



**ESCUELA SUPERIOR
POLITÉCNICA DEL LITORAL**

Tema: Investigación de Lenguajes

Materia: Lenguajes de Programación

Nombre del grupo en Sidweb: Lenguaje Ruby

Paralelo: 1

Fecha de entrega:
Miércoles, 23 de Octubre del 2013

Lenguaje Ruby

Autores:

- Cristina Barreno
- Sixto Castro
- Jordy Vasquez

20 de octubre de 2013

Tabla de Contenidos

1. Introducción
2. Características
3. Historia
4. Tutorial de instalación
5. Hola Mundo y otros programas introductorios
6. Referencias bibliográficas

1. Introducción

Hara Jordy yo jajajja

2. Características

- Ruby es un lenguaje multiplataforma (altamente portable).
- Es fácil de escribir.
- Es un lenguaje interpretado.
- Es un lenguaje orientado a objetos. Al igual que Java, cada tipo de dato es un objeto.
- Es una mezcla de varios lenguajes tales como Perl, Python, Smalltalk y otros, que definen a Ruby como un lenguaje que integra la programación funcional e imperativa.
- Posee expresiones similares(nivel de lenguaje) a las de Perl
- En Ruby, los métodos se pueden o no ser parte de una clase. Puede ser declarado en cualquier parte del archivo.
- Es capaz de manejar excepciones.
- Se puede manejar hilos (multihilos) sin depender del sistema operativo.

3. Historia

Hace Jordy

4. Tutorial de Instalación

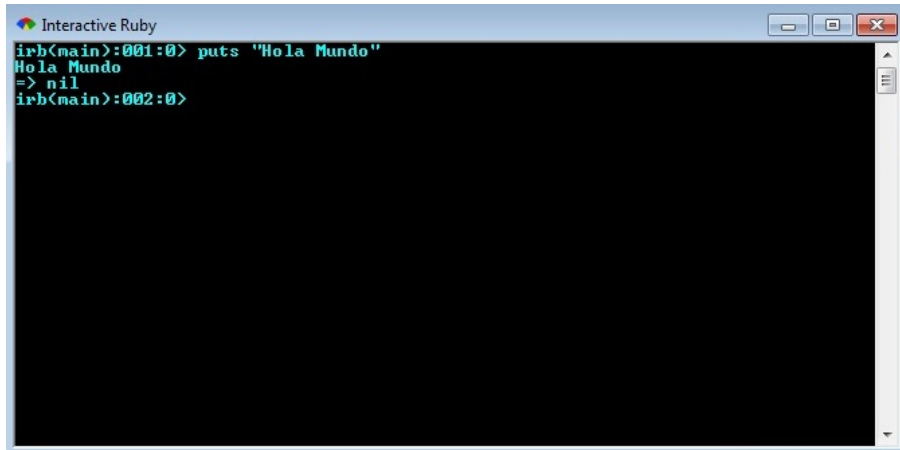
Hace Cristina

5. Hola Mundo y otros Programas Introductorios

A continuación revisaremos algunos programas básicos en ruby para esto solo necesitamos tener instalado en nuestro computador el interprete de ruby, abramos el terminal *InteractiveRuby* y empecemos a programar. Aunque solo trabajaremos desde consola recordemos que esta es una herramienta poderosa con la cual daremos nuestros primeros pasos en el mundo de ruby.

Ejemplo 1: Hola Mundo

Para hacer que nuestro programa escriba "*HolaMundo*" usaremos el comando *puts* "*HolaMundo*", siendo "*puts*" el comando básico para escribir algo en ruby.



```
irb(main):001:0> puts "Hola Mundo"
Hola Mundo
=> nil
irb(main):002:0>
```

Ejemplo 2: Operaciones básicas

aaa

Ejemplo 3: Trabajando con arreglos

aa

Ejemplo 4: Trabajando con Strings

- **Crear un array**

```
@names = Array.new
```

- **Método []**

```
@names = ["ivan", "luis", "pedro"]
```

- **Método: +**

Agregar contenido de un Array al final de otro Array.

```
@clientes = ["Juan", "Luis", "Roberto"]
```

```
@clientes2 = [ "Pierre", "Joe", "Rachel"]
```

```
@clientes + @clientes2
```

Resultado: ["Juan", "Luis", "Roberto", "Pierre", "Joe", "Rachel"]

- **Ordenar un arreglo**

```
notasdeberes = [10,7,5,8]
```

```
notasdeberes.sort
```

- **Obtener el tamaño de un arreglo**

```
notasdeberes.size()
```

- **Recorrer un arreglo: each**

```
equipos = ["Barcelona", "Milan", "PSG", "Monaco"]
equipos.each do equipo
  puts equipo
end
```

El método 'each' lo que va hacer es recorrer cada elemento del arreglo equipos. En cada iteración, cada elemento del array se guarda en la variable equipo y se va imprimir cada elemento(cadena) del arreglo equipos ya que se encuentra la función puts que se encarga de imprimir cada cadena.

- **Recorrer el arreglo: for in**

```
for i in equipos
  puts i
end
```

Este caso es parecido al ejercicio anterior, sino que ahora se utiliza un ciclo 'for' para recorrer el arreglo. En cada iteración, la variable i toma el valor actual del array equipos, y luego va imprimir dicho elemento ya que se hace un put en cada iteración.

6. Referencias

- <https://www.ruby-lang.org/es/about/>
- <http://es.wikipedia.org/wiki/Ruby#Caracter.C3.ADsticas>
- http://es.wikibooks.org/wiki/Programaci3n_en_Ruby#Caracter.C3.ADsticas_Especiales_del_Lenguaje
- http://www.ecured.cu/index.php/Lenguaje_de_Programaci3n_Ruby#Caracter.C3.ADsticas_generales_del_lenguaje