

Jeremias Parladorio, Juan Ignacio Alanis

Sprint 2



Ruby


```
app.rb:12:1: C: Layout/TrailingWhitespace: Trailing whitespace detected.
```

```
app.rb:269:121: C: Layout/LineLength: Line is too long. [153/120]
  if (User.find(username: params[:username]) && User.find(username: params[:username]).id != @current_user.id) || /\w{3,15}\z/ =~ params[:username]
  ~~~~~
```

```
app.rb:313:5: C: Style/IfUnlessModifier: Favor modifier if usage when having a single-line body. Another good alternative is the usage of control flow &&||.
  if User.find(email: params[:email]) || /\A.*@\.\.*\z/ =~ params[:email]
  ~~~~~
```

```
app.rb:339:3: C: Metrics/MethodLength: Method has too many lines. [11/10]
  def array_to_tag (users) ...
  ~~~~~
```

```
app.rb:11:1: C: Metrics/ClassLength: Class has too many lines. [535/100]
class App < Sinatra::Base ...
~~~~~
```

Corregir code smells

El primer paso para lograr el 3er tarea fue hacer uso de rubocop, ya que permite corregir ciertos errores automáticamente. Para esto corrimos:

```
Jere@Jeres-MacBook notifications % rubocop -A
```

Esto nos detecto 1089 ofensas, de las cuales pudo corregir automáticamente 1036. 10 no eran posibles corregir automáticamente.

El siguiente paso fue hacer las correcciones manuales: agregamos la documentación, arreglamos el posicionamiento de los include y cortamos las líneas que eran demasiado largas. El problema de tener clases y métodos con demasiadas líneas no fue posible solucionarlo debido a cómo esta estructurado nuestro proyecto.

El resultado de haber aplicado todas estas correcciones se puede ver en Docs/Sprint 2/Diff report.rtf. Recomendamos abrir este archivo con Sublime Text 2 para poder ver que es color-codee, lo que facilita ver los cambios hechos (tanto por nosotros mismos como por rubocop).

Otra parte importante de corregir errores fue el uso de branches durante el proceso: creamos dos branches extra. La primera, llamada 23-Rubocop-modifications fue creada para pusiera los cambios hechos automáticamente por rubocop. La segunda, llamada refactor, fue creada para pusiera los cambios manuales:

Active branches			
refactor	Updated 7 days ago by jereparla	5 0	#26 Merged 🗑
23-Rubocop-modifications	Updated 7 days ago by juanialanis	5 0	#24 Merged 🗑

Luego de tener todo listo, mergeamos ambas ramas con master.

También terminamos de realizar la integración con pivotal tracker, y movimos la story Refactor code que habíamos creado previamente a Done una vez terminada la tarea.

Referencias

1. <https://docs.rubocop.org/rubocop/index.html>
2. “Refactory Ruby Edition”, Fields, Harvie, Fowler, Beck, 2009.
3. <https://rubystyle.guide>