

Unidad 3

3.1) Métodos en Ruby

Temas: Definir métodos. Retorno y retorno implícito. Parámetros opcionales y obligatorios.

Métodos

Hasta acá hemos trabajado con variables, operadores lógicos y ciclos. Llega el momento de crear un método. Un método es un objeto que nos permite realizar una acción, sumar números, imprimir un texto, contar, especialmente útil si es una acción que se repetirá.

Los métodos pueden ser

Recibir parámetros o argumentos

No recibir parámetros o argumentos

Recibir y enviar parámetros y argumentos

No recibir y enviar parámetros y argumentos

Definición de un método

```
def
```

```
def nombreDelMetodo
```

```
end
```

Project	holamundo.rb	03Var.rb	09Math2.rb	09Estructura.rb	016While.rb	020Metodo.rb	017Case.rb	018For.rb	010Estructura.rb
03Var.rb	8								
04Time.rb	9								
05Error.rb	10								
06Hashes.rb	11								
07Hashes.rb	12								
08Math2.rb	13								
09Math2.rb	14								
holamundo.rb	15								
2	16								
bonus	17								
.DS_Store	18								
08Get.rb	19								
09Estructura.rb	20								
010Estructura.rb	21								
011Metodo.rb	22								
012Rescate.rb	23								
013For.rb	24								
014While.rb	25								
015Each.rb	26								
016While.rb	27								
017Case.rb	28								
018For.rb	29								
020Metodo.rb	30								
tallerSesion2.pdf	31								
programming-ruby.pdf	32								
ruhvininstaller-devkit-2.7.2-1-x64.exe	33								

```
8
9 principal = "SUMA DE DE NÚMEROS"
10 puts principal
11
12 textos = "Ingresa un número y presiona enter"
13 puts textos
14
15 numero1 = gets.chomp
16
17 textos2 = "Ingresa otro número y presiona enter"
18 puts textos2
19
20 numero2 = gets.chomp
21
22 entero1 = numero1.to_i
23 entero2 = numero2.to_i
24
25 def suma(numero1, numero2)
26   puts numero1+numero2
27   return 'correcto'
28 end
29
30 puts(suma(entero1,entero2))
31
32 def nada
33   puts "Soy la función texto y no hago nada más que imprimir este texto"
34 end
35
36 puts nada
37
```

Unidad 3

3.2) Variables locales y variables globales

Temas: Alcance de las variables locales. El problema de las variables globales.

Variable locales y globales

En la mayoría de los lenguajes de programación existen las variables locales y variables globales.

Las variables locales son aquellas que están delimitadas por un ámbito (generalmente una estructura de control o ciclo o método), por tanto solo pueden ser utilizadas por en ese ámbito.

Las variables globales pueden ser utilizadas en cualquier momento de la programación, independiente si están o no en el interior de una estructura de control.

Project

holamundo.rb

09Math2.rb

021Var.rb

03Var.rb

017Case.rb

018For.rb

010Estructura.rb

- 03Var.rb
- 04Time.rb
- 05Error.rb
- 06Hashes.rb
- 07Hashes.rb
- 08Math2.rb
- 09Math2.rb
- holamundo.rb

2

> bonus

- .DS_Store
- 08Get.rb
- 09Estructura.rb
- 010Estructura.rb
- 011Metodo.rb
- 012Rescate.rb
- 013For.rb
- 014While.rb
- 015Each.rb
- 016While.rb
- 017Case.rb
- 018For.rb
- 020Metodo.rb
- 021Var.rb
- tallerSesion2.pdf

programminio-ruby.pdf

```
2  titulo2 = "2021"
3  titulo3 = "CHILE"
4
5
6
7
8  puts titulo1
9  puts titulo2
10 puts titulo3
11
12 $global_variable = 10
13 variable_local = 5
14
15 class Global
16   def print_global
17     puts "Variable global es #{ $global_variable } "
18   end
19 end
20 class Local
21   def print_global
22     puts "Variable local es #{ variable_local }"
23   end
24 end
25
26 class1obj = Global.new
27 class1obj.print_global
28 class2obj = Local.new
29 class2obj.print_global
30
```

Material complementario de la unidad

Link a video relacionado

1. <https://www.youtube.com/watch?v=2A2RrMdXEpl>

Link a lectura complementaria

1. <https://makeitrealcamp.gitbook.io/ruby-book/metodos>

Link a investigación relacionada

1. https://docs.ruby-lang.org/en/2.7.0/doc/syntax/methods_rdoc.html