

Universidad Nacional Autónoma de Nicaragua Unan León



Facultad de Ciencias y Tecnología.

Componente: Software como un servicio

Tema: Practica3

Grupo:1

Docente: Erving Montes

Integrante:

-Harold Steven Garcia Ramos.

Fecha: 21 de agosto de 2024

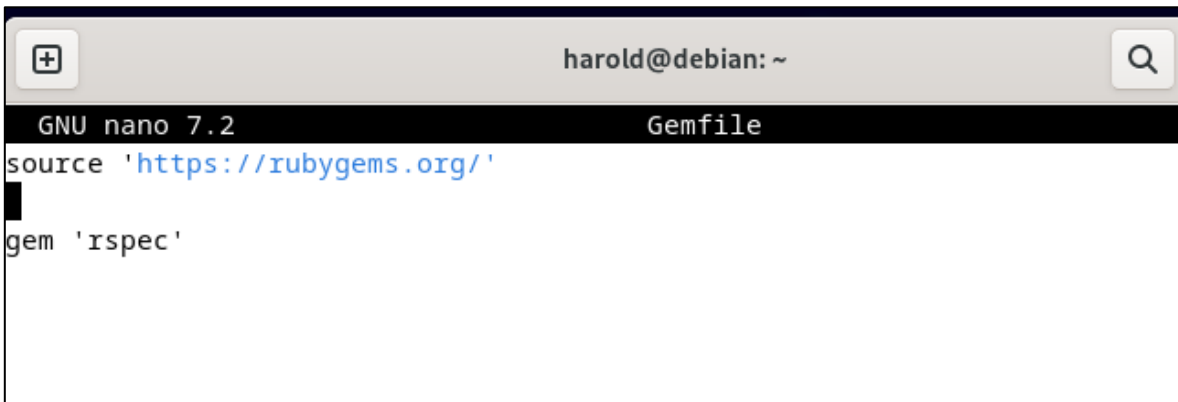
1. Crear un directorio donde almacenar el proyecto.

1.1. Crear un archivo llamado Gemfile dentro del directorio creado anteriormente, este archivo es donde se agregará la gema que necesita el proyecto, en este caso solo se utilizará la gema rspec para realizar los tests al código.

A terminal window titled 'harold@debian: ~' with search, menu, and close icons. The terminal shows the following commands and output:

```
root@debian:/home/harold/Nueva_practica# touch Gemfile
root@debian:/home/harold/Nueva_practica# ls
Gemfile
root@debian:/home/harold/Nueva_practica# nano Gemfile
root@debian:/home/harold/Nueva_practica#
```

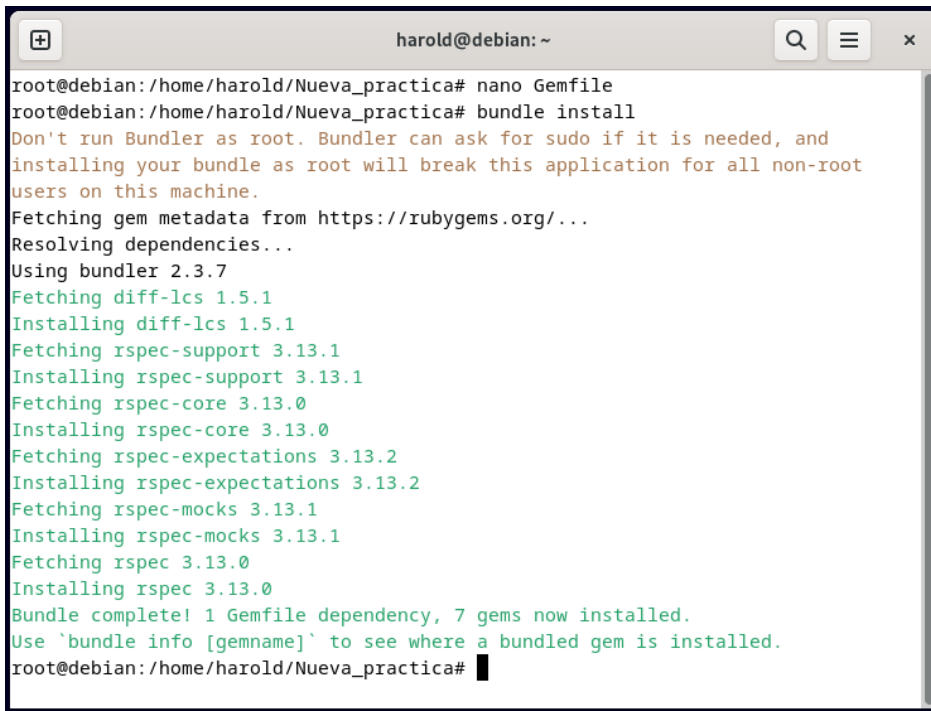
1.2. Abrir el archivo Gemfile y agregar las siguientes líneas, para poder integrar rspec al proyecto.

A terminal window titled 'harold@debian: ~' with search and close icons. The window shows the GNU nano 7.2 editor editing the Gemfile. The content of the file is:

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

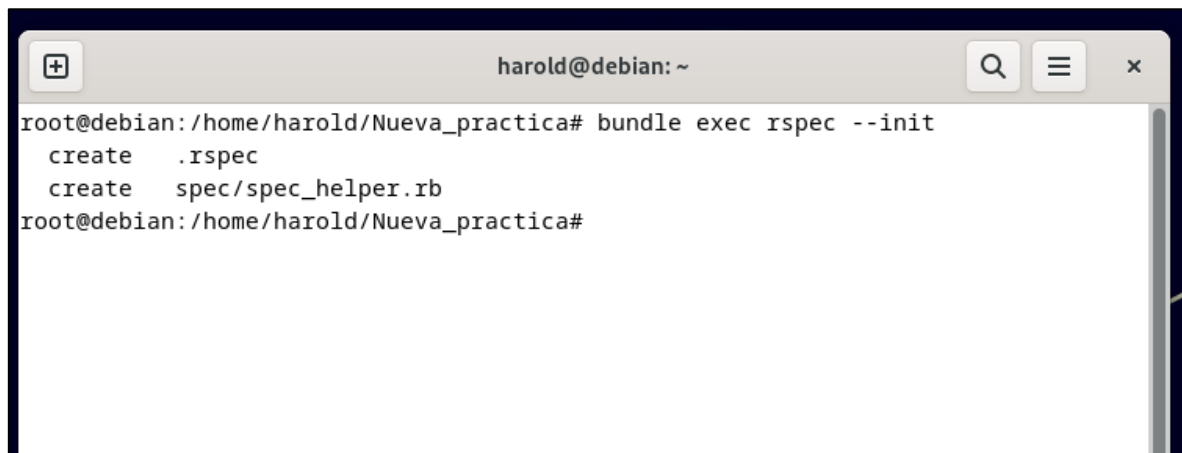
2. En el terminal, ubicarse dentro del directorio de trabajo y escribir los siguientes comandos.

2.1. Instalar la gema rspec, que es la que se utiliza para trabajar con TDD y que servirá para elaborar los tests.



```
harold@debian: ~  
root@debian:/home/harold/Nueva_practica# nano Gemfile  
root@debian:/home/harold/Nueva_practica# 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  
Installing diff-lcs 1.5.1  
Fetching rspec-support 3.13.1  
Installing rspec-support 3.13.1  
Fetching rspec-core 3.13.0  
Installing 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-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/harold/Nueva_practica#
```

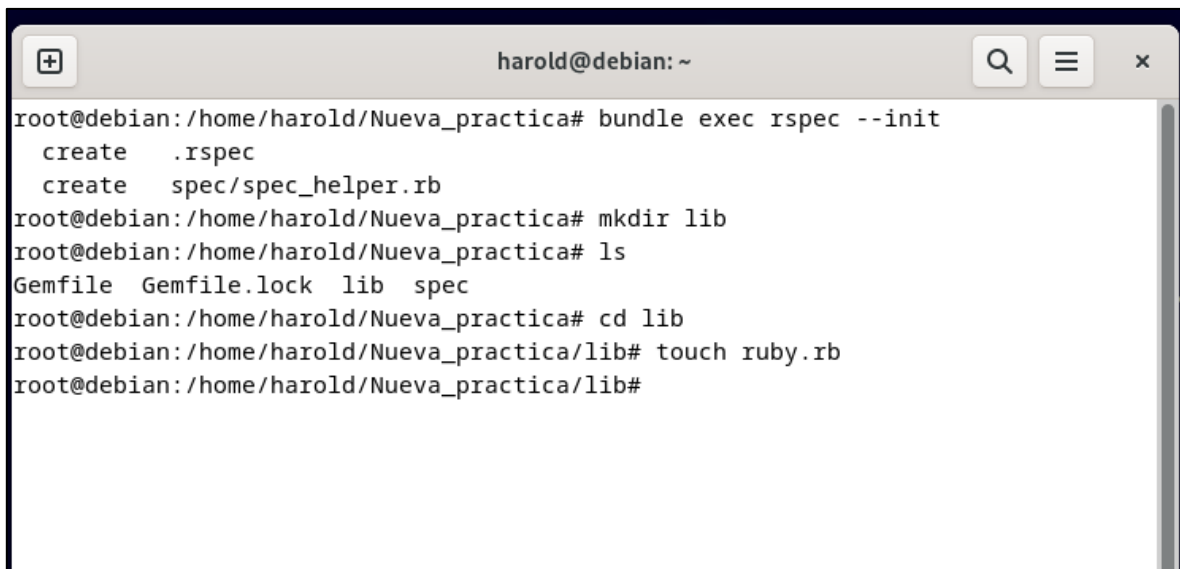
2.2. Ejecutar rspec. Se crearán automáticamente dos archivos: .rspec y spec/spec_helper.rb que son los archivos que utiliza rspec para correr los tests.



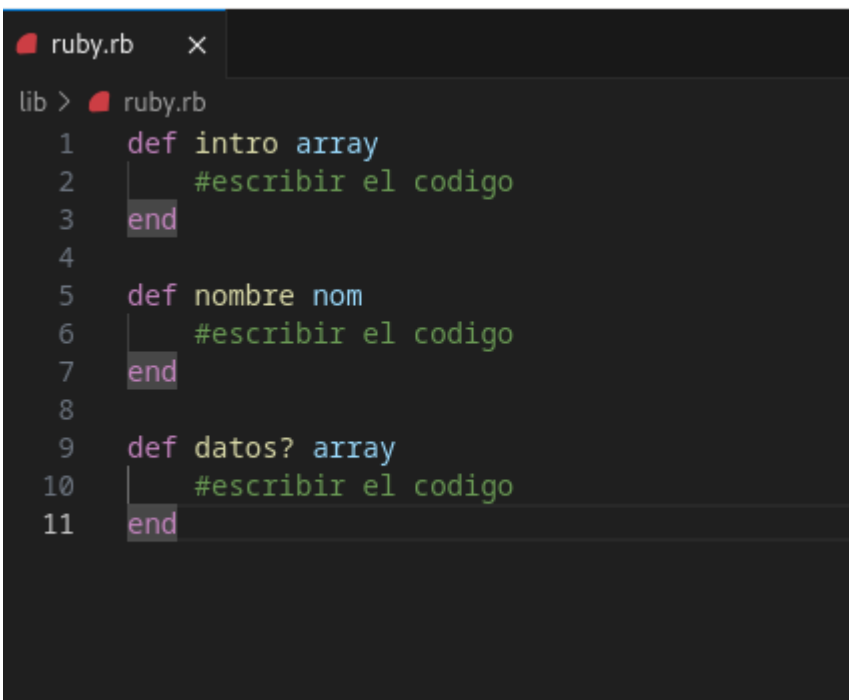
```
harold@debian: ~  
root@debian:/home/harold/Nueva_practica# bundle exec rspec --init  
create .rspec  
create spec/spec_helper.rb  
root@debian:/home/harold/Nueva_practica#
```

3. En el directorio raíz del proyecto, crear una carpeta nueva llamada lib

3.1. Dentro de la carpeta lib, crear un archivo ruby.rb y agregar lo siguiente.



```
harold@debian: ~  
root@debian:/home/harold/Nueva_practica# bundle exec rspec --init  
  create  .rspec  
  create  spec/spec_helper.rb  
root@debian:/home/harold/Nueva_practica# mkdir lib  
root@debian:/home/harold/Nueva_practica# ls  
Gemfile Gemfile.lock lib spec  
root@debian:/home/harold/Nueva_practica# cd lib  
root@debian:/home/harold/Nueva_practica/lib# touch ruby.rb  
root@debian:/home/harold/Nueva_practica/lib#
```



```
ruby.rb  
lib > ruby.rb  
1  def intro array  
2    #escribir el codigo  
3  end  
4  
5  def nombre nom  
6    #escribir el codigo  
7  end  
8  
9  def datos? array  
10   #escribir el codigo  
11 end
```

4. Dentro de la carpeta spec, crear un archivo rspec_1.rb y copiar el siguiente código, este código es cada uno de los tests aplicados, para escribir el código Ruby dentro de los métodos que están en el archivo ruby.rb.

```
harold@debian: ~/Nueva_practica/lib
root@debian:/home/harold/Nueva_practica/lib# cd ..
root@debian:/home/harold/Nueva_practica# cd spec/
root@debian:/home/harold/Nueva_practica/spec# touch rspec_1.rb
root@debian:/home/harold/Nueva_practica/spec#
```

```
File Edit Selection View Go Run Terminal Help
EXPLORER
  NUEVA_PRACTICA
    lib
    ruby.rb
    spec
      rspec_1.rb
      spec_helper.rb
      .rspec
      Gemfile
      Gemfile.lock
OUTLINE
TIMELINE
spec > rspec_1.rb
2 describe "#intro" do
14   it "retorna el valor del unico elemento del array" do
16   end
17 end
18 describe "#nombre" do
19   it "Debe iniciar con Mayúscula y concatena la cadena con la palabra r
20     expect(nombre("Pedro")).to eq("hello Pedro")
21     expect(nombre("Juan")).to eq("hello Juan")
22     expect(nombre("Maria")).to eq("hello Maria")
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
```

5. En el terminal, ejecutar el siguiente comando para verificar el funcionamiento de los test.

```
harold@debian: ~/Nueva_practica/spec
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 0.02214 seconds (files took 0.12711 seconds to load)
8 examples, 7 failures

Failed examples:

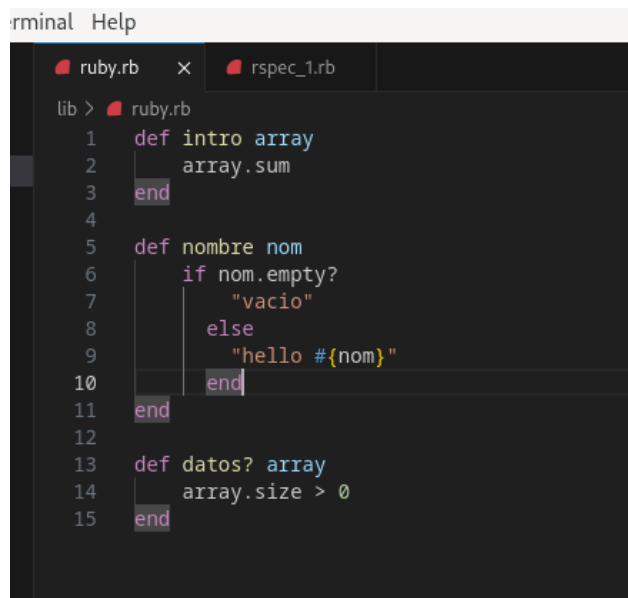
rspec ./spec/rspec_1.rb:6 # #intro retorna la suma del array
rspec ./spec/rspec_1.rb:11 # #intro retorna 0 si el array está vacío
rspec ./spec/rspec_1.rb:14 # #intro retorna el valor del único elemento del array
rspec ./spec/rspec_1.rb:19 # #nombre Debe iniciar con Mayúscula y concatena la cadena con 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/harold/Nueva_practica#
```

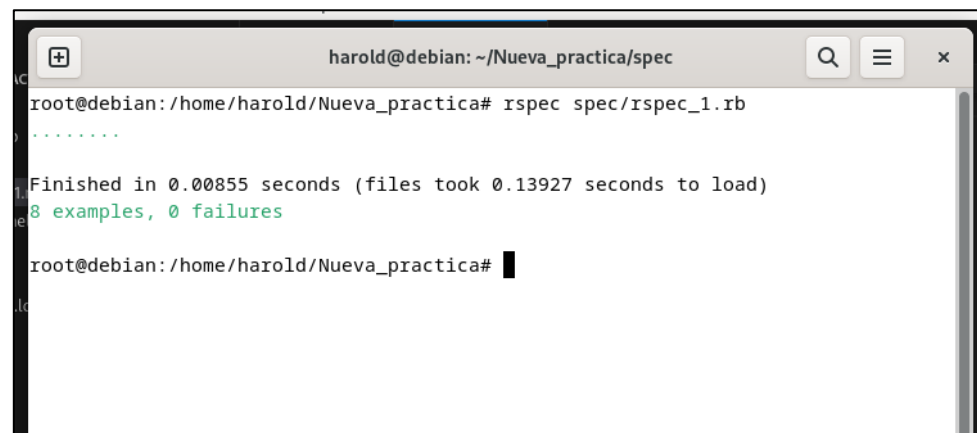
Al ejecutar el comando anterior mostrará un conjunto de mensajes en color rojo, los cuales corresponden a cada uno de los tests que se están aplicando en el ejercicio, cuando los tests se muestran en color rojo es debido a que los tests ejecutados están fallando, en relación a la funcionalidad del código al cual se le están aplicando cada uno de ellos.

Al final de todos los mensajes, se muestra exactamente los tests que están siendo ejecutados.

1. Realizar cada uno de los enunciados de la guía.
2. Escribir el código Ruby necesario dentro cada una de los métodos creados en el archivo lib/ruby.rb para aprobar todos los tests, y que muestre los mensajes en color verde como se observa.

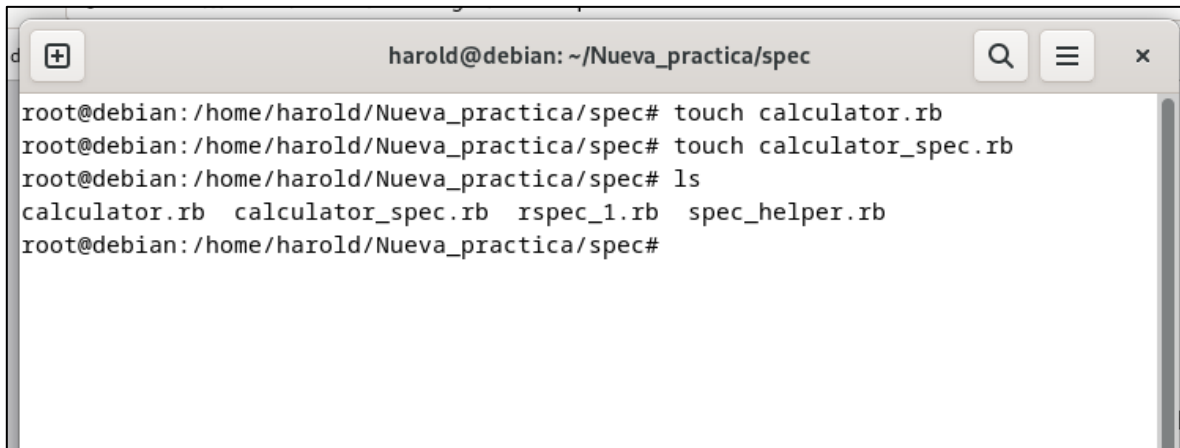


```
terminal Help
ruby.rb x rspec_1.rb
lib > ruby.rb
1 def intro array
2   array.sum
3 end
4
5 def nombre nom
6   if nom.empty?
7     "vacio"
8   else
9     "hello #{nom}"
10  end
11 end
12
13 def datos? array
14   array.size > 0
15 end
```



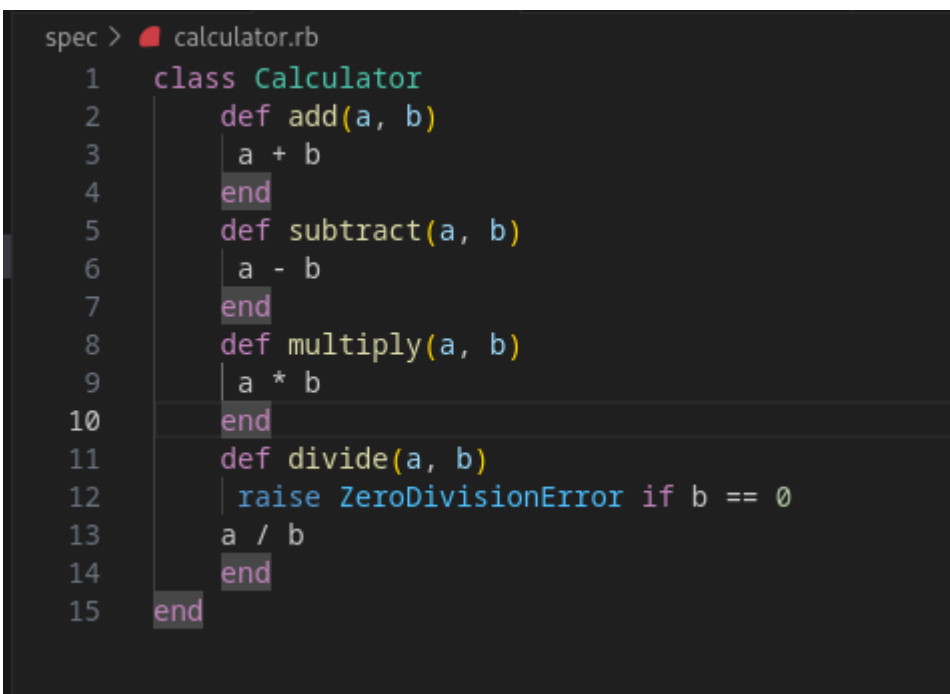
```
harold@debian: ~/Nueva_practica/spec
root@debian:/home/harold/Nueva_practica# rspec spec/rspec_1.rb
.....
Finished in 0.00855 seconds (files took 0.13927 seconds to load)
8 examples, 0 failures
root@debian:/home/harold/Nueva_practica#
```

Extra: escribir un conjunto de tests en relación a lo aprendido en la guía dentro del archivo `rspec_1.rb`, para luego crear un método y escribir código ruby de tal manera que pueda hacer funcionar los test a como se trabajó en la guía. 3 supongamos que tienes la siguiente clase `Calculator` en un archivo llamado `calculator.rb`:

A terminal window titled 'harold@debian: ~/Nueva_practica/spec'. It shows a series of commands and their outputs: 'touch calculator.rb', 'touch calculator_spec.rb', and 'ls'. The 'ls' command lists the files: 'calculator.rb', 'calculator_spec.rb', 'rspec_1.rb', and 'spec_helper.rb'.

```
harold@debian: ~/Nueva_practica/spec
root@debian:/home/harold/Nueva_practica/spec# touch calculator.rb
root@debian:/home/harold/Nueva_practica/spec# touch calculator_spec.rb
root@debian:/home/harold/Nueva_practica/spec# ls
calculator.rb  calculator_spec.rb  rspec_1.rb  spec_helper.rb
root@debian:/home/harold/Nueva_practica/spec#
```

Ahora, vamos a crear pruebas para esta clase utilizando RSpec. Crea un archivo llamado `calculator_spec.rb` para escribir tus pruebas:

A code editor showing the content of 'calculator_spec.rb'. The code defines a 'Calculator' class with methods for addition, subtraction, multiplication, and division. The division method includes a check for zero division. The code is numbered from 1 to 15.

```
spec > calculator.rb
1  class Calculator
2    def add(a, b)
3      a + b
4    end
5    def subtract(a, b)
6      a - b
7    end
8    def multiply(a, b)
9      a * b
10   end
11   def divide(a, b)
12     raise ZeroDivisionError if b == 0
13     a / b
14   end
15 end
```

```
spec > calculator_spec.rb
1 # calculator_spec.rb
2 require_relative 'calculator'
3
4 RSpec.describe Calculator do
5   let(:calculator) { Calculator.new }
6
7   describe "#add" do
8     it "sums two numbers" do
9       result = calculator.add(2, 3)
10      expect(result).to eq(5)
11    end
12  end
13
14  describe "#subtract" do
15    it "subtracts two numbers" do
16      result = calculator.subtract(5, 3)
17      expect(result).to eq(2)
18    end
19  end
20
21  describe "#multiply" do
22    it "multiplies two numbers" do
23      result = calculator.multiply(4, 3)
24      expect(result).to eq(12)
25    end
26  end
end
```

Ln 18, Col 8 Spaces:

```
> calculator_spec.rb
RSpec.describe Calculator do
  describe "#multiply" do
    it "multiplies two numbers" do
      result = calculator.multiply(4, 3)
      expect(result).to eq(12)
    end
  end

  describe "#divide" do
    it "divides two numbers" do
      result = calculator.divide(10, 2)
      expect(result).to eq(5)
    end

    it "raises an error when dividing by zero" do
      expect { calculator.divide(10, 0) }.to raise_error(ZeroDivisionError)
    end
  end
end
```

```
harold@debian: ~/Nueva_practica/spec
root@debian:/home/harold/Nueva_practica# rspec spec/calculator_spec.rb
.....

Finished in 0.00886 seconds (files took 0.13274 seconds to load)
5 examples, 0 failures

root@debian:/home/harold/Nueva_practica#
```