

# Actividades del Taller

## Actividad 1 - Informe de Cobertura

La primera cobertura de código arrojó los siguientes resultados:

All Files ( 77.56% )

Generated 15 minutes ago

All Files ( 77.56% covered at 0.78 hits/line )

19 files in total.  
254 relevant lines, 197 lines covered and 57 lines missed ( 77.56% )

Search:

File	% covered <sup>^</sup>	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
Q models/account.rb	50.00 %	54	36	18	18	0.50
Q specmodels/option_spec.rb	71.43 %	37	21	15	6	0.71
Q specmodels/lesson_spec.rb	72.22 %	32	18	13	5	0.72
Q specmodels/question_spec.rb	73.33 %	27	15	11	4	0.73
Q specmodels/test_spec.rb	73.33 %	27	15	11	4	0.73
Q specmodels/account_test_spec.rb	75.00 %	24	12	9	3	0.75
Q specmodels/account_spec.rb	77.14 %	65	35	27	8	0.77
Q specmodels/account_lesson_spec.rb	77.78 %	18	9	7	2	0.78
Q specmodels/account_question_spec.rb	77.78 %	19	9	7	2	0.78
Q specmodels/account_option_spec.rb	83.33 %	13	6	5	1	0.83
Q models/account_lesson.rb	87.50 %	14	8	7	1	0.88
Q models/account_option.rb	87.50 %	13	8	7	1	0.88
Q models/account_question.rb	87.50 %	14	8	7	1	0.88
Q models/account_test.rb	88.89 %	16	9	8	1	0.89
Q models/init.rb	100.00 %	11	10	10	0	1.00
Q models/lesson.rb	100.00 %	11	8	8	0	1.00
Q models/option.rb	100.00 %	12	9	9	0	1.00
Q models/question.rb	100.00 %	11	9	9	0	1.00
Q models/test.rb	100.00 %	11	9	9	0	1.00

Showing 1 to 19 of 19 entries

Partiendo desde este punto lo siguiente que hicimos fue hacer que los test corran en la base de datos test.sqlite3, dado que eso era lo que traía problemas con los expect (Los marcaba con "Missed line") que buscaban objetos en una base de datos vacia. Los pasos para solucionarlo fueron los siguientes:

1. Correr las migraciones para tener las tablas en la base de datos test.sqlite3

```
docker compose exec app bundle exec rake db:migrate RACK_ENV=test
```

2. Correr los test indicandole a la aplicacion que se ejecute en el entorno de prueba

```
docker compose exec -e RACK_ENV=test app bundle exec rspec --format doc
```

Haciendo esto y corriendo de nuevo la cobertura de código obtuvimos los siguientes resultados:

Code coverage for Src

Archivo | C:/Users/usuario/OneDrive/Escritorio/Universidad/Analisis\_Sist/codeQuest/src/coverage/index.html#/AllFiles

All Files ( 98.43% ) Generated 2 minutes ago

All Files ( 98.43% covered at 1.11 hits/line )

19 files in total  
254 relevant lines, 250 lines covered and 4 lines missed ( 98.43% )

Search:

File	% covered ^	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
Q models/account_lesson.rb	87.50 %	14	8	7	1	0.88
Q models/account_option.rb	87.50 %	13	8	7	1	0.88
Q models/account_question.rb	87.50 %	14	8	7	1	0.88
Q models/account_test.rb	88.89 %	16	9	8	1	0.89
Q models/account.rb	100.00 %	54	36	36	0	1.92
Q models/init.rb	100.00 %	11	10	10	0	1.00
Q models/lesson.rb	100.00 %	11	8	8	0	1.00
Q models/option.rb	100.00 %	12	9	9	0	1.00
Q models/question.rb	100.00 %	11	9	9	0	1.00
Q models/test.rb	100.00 %	11	9	9	0	1.00
Q specmodels/account_lesson_spec.rb	100.00 %	18	9	9	0	1.00
Q specmodels/account_option_spec.rb	100.00 %	13	6	6	0	1.00
Q specmodels/account_question_spec.rb	100.00 %	19	9	9	0	1.00
Q specmodels/account_spec.rb	100.00 %	65	35	35	0	1.00
Q specmodels/account_test_spec.rb	100.00 %	24	12	12	0	1.00
Q specmodels/lesson_spec.rb	100.00 %	32	18	18	0	1.00
Q specmodels/option_spec.rb	100.00 %	37	21	21	0	1.00
Q specmodels/question_spec.rb	100.00 %	27	15	15	0	1.00
Q specmodels/test_spec.rb	100.00 %	27	15	15	0	1.00

Showing 1 to 19 of 19 entries

Generated by simplecov v0.22.0 and simplecov-html v0.12.3

Para lograr que todos los test de los modelos estuvieran al 100% nos faltaba testear la forma en la que se actualiza el progreso de una cuenta al completar un test o una lección, para eso propusimos los siguientes test en los modelos Account\_lesson.rb y Account\_test.rb:

```

1 it "increases progress correctly when lesson is completed" do
2   account = Account.find_or_create_by(name: 'Juan', email: 'juanito@gmail.com', password: 'Juanito32', nickname: 'juanito', theme_light: 'dark')
3
4   test = Test.find_or_create_by(letter: 'A', description: "Description", cant_questions: 5)
5   test.save
6
7   lesson = Lesson.find_or_create_by(number: 1, title: "Title", description: "Description", test_letter: 'A')
8   lesson.save
9
10  account_lesson = AccountLesson.new(lesson_completed: false, account_id: 1, lesson_id: 1)
11  account_lesson.save
12
13  account = Account.find_by(id: 1)
14  initial_progress = account.progress
15
16  account_lesson.update(lesson_completed: true)
17  account_lesson.send(:actualizar_progreso_account) # Llamar al método usando send ya que es privado
18  account.reload
19
20  expect(account.progress).to be > initial_progress
21 end

```

```

1 it "increases progress correctly when test is completed" do
2   test = Test.find_or_create_by(letter: 'A', description: "Description", cant_questions: 5)
3   test.save
4
5   account_test = AccountTest.new(test_completed: false, account_id: 1, test_id: 1)
6   account_test.save
7
8   account = Account.find_by(id: 1)
9   initial_progress = account.progress
10
11  account_test.update(test_completed: true)
12  account_test.send(:actualizar_progreso_account) # Llamar al método usando send ya que es privado
13  account.reload
14
15  expect(account.progress).to be > initial_progress
16 end

```

Y con estos test funcionando ahora la cobertura de código arrojaba los siguientes resultados:

Code coverage for Src

Archivo | C:/Users/usuario/OneDrive/Escritorio/Universidad/Análisis\_Sist/codeQuest/src/coverage/index.html#\_AllFiles

All Files ( 100.0% )

Generated less than a minute ago

All Files ( 100.0% covered at 1.42 hits/line )

19 files in total  
271 relevant lines, 271 lines covered and 0 lines missed ( 100.0% )

Search:

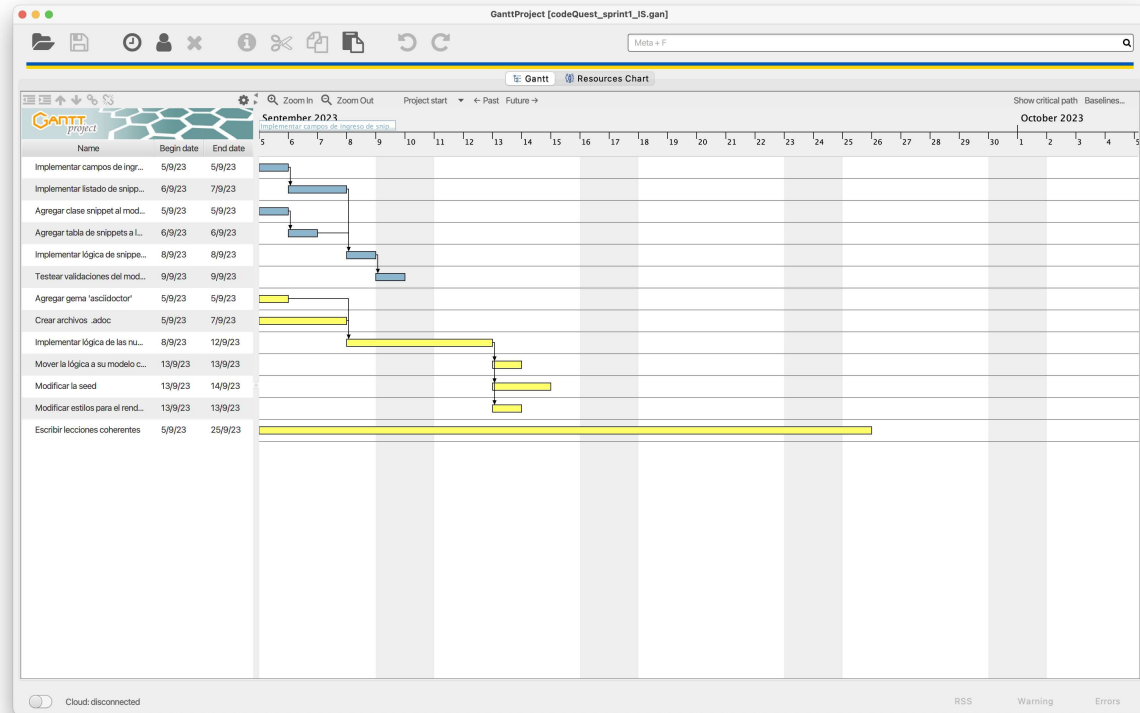
File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
Q models/account.rb	100.00 %	54	36	36	0	4.08
Q models/account_lesson.rb	100.00 %	14	8	8	0	1.25
Q models/account_option.rb	100.00 %	6	4	4	0	1.00
Q models/account_question.rb	100.00 %	7	4	4	0	1.00
Q models/account_test.rb	100.00 %	16	9	9	0	1.22
Q models/init.rb	100.00 %	11	10	10	0	1.00
Q models/lesson.rb	100.00 %	11	8	8	0	1.00
Q models/option.rb	100.00 %	12	9	9	0	1.00
Q models/question.rb	100.00 %	11	9	9	0	1.00
Q models/test.rb	100.00 %	11	9	9	0	1.00
Q spec/models/account_lesson_spec.rb	100.00 %	40	23	23	0	1.00
Q spec/models/account_option_spec.rb	100.00 %	13	6	6	0	1.00
Q spec/models/account_question_spec.rb	100.00 %	19	9	9	0	1.00
Q spec/models/account_spec.rb	100.00 %	65	35	35	0	1.00
Q spec/models/account_test_spec.rb	100.00 %	41	23	23	0	1.00
Q spec/models/lesson_spec.rb	100.00 %	32	18	18	0	1.00
Q spec/models/option_spec.rb	100.00 %	37	21	21	0	1.00
Q spec/models/question_spec.rb	100.00 %	27	15	15	0	1.00
Q spec/models/test_spec.rb	100.00 %	27	15	15	0	1.00

Showing 1 to 19 of 19 entries

Generated by simplecov v0.22.0 and simplecov.html v0.12.3

Una vez que todos los modelos estuvieran cubiertos y testeados (aunque en algunos solo probamos que se cumplen las validaciones definidas sobre ellos), procedimos a testear las rutas de nuestro servidor (server.rb).

## Actividad 2 - Diagrama de Gantt (Primer Sprint)



## Actividad 3 - Diagrama de Gantt (Segundo Sprint)

