Jeremias Parladorio, Juan Ignacio Alanis

Sprint 2



Preliminares

En este sprint el scrum master fue Jeremías Parladorio. El diagrama de Gantt de este sprint se puede encontrar en Docs/Sprint 2. Todas las stories fueron realizadas en conjunto por los dos miembros de este equipo, por lo cual no hay individuos específicamente asignados a cada story.

Instalar y correr rubocop

Esta era la primera tarea que planeamos para nuestro sprint. Instalar rubocop fue fácil, lo hicimos con:

Jere@Jeres-MacBook notifications % gem install rubocop

Luego, corrimos

Jere@Jeres-MacBook notifications % rubocop

Esto nos genero un reporte de errores. Además, para mejor documentación, obtuvimos este reporte en formato txt con el siguiente comando:

Jere@Jeres-MacBook notifications % rubocop --out result.txt

Este reporte se puede encontrar en Docs/Sprint 2/result.txt

Detectar code smells

Esta era nuestra segunda tarea para el sprint 2. Lo que hicimos fue hacer un análisis del reporte de rubocop mencionado anteriormente. Este análisis nos permitió darnos cuenta que nuestros errores eran principalmente:

- · Teníamos muchos espacios en blancos
- Teníamos muchas estructuras de control escritas en un formato no deseado
- Teníamos includes en lugares incorrectos
- Teníamos muchas clases con más lineas de las deseadas
- Teníamos muchos métodos con mas lineas de las deseadas
- · Nos faltaba escribir documentación
- Teníamos lineas muy largas

Algunos ejemplos de esto en base al reporte de rubocop:

```
app.rb:8:1: C: Style/MixinUsage: include is used at the top level. Use inside class or module. include ActionView::Helpers::DateHelper
```

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

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) || /\A\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.*Q.*\..*\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 ...</pre>
```

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 lineas 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