

Unidad 5

5.1) Hashes en Ruby

Temas: Creación de hashes. Diferencia entre índice y clave. Strings, símbolos y mutabilidad.

Hashes

Ya hemos conocido los arrays y las matrices, toca ahora conocer los hashes, que nuevamente son conjuntos de datos, pero permiten realizar operaciones o consultas a partir de su clave (key) o valor (value).

Un hash tiene la siguiente estructura, donde el número es la clave y el lenguaje de programación el valor.

```
lenguajes = { 1968 => "Logo", 1972 => "Smalltalk" }
```

```
hash = { clave => "value", clave => "value" }
```

La clave (key) es justo el dato que tenemos antes de del símbolo => y el valor el que va después de ese símbolo, se pueden agregar cuantos se necesitan, separándolos por una coma.

Project

▼ ruby

▼ 1

01Puts.rb

02Math.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

holamundo...

09Math2.rb

iteraciones...

ciclosFoArr...

arrayPushP...

matriz2.rb

07Hashes. •

ciclosArray...

arrayBasico...

017Case.rb

018For.rb

010Estructu...

```
1 titulo1 = "Módulo introducción a Ruby"
2 titulo2 = "2021"
3 titulo3 = "CHILE"
4
5
6 puts titulo1
7 puts titulo2
8 puts titulo3
9
10 #Hashes
11
12 #len = Hash.new
13
14 len = { 1968 => "Logo", 1972 => "Smalltalk", 1978 => "SQL", 1980 =>
15 "C++", 1984 => "Matlab", 1987 => "Perl", 1991 => "Python", 1993 => "Ruby!", 1995 => "Php" }
16
17 puts len.has_key? 1938 # comprueba si es que existe un registro con valor 1938, false
18 puts len.has_key? 1993 # comprueba si es que existe un registro con valor 1993, true
19 puts len.has_value? "Ruby" # comprueba si es que existe un registro con valor Ruby, true
20 puts len[1972] #imprime el valor del año 1972, samlltalk
21 puts len.keys #imprime todas las claves
22 puts len.values #imprime todos los valores
23 puts len.values_at 1968 #imprime el valor de 1968, Logo
24 puts len.key "SQL" #comprueba si hay una clave SQL, false
25 puts len.select { |key,val| key < 1980 } # consulta todas las claves y valores cuando la fecha sea menor a 1980
26
```

1\07Hashes.rb* 25:112



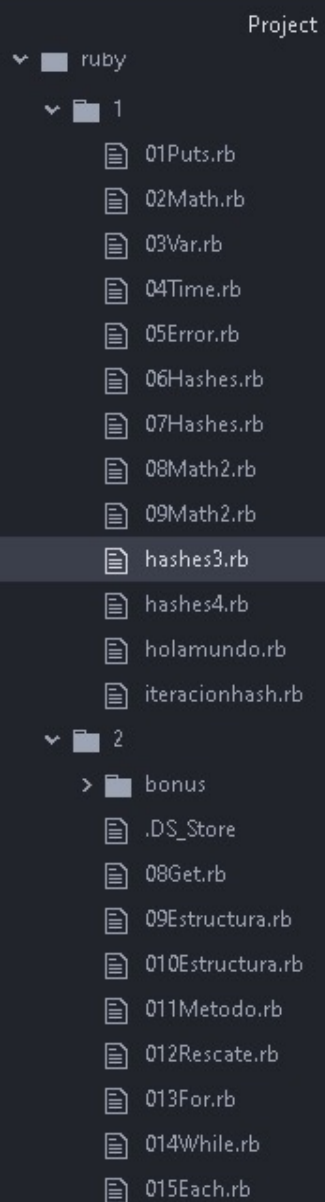
Unidad 5

5.2) Iteración de hashes

Temas: Invertir un hash. Merge de hashes. Agrupar elementos en un arreglo para generar un hash (con `.group_by`).
JSON y hashes.

Iteración de hashes

Para iterar un hash podemos hacerlo por su key y val, o solo por su key. también podemos **aplicar una condición**.



```
holamund... 09Math2.rb iteraciones... ciclosFoAr... arrayPush... matriz2.rb hashes4.rb iteracionh... ciclosArray... arrayBasic... 017Case.rb 018For.rb 010Estru...

1  titulo1 = "Módulo introducción a Ruby"
2  titulo2 = "2021"
3  titulo3 = "CHILE"
4
5
6  puts titulo1
7  puts titulo2
8  puts titulo3
9
10 #Hashes
11
12 #len = Hash.new
13
14 len = { 1968 => "Logo", 1972 => "Smalltalk", 1978 => "SQL", 1980 =>
15 "C++", 1984 => "Matlab", 1987 => "Perl", 1991 => "Python", 1993 => "Ruby!", 1995 => "Php" }
16
17 puts len.select { |key,val| key < 1980 } # consulta todas las claves y valores cuando la fecha sea menor a 1980
18 puts len.select { |key| key == 1972 } # consulta la claves y valores cuando la fecha sea igual a 1972
19
```

Invertir un hash, mutabilidad

Puede ser útil invertir un hash en algunos casos dependiendo de la naturaleza del conjunto de datos. Se puede invertir lo que era clave (key) a valor.

Project

▼ ruby

▼ 1

- 01Puts.rb
- 02Math.rb
- 03Var.rb
- 04Time.rb
- 05Error.rb
- 06Hashes.rb
- 07Hashes.rb
- 08Math2.rb
- 09Math2.rb
- hashes3.rb
- hashes4.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

holamundo... 09Math2.rb iteraciones... ciclosFoArr... arrayPushP... matriz2.rb hashes4.rb ciclosArray... arrayBasico... 017Case.rb 018For.rb 010Estructu...

```
1 titulo1 = "Módulo introducción a Ruby"
2 titulo2 = "2021"
3 titulo3 = "CHILE"
4
5
6 puts titulo1
7 puts titulo2
8 puts titulo3
9
10 #Hashes
11
12 #len = Hash.new
13
14 len = { 1968 => "Logo", 1972 => "Smalltalk", 1978 => "SQL", 1980 =>
15 "C++", 1984 => "Matlab", 1987 => "Perl", 1991 => "Python", 1993 => "Ruby!", 1995 => "Php" }
16
17
18
19 puts len.invert
20
```


Merge

La función merge nos permite unir dos hashes, se puede imprimir por consola directamente o crear nuevo hash.

Project

▼ ruby

▼ 1

- 01Puts.rb
- 02Math.rb
- 03Var.rb
- 04Time.rb
- 05Error.rb
- 06Hashes.rb
- 07Hashes.rb
- 08Math2.rb
- 09Math2.rb
- hashes3.rb
- hashes4.rb
- holamundo.rb
- iteracionhash.rb
- mergehash.rb

▼ 2

- > bonus
 - .DS_Store
 - 08Get.rb
 - 09Estructura.rb
 - 010Estructura.rb
 - 011Metodo.rb
 - 012Rescate.rb
 - 013For.rb
 - 014While.rb

- holamund...
- 09Math2.rb
- iteraciones...
- ciclosFoAr...
- arrayPush...
- matriz2.rb
- hashes4.rb
- mergehas...
- ciclosArray...
- arrayBasic...
- 017Case.rb
- 018For.rb
- 010Es...

```
1  titulo1 = "Módulo introducción a Ruby"
2  titulo2 = "2021"
3  titulo3 = "CHILE"
4
5
6  puts titulo1
7  puts titulo2
8  puts titulo3
9
10 #Hashes
11
12 #len = Hash.new
13
14 len1 = { 1968 => "Logo", 1972 => "Smalltalk" }
15 len2 = { 1978 => "SQL", 1980 => "C++" }
16
17 puts len1.merge(len2)
18
```



.json

Uno de los formatos de datos más extendidos en el actual escenario digital es JSON (Javascript object notation). En el siguiente ejemplo veremos cómo leer un .json

Necesitamos 2 gemas

```
gem install json
```

```
gem install pp
```

Tenemos un archivo que se llama pais.json con la siguiente estructura.

Project

session3

media

_.DS_Store

_.016crear_ficha.rb

_.navegador.rb

_.README

_.tallerS3.docx

_.DS_Store

016crear_ficha.rb

016jsonr.rb

017xml_node.rb

018xml_node.rb

019xml_gets.rb

020gosu_gui.rb

021gosu_gui.rb

022gosu_error.rb

023html.rb

024html.rb

025audio.rb

026gui.rb

027imagen.rb

demo.json

menu.xml

navegador.rb

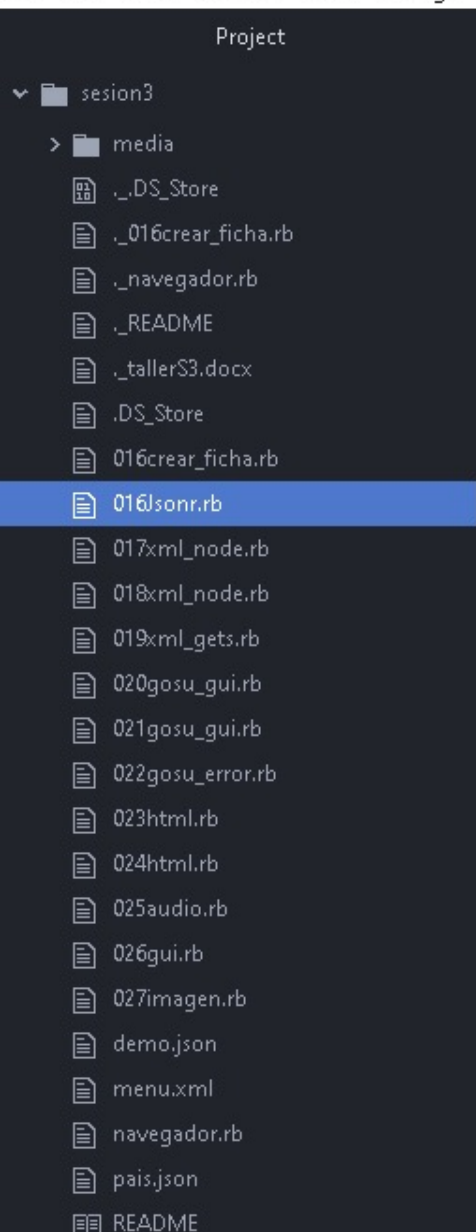
pais.json

README

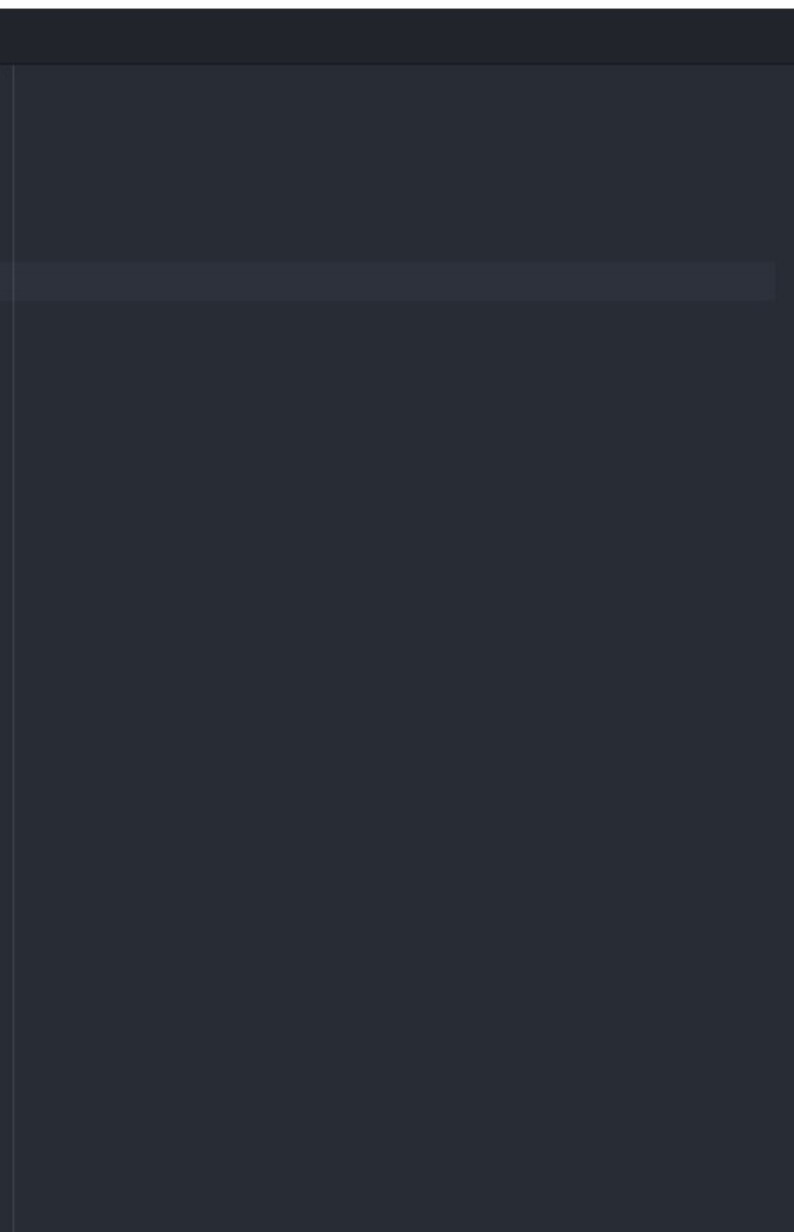
016jsonr.rb

pais.json

```
1 {
2   "Pais": "Chile",
3   "Presidente": "Sebastián Piñera",
4
5   "Santiago": [
6     "Felipe Alessandri",
7     "Carolina Toha"
8   ],
9
10  "Valparaiso": [
11    "Jorge Sharp",
12    "Marina Huerta"
13  ],
14
15  "Iquique": [
16    "Mauricio Soria",
17    "Jorge Soria"
18  ]
19
20 ]
21
22 }
23
```



```
016lsonr.rb
1  require 'rubygems'
2  require 'json'
3  require 'pp'
4
5  titulo1 = "Curso Ruby"
6  titulo2 = "2021"
7  titulo3 = "CHILE"
8
9  puts titulo1
10 puts titulo2
11 puts titulo3
12
13 json = File.read('pais.json')
14 obj = JSON.parse(json)
15 pp obj
16
17 candidato = obj['Santiago'][1]
18 puts candidato
19 pais = obj['Pais']
20 puts pais
21
```



Material complementario de la unidad

Link a video relacionado

1. <https://www.youtube.com/watch?v=YDTwvDADC9o>

Link a lectura complementaria

1. <https://ruby-doc.org/core-3.0.0/Hash.html>

Link a investigación relacionada

1. <https://www.digitalocean.com/community/tutorials/understanding-data-types-in-ruby>