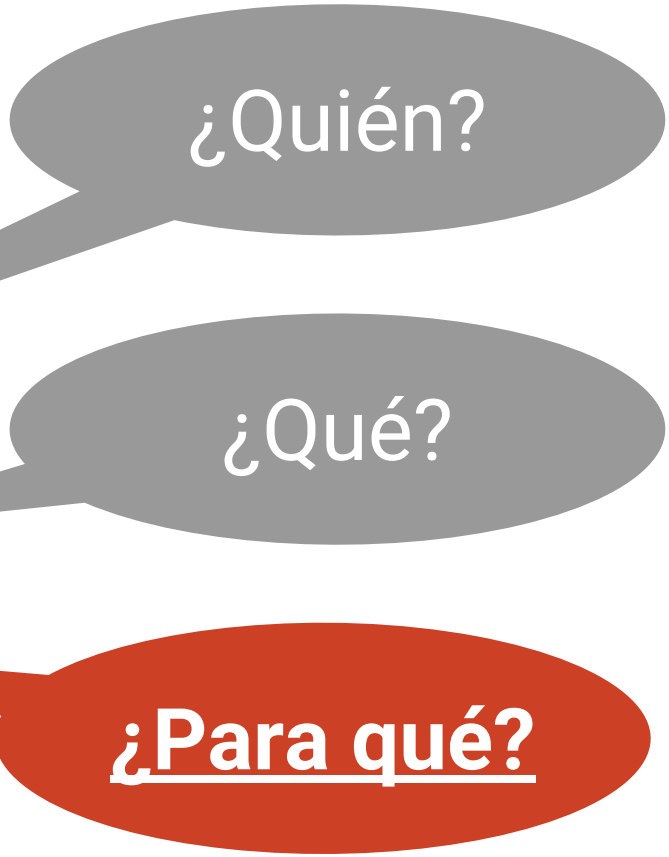
Relatos de usuario, Ruby, Rubocop y Github

IIC2143 Ingeniería de Software - Ayudantía 2

User Stories

<Título>

Yo como **<rol>**
quiero **<funcionalidad>**
para **<razón>**.



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

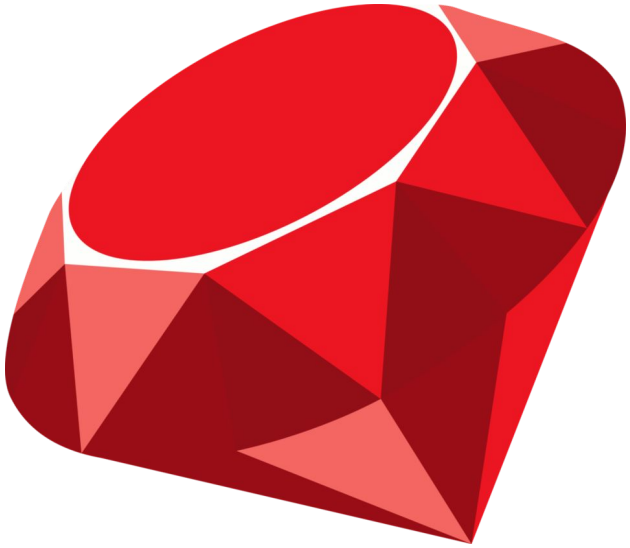
https://trello.com/b/PVUP6Mlu/ayudantia

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

Persona a cargo se marca en la tarjeta

IMPORTANCIA/URGENCIA

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

```ruby
# human.rb
class Human
  attr_accessor :first_name, :last_name

  @@all = []

  def initialize(first_name, last_name)
    @first_name = first_name
    @last_name = last_name
    @@all << self
  end

  def name
    "#{@first_name} #{@last_name}"
  end

  def self.all
    @@all
  end
end
```

Ejemplo básico

```ruby
# human.rb
class Human
  attr_accessor(:first_name, :last_name)

  @@all = []

  def initialize(first_name, last_name)
    @first_name = first_name
    @last_name = last_name
    @@all << self
  end

  def name()
    return "#{@first_name} #{@last_name}"
  end

  def self.all()
    return @@all
  end
end
```

Metaprogramación: agrega métodos para leer y actualizar esos atributos

Los paréntesis son opcionales para ejecutar un método/función

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

```ruby
# main.rb
require './human'

Human.new 'Juana', 'Pérez'
Human.new 'José', 'González'

Human.all.each { |human| puts human.name }









Human.all.each_with_index do |human, idx|
 list_item = "#{idx + 1}) #{human.name}"
 puts list_item
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 - Trial 10 dias
  - tryruby.org
- Learn X in Y minutes, where X = Ruby
  - https://learnxinyminutes.com/docs/ruby/

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
- Para ejecutar en el proyecto entrar al directorio

  `$ rubocop`
- 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`

GitHub

GitFlow