

Laboratorio de Computación IV


Clase 2

Andrés Fortier

¿Consultas?

- Organización de la materia y entregas.
- Comandos: `cd` y `ls`.

Ruby + rvm



- Ruby (MRI - Matz's Ruby Interpreter)
 - Lenguaje interpretado (RVM).
 - Dinámicamente tipado.
 - Manejo de memoria automático.
 - Soporta POO.
- Ruby Version Manager (rvm)
 - Permite manejar
 - Múltiples versiones de ruby (interpretes).
 - Múltiples gemsets.

rvm (<https://rvm.io>)

- Instalar rvm (<https://rvm.io/rvm/install>)

```
$ gpg --keyserver hkp://keys.gnupg.net --recv-keys  
409B6B1796C275462A1703113804BB82D39DC0E3
```

```
$ \curl -sSL https://get.rvm.io | bash -s stable
```

- Si hay problemas con keyserver
 - <https://rvm.io/rvm/security>

```
curl -sSL https://rvm.io/mpapis.asc | gpg --import -
```

rvm

- Ver los intérpretes instalados

```
$ rvm list
```

- Ver los intérpretes disponibles

```
$ rvm list known
# MRI Rubies
[ruby-]1.8.6[-p420]
[ruby-]1.8.7[-head] # security released on head
[ruby-]1.9.1[-p431]
[ruby-]1.9.2[-p330]
[ruby-]1.9.3[-p551]
[ruby-]2.0.0[-p643]
[ruby-]2.1.4
[ruby-]2.1[.5]
[ruby-]2.2[.1]
[ruby-]2.2-head
ruby-head
...
```

rvm

- Instalemos 2.0.0-p598

```
$ rvm install 2.0.0-p598
```

```
$ rvm list
```

```
rvm rubies
```

```
=* ruby-2.0.0-p598 [ x86_64 ]
```

```
# => - current
```

```
# *= - current && default
```

```
# * - default
```

```
$ rvm list default
```

```
Default Ruby (for new shells)
```

```
ruby-2.0.0-p598 [ x86_64 ]
```

rvm

- Instalemos 2.2.0

```
$ rvm install 2.2.0
```

```
$ rvm list
```

```
rvm rubies
```

```
  * ruby-2.0.0-p598 [ x86_64 ]  
=> ruby-2.2.0 [ x86_64 ]
```

```
# => - current
```

```
# *= - current && default
```

```
# * - default
```

rvm

- Verifiquemos la versión

```
$ ruby -v  
ruby 2.2.0p0 (2014-12-25 revision 49005) [x86_64-linux]
```

- Cambiando la versión de la sesión

```
$ rvm use ruby-2.0.0-p598  
Using /home/andres/.rvm/gems/ruby-2.0.0-p598
```

```
$ ruby -v  
ruby 2.0.0p598 (2014-11-13 revision 48408) [x86_64-  
linux]
```


rvm

- *Versión default*

```
$ rvm --default use ruby-2.0.0-p598  
Using /home/andres/.rvm/gems/ruby-2.0.0-p598
```

```
$ rvm list default
```

```
Default Ruby (for new shells)
```

```
ruby-2.0.0-p598 [ x86_64 ]
```

Ruby

- Irb == Interactive Ruby

```
$ irb  
2.0.0-p598 :001 >
```

```
2.0.0-p598 :001 > 5+5  
=> 10  
2.0.0-p598 :002 >
```

- Variables locales

```
2.0.0-p598 :002 > valor = 10  
=> 10  
2.0.0-p598 :003 > valor * 3  
=> 30  
2.0.0-p598 :004 > valor  
=> 10
```

Ruby

- Definiendo comportamiento (funciones?)

```
2.0.0-p598 :005 > def hola()  
2.0.0-p598 :006?>   return "Hola"  
2.0.0-p598 :007?>   end  
=> nil
```

```
2.0.0-p598 :008 > hola()  
=> "Hola"
```

- Investiguemos un poco mas en detalle

```
2.0.0-p598 :010 > self  
=> main
```

```
2.0.0-p598 :011 > self.class()  
=> Object
```

Ruby

```
2.0.0-p598 :012 > self.hola()  
NoMethodError: private method `hola' called for  
main:Object  
  from (irb):12  
  from /home/andres/.rvm/rubies/ruby-2.0.0-  
p598/bin/irb:12:in `<main>'
```

- Probemos crear un objeto nuevo

```
2.0.0-p598 :013 > miObjeto = Object.new()  
=> #<Object:0x00000002575e98>
```

```
2.0.0-p598 :014 > def miObjeto.nombre()  
2.0.0-p598 :015?>   return "Pepe"  
2.0.0-p598 :016?>   end  
=> nil
```

```
2.0.0-p598 :017 > miObjeto.nombre()  
=> "Pepe"
```

Ruby

- Verifiquemos que ese comportamiento es sólo de ese objeto

```
2.0.0-p598 :019 > nombre()  
NoMethodError: undefined method `nombre' for  
main:Object  
  from (irb):19  
  from /home/andres/.rvm/rubies/ruby-2.0.0-  
p598/bin/irb:12:in `<main>'
```

```
2.0.0-p598 :020 > Object.new().nombre()  
NoMethodError: undefined method `nombre' for  
#<Object:0x000000024bc2b8>  
  from (irb):20  
  from /home/andres/.rvm/rubies/ruby-2.0.0-  
p598/bin/irb:12:in `<main>'
```

Ruby

- Definamos una clase

```
2.0.0-p598 :008 > class Contador
2.0.0-p598 :009?>   def initialize()
2.0.0-p598 :010?>     @valor = 0
2.0.0-p598 :011?>   end
2.0.0-p598 :012?>
2.0.0-p598 :013 >   def siguiente()
2.0.0-p598 :014?>     @valor = @valor + 1
2.0.0-p598 :015 >   end
2.0.0-p598 :016?> end
=> nil
```

```
2.0.0-p598 :017 > c = Contador.new()
=> #<Contador:0x00000000ba51d0 @valor=0>
```

Ruby

```
2.0.0-p598 :018 > c.siguiente()  
=> 1
```

```
2.0.0-p598 :019 > c.siguiente()  
=> 2
```

```
2.0.0-p598 :020 > c  
=> #<Contador:0x00000000ba51d0 @valor=2>
```

- Probemos lo siguiente

```
2.0.0-p598 :021 > c.siguiente  
=> 3
```

- Salgamos de la sesión (`exit`)

Ruby

- Creemos un archivo `contador.rb`

```
class Contador
  def initialize
    limpiar
  end

  def limpiar
    @valor = 0
  end

  def siguiente
    @valor = @valor + 1
  end
end
```

- *Nota: lo vamos a usar la clase que viene*

Ruby

- `self` implícito

```
def initialize  
  limpiar  
end
```

- Definición de variables de instancia al usarlas

```
def limpiar  
  @valor = 0  
end
```

- `return` implícito

```
def siguiente  
  @valor = @valor + 1  
end
```

Ruby

- Probemos lo siguiente

```
$ irb -r ./contador.rb
2.0.0-p598 :001 > c = Contador.new
=> #<Contador:0x00000001cfff610 @valor=0>
2.0.0-p598 :002 > c.siguiente
=> 1
2.0.0-p598 :003 > c.siguiente
=> 2
2.0.0-p598 :004 > c.limpiar
=> 0
2.0.0-p598 :005 > c.siguiente
=> 1
```

Ruby

- Modificaciones

```
class Contador
  ...
  def vacio?
    return @valor == 0
  end
end
```

```
$ irb -r ./contador.rb
2.0.0-p598 :001 > c = Contador.new
=> #<Contador:0x0000000297e5a8 @valor=0>
2.0.0-p598 :002 > c.vacio?
=> true
2.0.0-p598 :003 > c.siguiente
=> 1
2.0.0-p598 :004 > c.vacio?
=> false
```

Ruby

- Getters y setters

```
class Contador
  ...
  def valor
    return @valor
  end

  def valor=(nuevoValor)
    @valor = nuevoValor
  end
end
```

Ruby

- Getters y setters

```
2.0.0-p598 :001 > c = Contador.new
=> #<Contador:0x00000001dfa510 @valor=0>
2.0.0-p598 :002 > c.valor
=> 0
2.0.0-p598 :003 > c.valor = 10
=> 10
2.0.0-p598 :004 > c.valor
=> 10
2.0.0-p598 :005 > c.siguiente
=> 11
2.0.0-p598 :006 > c.valor
=> 11
```

Ruby

- Borrar los getters y setters
- Agregar

```
class Contador  
  attr_accessor :valor  
  ...  
end
```

Ruby

- Volver a probar

```
2.0.0-p598 :001 > c = Contador.new
=> #<Contador:0x00000001dfa510 @valor=0>
2.0.0-p598 :002 > c.valor
=> 0
2.0.0-p598 :003 > c.valor = 10
=> 10
2.0.0-p598 :004 > c.valor
=> 10
2.0.0-p598 :005 > c.siguiente
=> 11
2.0.0-p598 :006 > c.valor
=> 11
```

Ruby



- Getters y setters
 - Sólo getter: `attr_reader :valor`
 - Sólo setter: `attr_writer :valor`
 - Ambos: `attr_accessor :valor`

Ruby

- Un poco mas del initialize

```
class Contador

  attr_accessor :valor

  def initialize(valorInicial)
    @valor = valorInicial
  end

  ...
end
```

```
$ irb -r ./contador.rb
2.0.0-p598 :001 > c = Contador.new(23)
=> #<Contador:0x00000002419c20 @valor=23>
```

Ruby

- Un poco mas del initialize

```
2.0.0-p598 :002 > c = Contador.new(23, "Segundo")
ArgumentError: wrong number of arguments (2 for 1)
  from /home/andres/Docencia/UTN/2015/Laboratorio
4/Clase03/contador.rb:5:in `initialize'
  from (irb):6:in `new'
  from (irb):6
  from /home/andres/.rvm/rubies/ruby-2.0.0-
p598/bin/irb:12:in `<main>'
```

Ruby

- Valores *default*

```
class Contador

  attr_accessor :valor

  def initialize(valorInicial = 0)
    @valor = valorInicial
  end

  ...
end
```

```
$ irb -r ./contador.rb
2.0.0-p598 :001 > Contador.new
=> #<Contador:0x00000002947058 @valor=0>
2.0.0-p598 :002 > Contador.new(10)
=> #<Contador:0x0000000293af38 @valor=10>
```

Ruby

- Subclasses

```
class ContadorLoco < Contador

  def siguiente
    super
    if @valor == 3
      @valor = 0
    end
    return @valor
  end

end
```

Ruby

- Subclases

```
$ irb -r ./contador.rb
2.0.0-p598 :001 > c = ContadorLoco.new
=> #<ContadorLoco:0x000000017cd2c8 @valor=0>
2.0.0-p598 :002 > c.siguiente
=> 1
2.0.0-p598 :003 > c.siguiente
=> 2
2.0.0-p598 :004 > c.siguiente
=> 0
```

Ruby

- Probemos un cambio

```
class ContadorLoco < Contador

  def siguiente
    super
    if valor == 3
      valor = 0
    end
    return valor
  end

end
```

Ruby

- ?!

```
$ irb -r ./contador.rb
2.0.0-p598 :001 > c = ContadorLoco.new
=> #<ContadorLoco:0x000000024e5190 @valor=0>
2.0.0-p598 :002 > c.siguiente
=> nil
2.0.0-p598 :003 > c.siguiente
=> nil
2.0.0-p598 :004 > c.siguiente
=> 0
2.0.0-p598 :005 > c.siguiente
=> nil
2.0.0-p598 :006 > c.valor
=> 4
```

Ruby

- Mensajes vs definición automática de variables temporales

```
class ContadorLoco < Contador

  def siguiente
    super
    if valor == 3
      self.valor = 0
    end
    return self.valor
  end

end
```


Ruby

- Probemos nuevamente

```
$ irb -r ./contador.rb
2.0.0-p598 :001 > c = ContadorLoco.new
=> #<ContadorLoco:0x00000001ed93a0 @valor=0>
2.0.0-p598 :002 > c.siguiente
=> 1
2.0.0-p598 :003 > c.siguiente
=> 2
2.0.0-p598 :004 > c.siguiente
=> 0
```

Tarea para el hogar



- Investigar cómo cargar un archivo en la consola irb una vez que ésta se encuentre abierta
 - <http://stackoverflow.com/questions/13112245/how-do-you-load-a-file-into-interactive-ruby-console>
 - <http://stackoverflow.com/questions/6149850/ruby-on-rails-running-a-rb-file-from-irb>

Tarea para el hogar

- Realizar una jerarquía de clases que resuelva los ítems 1 y 6 de la Práctica 2 de Programación 3. Probar su funcionamiento en la consola.
- Realizar el ejercicio 7 de la Práctica 2 de Programación 3. Probar su funcionamiento en la consola.

Links



- <https://www.ruby-lang.org/es/documentation/quickstart/>
- <http://www.codecademy.com/en/tracks/ruby>
- <https://rubymonk.com/>