

INSTITUTO TECNOLÓGICO DE COSTA RICA

TAREA 4 - RUBY

Ariel Herrera

Saúl Zamora

profesor

M. Sc. Saúl Calderón Ramírez

September 24, 2016

I. DATOS HISTÓRICOS

Ruby es un lenguaje de propósito general, dinámico, reflectivo y orientado a objetos. Fué diseñado y desarrollado a mediados de los años 90 por Yukihiro "Matz" Matsumoto en Japón.

De acuerdo con su creador, Ruby fue influenciado por Perl, Smalltalk, Eiffel, Ada y LISP. Soporta la programación en múltiples paradigmas incluyendo funcional, orientación a objetos e imperativo. También tiene un sistema de tipos dinámicos y un manejo de memoria automático.

II. IMPORTANCIA Y USOS

Ruby es un lenguaje fácil de aprender, con fuertes abstracciones de los detalles computacionales; dado esto es un buen lenguaje a considerar para comenzar a aprender a programar.

Ruby también es la entrada a Ruby On Rails. RoR es un framework con mucha popularidad que usa y depende de Ruby, dicho framework es ampliamente utilizado en la creación de aplicaciones web.

III. TIPOS DE DATOS

Ruby posee un tipado de datos dinámico, sin embargo posee los siguientes tipos de datos nativos:

- String
- Fixnum
- Integer
- Numeric
- Float
- NilClass
- Hash
- Symbol
- Array
- Range

IV. EXPRESIONES

A. Operadores aritméticos

+, -, *, /, %, ** (potencia)

B. Operadores de comparación

==, !=, <, >, <=, >=, <=>, eql?, equal?

C. Operadores de asignación

=, +=, -=, *=, /=, %=, **=

D. Operadores lógicos

and, or, &&, !, not

V. ESTRUCTURAS DE CONTROL

Estatuto IF:

```
# Generate a random number and print whether it's even or odd.
if rand(100) % 2 == 0
  puts "It's even"
else
  puts "It's odd"
end
```

Bloques e iteradores:

```
{ puts 'Hello, World!' } # note the braces
# or:
do
  puts 'Hello, World!'
end

File.open('file.txt', 'w') do |file| # 'w' denotes "write mode"
  file.puts 'Wrote some text.'
end # file is automatically closed here

File.readlines('file.txt').each do |line|
  puts line
end
# => Wrote some text.
```

VI. CARACTERÍSTICAS PRINCIPALES

- Fuerte orientación a objetos con herencia, mixins y metaclases.
- Tipado de datos dinámico y *duck typing* (si se ve como un pato y suena como un pato, es un pato).
- Todo es una expresión (hasta los estatutos) y todo se ejecuta imperativamente (hasta las declaraciones).
- Reflexión y alteración dinámica de objetos para facilitar metaprogramación.
- Sintaxis única de bloques para iteradores y generadores.
- Notación literal para arreglos, hashes, expresiones regulares y símbolos.
- Interpolación de hileras.
- Argumentos default.

VII. CARACTERÍSTICAS DISTINTIVAS

- Recolector de basura.
- Cuatro niveles de alcance para variables: globales, de clase, de instancia y local.
- Sobrecarga de operadores.
- Soporte nativo para números racionales, complejos y aritmética de precisión arbitraria.
- Soporte inicial para Unicode y múltiples codificaciones de caracteres.
- Consola Ruby interactiva.
- Manejo centralizado de paquetes a través de *RubyGems*.
- Implementaciones en todas las plataformas más conocidas.

VIII. VENTAJAS Y DESVENTAJAS

A. Ventajas

- Herramientas: utilizado con el framework de Ruby On Rails, Ruby es bueno para entregar más *features* en menos tiempo. Provee de una estructura estándar para web apps,

donde todos los patrones estándar son manejados por el usuario.

- Librerías: existen gemas (librerías externas) para casi todo lo que se le pueda ocurrir al usuario.
- Calidad de código: generalmente, la calidad de código Ruby es mejor que la de sus contrapartes en PHP o NodeJS.
- Pruebas automatizadas: la comunidad Ruby soporta mucho las pruebas, y aún más, las pruebas automatizadas. Lo cual ayuda a entregar software de buena calidad.
- Gran comunidad: en casi todas las grandes ciudades del mundo hay una comunidad Ruby que dirige reuniones regulares. Ruby es uno de los lenguajes más populares en Github.
- Es popular en el valle: la historia ha demostrado que las tecnologías que son populares en Silicon Valley son gradualmente aceptadas en todo el mundo. Muchas de las grandes startups de los años recientes como AirBnB, Etsy, Github y Shopify, están construidas con Ruby.
- Productividad: Ruby es un lenguaje elocuente, lo que combinado con la gran variedad de librerías externas, habilita al programador al desarrollo rápido.

B. Desventajas

- Velocidad de ejecución: el mayor argumento contra Ruby es que es lento. Sin embargo, estos problemas de velocidad no se notarán hasta que la aplicación tenga un gran volumen de tráfico.
- Velocidad de arranque: aplicable en el caso del framework, Ruby on Rails, dependiendo del número de dependencias (*gemas*), puede tomar una cantidad de tiempo significativa a una aplicación para arrancar, lo que resiente el tiempo de desarrollo.
- Documentación: puede ser difícil encontrar buena documentación, especialmente para las gemas y librerías que hacen uso excesivo de mixins.
- Múltiples hilos: Ruby on Rails soporta múltiples hilos, aunque algunas de las librerías de entrada y salida no. Esto significa que si no se es cuidadoso, los requests se apilarán detrás del request actual lo cual puede introducir problemas de performance.
- ActiveRecord: es una librería fuertemente utilizada en Ruby on Rails. Y si bien está bien diseñada, el más grande problema es que el dominio se amarra muy fuerte a los mecanismos de persistencia.

IX. EJEMPLO

A. Strings

```
a = "\nThis_is_a_double-quoted_string\n"
a = %Q{\nThis_is_a_double-quoted_string\n}
a = %{\nThis is a double-quoted string\n}
a = %/\nThis is a double-quoted string\n/
a = <<-BLOCK
```

```
This is a double-quoted string
BLOCK
```

```
var = 3.14159 # interpolacion de variables
"pi_is_#{var}"
=> "pi_is_3.14159"
```

B. Colecciones

```
a = [1, 'hi', 3.14, 1, 2, [4, 5]]

a[2]           # => 3.14
a.[2] (2)      # => 3.14
a.reverse      # => [[4, 5], 2, 1, 3.14, 'hi', 1]
a.flatten.uniq # => [1, 'hi', 3.14, 2, 4, 5]
```

X. REFERENCIAS

REFERENCES

- [1] Ruby (programming language) (2016). . In *Wikipedia*. Retrieved from [https://en.wikipedia.org/wiki/Ruby_\(programming_language\)](https://en.wikipedia.org/wiki/Ruby_(programming_language))
- [2] MacDonald, R., & Founder. (2015, September 8). Pros and cons of Ruby on rails. Retrieved September 24, 2016, from <https://www.madetech.com/blog/pros-and-cons-of-ruby-on-rails>
- [3] Ruby programming/data types - Wikibooks, open books for an open world. Retrieved September 24, 2016, from https://en.wikibooks.org/wiki/Ruby_Programming/Data_types
- [4] Ruby Operators. (2016). Retrieved September 24, 2016, from https://www.tutorialspoint.com/ruby/ruby_operators.htm
- [5] Retrieved September 25, 2016, from <http://www.skilledup.com/articles/4-reasons-learn-ruby-first-programming-language>