

Taller de Ingenieria de Software

Actividad 3

D'Autilio Joel, Rojo Jonathan, Rossi Pablo

14 de octubre de 2022

Link del repositorio en GitHub:

<https://github.com/Jota98daut/Prode-AYDS2022-DRR>

Link del proyecto en PivotalTracker:

<https://www.pivotaltracker.com/n/projects/2590781>

El rol de Scrum Master lo toma Jonathan.

1. Autocorrector

Al ejecutar una primera inspección con Rubocop, resulta en 584 ofensas, de las cuales 494 son autocorregibles. Esto se ve en la figura 1.

Al ejecutar el comando **rubocop -A**, se detectan 774 ofensas de las cuales se autocorrigen 685. En este caso se detectaron más ofensas que antes, porque se incluyen criterios que no se consideraron en la primera inspección. La salida se ve en la figura 2.

En una segunda inspección se observa que ya no hay ofensas autocorregibles, pero aún quedan 89 ofensas que deben ser tratadas de forma manual. (figura 3)

2. Ofensas en el cop Metrics

Una inspección con el comando **rubocop --only Metrics** (para mostrar sólo las ofensas en el cop Metrics) encuentra 19 ofensas en 53 archivos inspeccionados.

```
spec/user_spec.rb:40:1: C: [Correctable] Layout/TrailingWhitespace: Trailing whitespace detected.
spec/user_spec.rb:42:9: C: [Correctable] Layout/IndentationWidth: Use 2 (not 4) spaces for indentation.
    it 'shouldn\'t be added' do
      ^^^
spec/user_spec.rb:43:13: C: [Correctable] Layout/IndentationWidth: Use 2 (not 4) spaces for indentation.
    u = User.new(username: '', password: '1234', password_confirmation: '1234')
      ^^^
spec/user_spec.rb:48:4: C: [Correctable] Layout/TrailingEmptyLines: Final newline missing.
end

53 files inspected, 584 offenses detected, 494 offenses autocorrectable
```

Figura 1: Resultado del comando rubocop

```

spec/user_spec.rb:43:7: C: [Corrected] Layout/IndentationWidth: Use 2 (not 4) spaces for indentation.
  it 'shouldn\'t be added' do
  ^^^^
spec/user_spec.rb:45:9: C: [Corrected] Layout/IndentationWidth: Use 2 (not 4) spaces for indentation.
  u = User.new(username: '', password: '1234', password_confirmation: '1234')
  ^^^^
spec/user_spec.rb:46:11: C: [Corrected] Layout/IndentationConsistency: Inconsistent indentation detected.
  expect(u.save).to eq(false)
  ^^^^^^^^^^^^^^^^^
spec/user_spec.rb:48:4: C: [Corrected] Layout/TrailingEmptyLines: Final newline missing.
end

53 files inspected, 774 offenses detected, 685 offenses corrected

```

Figura 2: Resultado de rubocop -A

```

spec/user_spec.rb:5:1: C: Metrics/BlockLength: Block has too many lines. [39/25]
describe 'User' do ...
^^^^^^^^^^^^^^^^^^^^
spec/user_spec.rb:6:3: C: Metrics/BlockLength: Block has too many lines. [37/25]
  describe 'when an user is submitted' do ...
  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
spec/user_spec.rb:23:9: W: Lint/UselessAssignment: Useless assignment to variable - u.
  u = User.create(username: 'jlennon', password: 'yesterday', password_confirmation: '
  ^
53 files inspected, 89 offenses detected

```

Figura 3: Inspección posterior a la autocorrección

Muchas de las siguientes métricas no poseen una descripción de la solución, porque estas fueron arregladas como efecto colateral de solucionar otra ofensas.

2.1. Metrics/AbcSize

Descripción: Cantidad de Asignaciones, ramificaciones y condicionales en un método.

5 ofensas detectadas en tres archivos helper: match_helper, session_helper y bet_helper. En particular, en pocos métodos pertenecientes a esos archivos.

Las ofensas se solucionaron al re-pensar los métodos que las provocaban. En general, estaban pobremente pensados desde un principio y daban lugar para optimización. En algunos casos, debimos abstraer funcionalidades de los métodos escribiendo otros, distribuyendo la lógica.

Muchas de las ofensas sobre las demás métricas fueron solucionadas al resolver las ofensas sobre AbcSize, ya que pensamos mejor los métodos.

2.2. Metrics/BlockLength

Descripción: Comprueba si la longitud de un bloque excede algún valor máximo (25 por default). Las líneas de comentarios se pueden ignorar opcionalmente. La longitud máxima permitida es configurable.

7 ofensas.

Todas las ofensas sobre esta métrica fueron en los archivos rspec. En reiteradas ocasiones

utilizabamos un mismo ‘describe’ para abarcar muchos posibles escenarios. Se solucionaron estas ofensas a base de dividir en sub bloques de escenarios con pre-condiciones similares. Una de las 7 ofensas, se da en el archivo schema.rb que se genera automáticamente al crear la base de datos. Es por esto que decidimos ignorar esta última ofensa.

Al tirar nuevamente **rubocop –only Metrics/BlockLength**:

```
.....C.....
Offenses:

db/schema.rb:13:1: C: Metrics/BlockLength: Block has too many lines.
[88/25]
ActiveRecord::Schema[7.0].define(version: 2022_09_20_130054) do ...
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

53 files inspected, 1 offense detected
```

2.3. Metrics/BlockNesting

Descripción: Cantidad de anidamientos en ciclos y condicionales.
Sin ofensas.

2.4. Metrics/ClassLength

Descripción: Número de líneas de código en una clase.
1 ofensa.

La ofensa que marca es en la class app. Como ya modularizamos el server en una iteración anterior decidimos omitir esta ofensa pues no se puede achicar más el server.

```
.....C.....
Offenses:

server.rb:13:1: C: Metrics/ClassLength: Class has too many lines.
[152/100]
class App < Sinatra::Application ...
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

53 files inspected, 1 offense detected
```

Por último buscamos como ignorar archivos al ejecutar rubocop para evitar que salten ofensas en módulos donde no se pueden restar líneas de código (server.rb y schema.rb).

Al tirar **rubocop –only Metrics**

```
Inspecting 53 files
.....

53 files inspected, no offenses detected
```

2.5. Metrics/CyclomaticComplexity

Descripción: Complejidad ciclomática de los métodos, afectada por ciclos y statements condicionales.

1 ofensa.

2.6. Metrics/MethodLength

Descripción: Número de líneas no vacías de un método.

4 ofensas.

2.7. Metrics/ModuleLength

Descripción: Número de líneas no vacías de un módulo.

Sin ofensas.

2.8. Metrics/ParameterLists

Descripción: Número de parámetros que toma método.

Sin ofensas.

2.9. Metrics/PerceivedComplexity

Descripción: Medida de la complejidad que experiencia el lector cuando observa un método.

1 ofensa.