

Universidad Nacional de Río Cuarto

Materia: Ingeniería de Software

Tema:

“Análisis de Cobertura de Código del Proyecto Guesswhich”

Alumnos: Juarez Marcelo, Lingua Matías, Reynoso Juan Cruz

Fecha: 03/09/2024

1. Introducción

En el siguiente informe se desarrolla un análisis detallado sobre los datos obtenidos a través de la herramienta *SimpleCov*, utilizada para medir el porcentaje de cobertura de código alcanzado con los tests realizados.

El objetivo de este análisis es reflejar los resultados obtenidos antes y después de mejorar/agregar los tests necesarios para lograr la mayor cobertura posible. También se compara la cobertura previa con la posterior a agregar los nuevos tests, identificando las áreas específicas donde se logra un mayor incremento de cobertura y aquellas que aún necesiten de mejoras adicionales.

2. Configuración de SimpleCov

Detalles sobre la herramienta de medición de cobertura de código utilizada:

Instalación: para su instalación se añadió la gema “simplecov” al Gemfile y se ejecutó “bundle install”.

Configuración: se agregó al archivo `server_spec.rb` el siguiente código:

```
require 'simplecov'
SimpleCov.start
```

Para su utilización se corren los tests utilizando el comando `rspec`, al terminar se genera el archivo `index.html` en la ruta `coverage/assets/` con la información de cobertura especificada.

3. Resultados de la Cobertura

El informe de *SimpleCov* muestra una cobertura total del 47.76%. Donde se puede ver que sólo la mitad del código base se encuentra adecuadamente cubierto.

Los archivos totales son 4: `server.rb`, `models/user.rb`, `models/answer.rb` y `models/questions.rb`, de los cuales se observa un bajo porcentaje de cobertura que se encuentra principalmente en código del `server.rb` y `user.rb`. Los modelos `answer.rb` y `question.rb` se encuentran cubiertos al 100%.

El archivo `server.rb` es el que más líneas relevantes contiene por lo que la gran mayoría de tests que se deben agregar para una mayor cobertura serán en este archivo, ya que se encuentra todo el funcionamiento, rutas y lógica de la aplicación.

Los resultados obtenidos a través de SimpleCov previo a la actualización de tests se encuentran al final del informe.

4. Mejora de la Cobertura

Después de obtener los resultados iniciales de la cobertura de código a través de la herramienta SimpleCov y ver la parte de código que falta cubrir se realizaron mejoras y se agregaron nuevos tests.

4.1. Descripción de los Tests

4.1.1. Redirección a la página de restablecimiento de contraseñas

Verifica que la ruta `/password_resets/new` esté disponible y retorna un código de estado 200.

4.1.2. Confirmación de cambio de contraseña exitoso

Verifica que la ruta `/password_resets/notice` esté disponible y retorna un código de estado 200.

4.1.3. Redirección a la información del usuario cuando está autenticado

Simula un inicio de sesión , comprueba que el usuario autenticado pueda acceder a `/user` mostrando correctamente su información y retorna un código de estado 200.

4.1.4. Redirección a la página de inicio de sesión cuando no está autenticado

Verifica que el acceso a `/users` sin autenticación , retorna un código de estado 302 y redirige correctamente a `/login`.

4.1.5. Muestra la página de inicio

Verifica que la página de inicio (`/`) esté disponible y retorna un código de estado 200.

4.1.6. Redirección a la página de eventos cuando está autenticado

Simula un inicio de sesión ,retorna un código de estado 200 y comprueba que el usuario autenticado pueda acceder a `/events`.

4.1.7. Redirección a la página de inicio de sesión cuando no está autenticado

Verifica que el acceso a `/events` sin autenticación, retorna un código de estado 302 y redirige correctamente a `/login`.

4.1.8. Muestra el formulario de inicio de sesión

Verifica que la ruta `/login` muestre correctamente el formulario de inicio de sesión.

4.1.9. Inicio de sesión con credenciales válidas

Simula un inicio de sesión con credenciales válidas, retorna un código de estado 302 y verifica la redirección a `/menu`.

4.1.10. Falla de inicio de sesión con credenciales inválidas

Simula un intento de inicio de sesión con credenciales incorrectas, retorna un código de estado 302 y verifica la redirección a `/failed`.

4.1.11. Redirección a la página de inicio de sesión al intentar acceder a una ruta protegida sin autenticación

Verifica que el acceso a `/menu` sin autenticación, retorna un código de estado 302 y redirige correctamente a `/login`.

4.1.12. Muestra el formulario de registro

Verifica que la ruta `/register` muestre correctamente el formulario de registro.

4.1.13. Intento de registro con credenciales ya usadas

Verifica que al intentar registrarse con un nombre de usuario ya existente, se retorna un código de estado 302 y el programa redirige correctamente a `/failed`.

4.1.14. Registro con credenciales disponibles

Verifica que un registro con credenciales únicas, retorna un código de estado 302 y redirige correctamente a `/login`.

4.1.15. Intento de registro con contraseñas no coincidentes

Verifica que un intento de registro con contraseñas que no coinciden, retorna un código de estado 302 y redirige correctamente a `/failed`.

4.1.16. Falla de registro con un correo ya usado

Verifica que al intentar registrarse con un correo ya existente, se retorna un código de estado 302 y el programa redirige correctamente a `/failed`.

4.1.17. Muestra el ranking cuando está autenticado

Simula un inicio de sesión, retorna un código de estado 200 y verifica que el usuario autenticado pueda acceder a `/ranking`.

4.1.18. Finaliza el juego cuando el usuario ha respondido todas las preguntas

Simula un juego donde el usuario ha respondido todas las preguntas y verifica que es redirigido a `/finish`.

4.1.19. Elimina un usuario

Verifica que un usuario autenticado pueda eliminar su cuenta, redirigiendo correctamente después de la eliminación.

4.1.20. Redirección a la página de finalización cuando el usuario se queda sin tiempo

Simula un juego donde el usuario se queda sin tiempo y verifica la redirección a `/finish`.

4.1.21. Edición del perfil de usuario

Verifica que el usuario autenticado pueda acceder a la página de edición de perfil y realizar cambios.

4.1.22. Reinicio de la puntuación del usuario

Verifica que un usuario autenticado pueda resetear su puntuación, reflejando los cambios en la página de perfil.

4.1.23. Redirección a la página de finalización cuando se acaba el tiempo

Verifica la redirección a `/finish` cuando el tiempo del juego se agota.

4.1.24. Incremento de la puntuación y redirección a la próxima pregunta cuando la respuesta es correcta

Simula una respuesta correcta y verifica el incremento de la puntuación y la redirección a la siguiente pregunta.

4.1.25. Finalización del juego al alcanzar las 15 preguntas

Verifica que el juego finaliza y dirige a `/finish` al responder correctamente 15 preguntas.

4.1.26. Reinicio de la puntuación y redirección a la página de finalización cuando la respuesta es incorrecta

Simula una respuesta incorrecta y verifica que la puntuación se reinicia y se dirige a `/finish`.

4.1.27. Elección de un evento y acceso a su material

Verifica que un usuario autenticado pueda seleccionar un evento y acceder al material educativo asociado.

4.1.28. Restablecimiento de contraseña con un usuario existente

Verifica que un usuario registrado puede iniciar el proceso de restablecimiento de contraseña y recibir una notificación de éxito.

4.1.29. Restablecimiento de contraseña con un usuario no existente

Verifica que un intento de restablecimiento de contraseña con un usuario no registrado muestra un mensaje de error.

4.1.30. Renderizado del formulario de restablecimiento de contraseña cuando el token es válido

Verifica que un token de restablecimiento de contraseña válido permite acceder al formulario correspondiente.

4.1.31. Redirección a la página de noticias cuando el token usado para restablecer la contraseña expiró

Verifica que cuando el usuario intenta restablecer su contraseña y el token de restablecimiento de contraseña expiró, por ende es redirigido a `/password_resets/notice` informando que el token ha expirado.

4.1.32. Restablecimiento de contraseña exitoso cuando el token es válido

Verifica que cuando el token de restablecimiento de contraseña no ha expirado, el usuario puede restablecer su contraseña correctamente, por ende es redirigido a `/password_resets/notice` informando que la contraseña fue exitosamente restablecida.

4.1.33. Redirección a la página de noticias cuando el token usado para restablecer la contraseña es invalido

Verifica que cuando el usuario quiere recuperar su contraseña pero ingresa un email que no está registrado, el sistema lo redirecciona a `/password_resets/notice`.

4.1.34. Eliminación del token de restablecimiento

Verifica que el método `clear_password_reset_token!` de la clase `User` elimina correctamente el token de restablecimiento de contraseña y la marca de tiempo asociada

Los detalles de SimpleCov mostrando la nueva cobertura obtenida se encuentran al final de este informe.

5. Conclusión

Llegando al final de este informe, podemos concluir que los tests existentes previo a la utilización de la herramienta *SimpleCov*, eran altamente escasos en cobertura del código base y algunos modelos del proyecto. Por ello, esta herramienta fue realmente útil para poder ver en detalle las líneas de código relevantes que no estaban siendo cubiertas por algún test aún, para luego agregarlo y así, abarcar la cobertura necesaria.

Es importante alcanzar el 100% de la cobertura del código para la detección de errores o comportamientos inesperados del sistema, para poder actuar sobre los mismos y para finalmente lograr un producto de software de calidad.

All Files (47.76%)**All Files (47.76% covered at 1.15 hits/line)**

4 files in total.

201 relevant lines, 96 lines covered and 105 lines missed. (47.76%)

Search:

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
server.rb	45.00 %	325	180	81	99	1.13
models/user.rb	60.00 %	25	15	9	6	1.40
models/answer.rb	100.00 %	4	2	2	0	1.00
models/question.rb	100.00 %	7	4	4	0	1.00

Showing 1 to 4 of 4 entries

All Files (100.0%)

All Files (100.0% covered at 4.9 hits/line)

4 files in total.

201 relevant lines, 201 lines covered and 0 lines missed. (100.0%)

Search:

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
models/answer.rb	100.00 %	4	2	2	0	1.00
models/question.rb	100.00 %	7	4	4	0	1.00
models/user.rb	100.00 %	25	15	15	0	3.60
server.rb	100.00 %	325	180	180	0	5.14

Showing 1 to 4 of 4 entries