15,6,10,19])).to eq(50)
    expect(intro ([15,6,4,5])).to eq(30)
  end
  it "retorna 0 si el array está vacío" do
    expect(intro([])).to eq(0)
  end
  it "retorna el valor del único elemento del array" do
    expect(intro([5])).to eq(5)
  end
end
describe "#nombre" do
  it "Debe iniciar con Mayúscula y concatena la cadena con la palabra hello" do
    expect(nombre("Pedro")).to eq("hello Pedro")
    expect(nombre("Juan")).to eq("hello Juan")
    expect(nombre("María")).to eq("hello María")
  end
  it "No trabaja con un string vacío" do
    expect(nombre("")).to eq("vacío")
  end
end
end

```

```

describe "#datos?" do
  it "retorna true si el tamaño del array es > 0" do
    expect(datos?([2,1,5,6])).to be true
  end
  it "retorna false si el tamaño del array es <= 0" do
    expect(datos?([])).to be false
  end
end
end

```

```
1  require "ruby.rb"
2  describe "#intro" do
3      it "Debe estar definida" do
4          expect{intro([1,2,3,4])}.not_to raise_error
5      end
6      it "retorna la suma del array" do
7          expect(intro([2,6,10,5,2])).to eq(25)
8          expect(intro([15,6,10,19])).to eq(50)
9          expect(intro([15,6,4,5])).to eq(30)
10     end
11     it "retorna 0 si el array está vacío" do
12         expect(intro([])).to eq(0)
13     end
14     it "retorna el valor del único elemento del array" do
15         expect(intro([5])).to eq(5)
16     end
17 end
18 describe "#nombre" do
19     it "Debe iniciar con Mayúscula y concatena la cadena con la palabra hello" do
20         expect(nombre("Pedro")).to eq("hello Pedro")
21         expect(nombre("Juan")).to eq("hello Juan")
22         expect(nombre("María")).to eq("hello María")
23     end
24     it "No trabaja con un string vacío" do
25         expect(nombre("")).to eq("vacío")
26     end
27 end
```

```
28     describe "#datos?" do
29         it "retorna true si el tamaño del array es > 0" do
30             expect(datos?([2,1,5,6])).to be true
31         end
32         it "retorna false si el tamaño del array es <= 0" do
33             expect(datos?([])).to be false
34         end
35     end
```

```
josh@debian: ~/practica3
root@debian:/home/josh/practica3# rspec spec/rspec_1.rb
.FFFFFFFF

Failures:

  1) #intro retorna la suma del array
     Failure/Error: expect(intro ([2,6,10,5,2])).to eq(25)

     expected: 25
     got: nil

     (compared using ==)
     # ./spec/rspec_1.rb:7:in `block (2 levels) in <top (required)>'

  2) #intro retorna 0 si el array está vacío
     Failure/Error: expect(intro([])).to eq(0)

     expected: 0
     got: nil

     (compared using ==)
     # ./spec/rspec_1.rb:12:in `block (2 levels) in <top (required)>'
```

```
josh@debian: ~/practica3

  3) #intro retorna el valor del único elemento del array
     Failure/Error: expect(intro([5])).to eq(5)

     expected: 5
     got: nil

     (compared using ==)
     # ./spec/rspec_1.rb:15:in `block (2 levels) in <top (required)>'

  4) #nombre Debe iniciar con Mayúscula y concatena la cadena con la palabra hello
     Failure/Error: expect(nombre("Pedro")).to eq("hello Pedro")

     expected: "hello Pedro"
     got: nil

     (compared using ==)
     # ./spec/rspec_1.rb:20:in `block (2 levels) in <top (required)>'
```

```
x  +  josh@debian: ~/practica3  Q  ≡

5) #nombre No trabaja con un string vacío
Failure/Error: expect(nombre("")).to eq("vacío")

      expected: "vacío"
      got: nil

      (compared using ==)
# ./spec/rspec_1.rb:25:in `block (2 levels) in <top (required)>'

6) #datos? retorna true si el tamaño del array es > 0
Failure/Error: expect(datos?([2,1,5,6])).to be true

      expected true
      got nil
# ./spec/rspec_1.rb:30:in `block (2 levels) in <top (required)>'

7) #datos? retorna false si el tamaño del array es <= 0
Failure/Error: expect(datos?([])).to be false

      expected false
      got nil
# ./spec/rspec_1.rb:33:in `block (2 levels) in <top (required)>'

Finished in 0.07874 seconds (files took 0.83484 seconds to load)
```

```
josh@debian: ~/practica3

Finished in 0.07874 seconds (files took 0.83484 seconds to load)
8 examples, 7 failures

Failed examples:

rspec ./spec/rspec_1.rb:6 # #intro retorna la suma del array
rspec ./spec/rspec_1.rb:11 # #intro retorna 0 si el array está vacío
rspec ./spec/rspec_1.rb:14 # #intro retorna el valor del único elemento del array
rspec ./spec/rspec_1.rb:19 # #nombre Debe iniciar con Mayúscula y concatena la cadena con la palabra hello
rspec ./spec/rspec_1.rb:24 # #nombre No trabaja con un string vacío
rspec ./spec/rspec_1.rb:29 # #datos? retorna true si el tamaño del array es > 0
rspec ./spec/rspec_1.rb:32 # #datos? retorna false si el tamaño del array es <= 0

root@debian:/home/josh/practica3# rspec spec/rspec_1.rb
.FFFFFFFF

Failures:

  1) #intro retorna la suma del array
     Failure/Error: expect(intro ([2,6,10,5,2])).to eq(25)
```

```
x [terminal icon] josh@debian: ~/practica3 [search] [menu]

Failure/Error: expect(intro ([2,6,10,5,2])).to eq(25)

  expected: 25
   got: nil

  (compared using ==)
# ./spec/rspec_1.rb:7:in `block (2 levels) in <top (required)>'

2) #intro retorna 0 si el array está vacío
Failure/Error: expect(intro([])).to eq(0)

  expected: 0
   got: nil

  (compared using ==)
# ./spec/rspec_1.rb:12:in `block (2 levels) in <top (required)>'

3) #intro retorna el valor del único elemento del array
Failure/Error: expect(intro([5])).to eq(5)

  expected: 5
   got: nil

  (compared using ==)
```



```
josh@debian: ~/practica3
3) #intro retorna el valor del único elemento del array
Failure/Error: expect(intro([5])).to eq(5)

    expected: 5
     got: nil

    (compared using ==)
# ./spec/rspec_1.rb:15:in `block (2 levels) in <top (required)>'

4) #nombre Debe iniciar con Mayúscula y concatena la cadena con la palabra hello
Failure/Error: expect(nombre("Pedro")).to eq("hello Pedro")

    expected: "hello Pedro"
     got: nil

    (compared using ==)
# ./spec/rspec_1.rb:20:in `block (2 levels) in <top (required)>'

5) #nombre No trabaja con un string vacío
Failure/Error: expect(nombre("")).to eq("vacío")

    expected: "vacío"
     got: nil
```

```
josh@debian: ~/practica3

    got: nil

    (compared using ==)
# ./spec/rspec_1.rb:25:in `block (2 levels) in <top (required)>'

6) #datos? retorna true si el tamaño del array es > 0
Failure/Error: expect(datos?([2,1,5,6])).to be true

    expected true
     got nil
# ./spec/rspec_1.rb:30:in `block (2 levels) in <top (required)>'

7) #datos? retorna false si el tamaño del array es <= 0
Failure/Error: expect(datos?([])).to be false

    expected false
     got nil
# ./spec/rspec_1.rb:33:in `block (2 levels) in <top (required)>'

Finished in 0.02062 seconds (files took 0.13037 seconds to load)
8 examples, 7 failures
```

Failed examples:

```
rspec ./spec/rspec_1.rb:6 # #intro retorna la suma del array
rspec ./spec/rspec_1.rb:11 # #intro retorna 0 si el array está vacío
rspec ./spec/rspec_1.rb:14 # #intro retorna el valor del único elemento del array
rspec ./spec/rspec_1.rb:19 # #nombre Debe iniciar con Mayúscula y concatena la cadena con la palabra hello
rspec ./spec/rspec_1.rb:24 # #nombre No trabaja con un string vacío
rspec ./spec/rspec_1.rb:29 # #datos? retorna true si el tamaño del array es > 0
rspec ./spec/rspec_1.rb:32 # #datos? retorna false si el tamaño del array es <= 0
```

```
root@debian:/home/josh/practica3# ls
Gemfile Gemfile.lock lib spec
root@debian:/home/josh/practica3# cd spec
root@debian:/home/josh/practica3/spec# ls
rspec_1.rb spec_helper.rb
root@debian:/home/josh/practica3/spec#
```

```
ruby.rb
1 def intro array
2   #array=[2,6,10,5,2,]
3   return 0 if array.empty?
4   array.sum
5 end
```

Finished in 0.02258 seconds (files took 0.13524 seconds to load)
8 examples, 4 failures

Failed examples:

```
rspec ./spec/rspec_1.rb:19 # #nombre Debe iniciar con Mayúscula y concatena la cadena con la palabra hello
rspec ./spec/rspec_1.rb:24 # #nombre No trabaja con un string vacío
rspec ./spec/rspec_1.rb:29 # #datos? retorna true si el tamaño del array es > 0
rspec ./spec/rspec_1.rb:32 # #datos? retorna false si el tamaño del array es <= 0
```

```
root@debian:/home/josh/practica3#
```

```
def datos? array
  if(array>0)
    return true
  else
    return false
  end
end
```

8 examples, 3 failures

Failed examples:

```
rspec ./spec/rspec_1.rb:24 # #nombre No trabaja con un string vacío
rspec ./spec/rspec_1.rb:29 # #datos? retorna true si el tamaño del array es > 0
rspec ./spec/rspec_1.rb:32 # #datos? retorna false si el tamaño del array es <= 0
```

root@debian:/home/josh/practica3#

```
def nombre nom
  if nom.empty?
    "vacío"
  else
    "hello #{nom.capitalize}"
  end
end
```

Finished in 0.02116 seconds (files took 0.1417 seconds to load)

8 examples, 1 failure

Failed examples:

```
rspec ./spec/rspec_1.rb:24 # #nombre No trabaja con un string vacío
```

root@debian:/home/josh/practica3# rspec spec/rspec_1.rb

.....

Finished in 0.00797 seconds (files took 0.13793 seconds to load)

8 examples, 0 failures

root@debian:/home/josh/practica3#

EXTRA:

root@debian:/home/josh/practica3# bundle exec rspec spec/calculator_spec.rb

.....

Finished in 0.00597 seconds (files took 0.11698 seconds to load)

5 examples, 0 failures

root@debian:/home/josh/practica3#