

Actividad 4

Refactorización

Integrantes:

Agustin Giungi

Nicolas Ortiz

Joaquin Moran

Este informe resume todo el trabajo de refactorización aplicado al proyecto Test Vocacional. En base a la lectura de los capítulos 3, 6, 7 y 8 pudimos detectar los siguientes bad smells.

Bad smells:

Duplicated code: Tenemos código duplicado en 2 métodos los cuales pueden ser unificados en uno solo e invocados la cantidad de veces que sea necesaria.

La regla de refactorización que usaríamos sería Extract method ya que nos permitirá agrupar esas líneas en un solo método.

```
get "/career" do
  @career = params[:career_name]
  erb :info_career_index
end

get "/search-career" do
  @career = params[:career_name]
  erb :info_career_index
end
```

Si bien estos métodos son utilizados para distintas funcionalidades, tienen la misma lógica.

Este es el método que modulariza

```
def get_career_by_name
  @career = params[:career_name]
  erb :info_career_index
end
```

Invocación del método

```
get "/career" do
  get_career_by_name()
end

get "/search-career" do
  get_career_by_name()
end
```

Speculative generality: El endpoint el cual nunca tuvo utilidad, es /posts, el cual debería ser eliminado ya que no aporta nada al proyecto y solo es un gasto tenerlo.

La regla de refactorización que usaríamos sería inline class, el tema es que no deberíamos mover nada otra parte porque nada de eso sirve.

```
post "/posts" do
  request.body.rewind # in case someone already read it
  data = JSON.parse request.body.read
  post = Post.new(description: data['desc'])
  if post.save
    [201, { 'Location' => "posts/#{post.id}" }, 'CREATED']
  else
    [500, {}, 'Internal Server Error']
  end
end

get '/posts' do
  p = Post.where(id: 1).last
  p.description
end
```

Este es el método el cual eliminamos.

Estos cambios se realizan en el commit: <https://github.com/nico-ortiz/testVocacional-app/commit/52fec86e626e450a32219041fa10746571d77466>

Resultado del análisis de las métricas

Después de investigar y ejecutar el análisis de rubocop obtuvimos los siguientes resultados a partir de nuestro código

Errores de métricas detectadas (6 ofensas en total):

- **BlockLength (1 ofensa):** el bloque tiene muchas líneas de código.

db/migrations/000017_alter_tables_id_and_fk.rb:1:1: C: Metrics/BlockLength: Block has too many lines. [26/25]

- **AbcSize (4 ofensas):** el método realiza muchas acciones.

models/survey.rb:12:5: C: Metrics/AbcSize: Assignment Branch Condition size for career_obtained is too high. [<14, 17, 7> 23.11/17]

tests/models/choices_test.rb:39:3: C: Metrics/AbcSize: Assignment Branch Condition size for test_choice_has_many_response is too high. [<6, 19, 0> 19.92/17]

tests/models/questions_test.rb:57:5: C: Metrics/AbcSize: Assignment Branch Condition size for test_has_many_responses is too high. [<6, 19, 0> 19.92/17]

tests/models/responses_test.rb:39:5: C: Metrics/AbcSize: Assignment Branch Condition size for test_response_has_choice_survey_and_question is too high. [<8, 19, 1> 20.64/17]

- **MethodLength (1 ofensa):** el método tiene muchas líneas de código.

models/survey.rb:12:5: C: Metrics/MethodLength: Method has too many lines. [13/10]

Procedimos a autocorregirlas con rubocop y el resultado fue subido al repositorio.

Estos cambios se realizan en el commit:

<https://github.com/nico-ortiz/testVocacional-app/commit/82069e4cafce57295f3c3aa6b9ca14351f273a3a>

Y se documentan en este:

<https://github.com/nico-ortiz/testVocacional-app/commit/7ef4ddcc5b4ab8024b8c0305cdab52231f5a478f>