

Métricas y Refactorización

Anatomy Easy

Fecha: 05/11/2024

Ingeniería de Software (3304) - 2024

- Gutierrez, Marysol
- Palacios, Micaela

Introducción

Durante el análisis inicial del proyecto, las herramientas de estilo y calidad de código (como RuboCop) detectaron un total de **938 infracciones** en **25 archivos**. Estas infracciones incluían problemas de convención, estilo, organización, y modularización. La revisión de cada archivo y la aplicación de técnicas de refactorización permitieron reducir el número de infracciones, siguiendo buenas prácticas y principios de desarrollo limpio. A continuación, se detallan los problemas y las soluciones aplicadas en cada archivo o categoría.

Problemas Detectados y Acciones de Refactorización

Al instalar la herramienta RuboCop pudimos analizar los reportes generados:



Informe de inspección de RuboCop

25 archivos inspeccionados, 938 infracciones detectadas:

- [Gemfile - 14 delitos](#)
- [Rakefile - 1 infracción](#)
- [app.rb - 54 delitos](#)
- [config.ru - 1 delito](#)
- [db/migrate/20240515213018_create_users_table.rb - 4 delitos](#)
- [db/migrate/20240522203941_create_questions.rb - 2 delitos](#)
- [db/migrate/20240522204010_create_options.rb - 2 delitos](#)
- [db/migrate/20240919164913_add_avatar_to_users.rb - 3 delitos](#)
- [db/migrate/20240929144033_add_level_to_questions.rb - 2 delitos](#)
- [db/migrate/20240930145612_add_level_completed_to_user.rb - 2 delitos](#)
- [db/migrate/20241001123045_change_level_completed_to_string_in_users.rb - 5 delitos](#)
- [db/migrate/20241001142052_change_level_completed_to_in_users.rb - 3 delitos](#)
- [db/migrate/20241009155622_add_role_to_users.rb - 2 delitos](#)
- [db/migrate/20241102195335_add_counts_to_questions.rb - 2 delitos](#)
- [db/schema.rb - 33 delitos](#)
- [db/seeds.rb - 662 delitos](#)
- [lib/tasks/user_roles.rake - 5 delitos](#)
- [modelos/opción.rb - 6 infracciones](#)
- [modelos/question.rb - 5 delitos](#)
- [modelos/user.rb - 6 ofensas](#)
- [spec/app_spec.rb - 86 delitos](#)
- [spec/models/option_spec.rb - 8 infracciones](#)
- [spec/models/question_spec.rb - 13 delitos](#)
- [spec/models/user_spec.rb - 13 delitos](#)
- [spec/spec_helper.rb - 4 delitos](#)

El proceso de corrección comenzó con la ejecución del comando *bundle exec rubocop -a* y *bundle exec rubocop -A*, el cual corrigió automáticamente algunas infracciones de estilo. Sin embargo, ciertas ofensas requerían ajustes manuales.



RuboCop Inspection Report

25 files inspected, 93 offenses detected:

- [Gemfile - 1 offense](#)
- [Rakefile - 1 offense](#)
- [app.rb - 9 offenses](#)
- [config.ru - 1 offense](#)
- [db/migrate/20240515213018 create_users_table.rb - 1 offense](#)
- [db/migrate/20240522203941 create_questions.rb - 1 offense](#)
- [db/migrate/20240522204010 create_options.rb - 1 offense](#)
- [db/migrate/20240919164913 add_avatar_to_users.rb - 2 offenses](#)
- [db/migrate/20240929144033 add_level_to_questions.rb - 2 offenses](#)
- [db/migrate/20240930145612 add_level_completed_to_user.rb - 2 offenses](#)
- [db/migrate/20241001123045 change_level_completed_to_string_in_users.rb - 2 offenses](#)
- [db/migrate/20241001142052 change_level_completed_to_in_users.rb - 2 offenses](#)
- [db/migrate/20241009155622 add_role_to_users.rb - 2 offenses](#)
- [db/migrate/20241102195335 add_counts_to_questions.rb - 2 offenses](#)
- [db/schema.rb - 2 offenses](#)
- [db/seeds.rb - 29 offenses](#)
- [lib/tasks/user_roles.rake - 1 offense](#)
- [models/option.rb - 2 offenses](#)
- [models/question.rb - 2 offenses](#)
- [models/user.rb - 3 offenses](#)
- [spec/app_spec.rb - 14 offenses](#)
- [spec/models/option_spec.rb - 2 offenses](#)
- [spec/models/question_spec.rb - 5 offenses](#)
- [spec/models/user_spec.rb - 3 offenses](#)
- [spec/spec_helper.rb - 1 offense](#)

Las correcciones se dividieron, donde:

1. Archivos con Infracciones y Problemas Identificados



RuboCop Inspection Report

28 files inspected, 56 offenses detected:

- [db/migrate/20240919164913 add_avatar_to_users.rb - 1 offense](#)
- [db/migrate/20240929144033 add_level_to_questions.rb - 1 offense](#)
- [db/migrate/20240930145612 add_level_completed_to_user.rb - 1 offense](#)
- [db/migrate/20241001123045 change_level_completed_to_string_in_users.rb - 1 offense](#)
- [db/migrate/20241001142052 change_level_completed_to_in_users.rb - 1 offense](#)
- [db/migrate/20241009155622 add_role_to_users.rb - 1 offense](#)
- [db/migrate/20241102195335 add_counts_to_questions.rb - 1 offense](#)
- [db/schema.rb - 1 offense](#)
- [db/seeds.rb - 28 offenses](#)
- [spec/app_spec.rb - 13 offenses](#)
- [spec/models/option_spec.rb - 1 offense](#)
- [spec/models/question_spec.rb - 4 offenses](#)
- [spec/models/user_spec.rb - 2 offenses](#)

Archivo Gemfile (1 infracción):

- Problema:
Probablemente se deba a un problema menor de formato o alineación de gemas.

- Acción:
Revisar y corregir la alineación de las gemas o cualquier estilo inconsistente.

Archivo Rakefile (1 infracción):

- Problema: Relacionado con la falta de documentación y comentarios en bloques de código.

- Acción: Añadir comentarios que describen la funcionalidad del archivo o alinear adecuadamente.

app.rb (9 infracciones):

- Problemas:
 - Comentarios faltantes para la documentación de métodos o clases.
 - Problemas de alineación en líneas o excedentes de longitud en algunas líneas.
- Acción: Documentar métodos faltantes, dividir líneas largas en varias líneas y ajustar la alineación.

RuboCop Inspection Report

24 files inspected, 56 offenses detected:

- [app.rb - 4 offenses](#)
- [db/schema.rb - 1 offense](#)
- [db/seeds.rb - 28 offenses](#)
- [models/option.rb - 1 offense](#)
- [models/question.rb - 1 offense](#)
- [models/user.rb - 1 offense](#)
- [spec/app_spec.rb - 13 offenses](#)
- [spec/models/option_spec.rb - 1 offense](#)
- [spec/models/question_spec.rb - 4 offenses](#)
- [spec/models/user_spec.rb - 2 offenses](#)

Revisar el archivo y realizar correcciones menores de alineación si Rubocop lo permite.

Archivos en db/migrate

(varias infracciones en cada archivo):

- Problemas: Falta de comentarios en ciertos bloques.

- Acción: Alinear correctamente las columnas y formato consistente

db/schema.rb (2 infracciones):

- Problema: Problemas de formato y alineación.

- Acción:

Informe de inspección de RuboCop

27 expedientes inspeccionados, 21 infracciones detectadas:

- [db/schema.rb - 1 delito](#)
- [spec/app_spec.rb - 13 delitos](#)
- [spec/models/option_spec.rb - 1 infracción](#)
- [spec/models/question_spec.rb - 4 infracciones](#)
- [spec/models/user_spec.rb - 2 infracciones](#)

db/seeds.rb (29 infracciones):

- Problemas: Infracciones por exceso de líneas

- Acción: Dividir las líneas largas, usar alineación consistente y simplificar bloques repetitivos para mejorar la legibilidad.

Archivos en models (1-3 infracciones en cada archivo):

- Problemas: Documentación faltante en algunos métodos y problemas de alineación en option.rb, question.rb, y user.rb.
- Acción: Documentar métodos faltantes y ajustar cualquier desalineación.

Archivos en spec (varias infracciones):

- Problemas:
 - En app_spec.rb y otros archivos de prueba, las infracciones esta relacionadas con métodos de prueba largos o configuraciones repetitivas.
 - Acción: Dividir métodos largos y reorganizar bloques repetidos en helpers para reducir la longitud de las pruebas.

2. Principios y Justificación de Refactorización

Estado Inicial

Inicialmente, la clase app.rb presentaba múltiples problemas de diseño:

1. Clase Grande (Large Class): app.rb centralizaba todas las rutas de la aplicación, acumulando demasiadas responsabilidades y afectando la legibilidad y mantenibilidad del código.
2. Complejidad de Mantenimiento: Con el crecimiento de la aplicación, esta clase se volvería cada vez más difícil de modificar sin riesgo de introducir errores.

El análisis inicial de RuboCop mostró 938 infracciones en el proyecto, con 54 infracciones sólo en app.rb, lo cual reflejaba la necesidad de una refactorización.

Seguimos las reglas y el pdf adjunto del profesor, donde:

Reglas y Code Smells Detectados

- ❖ Code Smell: Clase Grande (Large Class).
- ❖ Reglas de Refactorización Aplicadas:
 - Extraer Clase (Extract Class): Se separaron las rutas en controladores específicos, organizando el proyecto en las clases adminController y UsersController. Separación de Responsabilidades (Single Responsibility Principle): Se organizaron los controladores para que cada uno gestione una responsabilidad clara.

Proceso de Refactorización

Después de la aplicación de bundle exec rubocop -a, logramos reducir considerablemente las infracciones en app.rb. Posteriormente, se refactorizó app.rb mediante los siguientes pasos:

- ❖ Extracción de Controladores: Cada grupo de rutas se movió a una clase controladora dedicada:
 - adminController: Maneja las rutas generales de la aplicación.
 - UsersController: Gestiona las rutas relacionadas con usuarios.

- ❖ Separación de Lógica: Se mejoró la cohesión de cada controlador, asignando responsabilidades específicas. Esto permitió un enfoque modular y facilitó la mantenibilidad y extensibilidad de la aplicación.

Estado Final

Al finalizar el proceso de refactorización:

- ❖ Infracciones reducidas: La cantidad total de infracciones se redujo drásticamente.
- ❖ Mejor organización del código: Cada controlador ahora sigue el Principio de Responsabilidad Única.
- ❖ Legibilidad mejorada: La estructura de los archivos se simplificó, lo que hace que el código sea más fácil de leer y entender.

rubocop_report_app_model_new




Informe de inspección de RuboCop

31 expedientes inspeccionados, 26 infracciones detectadas:

- [app.rb - 1 delito](#)
- [controladores/admin_controller.rb - 1 infracción](#)
- [controladores/game_controller.rb - 1 infracción](#)
- [controladores/main_controller.rb - 1 infracción](#)
- [controladores/users_controller.rb - 1 delito](#)
- [db/schema.rb - 1 delito](#)
- [spec/app_spec.rb - 13 delitos](#)
- [spec/models/option_spec.rb - 1 infracción](#)
- [spec/models/question_spec.rb - 4 infracciones](#)
- [spec/models/user_spec.rb - 2 infracciones](#)

rubocop_report_app_refactorizar



Informe de inspección de RuboCop

29 expedientes inspeccionados, 24 infracciones detectadas:

- [controladores/admin_controller.rb - 1 infracción](#)
- [controladores/user_controller.rb - 2 infracciones](#)
- [db/schema.rb - 1 delito](#)
- [spec/app_spec.rb - 13 delitos](#)
- [spec/models/option_spec.rb - 1 infracción](#)
- [spec/models/question_spec.rb - 4 infracciones](#)
- [spec/models/user_spec.rb - 2 infracciones](#)

rubocop_report_final



Informe de inspección de RuboCop

29 expedientes inspeccionados, 14 infracciones detectadas:

- [controladores/user_controller.rb](#) - 1 infracción
- [db/schema.rb](#) - 1 delito
- [spec/app_spec.rb](#) - 8 delitos
- [spec/models/option_spec.rb](#) - 1 infracción
- [spec/models/question_spec.rb](#) - 1 delito
- [spec/models/user_spec.rb](#) - 2 infracciones

Estado Final (Infracciones No Corregibles)

A pesar de la refactorización y la reducción de infracciones en el código, algunas infracciones persisten y no se pueden corregir debido a diversas limitaciones técnicas o a decisiones de diseño que no afectan la funcionalidad general del proyecto. Las infracciones que quedan son las siguientes:

- ❖ controladores/user_controller.rb: 1 infracción
- ❖ db/schema.rb: 1 infracción
- ❖ spec/app_spec.rb: 8 infracciones
- ❖ spec/models/option_spec.rb: 1 infracción
- ❖ spec/models/question_spec.rb: 1 infracción
- ❖ spec/models/user_spec.rb: 2 infracciones

Justificación de Infracciones no corregibles

1. controladores/user_controller.rb: La infracción se debe a un estilo de codificación que, no sigue la convención, pero no afecta la lógica del controlador.
2. db/schema.rb: Las convenciones en este archivo son generadas automáticamente por migraciones y pueden no alinearse completamente con las convenciones de RuboCop.
3. spec/app_spec.rb, spec/models/option_spec.rb, spec/models/question_spec.rb, spec/models/user_spec.rb: Las infracciones en las pruebas suelen ser complejas de corregir sin modificar el comportamiento de las pruebas, lo que podría afectar su validez.

Conclusión

Esta refactorización mejora la calidad del código, alineándose con los principios de diseño orientado a objetos. La estructura modular de los controladores permite que el código sea extensible y más fácil de mantener, garantizando así un desarrollo más escalable y ágil.