

SAAS

practica 1 Ruby

JOSE ABRAHAM SCHIFFMANN VANEGA 19-00501-0

Objetivos

- Aprender a declarar variables en Ruby.
- Conocer el funcionamiento de los métodos del lenguaje Ruby.
- Distinguir los tipos de objetos que contiene el lenguaje Ruby.

Requerimientos

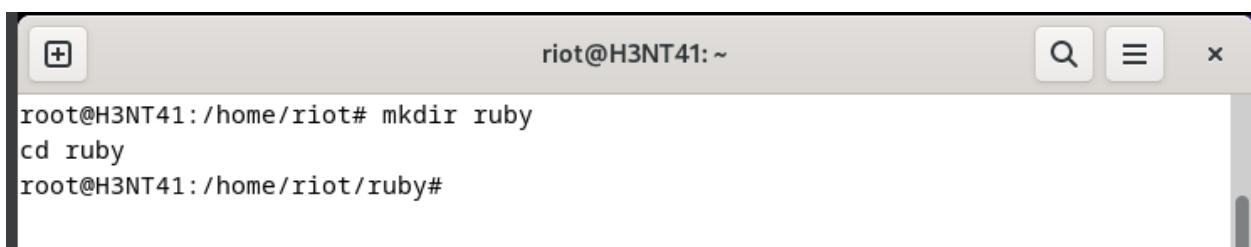
- **Software:** Sistema Debian 11 en Virtual Box con Ruby 2.4.1, Rails 5.4.1, Nodejs, y Vscode.
- **Hardware:** Computadora con mínimo 2GB de RAM y procesador de 2.1 GHz.

Desarrollo

1. Strings

1. Crear directorio:

```
mkdir ruby  
cd ruby
```

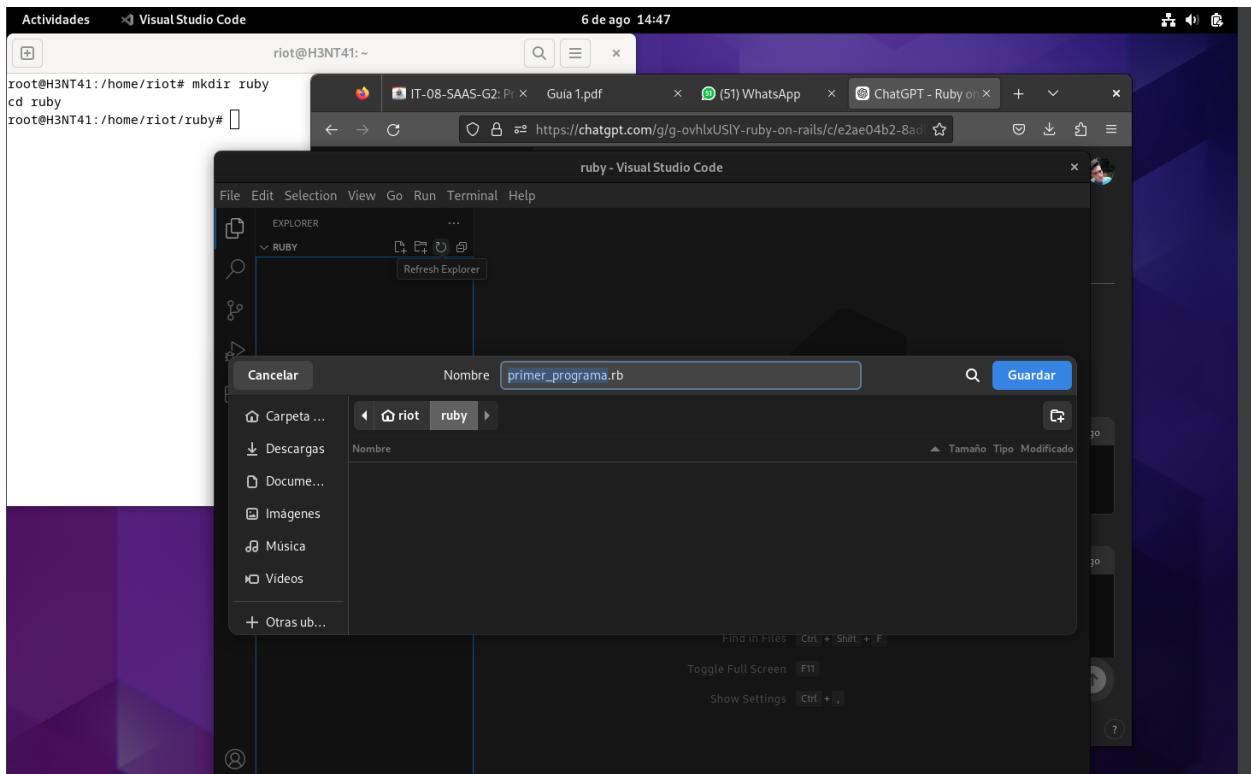


```
riot@H3NT41:~
```

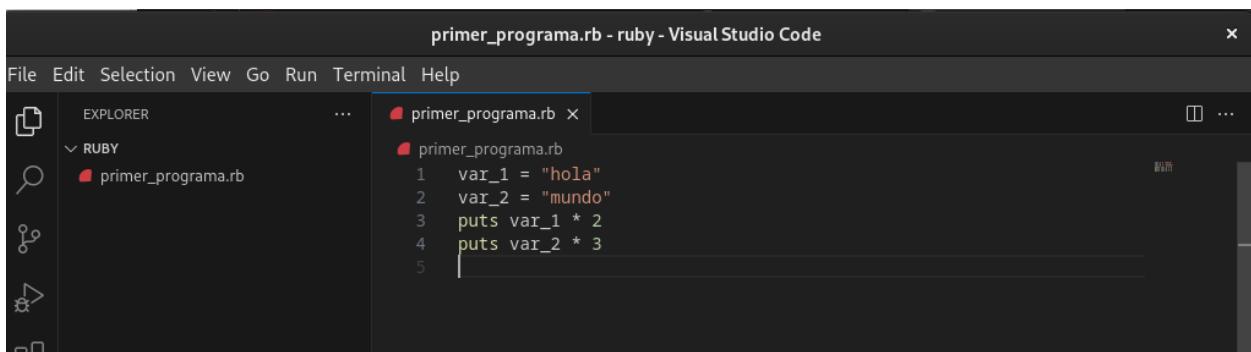
```
root@H3NT41:/home/riot# mkdir ruby
cd ruby
root@H3NT41:/home/riot/ruby#
```

2.Crear archivo

primer_programa.rb :



```
var_1 = "hola"
var_2 = "mundo"
puts var_1 + var_2
```



Ejecutar en terminal:

```
ruby primer_programa.rb
```

```
root@H3NT41:/home/riot/ruby# ruby primer_programa.rb
holahola
mundomundomundo
root@H3NT41:/home/riot/ruby#
```

2. Números

- **Crear archivo** `programa_numero.rb`:

```
var_1 = 20
var_2 = 5

# Suma
puts var_1 + var_2

# Resta
puts var_1 - var_2

# Multiplicar
puts var_1 * var_2

# Dividir
puts var_1 / var_2

# Módulo
puts var_1 % var_2

# Números aleatorios
puts rand(100)
```

```
primer_programa.rb
programa_numero.rb x

programa_numero.rb
1 var_1 = 20
2 var_2 = 5
3
4 # Suma
5 puts var_1 + var_2
6
7 # Resta
8 puts var_1 - var_2
9
10 # Multiplicar
11 puts var_1 * var_2
12
13 # Dividir
14 puts var_1 / var_2
15
16 # Módulo
17 puts var_1 % var_2
18
19 # Números aleatorios
20 puts rand(100)
21
```

- Ejecutar en terminal:

```
ruby programa_numero.rb
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code [Running] ruby "/home/riot/ruby/programa_numero.rb"
25
15
100
4
0
54

[Done] exited with code=0 in 0.199 seconds
```

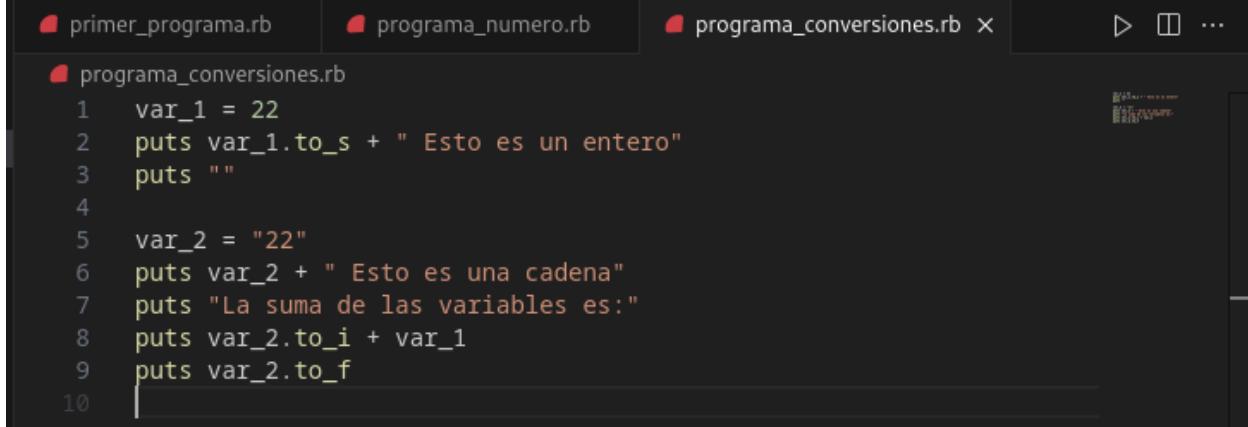
Ln 21, Col 1 Spaces: 2 UTF-8 LF ↻ Ruby ⚙

3. Conversiones

1. Crear archivo `programa_conversiones.rb`:

```
var_1 = 22
puts var_1.to_s + " Esto es un entero"
puts ""
```

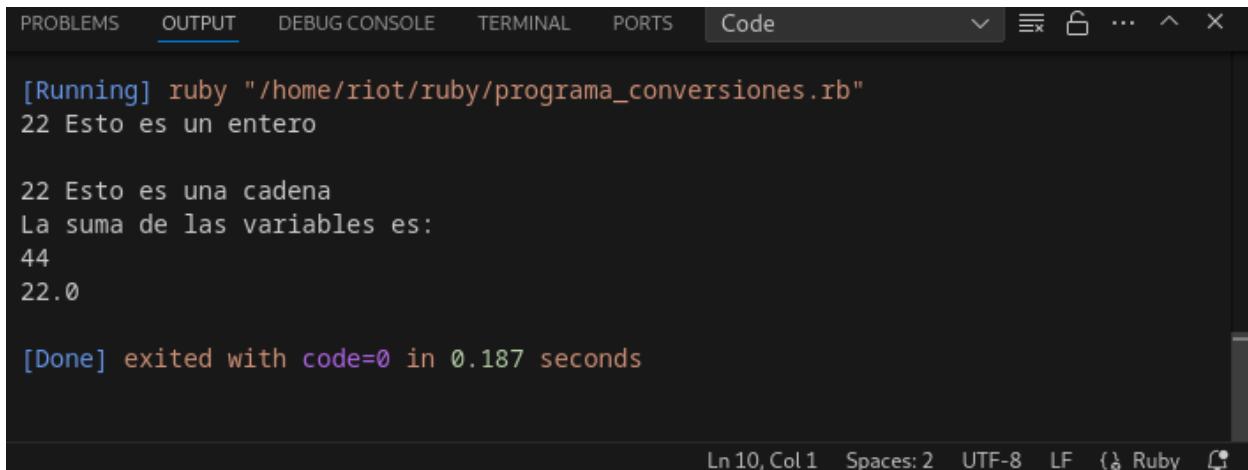
```
var_2 = "22"
puts var_2 + " Esto es una cadena"
puts "La suma de las variables es:"
puts var_2.to_i + var_1
puts var_2.to_f
```



```
primer_programa.rb | programa_numero.rb | programa_conversiones.rb ×
programa_conversiones.rb
1 var_1 = 22
2 puts var_1.to_s + " Esto es un entero"
3 puts ""
4
5 var_2 = "22"
6 puts var_2 + " Esto es una cadena"
7 puts "La suma de las variables es:"
8 puts var_2.to_i + var_1
9 puts var_2.to_f
10
```

2. Ejecutar en terminal:

```
ruby programa_conversiones.rb
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code
[Running] ruby "/home/riot/ruby/programa_conversiones.rb"
22 Esto es un entero

22 Esto es una cadena
La suma de las variables es:
44
22.0

[Done] exited with code=0 in 0.187 seconds
```

Ln 10, Col 1 Spaces: 2 UTF-8 LF {Ruby} ⚡

4. Métodos `gets` y `chomp`

1. Crear archivo `leer.rb`:

```
puts "Ingrese su primer nombre"
nombre = gets.chomp
puts "Bienvenido #{nombre} disfrute!"
```

2. Ejecutar en terminal:

ruby leer.rb

captura de vs code



```
leer.rb
1 puts "Ingrese su primer nombre"
2 nombre = gets.chomp
3 puts "Bienvenido #{nombre} disfrute!"
4
```

captura de la ejecución:

```
root@H3NT41:/home/riot/ruby# ruby leer.rb
Ingrese su primer nombre
abraham
Bienvenido abraham disfrute!
root@H3NT41:/home/riot/ruby#
```

5. Métodos de String

1. Crear archivo `string.rb`:

```
puts "Ingrese su nombre"
nombre = gets.chomp

puts "Nombre => " + nombre
puts "Método reverse => " + nombre.reverse
puts "Método upcase => " + nombre.upcase
puts "Método downcase => " + nombre.downcase
puts "Método swapcase => " + nombre.swapcase
puts "Método capitalize => " + nombre.capitalize
puts "Método length => " + nombre.length.to_s
```

2. Ejecutar en terminal:

ruby string.rb

captura de vs code

```
string.rb - ruby - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER ... irograma_numero.rb programa_conversiones.rb leer.rb string.rb ...
RUBY
leer.rb
primer_programa.rb
programa_conversiones.rb
programa_numero.rb
string.rb
```

```
1 puts "Ingrese su nombre"
2 nombre = gets.chomp
3
4 puts "Nombre => " + nombre
5 puts "Método reverse => " + nombre.reverse
6 puts "Método upcase => " + nombre.upcase
7 puts "Método downcase => " + nombre.downcase
8 puts "Método swapcase => " + nombre.swapcase
9 puts "Método capitalize => " + nombre.capitalize
10 puts "Método length => " + nombre.length.to_s
```

captura de ejecución

```
root@H3NT41:/home/riot/ruby# ruby string.rb
Ingrese su nombre
pepito
Nombre => pepito
Método reverse => otipep
Método upcase => PEPITO
Método downcase => pepito
Método swapcase => PEPITO
Método capitalize => Pepito
Método length => 6
root@H3NT41:/home/riot/ruby#
```

6. Condicionales y bucles

1. Crear archivo `condicionales_bucles.rb`:

```
iterador = " "

while iterador.downcase != "s"
    puts "Ingrese un nombre"
    nombre = gets.chomp
    tamaño = nombre.length

    if tamaño >= 5
        puts "Su nombre tiene más de 5 caracteres"
    else
        puts "Su nombre tiene menos de 5 caracteres"
    end

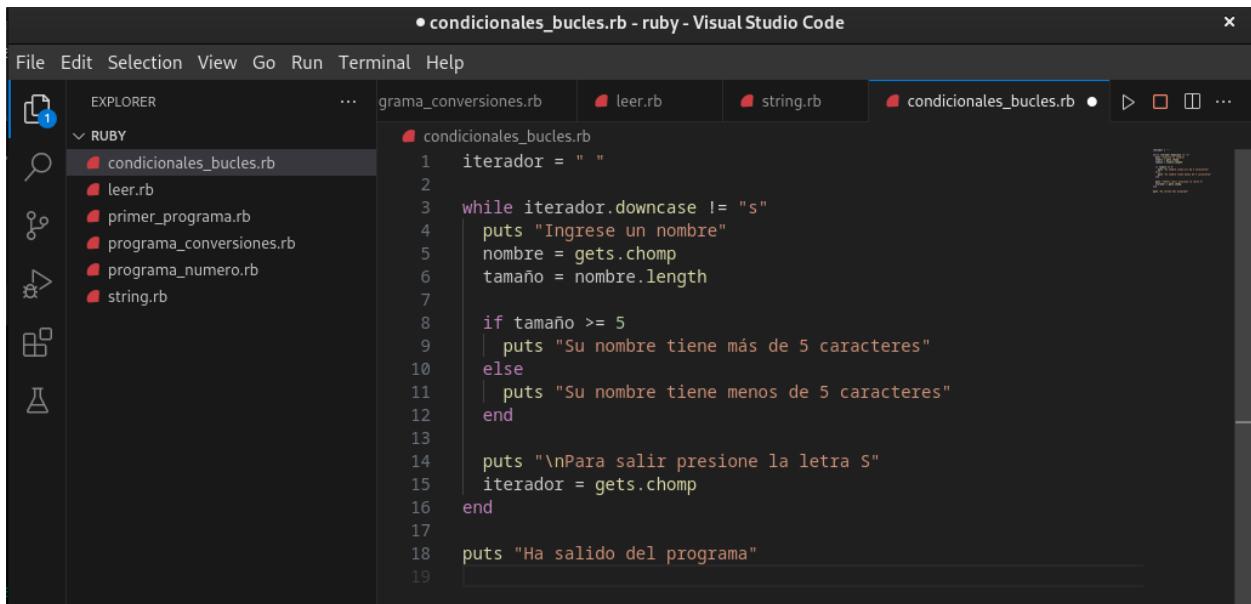
    puts "\nPara salir presione la letra S"
    iterador = gets.chomp
end

puts "Ha salido del programa"
```

2. Ejecutar en terminal:

```
ruby condicionales_bucles.rb
```

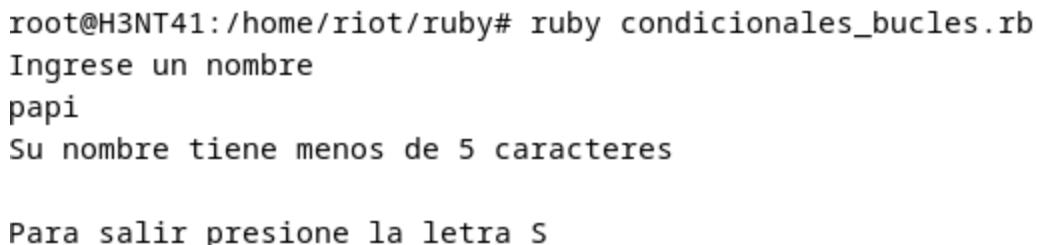
captura de vs code



```
• condicionales_bucles.rb - ruby - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER ... grama_conversiones.rb leer.rb string.rb condicionales_bucles.rb • ⌂ ⌂ ⌂ ...
RUBY
condicionales_bucles.rb
leer.rb
primer_programa.rb
programa_conversiones.rb
programa_numero.rb
string.rb

1 iterador = " "
2
3 while iterador.downcase != "s"
4   puts "Ingrese un nombre"
5   nombre = gets.chomp
6   tamaño = nombre.length
7
8   if tamaño >= 5
9     | puts "Su nombre tiene más de 5 caracteres"
10  else
11    | puts "Su nombre tiene menos de 5 caracteres"
12  end
13
14  puts "\nPara salir presione la letra S"
15  iterador = gets.chomp
16 end
17
18 puts "Ha salido del programa"
19
```

captura de ejecución



```
root@H3NT41:/home/riot/ruby# ruby condicionales_bucles.rb
Ingrese un nombre
papi
Su nombre tiene menos de 5 caracteres

Para salir presione la letra S
```

Ejercicio Propuesto

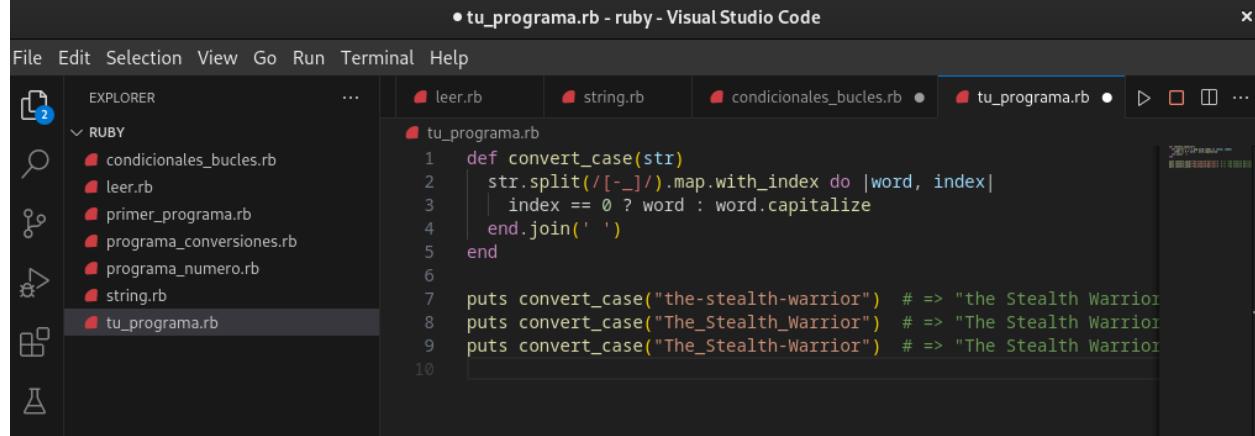
Completa una función que convierta las palabras delimitadas por guiones o guiones bajos en mayúsculas y minúsculas, manteniendo la primera palabra en mayúsculas solo si estaba originalmente en mayúsculas.

Ejemplo de implementación:

```
def convert_case(str)
  str.split(/[-_]/).map.with_index do |word, index|
    index == 0 ? word : word.capitalize
  end.join(' ')
end

puts convert_case("the-stealth-warrior") # => "The Stealth W
```

```
arrior"
puts convert_case("The_Stealth_Warrior") # => "The Stealth W
arrior"
puts convert_case("The_Stealth-Warrior") # => "The Stealth W
arrior"
```



The screenshot shows the Visual Studio Code interface with the title bar "tu_programa.rb - ruby - Visual Studio Code". The menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The Explorer sidebar on the left shows a folder named "RUBY" containing several Ruby files: condicionales_bucles.rb, leer.rb, primer_programa.rb, programa_conversiones.rb, programa_numero.rb, string.rb, and tu_programa.rb. The "tu_programa.rb" file is currently selected and its content is displayed in the main editor area:

```
tu_programa.rb
1 def convert_case(str)
2   str.split(/[-_]/).map.with_index do |word, index|
3     index == 0 ? word : word.capitalize
4   end.join(' ')
5 end
6
7 puts convert_case("the-stealth-warrior") # => "the Stealth Warrior
8 puts convert_case("The_Stealth_Warrior") # => "The Stealth W
9 puts convert_case("The_Stealth-Warrior") # => "The Stealth W
```

Ejecutar en terminal:

```
ruby tu_programa.rb
```

```
root@H3NT41:/home/riot/ruby# ruby tu_programa.rb
the Stealth Warrior
The Stealth Warrior
The Stealth Warrior
root@H3NT41:/home/riot/ruby#
```