


Nelson Jiménez (@_patojimenez)

Conociendo Ruby



- 28 Años
- Programando **en Ruby desde el 2011**
- CTO en **Shipit (Chile)** 

github.com/patojimenez
linkedin.com/in/nelsonpatriciojimenez



Tú vendes. Nosotros hacemos el resto.

Ahorra tiempo y dinero con nuestro servicio integral de logística para tu tienda online.

[Solicita tu cuenta gratis](#)



Soluciones Shipit





Shipit



Shipit

- 28 Años
 - Programando **en Ruby desde el 2011**
 - CTO en **Shipit (Chile)** 🐦
 - Ruby Backend / API en **Logan Mobile (Argentina)**
-

github.com/patojimenez

linkedin.com/in/nelsonpatriciojimenez



Agenda

- Qué es Ruby ?
- Configuración
- Comentarios
- Operaciones Matemáticas
- Variables y nombres
- Cadenas y textos
- Lectura de archivos de Texto
- Escritura de archivos de Texto
- Funciones
- Declaraciones (If, Else, Case)
- Bucles y Arreglos
- Hashes
- Clases e Instancias de clases

“Ruby es un lenguaje **diseñado para la felicidad del programador**”

Yukihiro Matsumoto (Matz) 

Qué es Ruby ?

- Lenguaje **Orientado a Objetos**
- La primera versión fue **liberada en 1995** (Se empezó a trabajar en 1993)
- El creador es Yukihiro Matsumoto (**Matz**)
- Se basa en: Perl, Smalltalk, Eiffel, Ada, Lisp
- Sintaxis fácil de entender
- Se empieza a adoptar fuera de Japón en el año 2001 (Se publica el primer libro en inglés)
- En el año **2004 nace RubyGems** 

Qué es Ruby ?

```
[9] pry(main)> 1.class  
=> Fixnum  
[10] pry(main)> true.class  
=> TrueClass  
[11] pry(main)> "Hola Mundo".class  
=> String  
[12] pry(main)> [2,2,"tres"].class  
=> Array  
[13] pry(main)> |
```



“En Ruby todo es un
objeto”

Configuración

- En sistemas basados en Unix (Distros de linux y IOs) siempre trae instalada una versión de Ruby.
- **Recomendado:** Usar RVM o RBENV (Manejador de versiones de Ruby)
 - **RVM:** <https://rvm.io/rvm/install>
 - **RBENV:** <https://github.com/rbenv/rbenv>
- Para Windows usar RailsInstaller: <http://railsinstaller.org/en>

Configuración

- En sistemas basados en Unix (Distros de linux y IOs) siempre trae instalada una versión de Ruby.
- **Recomendado:** Usar RVM o RBENV (Manejador de versiones de Ruby)
 - **RVM:** <https://rvm.io/rvm/install>
 - **RBENV:** <https://github.com/rbenv/rbenv>
- ~~- Para Windows usar RailsInstaller: <http://railsinstaller.org/en>~~
- **Quemar disco duro y pedir perdón a Dios por usar Windows**

Mensaje Subliminal



“Repitan conmigo: No usar Windows, No Usar Windows”

Configuración

- En sistemas basados en Unix (Distros de linux y IOs) siempre trae instalada una versión de Ruby.
- **Recomendado:** Usar RVM o RBENV (Manejador de versiones de Ruby)
 - **RVM:** <https://rvm.io/rvm/install>
 - **RBENV:** <https://github.com/rbenv/rbenv>
- ~~- Para Windows usar RailsInstaller: <http://railsinstaller.org/en>~~
- **Crear máquina Virtual con distro de Linux (Compatibilidad con algunas gemas)**

Comentarios

```
1 class ShipService
2   #TODO: Revisar método get y verificar que $redis no es empty
3   KEY = "shipping_prices"
4   STORE_LIMIT = 1
5
6   def self.send(package)
7     | package = Package.new(Package)
8     | Publisher.publish("shipping", package)
9   end
10
11 =begin
12   def self.get(package)
13     | $redis.lrange(KEY, 0, limit-1).map do |raw_package|
14       | | JSON.parse(raw_package).with_indifferent_access
15     | end
16   end
17 =end
18
19   def self.push(package)
20     | $redis.lpush(KEY, package)
21     | $redis.ltrim(KEY, 0, STORE_LIMIT-1)
22   end
23 end
```

**“Good code is like a good joke:
it needs no explanation”**

Russ Olsen - Eloquent Ruby

Operaciones Matemáticas

2.2.3 :001 > 3 + 1

=> 4

2.2.3 :002 > 2 * 2

=> 4

2.2.3 :003 > 3 - 1

=> 2

2.2.3 :004 > 3/3

=> 1

2.2.3 :005 > 3 < 3

=> false

2.2.3 :006 > 4 < 2

=> false

2.2.3 :007 > 2**3

=> 8

2.2.3 :008 > |

Variables y Nombres

```
1  class Ship
2  ; def initialize(package, settings=nil)
3  ; | company_id = settings.nil? ? 1 : settings['company_id']
4  ; | @package = package
5  ; | @settings = settings
6  ; | @couriers = couriers(company_id)
7  ; end
8
9  ; def cost
10 ; | courier = CourierPrice.get_range(@package, @settings, @couriers)
11 ; | delivery_factor = self.speed_delivery
12 ; | courier_cost = courier[:total] - self.discount(courier[:courier], courier[:total]).to_i
13 ; | total = self.is_payable(self.with_iva(courier_cost)) * delivery_factor
14 ; | {total: total.ceil, courier: courier[:courier], delivery_time: courier[:delivery_time]}
15 ; end
16
17 ; def price
18 ; | courier = CourierPrice.get_range(@package, @settings, @couriers)
19 ; | delivery_factor = self.speed_delivery
20 ; | total = self.is_payable(self.with_iva(courier[:total])) * delivery_factor
21 ; | { total: total.ceil, courier: courier[:courier], delivery_time: courier[:delivery_time]}
22 ; end
23 end
24
```

- Para el **nombre de las variables** es altamente recomendado **usar upper_case**
-
- Nombres de **variables con nombres descriptivos**

Cadenas y Textos

```
2.2.3 :005 > "Hola Mundo".downcase  
=> "hola mundo"  
2.2.3 :006 > "Hola Mundo".upcase  
=> "HOLA MUNDO"  
2.2.3 :007 > "hola mundo".capitalize  
=> "Hola mundo"  
2.2.3 :008 > "asd" + "asd"  
=> "asdasd"  
2.2.3 :009 > "asd".reverse  
=> "dsa"  
2.2.3 :010 >
```



“Funciones más utilizadas”

Más información:

<https://ruby-doc.org/core-2.3.0/String.html>

Declaraciones (If, Else)

- **Buena práctica:** Cuando el bloque de código que ejecuta el if es equivalente a una linea es **mucho mejor ejecutar todo en una linea.**
-
- ':' es el equivalente a 'else' en el segundo ejemplo
-
- '?' es equivalente a 'then'

```
+ 15 valor = 1
+ 16
+ 17 if valor > 1
+ 18 | mensaje = "#{valor} es mayor a 1"
+ 19 else
+ 20 | mensaje = "#{valor} es menor igual a 1"
+ 21 end
+ 22 puts mensaje
+ 23
+ 24 mensaje = valor > 1 ? "#{valor} es mayor a 1" : "#{valor} es menor igual a 1"
+ 25 puts mensaje
+ 26
+ 27 |
```

Declaraciones (Case)

```
| tracking =  
| | case package['courier_for_client'].capitalize  
| | when 'Chilexpress'  
| | | Chilexpress.create_tracking  
| | when 'Starken'  
| | | Starken.create_tracking  
| | when 'Correos'  
| | | Correos.create_tracking  
| | else  
| | | nil  
| | end  
|  
| puts ticket # => retorna un número de tracking: 9989728192
```

Buena práctica

```
| tracking =  
| | case package['courier_for_client'].capitalize  
| | when 'Chilexpress' then Chilexpress.create_tracking  
| | when 'Starken' then Starken.create_tracking  
| | when 'Correos' then Correos.create_tracking  
| | else nil  
| | end  
|  
| puts ticket # => retorna un número de tracking: 9989728192
```

Cuando el bloque de código es equivalente **a sólo acción**, es una buena práctica **ejecutarlo como aparece en el ejemplo de la derecha**

Bucles y Arreglos

```
| datos_sabado = [1, 2, 3, 4]
| datos_sabado.each do |dato|
| | puts "#{dato + 1}"
| end

| # Imprimirá: 2, 3, 4, 5|
|
```



“Esto es lo que más se repite en una aplicación Rails”

Bucles y Arreglos

```
! datos_sabado = [3, 2, 1, 4,0]
! datos_domingo = [2,3,4,5,5]

! #comparar
! datos_sabado.compare(datos_domingo) #retorna falso

! #Ordenar
! datos_sabado.sort # retorna [0, 1,2,3,4]

! #Ordenar de mayor a menor
! datos_sabado.sort.reverse #retorna [4,3,2,1,0]

! #Concatenar dos arrays
! datos_sabado.concat(datos_domingo) # retorna 3,2,1,4,0,2,3,4,5,5

! #Eliminar un numero del array
! datos_sabado.drop(2) # retorna 3,1,4,0

! #Eliminar Repetidos
! datos_domingo.uniq #retorna 2,3,4,5
!
! #Sumar información de dos arrays
! [datos_sabado, datos_domingo].transpose.map {|position| position.inject(:+)}
! #retorna [5,5,5,9,0]
!
```



Funciones más utilizadas en Array

Más información:
<https://ruby-doc.org/core-2.3.0/Array.html>

Hashes

```
| edades = {"Nelson": 28, "Javier": 25, "Daniela": 26}  
| puts edades["Nelson"] #retorna 28  
  
| edades["Nelson"] = 25  
| puts edades["Nelson"] #retorna 25  
  
| edades.class # retorna Hash
```



“Es como un diccionario de datos con valores y llaves únicos”

Preguntas ? 

github.com/patojimenez
linkedin.com/in/nelsonpatriciojimenez

