DP2 2022-2023

Documento de Testing

ACME-L3-D4

Repositorio: https://github.com/Miguel-Angel-Roldan-Garcia/Acme-L3

GRUPO C1.04.09

-**Estudiante**:

Roldán García, Miguel Ángel (migrolgar2@alum.us.es)

1.0.0

Índice

1. Resumen ejecutivo

El grupo C1.04.09 tiene como objetivo entregar el proyecto **Acme-L3-D04** en el plazo indicado, y con las funcionalidades y documentos solicitados por el Product Owner. Para alcanzar el objetivo, nuestro grupo asignó los siguientes roles a los miembros:

- Manuel Otero Barbasán como Manager
- Javier Fernández Castillo y Álvaro Urquijo Martínez como Analistas
- Todos los miembros del grupo como Desarrolladores
- Andrés Domínguez Ruiz y Miguel Ángel Roldán García como Testers
- Miguel Ángel Roldán García como Operador

Esta asignación se realizó acorde a las especialidades y puntos fuertes de cada uno, cubriendo las posibles debilidades que puedan afectar a la realización del proyecto.

Ante cualquier inconveniente y/o duda durante el desarrollo del proyecto, nos pondremos en contacto inmediatamente con el Product Owner.

2. Historial de versiones

Versión	Fecha	Descripción	Sprint
1.0.0	22-05-2023	• Creación del documento	4

3. Introducción

El objetivo de este documento es mostrar la efectividad del trabajo realizado por el estudiante en labores de testing formal y automatizado.

En primer lugar se mostrará el testing funcional. Se agruparán los casos de test realizados por características y se proveerá una breve descripción. Finalmente se evaluará la efectividad de cada caso de test. Para ello se usará la siguiente métrica: porcentaje de líneas de código evaluadas. Cuantas más líneas de código haya evaluado satisfactoriamente nuestro test, menos probable es que existan errores en esa parte del código.

En segundo lugar se evaluará el testing de rendimiento. Se proveerán gráficos y datos sobre el rendimiento temporal de los tests funcionales en 2 equipos diferentes y se evaluará cuál de ellos es más potente.

4. Contenido

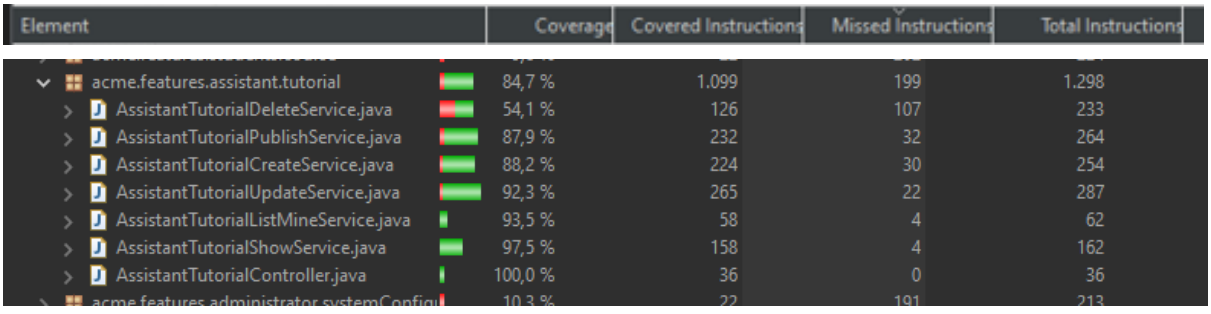
4.1. Testing funcional

Nota: En todos los casos de test, exceptuando los de hacking, se comprueba que la operación correspondiente se haya realizado correctamente.

4.1.1. Tests de Tutorial:

1. Create test.
    a. 100 positive: Como un asistente, crea tutorías con datos válidos.

   b. 200 negative: Como un asistente, crea tutorías con datos no válidos.
   c. 300 hacking: Intenta crear tutorías con un usuario que no es un asistente.
2. Delete test.
   a. 100 positive: Como un asistente, borra una tutoría.
   b. 200 negative: No necesario dado que no hay un formulario involucrado.
   c. 300 hacking: Intenta borrar una tutoría con un usuario que no es un asistente, ó con un asistente que no es dueño de la tutoría.
   d. 301 hacking: Intenta borrar una tutoría ya publicada.
3. List mine test.
   a. 100 positive: Como un asistente, lista sus tutorías.
   b. 200 negative: No necesario dado que no hay un formulario involucrado.
   c. 300 hacking: Intenta listar las tutorías de usuarios que no son asistentes.
4. Publish test.
   a. 100 positive: Como un asistente, publica una tutoría.
   b. 200 negative: Como un asistente, intenta publicar una tutoría sin sesiones.
   c. 300 hacking: Intenta publicar una tutoría con un usuario que no es un asistente.
   d. 301 hacking: Como un asistente, intenta publicar una tutoría ya publicada de ese asistente.
   e. 302 hacking: Como un asistente, intenta publicar tutorías que no pertenecen a ese asistente, estuvieran publicadas o sin publicar.
5. Show test.
   a. 100 positive: Como un asistente, muestra una tutoría.
   b. 200 negative: No necesario dado que no hay un formulario involucrado.
   c. 300 hacking: Intenta mostrar una tutoría no publicada como un usuario que no es un asistente ó un asistente que no es dueño de la tutoría.
6. Update test.
   a. 100 positive: Como un asistente, actualiza una tutoría con datos válidos.
   b. 200 negative: Como un asistente, actualiza una tutoría con datos no válidos.
   c. 300 hacking: Intenta actualizar una tutoría como un usuario que no es un asistente ó un asistente que no es dueño de la tutoría.
   d. 301 hacking: Como un asistente, intenta actualizar una tutoría ya publicada.



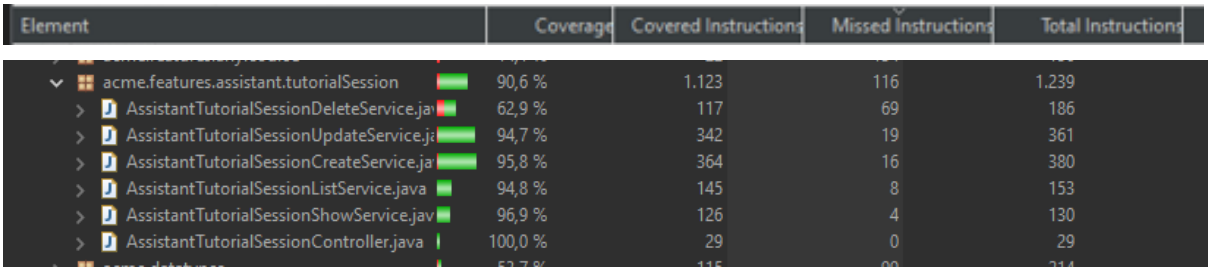
Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
acme.features.assistant.tutorial	84,7 %	1.099	199	1.298
AssistantTutorialDeleteService.java	54,1 %	126	107	233
AssistantTutorialPublishService.java	87,9 %	232	32	264
AssistantTutorialCreateService.java	88,2 %	224	30	254
AssistantTutorialUpdateService.java	92,3 %	265	22	287
AssistantTutorialListMineService.java	93,5 %	58	4	62
AssistantTutorialShowService.java	97,5 %	158	4	162
AssistantTutorialController.java	100,0 %	36	0	36

Como podemos ver aquí, la efectividad de los casos de test al probar las clases referentes a las features para las que fueron creados es buena. Se ha probado la mayoría de las instrucciones de esas clases, exceptuando el servicio de borrado, para el cuál es imposible comprobar el método unbind, al no tener la posibilidad de devolver errores de validación, único caso en el que se usaría dicho método.

4.1.2. Tests de TutorialSession:

1. Create test.
    a. 100 positive: Como un asistente, crea sesiones de tutoría con datos válidos.
    b. 200 negative: Como un asistente, crea sesiones de tutoría con datos no válidos.
    c. 300 hacking: Intenta crear sesiones de tutoría con un usuario que no es un asistente.
    d. 301 hacking: Intenta crear sesiones de tutoría para una tutoría ya publicada.
    e. 302 hacking: Intenta crear sesiones de tutoría para una tutoría que no pertenece al asistente, ya sea publicada o no.

2. Delete test.
    a. 100 positive: Como un asistente, borra una sesión de tutoría.
    b. 200 negative: No necesario dado que no hay un formulario involucrado.
    c. 300 hacking: Intenta borrar una sesión de tutoría con un usuario que no es un asistente, ó con un asistente que no es dueño de la tutoría.
    d. 301 hacking: Intenta borrar una tutoría ya publicada.
    e. 302 hacking: Intenta borrar sesiones de tutoría de una tutoría que no pertenece al asistente, ya sea publicada o no.
3. List test.
    a. 100 positive: Como un asistente, lista las sesiones de sus tutorías.
    b. 200 negative: No necesario dado que no hay un formulario involucrado.
    c. 300 hacking: Intenta listar las sesiones de tutoría de usuarios que no son asistentes.
4. Show test.
    a. 100 positive: Como un asistente, muestra sus sesiones de tutoría.
    b. 200 negative: No necesario dado que no hay un formulario involucrado.
    c. 300 hacking: Intenta mostrar una sesión de tutoría de una tutoría no publicada como un usuario que no es un asistente ó un asistente que no es dueño de la tutoría.
5. Update test.
    a. 100 positive: Como un asistente, actualiza una sesión de tutoría con datos válidos.
    b. 200 negative: Como un asistente, actualiza una sesión de tutoría con datos no válidos.
    c. 300 hacking: Intenta actualizar una sesión de tutoría como un usuario que no es un asistente ó un asistente que no es dueño de la tutoría.
    d. 301 hacking: Como un asistente, intenta actualizar una sesión de tutoría de una tutoría ya publicada.
    e. 302 hacking: Como un asistente, intenta actualizar una sesión de tutoría de una tutoría que pertenece a otro asistente.



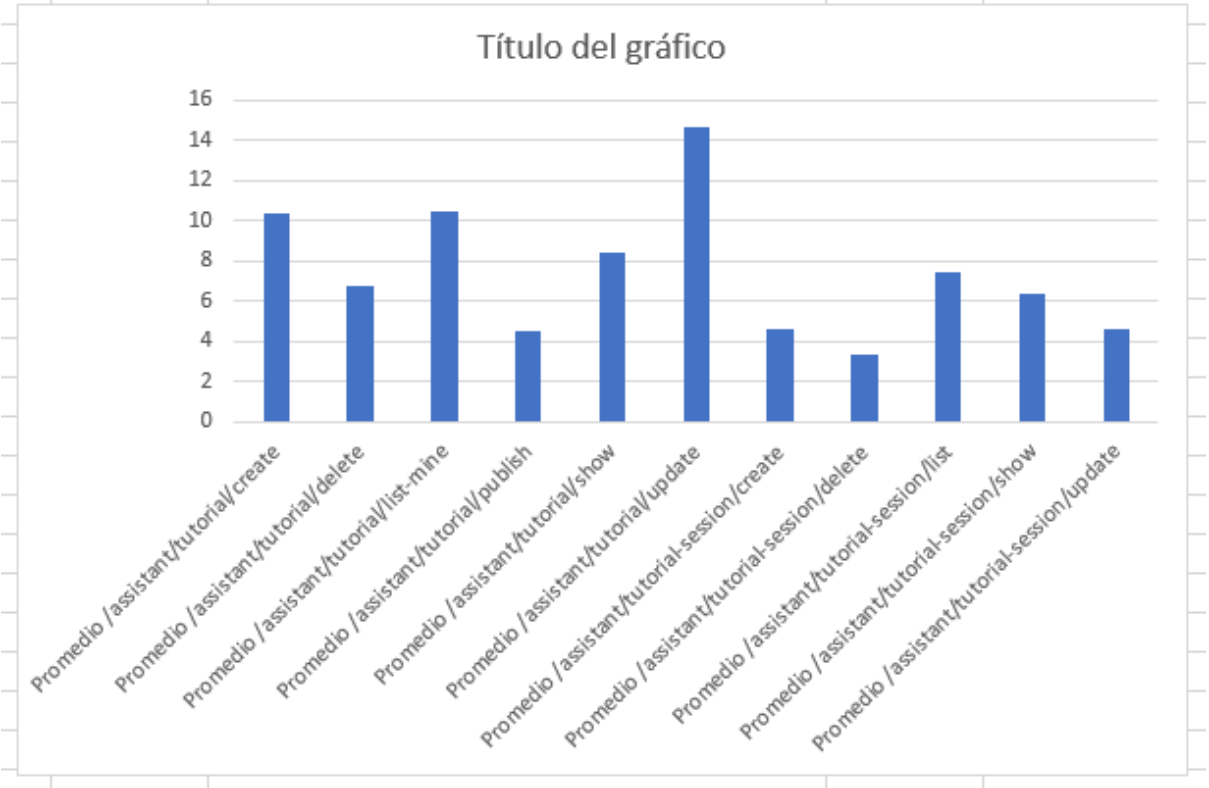
Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
acme.features.assistant.tutorialSession	90,6 %	1.123	116	1.239
AssistantTutorialSessionDeleteService.ja	62,9 %	117	69	186
AssistantTutorialSessionUpdateService.ja	94,7 %	342	19	361
AssistantTutorialSessionCreateService.ja	95,8 %	364	16	380
AssistantTutorialSessionListService.java	94,8 %	145	8	153
AssistantTutorialSessionShowService.jav	96,9 %	126	4	130
AssistantTutorialSessionController.java	100,0 %	29	0	29

Como podemos ver aquí, la efectividad de los casos de test al probar las clases referentes a las features para las que fueron creados es buena. Se ha probado la mayoría de las instrucciones de esas clases, exceptuando el servicio de borrado, para el cuál es imposible comprobar el método unbind, al no tener la posibilidad de devolver errores de validación, único caso en el que se usaría dicho método.

4.2. Testing de rendimiento

4.2.1 Análisis de rendimiento de peticiones

	method	feature	time	status
		Promedio /assistant/tutorial/create	10,3333333	
		Promedio /assistant/tutorial/delete	6,8125	
		Promedio /assistant/tutorial/list-mine	10,4672619	
		Promedio /assistant/tutorial/publish	4,52941176	
		Promedio /assistant/tutorial/show	8,42465753	
		Promedio /assistant/tutorial/update	14,6470588	
		Promedio /assistant/tutorial-session/create	4,67092652	
		Promedio /assistant/tutorial-session/delete	3,3125	
		Promedio /assistant/tutorial-session/list	7,4744186	
		Promedio /assistant/tutorial-session/show	6,35483871	
		Promedio /assistant/tutorial-session/update	4,58823529	
		Promedio general	7,78663793	



Como podemos ver se ha conseguido un promedio general de 7.78, un valor aparentemente bueno. En el gráfico podemos ver que las características de creación y edición de los tutoriales han sido las predominantes en cuanto a media de tiempo, siendo el valor más alto 14,647 para las actualizaciones.

4.2.2 Análisis del rendimiento de tests

Se ha conseguido un promedio general de 8822,35. Además podemos ver en el gráfico que todos los tests tardan entre 0 y 45 segundos.

4.2.3 Análisis del intervalo de confianza

4.2.3.1 Análisis PC 1

Columna1	
Media	7,7839196
Error típico	0,14892899
Mediana	8
Moda	8
Desviación estándar	5,55846407
Varianza de la muestra	30,8965228
Curtosis	96,2947958
Coeficiente de asimetría	7,0606991
Rango	104
Mínimo	1
Máximo	105
Suma	10843
Cuenta	1393
Nivel de confianza(95,0%)	0,29214948

Interval(ms)	7,491770117	8,07606908
Interval(s)	0,00749177	0,00807607

4.2.3.2 Análisis PC 2

Columna1	
Media	8,45513281
Error típico	0,16912212
Mediana	8
Moda	9
Desviación estándar	6,3121306
Varianza de la muestra	39,8429927
Curtosis	91,4595563
Coeficiente de asimetría	6,80022404
Rango	117
Mínimo	1
Máximo	118
Suma	11778
Cuenta	1393
Nivel de confianza(95,0%)	0,33176173

Interval(ms)	8,12337107	8,78689454
Interval(s)	0,00812337	0,00878689

4.2.4 Hypothesis contrast

Prueba z para medias de dos muestras		
	PC1	*PC2*
Media	7,7839196	8,45513281
Varianza (conocida)	30,8965228	39,8429927
Observaciones	1393	1393
Diferencia hipotética de las medias	0	
z	-2,9785508	
P(Z<=z) una cola	0,00144808	
Valor crítico de z (una cola)	1,64485363	
P(Z<=z) dos colas	0,00289615	
Valor crítico de z (dos colas)	1,95996398	

Tras realizar un estudio de Z-test (En el que se ha usado PC1 como antes y PC2 como después), podemos notar que el two-tail p-value, es aproximadamente 0 '0029. Dado que este valor está entre 0 y 1-0’95, se pueden comparar las medias de ambos conjuntos de datos para evaluar la diferencia de rendimiento.
Como la media de PC2 es significativamente mayor a la de PC1, podemos determinar que el PC2 ha conseguido un menor rendimiento al ejecutar los tests.

5. Conclusiones

Durante el sprint 4, el estudiante ha realizado una test suite para probar formalmente que los requisitos del cliente se han cumplido correctamente. Para ello ha estructurado las test classes por características y ha creado test cases para las tres tipos de comprobaciones principales que se deben hacer en un sistema. Estas serían las positivas nombradas como test1XXpositive, las negativas, test2XXnegative y las de hacking, test3XXhacking. Con esto se ha conseguido reducir la probabilidad de que existan errores o bugs dentro del código desarrollado y lo más probable es que nos hayamos asegurado de que el cliente no va a tener que pedir ningún mantenimiento para arreglar errores que pudiera haber encontrado de otra forma.

Finalmente, el estudiante ha usado las herramientas proporcionadas para evaluar el rendimiento del sistema desarrollado. Esto ha implicado recoger los datos obtenidos por el framework de desarrollo, analizarlos con herramientas estadísticas y proveer observaciones respecto a los resultados. Todo con el fin de asegurar que el cliente recibe una aplicación eficiente y quede demostrado que ésta opera dentro de los valores de rendimiento provistos en los requisitos no funcionales, si los hubiera dentro del marco de trabajo de Acme L3.

6. Anexos

Intencionalmente en blanco.

7. Bibliografía

Intencionalmente en blanco.