



# INGENIERÍA DE SOFTWARE

*Informe de Proceso de Pruebas y Cobertura  
de Código*

Hernandez Florencia, Galucci Esteban, Trepin Gonzalo.

## Introducción

Este informe proporciona un análisis detallado del proceso de pruebas y del estado de la cobertura de código para el proyecto desarrollado con Ruby, Sinatra y ActiveRecord. El objetivo primordial de estas actividades fue elevar la calidad del código mediante la implementación de pruebas rigurosas y asegurar una cobertura de código adecuada que respalde la estabilidad y la fiabilidad del software.

El proyecto en cuestión emplea Ruby como lenguaje de programación, junto con el framework Sinatra para la creación de aplicaciones web y ActiveRecord para la gestión de la base de datos. Dado el alcance y la complejidad del proyecto, se ha realizado un esfuerzo considerable para integrar un conjunto completo de pruebas unitarias, de integración y funcionales. La finalidad de este enfoque es garantizar que todas las partes del código funcionen correctamente y que los posibles errores sean detectados y corregidos de manera proactiva.

### Estrategia de Pruebas:

La estrategia de pruebas se diseñó para cubrir varios aspectos críticos del proyecto:

- **Pruebas Unitarias:** Estas pruebas están enfocadas en validar el comportamiento de las unidades individuales del código, como clases y métodos. Se utilizó el marco de pruebas **Rspec** para escribir y ejecutar estas pruebas, lo cual permite verificar que cada componente funcione de manera independiente según lo esperado.
- **Pruebas de Integración:** Estas pruebas aseguran que diferentes componentes del sistema interactúen correctamente entre sí. En este caso, se llevaron a cabo para validar la integración entre **Sinatra y ActiveRecord**, así como para garantizar la correcta comunicación entre la aplicación y la base de datos.

### Cobertura de Código:

La cobertura de código se ha medido utilizando la herramienta **SimpleCov**, que permite identificar las partes del código que están cubiertas por las pruebas y aquellas que no lo están. **SimpleCov** proporciona un informe detallado de la cobertura de código en varias métricas clave:

- **Cobertura de Declaraciones:** Mide el porcentaje de líneas de código que se ejecutan durante las pruebas. Una cobertura alta en este aspecto indica que la mayoría de las líneas de código han sido probadas.

- **Cobertura de Rama:** Evalúa el porcentaje de ramas de control (como if, else, switch) que han sido ejecutadas durante las pruebas. Esto ayuda a asegurar que todas las posibles rutas de ejecución sean verificadas.
- **Cobertura de Funciones:** Determina el porcentaje de funciones o métodos que han sido llamados durante las pruebas. Es crucial para garantizar que todos los métodos importantes sean probados.

\*Aclarado esto, vamos con los resultados del informe

# Análisis de Cobertura Previa

## 1. Cobertura Inicial

Fecha de Evaluación: 25/08/2024  
Porcentaje de Cobertura Total: 60.65%  
Áreas Evaluadas:  
Archivos Evaluados: app.rb, helpers.rb, models/\*.rb, auth\_middleware.rb, etc.  
Cobertura por Archivo/Módulo:  
  
app.rb: 50.57%  
helpers.rb: 47.37%  
models/user.rb: 69.23%  
Otros archivos con cobertura del 100% incluyen models/country.rb, models/exam.rb, etc.

Imagen que muestra el análisis de cobertura del proyecto antes de las mejoras:

All Files ( 60.65% covered at 1.65 hits/line )  
14 files in total.  
277 relevant lines, 168 lines covered and 109 lines missed. ( 60.65% )

File	% covered <sup>▲</sup>	Lines <sup>⌵</sup>	Relevant Lines <sup>⌵</sup>	Lines covered <sup>⌵</sup>	Lines missed <sup>⌵</sup>	Avg. Hits / Line <sup>⌵</sup>
helpers.rb	47.37 %	31	19	9	10	0.84
app.rb	50.57 %	311	174	88	86	1.12
models/user.rb	69.23 %	64	39	27	12	4.05
auth_middleware.rb	88.89 %	17	9	8	1	5.89
models/country.rb	100.00 %	3	2	2	0	1.00
models/exam.rb	100.00 %	5	4	4	0	1.00
models/lesson.rb	100.00 %	5	4	4	0	1.00
models/level.rb	100.00 %	5	4	4	0	1.00
models/material.rb	100.00 %	4	3	3	0	1.00
models/option.rb	100.00 %	4	3	3	0	1.00
models/progresslesson.rb	100.00 %	5	4	4	0	1.00
models/qa.rb	100.00 %	5	4	4	0	1.00
models/question.rb	100.00 %	5	4	4	0	1.00
models/topic.rb	100.00 %	5	4	4	0	1.00

## 2. Áreas con Baja Cobertura

- Descripción:
- app.rb: Muchas rutas y métodos no estaban suficientemente cubiertos, con un 50.57% de cobertura.
  - helpers.rb: Pruebas insuficientes para algunas funciones de ayuda, con un 47.37% de cobertura.
  - models/user.rb: Aunque tenía una cobertura del 69.23%, aún había áreas que no estaban completamente cubiertas.

## Cambios Realizados

### 1. Mejoras en los Tests

Descripción de Cambios:

- Nuevas Pruebas Añadidas:

Se añadieron pruebas para todas las rutas en app.rb que antes no estaban cubiertas, junto a otras, las cuales como estaban relacionadas con los otros dos archivos que presentaban una baja cobertura e hicieron que aumente la cobertura de los tres.

- Pruebas Mejoradas:

Se refactorizaron las pruebas existentes para app.rb para cubrir más casos de borde.

Razón de los Cambios:

Mejorar la cobertura de código para asegurar que las principales funcionalidades y validaciones estén correctamente probadas.

### 2. Elementos Evaluados

Archivos/Módulos Afectados:

- app.rb
- helpers.rb
- models/user.rb

Métodos y Funciones Cubiertas:

- app.rb: Métodos de controladores para autenticación, gestión de lecciones y exámenes.
- helpers.rb: Métodos de ayuda para autenticación y formateo de datos.
- models/user.rb: Validaciones y asociaciones, pruebas extensas para todos los métodos.

### Descripción detallada de tests agregados en archivo app.rb

#### 1. Contexto 'GET /'

- it 'renders the index page with the correct layout':  
Verifica que la página de inicio (/) se renderice correctamente y contenga los elementos esperados en el diseño, como "Contacto", "Donaciones" e "Ingresar".

#### 2. Contexto 'GET /signup'

- it 'renders the signup page for nonlogged in users':  
Comprueba que la página de registro (/signup) se renderice correctamente para los usuarios que no han iniciado sesión.
- it 'redirects logged in users away from the signup page':  
Asegura que los usuarios que ya están autenticados sean redirigidos desde la página de registro (/signup) hacia el panel de control (/dashboard).

#### 3. Contexto 'API Error Handling'

- it 'returns 404 for nonexistent QA correct answer':  
Verifica que la API devuelva un error 404 cuando se solicita la respuesta correcta de una QA que no existe.

- it 'returns an error if the exam is not found':

Asegura que la API devuelva un error si se solicita un examen que no se encuentra.

- it 'returns 401 for unauthenticated access to API':

Comprueba que la API devuelva un error 401 para solicitudes no autenticadas.

#### 4. Contexto 'GET /exam/:lesson\_id/:level\_id'

- it 'renders the exam page if the level is unlocked':

Verifica que la página del examen se renderice correctamente cuando el nivel está desbloqueado.

#### 5. Contexto 'GET /api/exam/:exam\_id/:correct\_answers'

- it 'returns "No higher level found" if no more levels exist':

Asegura que la API devuelva un mensaje adecuado si no hay niveles superiores disponibles.

- it 'returns an error if progress is not found for the user and lesson':

Comprueba que la API devuelva un error si no se encuentra el progreso del usuario para una lección.

- it 'returns 404 if the question is not found for the QA record':

Verifica que la API devuelva un error 404 si no se encuentra la pregunta asociada a un registro QA.

- it 'returns 404 if the correct answer is not found for the QA record':

Asegura que la API devuelva un error 404 si no se encuentra la respuesta correcta para un registro QA.

#### 6. Contexto 'POST /completed\_lesson'

- it 'successfully completes a lesson and updates progress':

Verifica que al completar una lección, el progreso del usuario se actualice correctamente.

#### 7. Contexto 'GET /profile'

- it 'shows user profile page':

Comprueba que la página de perfil del usuario se renderice correctamente.

- it 'redirects to login if not authenticated':

Verifica que los usuarios no autenticados sean redirigidos a la página de inicio de sesión cuando intentan acceder a la página de perfil.

#### 8. Contexto 'POST /update\_streak'

- it 'successfully updates streak':

Asegura que el rastro del usuario se actualice correctamente cuando se envíe una solicitud para actualizarlo.

#### 9. Contexto 'GET /leaderboard'

- it 'shows the leaderboard with users sorted by highest streak':

Verifica que la página de la tabla de clasificación muestre a los usuarios ordenados por su racha más alta.

10. Contexto 'GET /materials/:lesson\_id/:level\_id'

- it 'shows materials for a lesson and level':

Comprueba que se muestren los materiales para una lección y nivel específicos.

11. Contexto 'GET /lessons/levels/:lesson\_id/:level'

- it 'renders the level page if the level is unlocked':

Verifica que la página del nivel se renderice correctamente si el nivel está desbloqueado.

- it 'returns 404 if lesson\_id is invalid':

Asegura que se devuelva un error 404 si el lesson\_id proporcionado es inválido.

12. Contexto 'GET /quiz/:exam\_id'

- it 'renders the quiz page for a valid exam ID':

Verifica que la página del cuestionario se renderice correctamente para un ID de examen válido.

13. Contexto 'GET /lessons'

- it 'renders the lessons page with lessons':

Asegura que la página de lecciones muestre las lecciones existentes con sus descripciones.

## **Análisis de Cobertura Posterior**

### **1. Cobertura Después de las Mejoras**

Fecha de Evaluación: 28/08/2024

Porcentaje de Cobertura Total: 100%

Áreas Evaluadas:

Archivos Evaluados: app.rb, helpers.rb, models/\*.rb, auth\_middleware.rb, config/environment.rb, etc.

Cobertura por Archivo/Módulo:

app.rb: 100%

helpers.rb: 100%

models/user.rb: 100%

Otros archivos con cobertura del 100% incluyen auth\_middleware.rb, config/environment.rb, etc.

Imagen que muestra el análisis de la cobertura del proyecto después de las mejoras:

All Files ( 100.0% covered at 9.31 hits/line )							
15 files in total. 282 relevant lines, 282 lines covered and 0 lines missed. ( 100.0% )							
Search: <input type="text"/>							
File	% covered <sup>▲</sup>	Lines <sup>▾</sup>	Relevant Lines <sup>▾</sup>	Lines covered <sup>▾</sup>	Lines missed <sup>▾</sup>	Avg. Hits / Line <sup>▾</sup>	
app.rb	100.00 %	310	174	174	0	5.85	
auth_middleware.rb	100.00 %	17	9	9	0	32.78	
config/environment.rb	100.00 %	6	5	5	0	1.00	
helpers.rb	100.00 %	31	19	19	0	6.95	
models/country.rb	100.00 %	3	2	2	0	1.00	
models/exam.rb	100.00 %	5	4	4	0	1.00	
models/lesson.rb	100.00 %	5	4	4	0	1.00	
models/level.rb	100.00 %	5	4	4	0	1.00	
models/material.rb	100.00 %	4	3	3	0	1.00	
models/option.rb	100.00 %	4	3	3	0	1.00	
models/progresslesson.rb	100.00 %	5	4	4	0	1.00	
models/qa.rb	100.00 %	5	4	4	0	1.00	
models/question.rb	100.00 %	5	4	4	0	1.00	
models/topic.rb	100.00 %	5	4	4	0	1.00	
models/user.rb	100.00 %	64	39	39	0	29.21	

## 2. Incrementos en la Cobertura

Áreas con Mayor Incremento:

- app.rb: Incremento del 49.43%, alcanzando una cobertura total del 100%.
- helpers.rb: Incremento del 52.63%, logrando una cobertura total del 100%.
- models/user.rb: Incremento del 30.77%, con cobertura total del 100%.

Razones del Incremento:

Las nuevas pruebas cubrieron casos previamente no contemplados y mejoraron la verificación de la funcionalidad existente.

## 3. Áreas que Requieren Mejora Adicional:

No se identificaron áreas con cobertura insuficiente, ya que se alcanzó el 100% de cobertura en todos los archivos evaluados.

## Conclusión

### 1. Resumen de Observaciones

Mejoras Logradas:

- Se logró un incremento general del 39.35% en la cobertura de código, alcanzando un 100% de cobertura total.
- Se cubrieron rutas y métodos previamente no testeados, aumentando la robustez de las pruebas.

### 2. Impacto en el Proyecto

Beneficios de la Mejora en la Cobertura:

- La mejora en la cobertura de pruebas asegura una mayor estabilidad y fiabilidad del código, reduciendo el riesgo de errores en producción.

Lecciones Aprendidas:

- La implementación de pruebas exhaustivas es crucial para la calidad del software.
- La cobertura del 100% proporciona una mayor confianza en la solidez del proyecto.