



Universidad Nacional de Río Cuarto

Facultad de Ciencias Exactas, Físico-Químicas y Naturales

Departamento de Computación

Asignatura: **Ingeniería de Software (Cód. 3304) - 2024**

Año 2024

# **TALLER IS 2024**

## **ACTIVIDAD NRO: 4**

**INTEGRANTES:**

CAMPREGHER BRUNO

CENTENO NICOLÁS

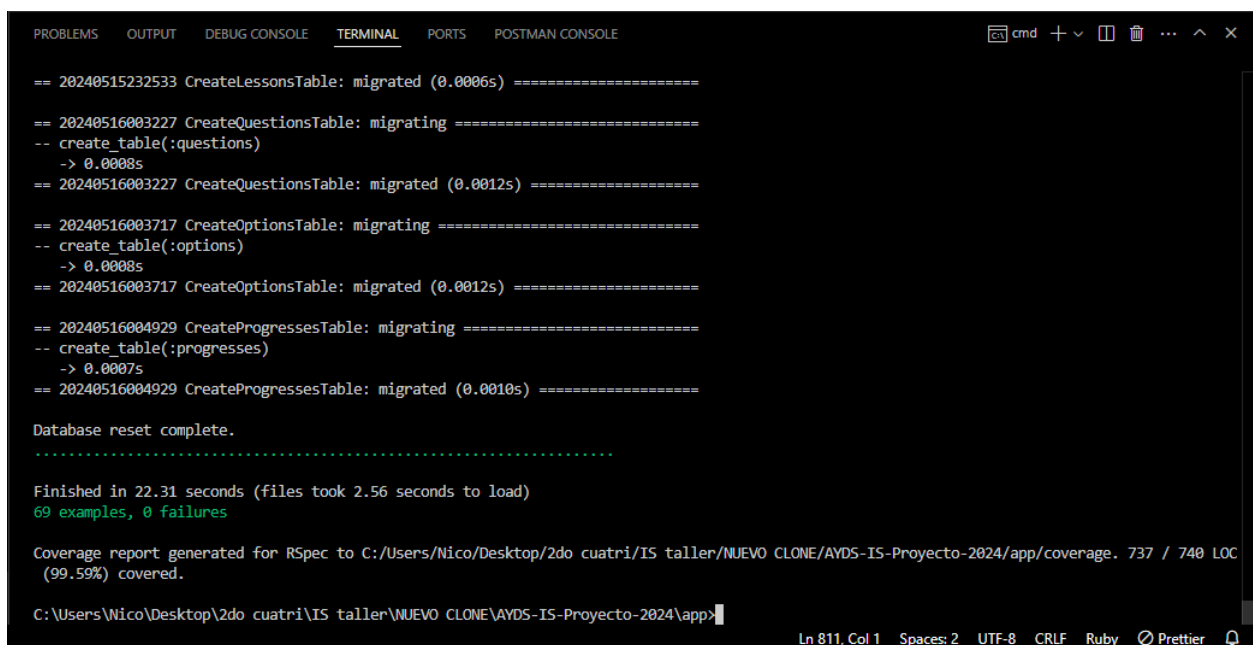
SERAFIN AGUSTÍN

# Taller IS 2024 - Actividad Nro: 4

## Análisis de reporte

### Tests iniciales

Iniciamos el ciclo de refactorización en nuestro proyecto con nuestro código pasando los tests correspondientes.



```
== 20240515232533 CreateLessonsTable: migrated (0.0006s) =====
== 20240516003227 CreateQuestionsTable: migrating =====
-- create_table(:questions)
-> 0.0008s
== 20240516003227 CreateQuestionsTable: migrated (0.0012s) =====
== 20240516003717 CreateOptionsTable: migrating =====
-- create_table(:options)
-> 0.0008s
== 20240516003717 CreateOptionsTable: migrated (0.0012s) =====
== 20240516004929 CreateProgressesTable: migrating =====
-- create_table(:progresses)
-> 0.0007s
== 20240516004929 CreateProgressesTable: migrated (0.0010s) =====

Database reset complete.
.....

Finished in 22.31 seconds (files took 2.56 seconds to load)
69 examples, 0 failures

Coverage report generated for RSpec to C:/Users/Nico/Desktop/2do cuatri/IS taller/NUEVO CLONE/AYDS-IS-Proyecto-2024/app/coverage. 737 / 740 LOC
(99.59%) covered.

C:\Users\Nico\Desktop\2do cuatri\IS taller\NUEVO CLONE\AYDS-IS-Proyecto-2024\app>
```

Resultado de ejecutar los tests de nuestra aplicación antes de iniciar con la refactorización

Al asegurarnos de ello, podemos continuar con el siguiente paso del ciclo, el cual es incluir la gema Rubocop a nuestro Gemfile, instalarla, y ejecutar Rubocop.

### Ejecución de Rubocop

Luego de ejecutar Rubocop, obtenemos el siguiente reporte.

```
puts "Database reset complete."
=====
spec/spec_helper.rb:68:1: C: [Correctable] Style/BlockComments: Do not use block comments.
=begin ...
=====
spec/spec_helper.rb:118:38: C: [Correctable] Layout/CommentIndentation: Incorrect indentation detected (column 0 instead of 2).

22 files inspected, 682 offenses detected, 631 offenses autocorrectable

Tip: Based on detected gems, the following RuboCop extension libraries might be helpful:
* rubocop-rake (https://rubygems.org/gems/rubocop-rake)
* rubocop-rspec (https://rubygems.org/gems/rubocop-rspec)

You can opt out of this message by adding the following to your config (see https://docs.rubocop.org/rubocop/extensions.html#extension-suggesti
ons for more options):
AllCops:
  SuggestExtensions: false

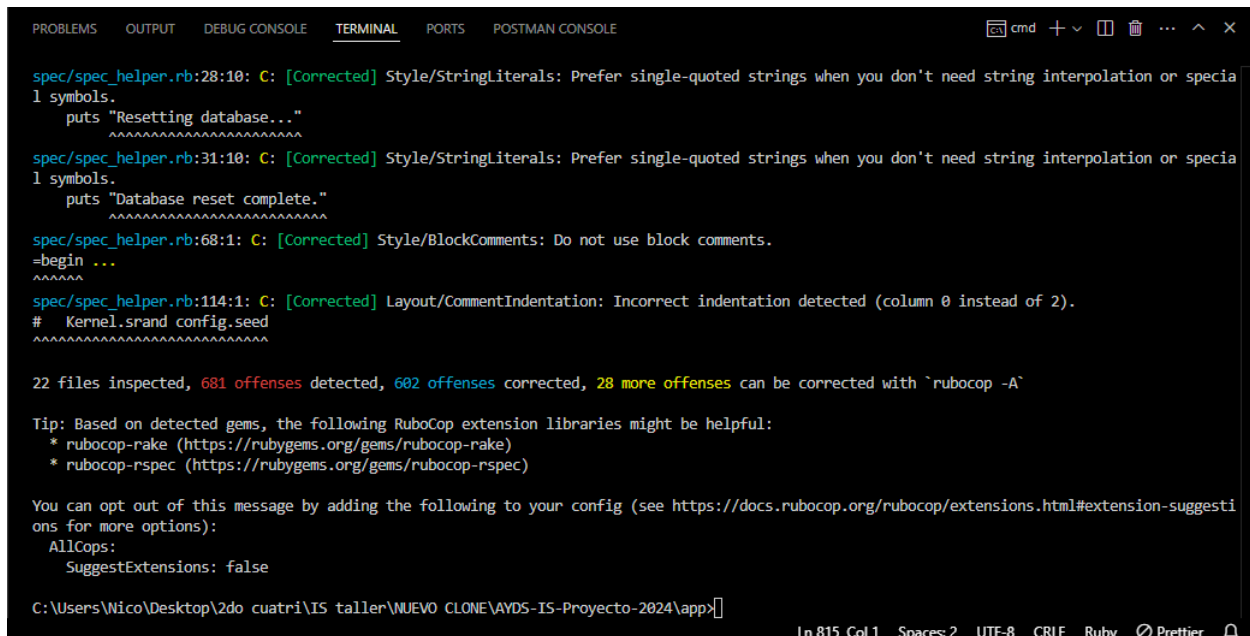
C:\Users\Nico\Desktop\2do cuatri\IS taller\NUEVO CLONE\AYDS-IS-Proyecto-2024\app>
```

Reporte dado por Rubocop luego de su ejecución

En la imagen se puede ver, separadas por una línea amarilla, las dos partes principales del informe provisto por Rubocop posterior a su ejecución.

Por encima de la línea amarilla se nos informa sobre cada una de las líneas que deben ser corregidas en cada archivo del proyecto, también si son autocorregibles o no y el tipo de "ofensa" cometido en ella. Por debajo, obtenemos el conteo total de archivos inspeccionados, las ofensas detectadas y las ofensas autocorregibles.

Empezamos corrigiendo las ofensas autocorregibles ejecutando Rubocop con la flag "-a" para autocorregir las ofensas "de manera segura", es decir, solo las ofensas que Rubocop nos asegura que es seguro corregir de manera automática evitando realizar modificaciones que potencialmente puedan generar errores en la aplicación.



```
spec/spec_helper.rb:28:10: C: [Corrected] Style/StringLiterals: Prefer single-quoted strings when you don't need string interpolation or special symbols.
  puts "Resetting database..."
  ~~~~~

spec/spec_helper.rb:31:10: C: [Corrected] Style/StringLiterals: Prefer single-quoted strings when you don't need string interpolation or special symbols.
  puts "Database reset complete."
  ~~~~~

spec/spec_helper.rb:68:1: C: [Corrected] Style/BlockComments: Do not use block comments.
=begin ...
~~~~~

spec/spec_helper.rb:114:1: C: [Corrected] Layout/CommentIndentation: Incorrect indentation detected (column 0 instead of 2).
# Kernel.srand config.seed
~~~~~

22 files inspected, 681 offenses detected, 602 offenses corrected, 28 more offenses can be corrected with `rubocop -A`

Tip: Based on detected gems, the following RuboCop extension libraries might be helpful:
* rubocop-rake (https://rubygems.org/gems/rubocop-rake)
* rubocop-rspec (https://rubygems.org/gems/rubocop-rspec)

You can opt out of this message by adding the following to your config (see https://docs.rubocop.org/rubocop/extensions.html#extension-suggestions for more options):
AllCops:
  SuggestExtensions: false

C:\Users\Nico\Desktop\2do cuatri\IS taller\NUEVO CLONE\AYDS-IS-Proyecto-2024\app>
```

Reporte dado por Rubocop finalizada la autocorrección de ofensas con flag -a

De manera similar al reporte anterior, en este nuevo reporte se nos presentan las líneas de cada archivo que fueron corregidas de manera automática por Rubocop y la ofensa correspondiente a cada una. Por último, el conteo total de archivos inspeccionados, ofensas detectadas, ofensas corregidas y ofensas que pueden ser corregidas utilizando el flag "-A" donde Rubocop autocorrigie tanto las ofensas "seguras" como las "inseguras" de corregir.

Luego de ejecutar la autocorrección de ofensas segura de Rubocop, nos aseguramos de que nuestra aplicación siga cumpliendo con los tests correspondientes.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  POSTMAN CONSOLE

== 20240515232533 CreateLessonsTable: migrated (0.0010s) =====
== 20240516003227 CreateQuestionsTable: migrating =====
-- create_table(:questions)
-> 0.0012s
== 20240516003227 CreateQuestionsTable: migrated (0.0017s) =====
== 20240516003717 CreateOptionsTable: migrating =====
-- create_table(:options)
-> 0.0008s
== 20240516003717 CreateOptionsTable: migrated (0.0012s) =====
== 20240516004929 CreateProgressesTable: migrating =====
-- create_table(:progresses)
-> 0.0008s
== 20240516004929 CreateProgressesTable: migrated (0.0014s) =====

Database reset complete.
.....

Finished in 22.33 seconds (files took 2.65 seconds to load)
69 examples, 0 failures

Coverage report generated for RSpec to C:/Users/Nico/Desktop/2do cuatri/IS taller/NUEVO CLONE/AYDS-IS-Proyecto-2024/app/coverage. 737 / 740 LOC
(99.59%) covered.

C:\Users\Nico\Desktop\2do cuatri\IS taller\NUEVO CLONE\AYDS-IS-Proyecto-2024\app>

Ln 815, Col 1  Spaces: 2  UTF-8  CRLF  Ruby  Prettier
```

Resultado de ejecutar los tests de nuestra aplicación después de ejecutar la autocorrección de ofensas segura de Rubocop

# Ofensas

## Style/FrozenStringLiteralComment: Missing frozen string literal comment.

### Ofensa cometida

```
models/lesson.rb:1:1: C: [Correctable] Style/FrozenStringLiteralComment: Missing frozen string literal comment.
class Lesson < ActiveRecord::Base
```

### Porción de código que causa la ofensa

```
class Lesson < ActiveRecord::Base
  # referencia a la relación 1 a N entre Lesson y Question
```

```
has_many :questions  
end
```

## Solución

Rubocop está sugiriendo agregar un comentario al inicio del archivo para activar el "string freezing" de Ruby. Esto mejora la eficiencia del código al hacer que todas las literales de cadena tipo String sean inmutables por defecto.

Se agrega esta línea al inicio del archivo:

```
# frozen_string_literal: true
```

Quedándonos de esta forma

```
# frozen_string_literal: true      <-- se agrega este string  
  
class Lesson < ActiveRecord::Base  
  # referencia a la relación 1 a N entre Lesson y Question  
  has_many :questions  
end
```

Código actualizado

---

## **Style/SafeNavigation: Use safe navigation (&.) instead of checking if an object exists before calling the method.**

### Ofensa cometida

```
app.rb:228:8: C: [Correctable] Style/SafeNavigation: Use safe  
navigation (&.) instead of checking if an object exists befor  
e calling the method.  
if @user && @user.authenticate(params[:current_password])
```

## Porción de código que causa la ofensa

```
if @user && authenticate(params[:current_password])
  @user.destroy
  session.clear
  redirect '/'
else
  erb :profile, locals: { error_message: 'Incorrect current password.' }
end
end
```

## Solución

Rubocop indica que se puede simplificar parte del código usando la navegación segura (*safe navigation operator*) en lugar de verificar si el objeto existe antes de llamar a un método. Esto hace que el código sea más limpio y legible.

Para solucionarlo, podemos usar el operador de navegación segura &. en vez de &&.

```
if @user&.authenticate(params[:current_password])
  @user.destroy
  session.clear
  redirect '/'
else
  erb :profile, locals: { error_message: 'Incorrect current password.' }
end
end
```

Código actualizado

---

## Style/MutableConstant: Freeze mutable objects assigned to constants.

### Ofensa cometida y porción de código que causa la ofensa

```
models/progress.rb:8:21: C: [Correctable] Style/MutableConstant: Freeze mutable objects assigned to constants.  
  PROGRESS_LEVELS = { ...          <-- causa la ofensa
```

### Solución

Rubocop indica que se está asignando un objeto mutable a una constante. Para evitar problemas de mutabilidad, se nos recomienda "congelar" el objeto, lo que lo convierte en inmutable.

Para solucionarlo, utilizamos el método freeze en la constante.

```
PROGRESS_LEVELS = {  
  1 => 'Junior',  
  2 => 'Semi-Senior',  
  3 => 'Senior'  
}.freeze          <-- se agrega este método
```

Código actualizado

---

## Layout/LineLength: Line is too long.

### Ofensa cometida y porción de código que comete la ofensa

```
db/seeds.rb:624:121: C: Layout/LineLength: Line is too long.  
[124/120]  
  description: 'Which of the following keywords is used to  
skip the rest of the code in the current iteration of a loop?',
```



## Solución

La advertencia de Rubocop indica que la línea de código supera la longitud máxima permitida (120 caracteres por defecto).

Para solucionarlo, dividimos la línea larga en segmentos más cortos y manejables, o simplemente la achicamos.

---

## Style/Documentation: Missing top-level documentation comment.

### Ofensa cometida

```
models/progress.rb:3:1: C: Style/Documentation: Missing top-level documentation comment for class Progress.  
class Progress < ActiveRecord::Base
```

### Porción de código que comete la ofensa

```
# frozen_string_literal: true  
  
class Option < ActiveRecord::Base  
  # referencia a la relacion 1 a N entre Question y Option  
  belongs_to :question  
end
```

*Rubocop también detecta esta ofensa en las clases Lesson, Progress, Question y User.*

## Solución

Para solucionarlo, se agrega el siguiente comentario encima de la clase:

```
# Option model
```

```
# frozen_string_literal: true

# Option model  <-- se agrega este string
class Option < ActiveRecord::Base
  # referencia a la relación 1 a N entre Question y Option
  belongs_to :question
end
```

Código actualizado

---

## **Naming/VariableNumber: Use normalcase for symbol numbers.**

Se asegura de que todas las variables numeradas utilicen el estilo configurado, snake\_case, normalcase o non\_integer, para su numeración.

### **Ofensa cometida y porción de código que comete la ofensa**

```
spec/app_spec.rb:777:9: C: Naming/VariableNumber: Use normalc
ase for symbol numbers.
      option_description_1: 'True',
```

### **Solución**

Para solucionarlo, se cambia el nombre de la variable por option\_description1:

```
option_description1: 'True',
```

*Se corrige esta ofensa en otras tres oportunidades.*

---

## **Naming/VariableName: Use snake\_case for variable names.**

Se asegura de que todas las variables utilicen el estilo snake\_case

### **Ofensa cometida y porción de código que comete la ofensa**

```
db/seeds.rb:717:3: C: Naming/VariableName: Use snake_case for
variable names.
  userProgress = Progress.create(current_lesson: 1)
```

### **Solución**

Para solucionarlo, se cambia el nombre de la variable por user\_progress:

```
user_progress = Progress.create(current_lesson: 1)
```

*Se corrige esta ofensa en otras tres oportunidades.*

---

## **Style/CaseEquality: Avoid the use of the case equality operator ===.**

Evita el uso explícito del operador de igualdad idéntica ===. Como su nombre indica, está destinado a ser utilizado implícitamente por expresiones case y fuera de ellas provee código bastante confuso.

### **Ofensa cometida**

```
app.rb:23:46: C: Style/CaseEquality: Avoid the use of the cas
e equality operator ===.
  if protected_routes.any? { |route| route === request.path
_info } && session[:username].nil?
```

### **Porción de código que comete la ofensa**

```
before do
  if protected_routes.any? { |route| route === request.path_info } && session[:username].nil?
    # La ruta está protegida y no hay sesión activa.
    redirect '/login'
  end
end
```

## Solución

Para solucionarlo, fue necesario realizar modificaciones para evitar el uso del operador ===

```
before do
  if protected_routes.any? do |route|
    route.is_a?(String) ? (request.path_info == route) : route.match(request.path_info)
  end && session[:username].nil?
    redirect '/login'
  end
end
```

Código actualizado

---

## Metrics/BlockLength: Block has too many lines.

Detecta clases que son muy extensas (exceden un número predeterminado de líneas de código).

Busca evitar que los bloques de código sean muy largos y difícil de entender.

## Ofensa cometida

```
spec/app_spec.rb:9:1: C: Metrics/BlockLength: Block has too m
```

```
any lines. [592/25]
```

En este ejemplo, `app_spec` supera el límite predeterminado de 25 líneas de código por clase.

## Solución

Para solucionarlo, se divide el bloque de código en bloques más cortos en caso de que sea posible.

```
|— spec
| |— app_spec                (clase que se divide en bloques más
chicos)
| |— admin_add_question_spec (clase más chica generada al refactorizar)
| |— admin_panel_spec
| |— admin_top_questions_spec
| |— game_spec
| |— lesson_progression_spec
| |— login_process_spec
| ...
|   └─ progress_spec
└─ config.ru
```

Caso contrario, se puede optar por ignorar esta regla (indicándolo en el archivo de configuración de Rubocop o dentro de la clase agregando un string especial) en casos donde sea absolutamente necesario contar con un bloque más extenso, documentando la razón detrás de tal decisión.

```
# rubocop:disable Metrics/BlockLength <-- se agregan estos
strings al bloque
```

```
RSpec.describe 'Login Process' do
  # ...código
end
# rubocop:enable Metrics/BlockLength <-- se agregan estos
strings al bloque
```

Código actualizado para que Rubocop ignore un bloque de código específico

```
Metrics/BlockLength:
  Max: 75
```

Modificación a .rubocop.yml para subir el límite global predeterminado de líneas de código por clase a 75

Esta última modificación se implementó en la configuración de Rubocop dentro de nuestro proyecto porque algunos de nuestros archivos de test sobrepasaban el límite de 25 líneas de código por clase aún luego de refactorizar nuestro código.

Consideramos que un límite tan bajo restringe nuestra capacidad para testear la aplicación adecuadamente y garantizar que funcione correctamente, cubriendo la mayor cantidad del código posible, es por eso que decidimos subir el límite a 75 líneas de código por clase.

---

## Code smells detectados

### Clase grande (large class)

Actualmente, la clase App maneja toda la lógica de la aplicación.

#### Regla de refactorización:

- **Extraer clase (extract class):** se recomienda dividir la clase principal en clases más pequeñas y especializadas donde cada nueva clase debería

enfocarse en una única responsabilidad o funcionalidad específica.

Luego de refactorizar la clase App, creando un controlador específico por cada grupo de rutas, el directorio de la aplicación se actualiza de la siguiente forma con los nuevos controladores ubicados en la carpeta controllers

```
|— app.rb
|— controllers
|   |— app_controller
|   |— user_controller
|   |— learning_controller
|   |— time_trial_controller
|   |— admin_controller
|— config.ru
```

Esta estructura reduce la complejidad y hace que cada clase tenga una responsabilidad bien definida.

---

## Duplicated code

### Regla de refactorización:

- **Extract Method:** implica mover código duplicado en un método separado y luego llamar a ese método cuando sea necesario.

En los siguientes casos se puede observar código repetido:

- **Primer caso**

Antes de la refactorización:

```
get '/timetrial/end_game' do
  @user = User.find_by(username: session[:username])
  progress = @user.progress

  progress.act(session[:points], session[:max_streak])
```

```

    @points = session[:points]
    @streak = session[:max_streak]

    session[:streak] = 0
    session[:points] = 0
    session[:answered_questions] = []
    session[:max_streak] = 0

    erb :end_game_time
end

```

```

get '/timetrial' do
  session[:streak] = 0
  session[:points] = 0
  session[:answered_questions] = []
  session[:max_streak] = 0
  session[:game_started_at] = Time.now.to_i
  redirect '/timetrial/play'
end

```

Después de la refactorización:

```

get '/timetrial' do
  reset_game_session
  session[:game_started_at] = Time.now.to_i
  redirect '/timetrial/play'
end

```

```

get '/timetrial/end_game' do
  @user = current_user
  progress = @user.progress

```



```

        progress.update_progress(session[:points], session[:max_streak])

        @points = session[:points]
        @streak = session[:max_streak]

        reset_game_session

        erb :end_game_time
    end

```

```

def reset_game_session
  session[:streak] = 0
  session[:points] = 0
  session[:answered_questions] = []
  session[:max_streak] = 0
end

```

- **Segundo caso**

Antes de la refactorización:

```

get '/timetrial/end_game' do
  @user = User.find_by(username: session[:username])

```

```

post '/user/change_password' do
  @user = User.find_by(username: session[:username])

```

```

post '/user/change_email' do

```

```
@user = User.find_by(username: session[:username])
```

Después de la refactorización:

Se utiliza un método existente perteneciente a `helpers.rb` evitando repetir código.

```
def current_user  
  @current_user ||= User.find_by(username: session[:username])  
end
```

```
get '/timetrial/end_game' do  
  @user = current_user
```

```
post '/user/change_password' do  
  @user = current_user
```

```
post '/user/change_email' do  
  @user = current_user
```

---

## Long Method

### Regla de refactorización:

- **Inline Method:** eliminar un método que tiene poca lógica y sustituirlo por su contenido en cada lugar donde es llamado

Antes de la refactorización:

```
def obtener_ranking_usuarios  
  usuarios = User.includes(:progress).all.to_a  
  usuarios_with_progress = usuarios.select { |usuario| usuario.progress }
```

```

      usuarios_with_progress.sort_by do |usuario|
        [-usuario.progress.points, -usuario.progress.streak]
      end
    end
  end
end

```

```

get '/ranking' do
  @ranking_usuarios = obtener_ranking_usuarios
  current_user
  erb :ranking
end

```

Este método, aunque cumplía la función de obtener y ordenar usuarios según su progreso, no proporcionaba un valor agregado significativo en comparación con la simplicidad que podría lograrse al incluir directamente la lógica en el manejador de la ruta '/ranking'.

Después de la refactorización:

```

get '/ranking' do
  @ranking_usuarios = Progress.order(points: :desc, streak: :desc)
  current_user
  erb :ranking
end

```

## Dead code

El siguiente método ya no se utiliza y se considera código muerto.

```

get '/learning/lesson_completed' do
  if session[:completed_user_id]
    @user = User.find(session[:completed_user_id])
    session[:success] = nil
  end
end

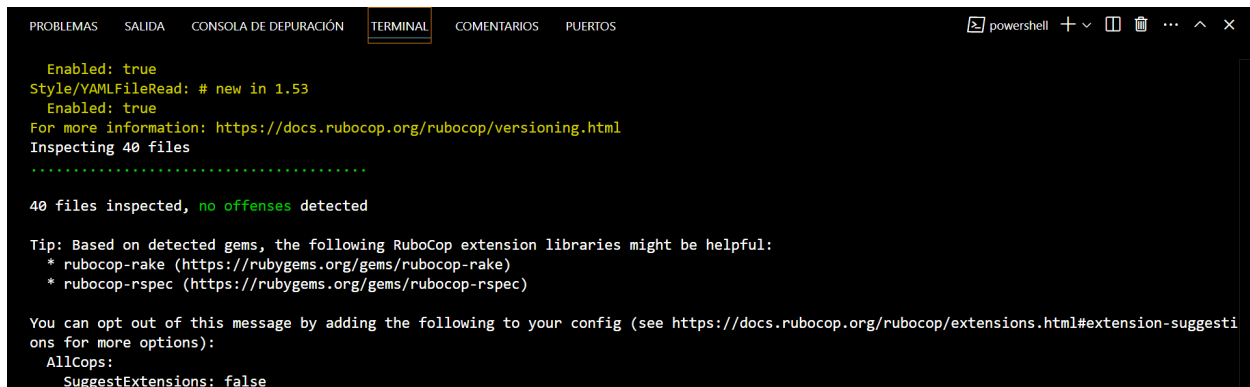
```

```
    session[:error] = nil
    erb :lesson_completed
  else
    redirect "/dashboard"
  end
end
```

## Comentarios finales

Durante el desarrollo del proyecto, se han identificado inconscientemente varios code smells, incluyendo código duplicado, métodos muy largos, código muerto y generalidad especulativa. Estos se han ido corrigiendo a medida que se han detectado, lo que ha mejorado la calidad y mantenibilidad del código.

A continuación se muestra el resultado de la ejecución de Rubocop una vez finalizada la etapa de refactorización.

A screenshot of a terminal window with a dark background. The terminal shows the output of a RuboCop command. At the top, there's a header bar with tabs: PROBLEMAS, SALIDA, CONSOLA DE DEPURACIÓN, TERMINAL (which is active), COMENTARIOS, and PUERTOS. The output text is as follows:  
Enabled: true  
Style/YAMLFFileRead: # new in 1.53  
Enabled: true  
For more information: <https://docs.rubocop.org/rubocop/versioning.html>  
Inspecting 40 files  
.....  
40 files inspected, no offenses detected  
  
Tip: Based on detected gems, the following RuboCop extension libraries might be helpful:  
\* rubocop-rake (<https://rubygems.org/gems/rubocop-rake>)  
\* rubocop-rspec (<https://rubygems.org/gems/rubocop-rspec>)  
  
You can opt out of this message by adding the following to your config (see <https://docs.rubocop.org/rubocop/extensions.html#extension-suggestions> for more options):  
AllCops:  
 SuggestExtensions: false

Resultado de ejecutar los tests de nuestra aplicación luego de finalizar la refactorización