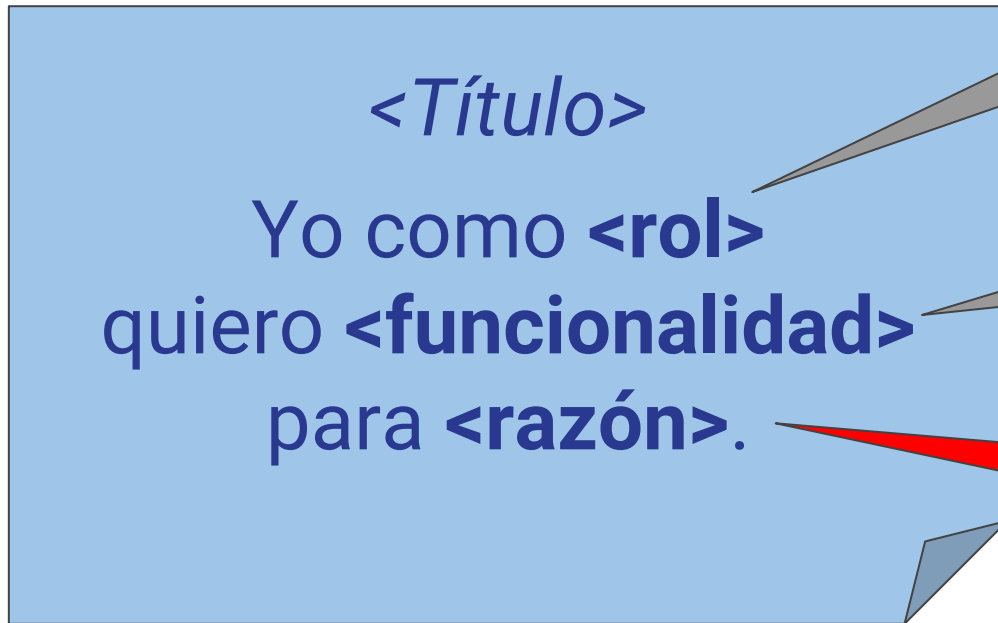


Relatos de usuario, Ruby y Rubocop

IIC2143 Ingeniería de Software - Ayudantía 2

User Stories



¿Quién?

¿Qué?

¿Para qué?

User Stories

Condiciones de Satisfacción

- Test en lenguaje natural
- Ejecutable por el usuario

Características generales

La aplicación a desarrollar debe permitir a los usuarios acceder, gestionar canciones y explorar canciones, álbumes y artistas. Las canciones se encuentran relacionadas a algún álbum, el cual fue lanzado por algún artista o banda. Para esto, su aplicación debe permitir agregar canciones, álbumes y artistas, junto a la información asociada a cada uno. Además, se pueden crear listas temáticas, que agrupen un conjunto de canciones (por ejemplo: “90’s Grunge”, que contiene canciones como “Smells like teen spirit” de Nirvana y “Jeremy” de Pearl Jam), las cuales pueden ser seguidas por otros usuarios si es que esta se encuentra pública. En caso de ser privada, solo podrán verla los usuarios con los permisos adecuados.

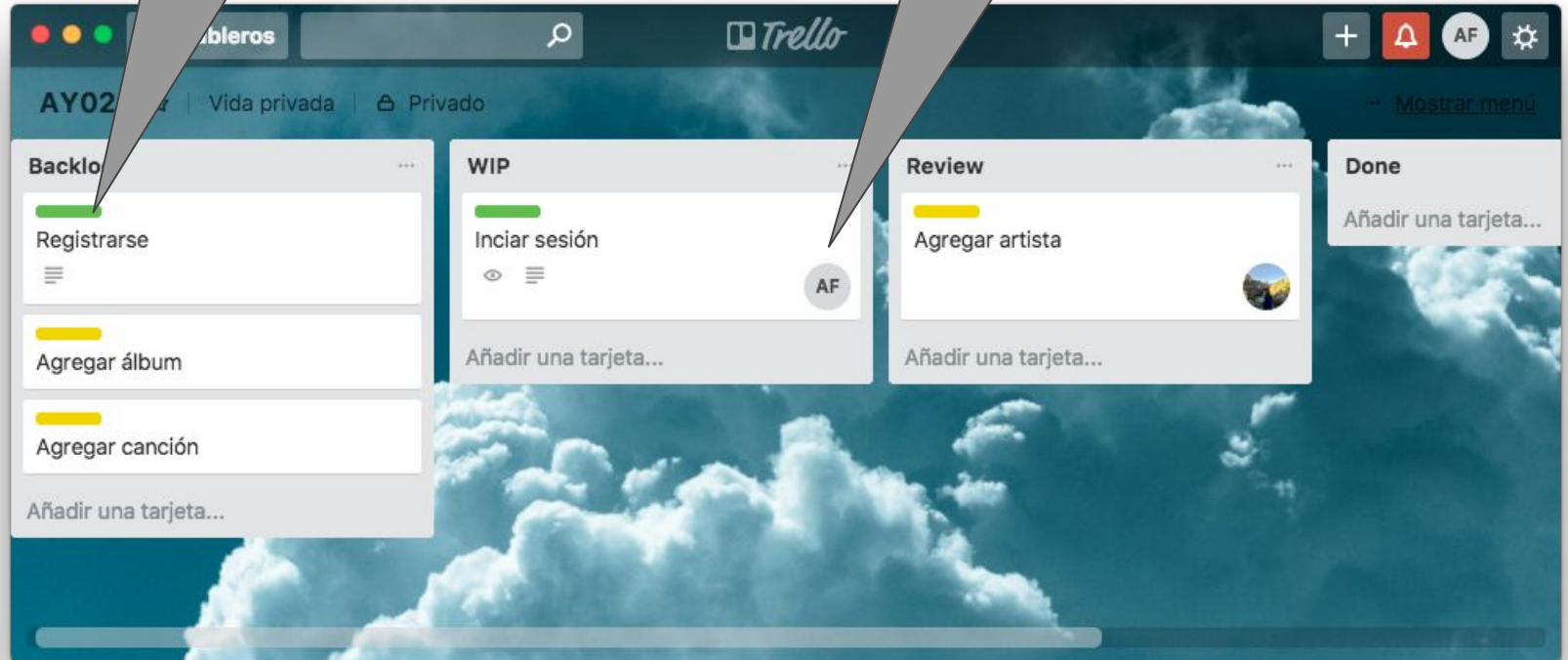
Enunciado 2017-2: <http://bit.ly/SoftwareAY02>



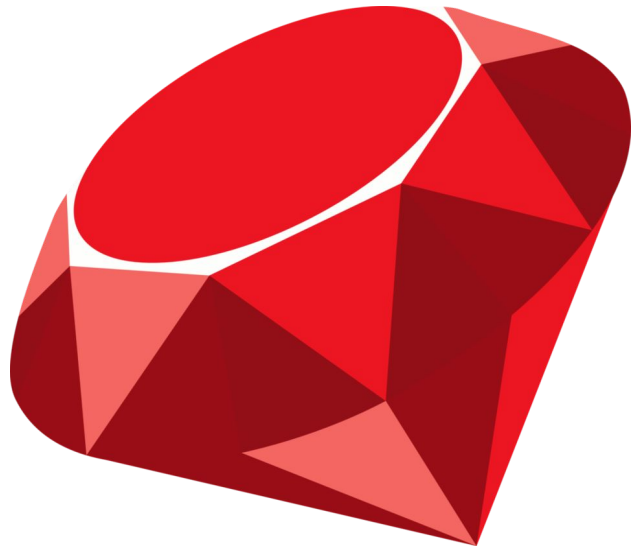
IMPORTANCIA/URGENCIA

Etiquetas de colores pueden indicar relatos (tarjetas/tickets) de una misma épica

Persona a cargo se marca en la tarjeta



SECUENCIA



Ruby

Uso básico

Para ejecutar un script

```
$ ruby <nombre_del_archivo>.rb
```

Para iniciar la consola de Ruby

```
$ irb
```

```
irb(main):001:0>
```



Ejemplo básico

```
# human.rb
1  class Human
2    attr_accessor :first_name, :last_name
3
4    @@all = []
5
6    def initialize(first_name, last_name)
7      @first_name = first_name
8      @last_name = last_name
9      @@all << self
10   end
11
12   def name
13     "#{@first_name} #{@last_name}"
14   end
15
16   def self.all
17     @@all
18   end
19 end
```

Ejemplo básico

Metaprogramación: agrega métodos para leer y actualizar esos atributos

Los paréntesis son opcionales para ejecutar un método/función

```
# human.rb
1 class Human
2   attr_accessor(:first_name, :last_name)
3
4   @@all = []
5
6   def initialize(first_name, last_name)
7     @first_name = first_name
8     @last_name = last_name
9     @@all << self
10  end
11
12  def name()
13    return "#{@first_name} #{@last_name}"
14  end
15
16  def self.all()
17    return @@all
18  end
19 end
```

Los símbolos parten con `:`. Corresponden a objetos de texto inmutables y únicos en memoria para un mismo valor.

Los atributos de instancia son privados y empiezan con `@`

Interpolación de *strings*, para insertar variables.


Los métodos de clase se declaran con `self`.

Los atributos de clase empiezan con `@@`

Siempre se retorna lo evaluado en la última línea. En ese caso, el `return` es opcional.

Ejemplo básico

```
# main.rb
1  require './human'
2
3  Human.new 'Juana', 'Pérez'
4  Human.new 'José', 'González'
5
6  Human.all.each { |human| puts human.name }
7
8
9
10
11
12
13
14  Human.all.each_with_index do |human, idx|
15    list_item = "#{idx + 1}) #{human.name}"
16    puts list_item
17  end
```



Ejemplo básico

Para importar

Se instancia con el método new

each permite iterar sobre cada elemento de un arreglo. Necesita un bloque.

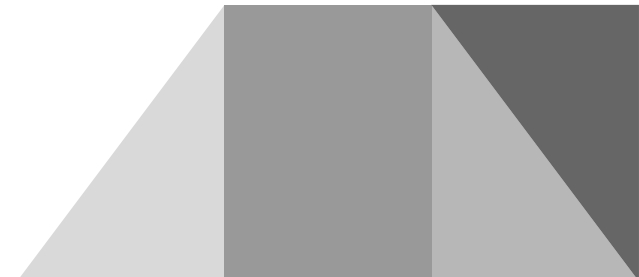
Un bloque es un procedimiento que se puede entregar a algunas funciones. Se pueden demarcar con { } o con do ... end

Los argumentos de un bloque se demarcan con | |

```
# main.rb
1  require './human'
2
3  Human.new('Juana', 'Pérez')
4  Human.new('José', 'González')
5
6  Human.all().each() { |human| puts(human.name()) }
7
8
9
10
11
12
13
14  Human.all().each_with_index() do |human, idx|
15    list_item = "#{idx + 1}) #{human.name()}"
16    puts(list_item)
17  end
```

Tutoriales recomendados

- Try Ruby
 - tryruby.org
- Learn X in Y minutes, where X = Ruby
 - <https://learnxinyminutes.com/docs/ruby/>





RuboCop

<https://rubocop.readthedocs.io/en/latest/>

Rubocop

- Es una gema (librería/paquete) de Ruby
- Es un *linter*: programa que revisa el código
- Se puede integrar con plugins a tu editor de texto
- Se puede ejecutar en la terminal con

```
$ rubocop <archivo_a_revisar>.rb
```

- Tiene reglas (cops) para el estilo del código
 - <https://rubocop.readthedocs.io/en/latest/>
- Se puede personalizar su configuración para un proyecto específico con el archivo `.rubocop.yml`

```
# .rubocop.yml
Layout/DotPosition:
  Enabled: true
  EnforcedStyle: leading
```