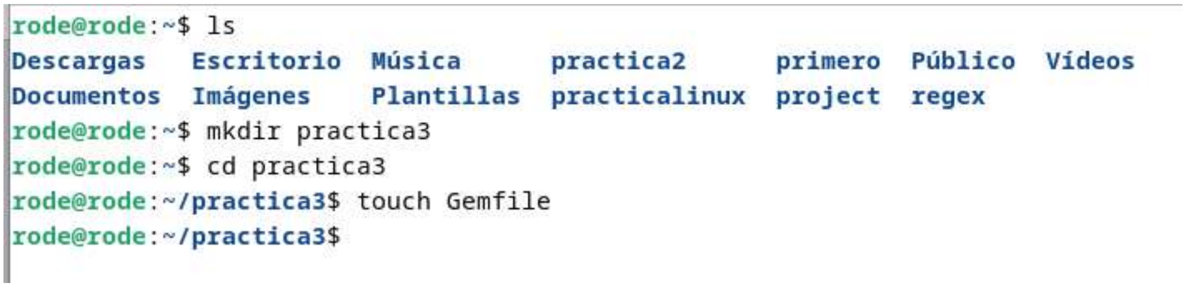
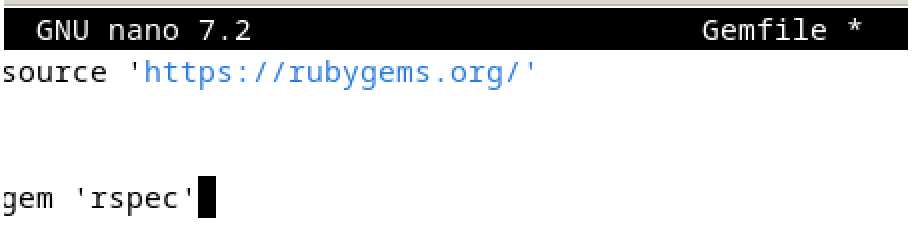
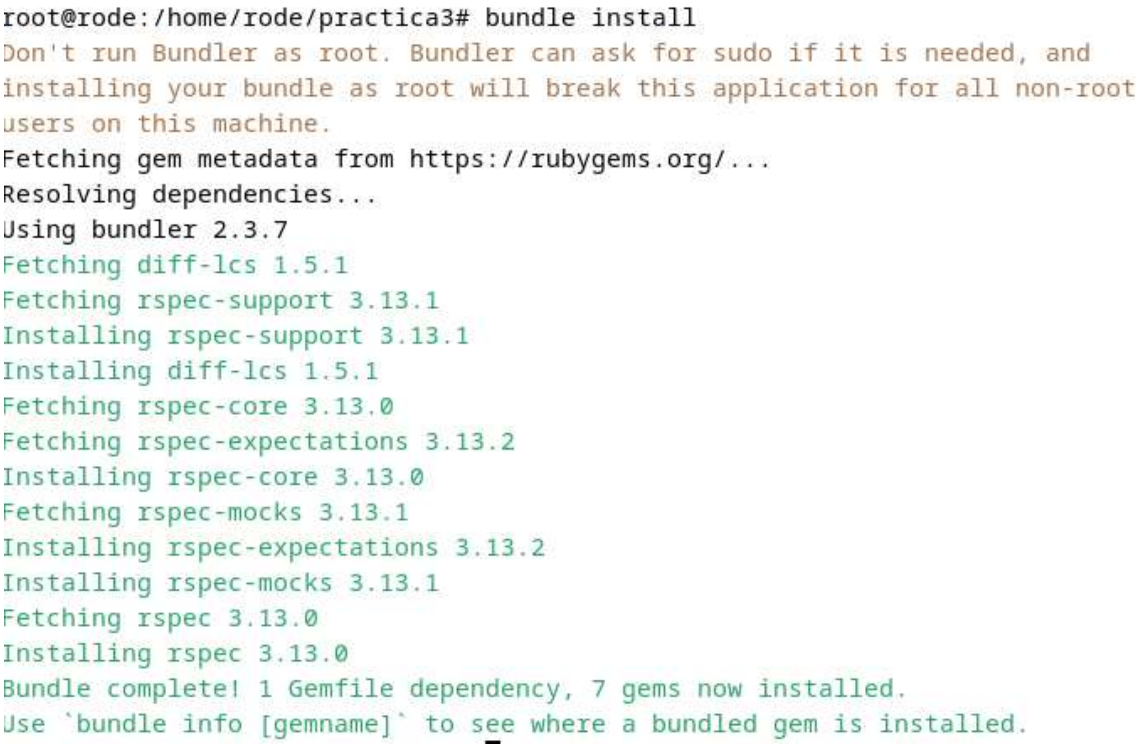


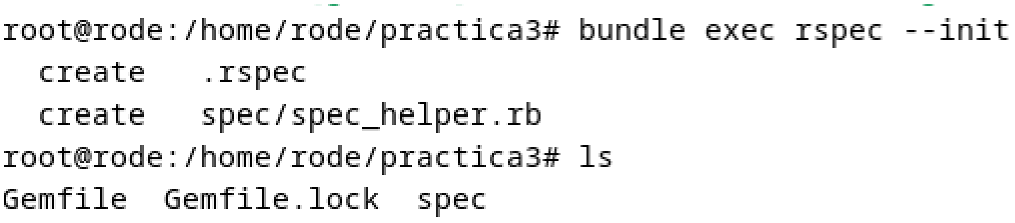
```
rode@rode:~$ ls
Descargas   Escritorio  Música      practica2     primero  Público  Vídeos
Documentos  Imágenes    Plantillas  practicalinux project  regex
rode@rode:~$ mkdir practica3
rode@rode:~$ cd practica3
rode@rode:~/practica3$ touch Gemfile
rode@rode:~/practica3$
```



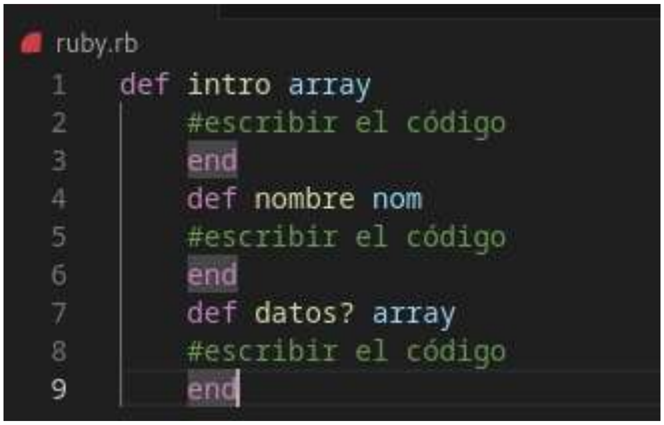
```
  GNU nano 7.2                        Gemfile *
source 'https://rubygems.org/'


gem 'rspec'
```



```
root@rode:/home/rode/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-core 3.13.0
Fetching rspec-mocks 3.13.1
Installing rspec-expectations 3.13.2
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.
```

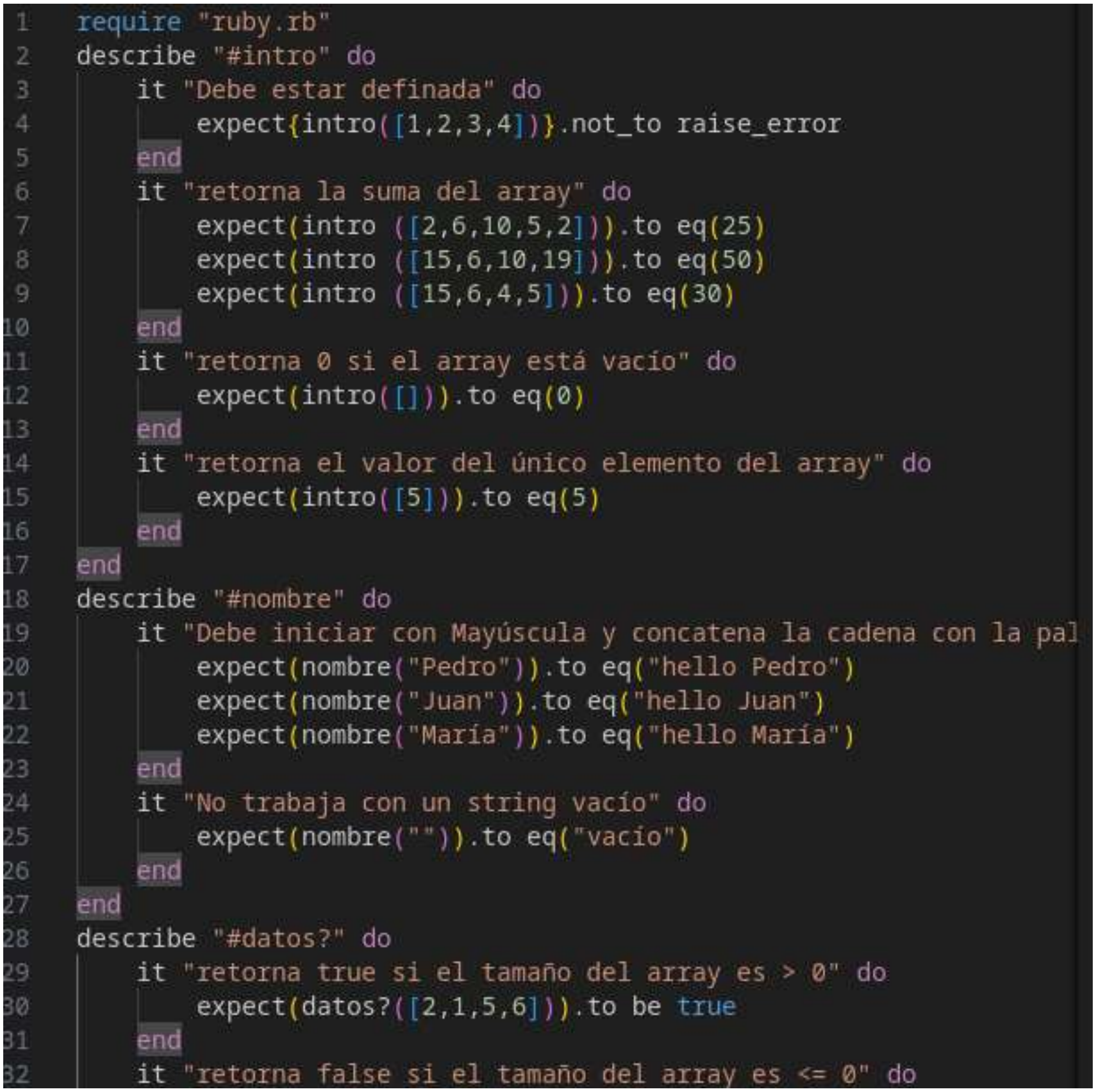


```
root@rode:/home/rode/practica3# bundle exec rspec --init
  create   .rspec
  create   spec/spec_helper.rb
root@rode:/home/rode/practica3# ls
Gemfile  Gemfile.lock  spec
```

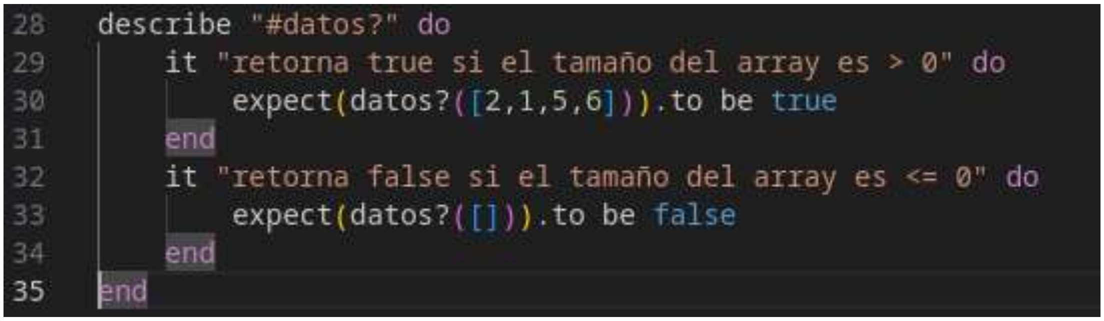


```ruby
def intro array
    #escribir el código
    end
    def nombre nom
    #escribir el código
    end
    def datos? array
    #escribir el código
    end
```

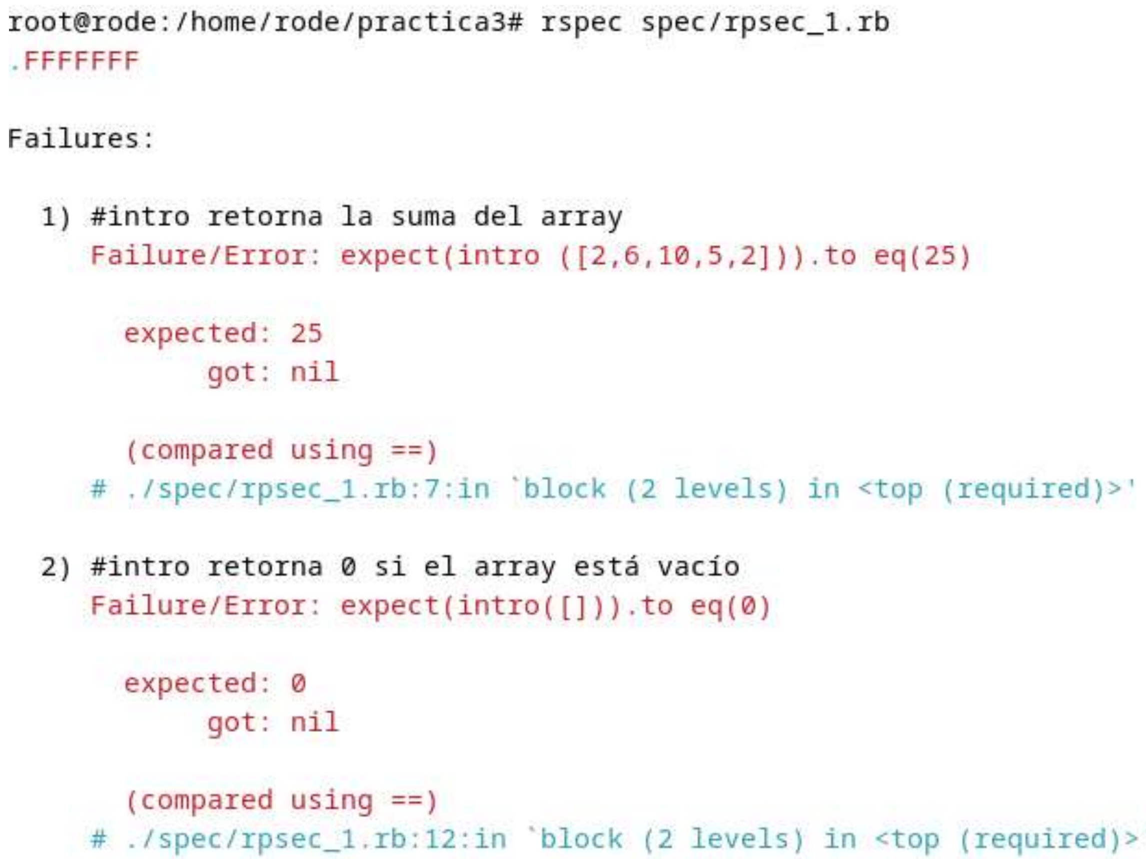
```
root@rode:/home/rode/practica3/lib# cd ..
root@rode:/home/rode/practica3# ls
Gemfile  Gemfile.lock  lib  spec
root@rode:/home/rode/practica3# cd spec
root@rode:/home/rode/practica3/spec# touch rpsec_1.rb
```



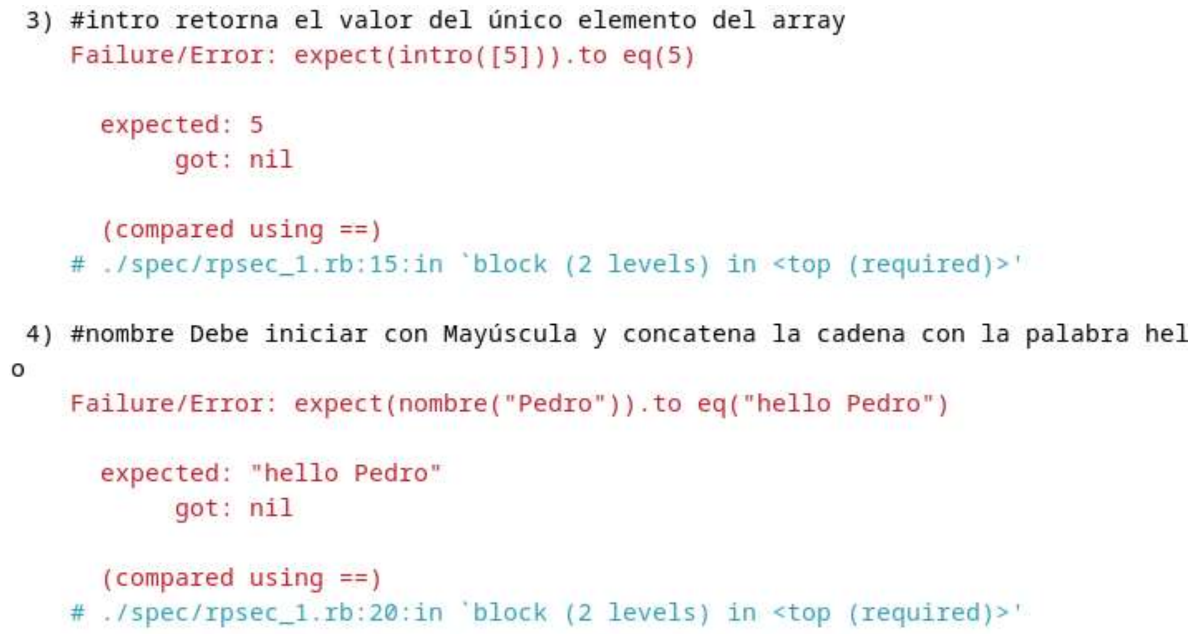
```ruby
require "ruby.rb"
describe "#intro" do
    it "Debe estar definada" do
        expect{intro([1,2,3,4])}.not_to raise_error
    end
    it "retorna la suma del array" do
        expect(intro ([2,6,10,5,2])).to eq(25)
        expect(intro ([15,6,10,19])).to eq(50)
        expect(intro ([15,6,4,5])).to eq(30)
    end
    it "retorna 0 si el array está vacio" 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 pal
        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("vacio")
    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
```



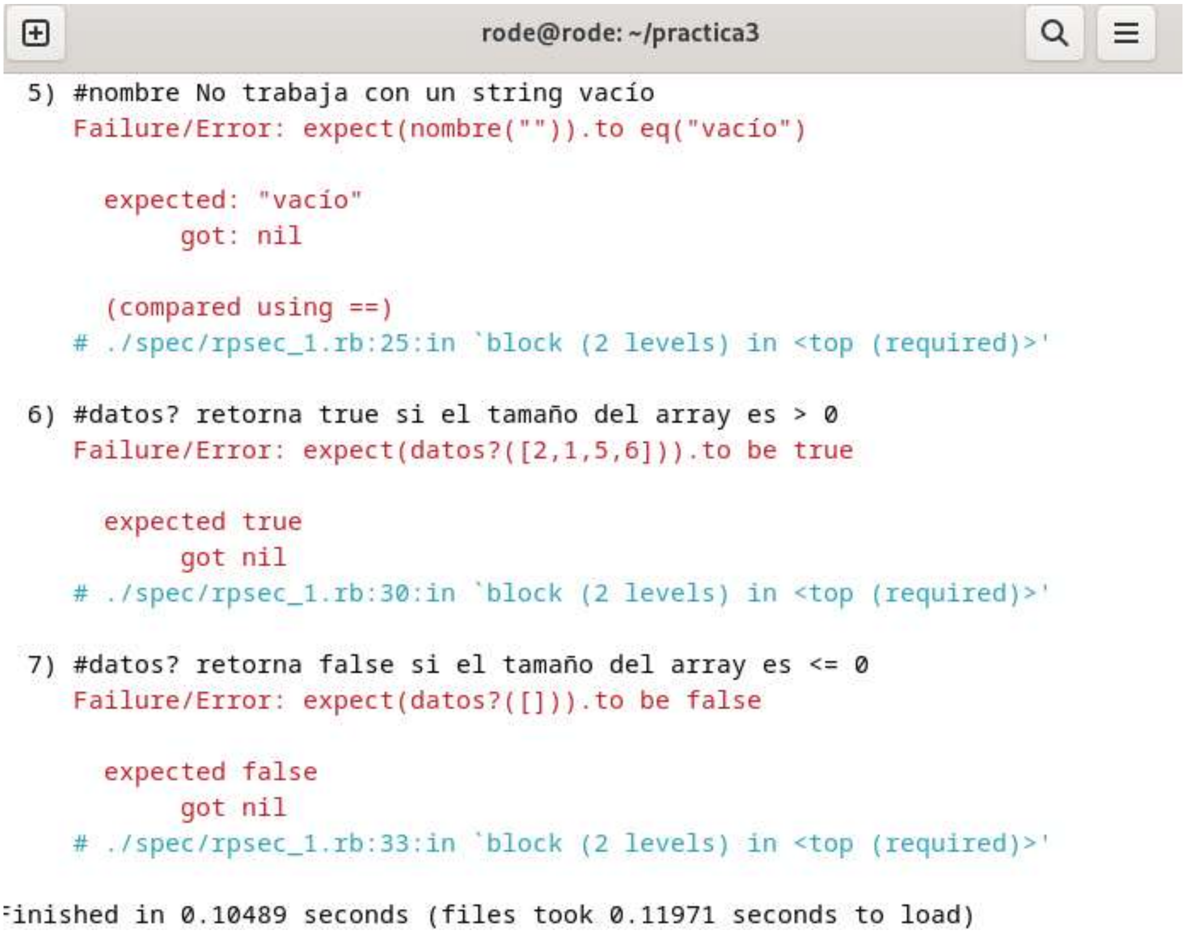
```ruby
describe "#datos?" do
    it "retorna true si el tamaño del array es > 0" do
        expect(datos?([2,1,5,6])).to be true
    end
    it "retorna false si el tamaño del array es <= 0" do
        expect(datos?([])).to be false
    end
end
```



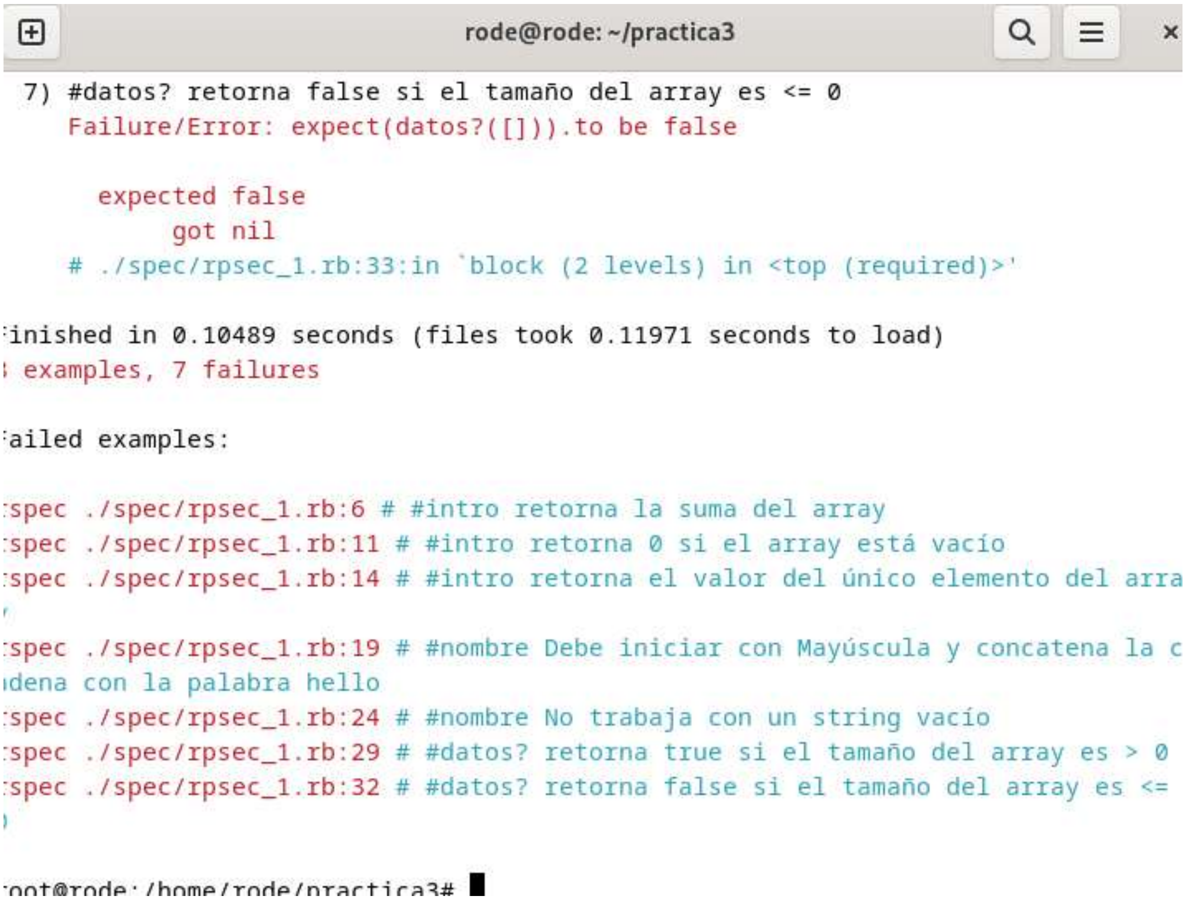
```
root@rode:/home/rode/practica3# rspec spec/rpsec_1.rb
.FFFFFFF

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/rpsec_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/rpsec_1.rb:12:in `block (2 levels) in <top (required)>'
```



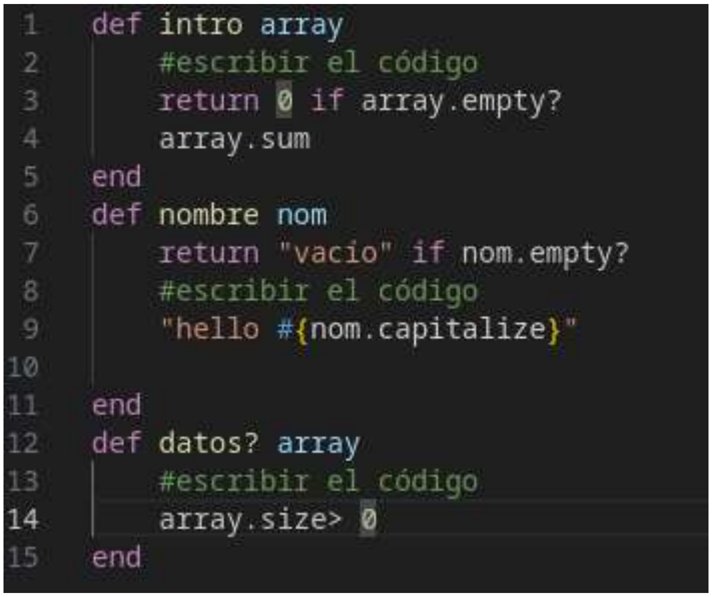
```
 3) #intro retorna el valor del único elemento del array
    Failure/Error: expect(intro([5])).to eq(5)

      expected: 5
           got: nil

      (compared using ==)
    # ./spec/rpsec_1.rb:15:in `block (2 levels) in <top (required)>'

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

      expected: "hello Pedro"
           got: nil

      (compared using ==)
    # ./spec/rpsec_1.rb:20:in `block (2 levels) in <top (required)>'
```



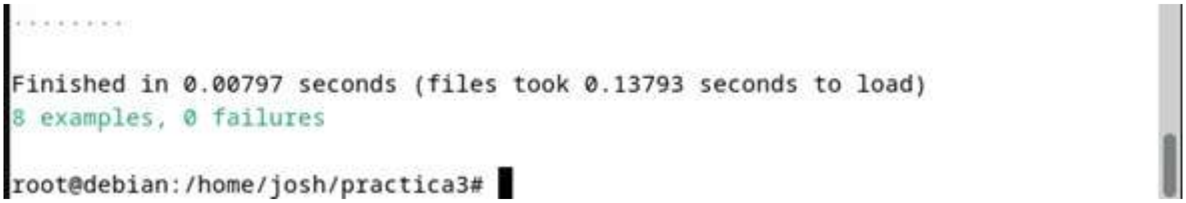
```
  5) #nombre No trabaja con un string vacío
     Failure/Error: expect(nombre("")).to eq("vacío")

       expected: "vacío"
            got: nil

       (compared using ==)
     # ./spec/rpsec_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/rpsec_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/rpsec_1.rb:33:in `block (2 levels) in <top (required)>'

Finished in 0.10489 seconds (files took 0.11971 seconds to load)
```



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

      expected false
           got nil
    # ./spec/rpsec_1.rb:33:in `block (2 levels) in <top (required)>'

inished in 0.10489 seconds (files took 0.11971 seconds to load)
 examples, 7 failures

ailed examples:

spec ./spec/rpsec_1.rb:6 # #intro retorna la suma del array
spec ./spec/rpsec_1.rb:11 # #intro retorna 0 si el array está vacío
spec ./spec/rpsec_1.rb:14 # #intro retorna el valor del único elemento del arra

spec ./spec/rpsec_1.rb:19 # #nombre Debe iniciar con Mayúscula y concatena la c
dena con la palabra hello
spec ./spec/rpsec_1.rb:24 # #nombre No trabaja con un string vacío
spec ./spec/rpsec_1.rb:29 # #datos? retorna true si el tamaño del array es > 0
spec ./spec/rpsec_1.rb:32 # #datos? retorna false si el tamaño del array es <=


oot@rode:/home/rode/practica3#
```

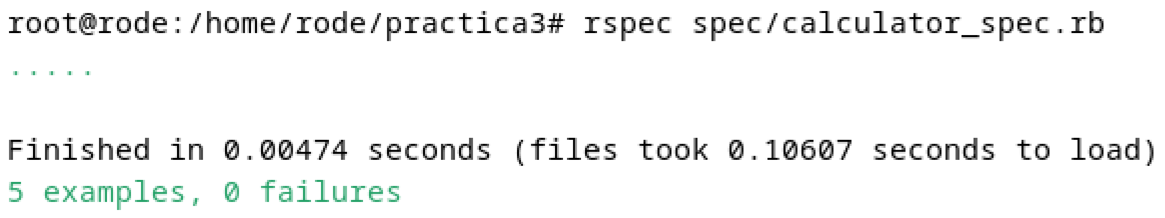


```ruby
def intro array
    #escribir el código
    return 0 if array.empty?
    array.sum
end
def nombre nom
    return "vacio" if nom.empty?
    #escribir el código
    "hello #{nom.capitalize}"

end
def datos? array
    #escribir el código
    array.size> 0
end
```

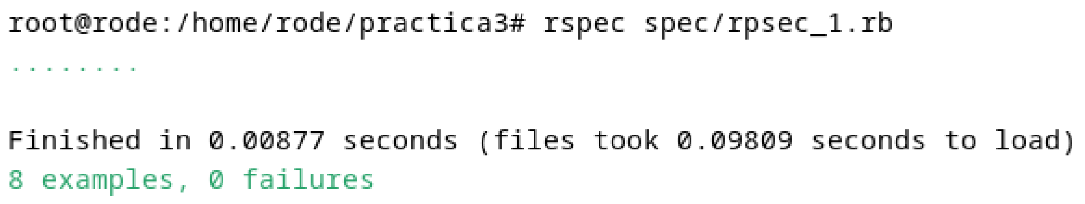


```
.........

Finished in 0.00797 seconds (files took 0.13793 seconds to load)
8 examples, 0 failures

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

Extra



```
root@rode:/home/rode/practica3# rspec spec/calculator_spec.rb
.....

Finished in 0.00474 seconds (files took 0.10607 seconds to load)
5 examples, 0 failures
```



```
root@rode:/home/rode/practica3# rspec spec/rpsec_1.rb
........

Finished in 0.00877 seconds (files took 0.09809 seconds to load)
8 examples, 0 failures
```