

Ingeniería de software 2021

Actividad n° 4

Informe: Refactorización proyecto grupo 4.

Integrantes: Danielle Rubén Rafael, Garcia Maria Ayelen, Urricelqui Irina.

## 2) Bad smells que reconocemos:

- Líneas de código muy largas. De más de 80 caracteres.
- Espacios después del último carácter en línea.
- Comentarios muy largos.
- Métodos con nombres largos.
- Comentarios ambiguos.
- No usar snake\_case para nombrar variables, métodos.
- No dejar espacios entre las definiciones de métodos.
- No dejar espacios alrededor de operadores.
- Métodos con demasiadas líneas de código.

Espacio entre operadores:

```
@careers=Career.all
@surveys=Survey.all
```

El método `post "/surveys"`, tiene demasiadas líneas de código.

Problema: long method

```
post "/surveys" do
  results = {}

  # calcular la puntuacion del usuario en cada carrera
  results = Career.countUserPointsOnEachCareer(results, params)

  # guardar como resultado del cuestionario realizado por el usuario la carrera con la max puntuacion
  resulting_career = results.key(results.values.max)
  survey = Survey.new(career_id: resulting_career, username: params['username'])

  if survey.save

    # guardar todas las respuestas dadas por el usuario
    load_responses(params, survey.id)
    @resultadoFinal = Career[resulting_career][:name]
    # mostrar la carrera resultante
    erb : 'surveys/result'

  else
    [500, {}, 'Internal Server Error']
  end
end
```

Solución: eliminar comentarios ambiguos.

```
post '/surveys' do
  results = {}
  results = Career.points_by_career(results, params)
  survey = Survey.new(career_id: results.key(results.values.max),
    username: params['username'])
  if survey.save
    load_responses(params, survey.id)
    @resultadoFinal = Career[resulting_career][:name]
    erb : 'surveys/result'
  else
    [500, {}, 'Internal Server Error']
  end
end
```

El método `countUserPointsOnEachCareer` tiene los siguientes problemas:

Problemas: nombre del método muy largo, en formato lowerCamelCase, y con demasiadas líneas de código.

```
10 # funcion que toma un hash y devuelve un contador de
11 # la cantidad de puntos de cada carrera en base a las respuestas del cuestionario
12 def self.countUserPointsOnEachCareer(set, parameters)
13   each do |career|
14     set[career.id] = 0
15   end
16
17   # sumar el contador de cada carrera dependiendo de las respuestas dadas
18   Question.all.each do |question|
19     choice_selected = parameters[: "#{question.id}" ].to_i
20     Outcome.all.select do |out|
21       choice_selected == out.choice_id
22     end.each do |outcome|
23       set[outcome.career_id] += 1
24     end
25   end
26
27   set
28 end
```

Solución: Acortar el nombre, en formato snake\_case y para acortar el código, utilizamos la técnica “Replace Loop with Collection Closure Method” extraída del capítulo 6 del libro “Refactoring: ruby edition” en los ciclos de las líneas 20 y 22, reemplazándolos por una consulta y un mapeo, respectivamente.

```
def self.points_by_career(set, parameters)
  each do |career|
    set[career.id] = 0
  end
  Question.all.each do |question|
    choice_selected = parameters[: "#{question.id}" ].to_i
    outcomes = Outcome.where(choice_id: choice_selected)
    outcomes.map { |outcome| set[outcome.career_id] += 1 }
  end
  set
end
```

Otras ofensas comunes (auto solucionadas por rubocop):

Líneas de código muy largas:

```
get '/surveys/search' do
  @career_name = Career.where(id: params['careerId'].to_i).last.name
  @surveys = Survey.where(Sequel.lit('career_id = ? AND created_at > ? AND created_at < ?', params[:ca
  @first_date = Time.parse(params['firstDate']))
  @last_date = Time.parse(params['lastDate']))
  erb : 'surveys/surveys_by_date_and_career'
end
end
```

196 characters selected

Llamada a método sin argumentos no debe llevar paréntesis:

```
params[:careerId].to_i()
```

Falta de espacio entre las definiciones de métodos:

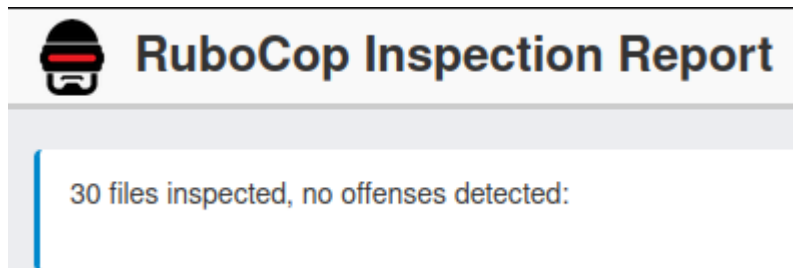
```
get '/surveys/get_career_and_dates' do
  @surveys = Survey.all
  @careers = Career.all
  erb : 'surveys/get_career_and_dates'
end
get '/surveys/search' do
  @career_name = Career.where(id: params['careerId']).to_i().last.name
  @surveys = Survey.where(Sequel.lit('career_id = ? AND created_at > ? AND created_at < ?', params[:career_id], params[:first_date], params[:last_date]))
  @first_date = Time.parse(params['firstDate'])
  @last_date = Time.parse(params['lastDate'])
  erb : 'surveys/surveys_by_date_and_career'
end
```

Entre otras. Hay muchas otras ofensas, las cuales se encuentran en el archivo *refactor\_result.html* en la carpeta *Refactor/html* del proyecto

30 files inspected, 336 offenses detected:

- [Gemfile - 1 offense](#)
- [app.rb - 30 offenses](#)
- [config.ru - 4 offenses](#)
- [db/migrations/00001\\_create\\_posts\\_table.rb - 1 offense](#)
- [db/migrations/00003\\_create\\_outcomes\\_table.rb - 20 offenses](#)
- [db/migrations/00004\\_create\\_careers\\_table.rb - 22 offenses](#)
- [db/migrations/00005\\_create\\_surveys\\_table.rb - 5 offenses](#)
- [db/migrations/00006\\_create\\_responses\\_table.rb - 5 offenses](#)
- [db/migrations/00007\\_create\\_questions\\_table.rb - 24 offenses](#)
- [db/migrations/00008\\_create\\_choices\\_table.rb - 22 offenses](#)
- [db/migrations/00009\\_add\\_foreign\\_keys.rb - 10 offenses](#)
- [db/migrations/00010\\_add\\_type\\_field\\_to\\_questions.rb - 3 offenses](#)
- [db/migrations/00011\\_add\\_primary\\_keys\\_to\\_responses.rb - 3 offenses](#)
- [db/seeds.rb - 23 offenses](#)
- [models/career.rb - 17 offenses](#)
- [models/choice.rb - 16 offenses](#)
- [models/init.rb - 1 offense](#)
- [models/outcome.rb - 10 offenses](#)
- [models/question.rb - 16 offenses](#)
- [models/response.rb - 9 offenses](#)
- [models/survey.rb - 8 offenses](#)
- [tests/models/career\\_test.rb - 7 offenses](#)
- [tests/models/choice\\_test.rb - 11 offenses](#)
- [tests/models/outcome\\_test.rb - 9 offenses](#)
- [tests/models/question\\_test.rb - 17 offenses](#)
- [tests/models/response\\_test.rb - 20 offenses](#)
- [tests/models/survey\\_test.rb - 11 offenses](#)
- [tests/test\\_helper.rb - 11 offenses](#)

Además, creamos un informe con el resultado de análisis de las métricas del proyecto, el cual también se encuentra en la carpeta anterior, con el nombre *metrics.html*. El informe dice lo siguiente:



Después de ejecutar con rubocop, nos quedaron las siguientes ofensas que no son autocorregibles, las cuales corregimos manualmente.

```
Inspecting 30 files
.C.....C.....

Offenses:

app.rb:15:81: C: Layout/LineLength: Line is too long. [83/80]
# Dado un id de una carrera, la obtiene de la base de datos y muestra sus datos
app.rb:40:81: C: Layout/LineLength: Line is too long. [103/80]
# guardar como resultado del cuestionario realizado por el usuario la carrera con la max puntuacion
app.rb:58:81: C: Layout/LineLength: Line is too long. [88/80]
# funcion utilizada para cargar las respuestas dadas por un usuario a las preguntas de
app.rb:59:81: C: Layout/LineLength: Line is too long. [83/80]
# un cuestionario, se deben pasar como parametros un hash con las respuestas y el
app.rb:93:81: C: Layout/LineLength: Line is too long. [187/80]
'career_id = ? AND created_at > ? AND created_at < ?', params[:careerId].
to_i, DateTime.parse(params[:firstDate]), DateTime.parse(params[:lastDate]).next_day
models/career.rb:11:81: C: Layout/LineLength: Line is too long. [83/80]
# la cantidad de puntos de cada carrera en base a las respuestas del cuestionario
models/career.rb:12:3: C: Metrics/MethodLength: Method has too many lines. [12/10]
def self.countUserPointsOnEachCareer(set, parameters) ...

30 files inspected, 7 offenses detected
```