

DP2 2022-2023
Documento de Testing Grupal

ACME-L3-D04

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

- Miembros:

GRUPO C1.04.09

Fernández Castillo, Javier (javidelatorre400@gmail.com)

Domínguez Ruíz, Andrés (andresdominguezruiz.ad@gmail.com)

Otero Barbasán, Manuel (m.otero.barbasan@gmail.com)

Urquijo Martínez, Álvaro (alvarourma@gmail.com)

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

1.0.0

Índice

1. Resumen ejecutivo	3
2. Historial de versiones	4
3. Introducción	4
4. Contenido	6
4.1. Testing funcional	6
4.1.1. Tests de Peep:	6
4.2. Testing de rendimiento	8
4.2.1 Análisis de rendimiento de peticiones	8
4.2.2 Análisis del rendimiento de tests	9
4.2.3 Análisis del intervalo de confianza	10
4.2.3.1 Análisis PC 1	10
4.2.3.2 Análisis PC 2	10
4.2.4 Hypothesis contrast	11
5. Conclusiones	12
6. Anexos	13
7. Bibliografía	14

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	24-05-2023	<ul style="list-style-type: none">Creación del documento	4

3. Introducción

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

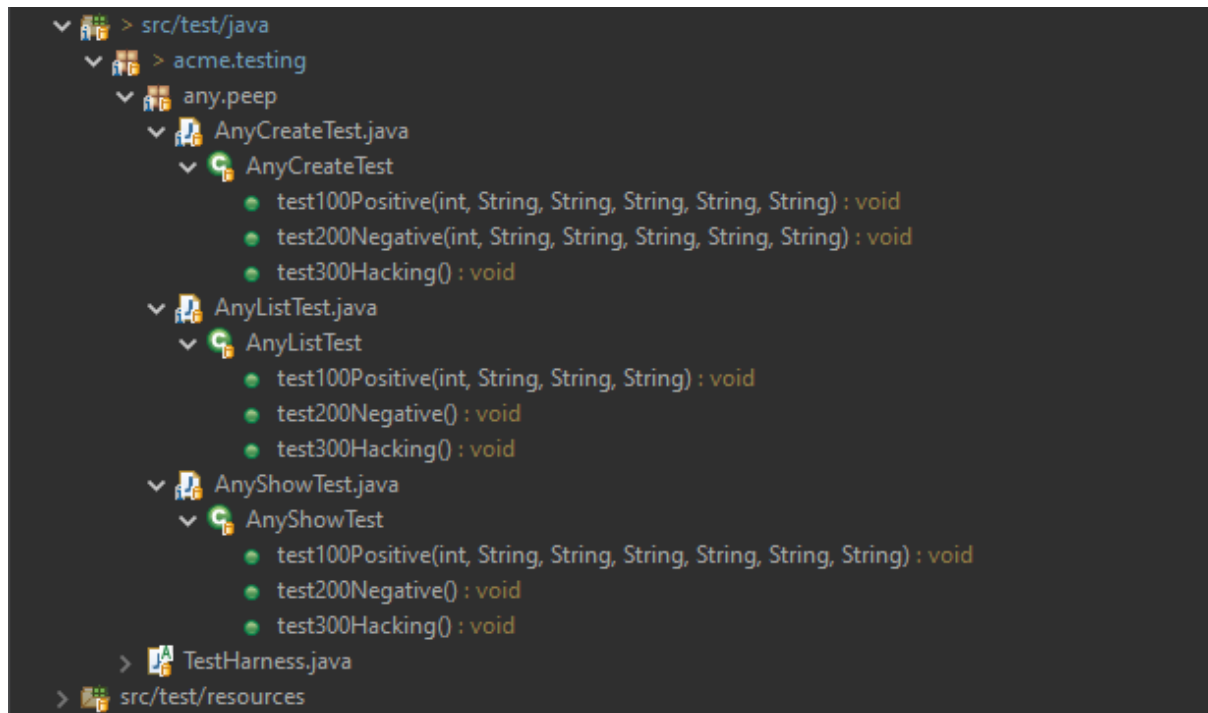
En primer lugar se mostrará el testing funcional. Se agruparán los casos de test realizados por características y se proveerá una breve descripción. Finalmente se evaluará la efectividad de cada caso de test. Para ello se usará la siguiente métrica: porcentaje de líneas de código evaluadas. Cuantas más 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 Peep:

1. Create test.
 - a. 100 positive: Crea peeps con datos correctos.
 - b. 200 negative: Intenta crear peeps con datos no válidos.
 - c. 300 hacking: No hay tests de hacking puesto que es una característica que cualquiera puede acceder, independientemente de registro o rol.
2. List test.
 - a. 100 positive: Lista todos los peeps.
 - b. 200 negative: No necesario dado que no hay un formulario involucrado.
 - c. 300 hacking: No hay tests de hacking puesto que es una característica que cualquiera puede acceder, independientemente de registro o rol.
3. Show test.
 - a. 100 positive: Lista todos los peeps, muestra los detalles de uno y comprueba que los valores sean los esperados.
 - b. 200 negative: No necesario dado que no hay un formulario involucrado.
 - c. 300 hacking: No hay tests de hacking puesto que es una característica que cualquiera puede acceder, independientemente de registro o rol.

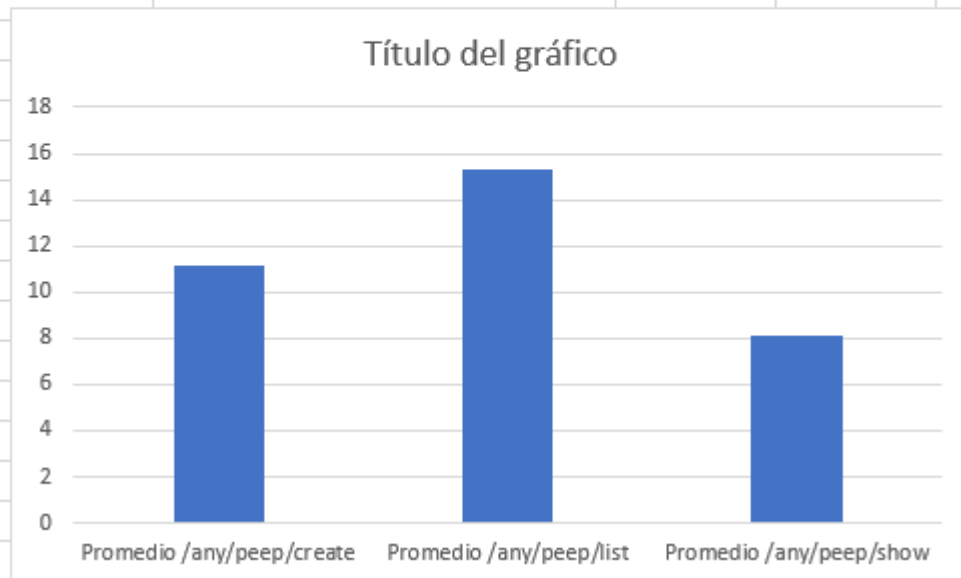
Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
acme.features.any.peep	 86,6 %	277	43	320
> AnyPeepCreateService.java	 79,0 %	132	35	167
> AnyPeepListService.java	 92,5 %	49	4	53
> AnyPeepShowService.java	 95,1 %	77	4	81
> AnyPeepController.java	 100,0 %	19	0	19

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.

4.2. Testing de rendimiento

4.2.1 Análisis de rendimiento de peticiones

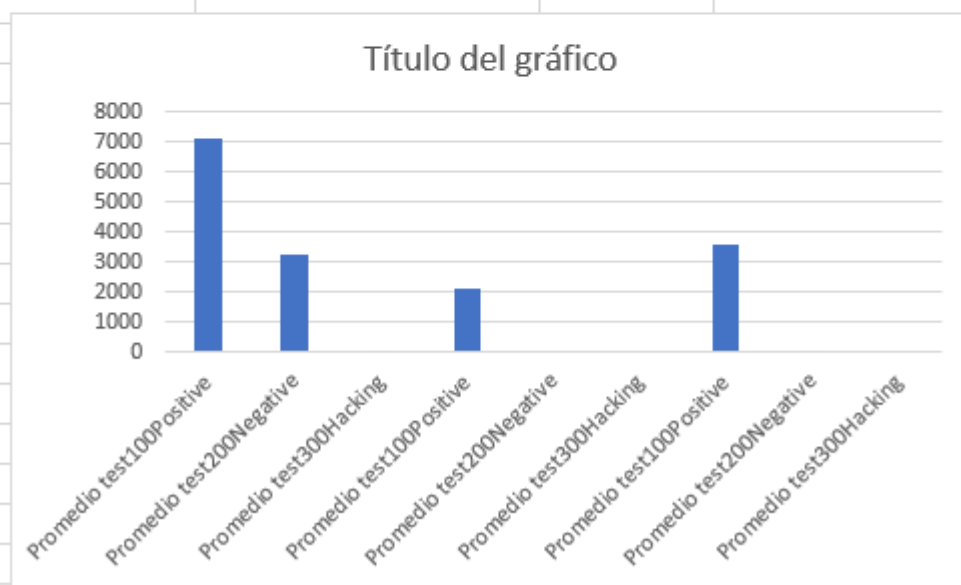
method	feature	time	status
	Promedio /any/peep/create	11,1730769	
	Promedio /any/peep/list	15,3235294	
	Promedio /any/peep/show	8,12903226	
	Promedio general	12,7931034	



Como podemos ver se ha conseguido un promedio general de 12,79, un valor aparentemente bueno. En el gráfico podemos ver que la característica que más tarda de media es el listado con 15,32.

4.2.2 Análisis del rendimiento de tests

test-method	time	description
Promedio test100Positive	7094,227273	
Promedio test200Negative	3240,625	
Promedio test300Hacking	1	
Promedio test100Positive	2107,040816	
Promedio test200Negative	0	
Promedio test300Hacking	0	
Promedio test100Positive	3596,530612	
Promedio test200Negative	1	
Promedio test300Hacking	0	
Promedio general	3984,180645	



Se ha conseguido un promedio general de 3984,53. Además podemos ver en el gráfico que todos los tests tardan de media entre 0 y 7 segundos.

4.2.3 Análisis del intervalo de confianza

4.2.3.1 Análisis PC 1

Columna1					
Media	12,79310345		Intervalo(ms)	13,4658599	12,120347
Error típico	0,342292178		Intervalo(s)	0,01346586	0,01212035
Mediana	13				
Moda	15				
Desviación estándar	7,139069391				
Varianza de la muestra	50,96631177				
Curtosis	67,74666742				
Coefficiente de asimetría	6,242125041				
Rango	99				
Mínimo	4				
Máximo	103				
Suma	5565				
Cuenta	435				
Nivel de confianza(95,0%)	0,672756472				

4.2.3.2 Análisis PC 2

Columna1					
Media	11,4		Intervalo(ms)	12,0316095	10,7683905
Error típico	0,321356967		Intervalo(s)	0,01203161	0,01076839
Mediana	11				
Moda	13				
Desviación estándar	6,70243095				
Varianza de la muestra	44,92258065				
Curtosis	70,33550429				
Coefficiente de asimetría	6,387411788				
Rango	94				
Mínimo	3				
Máximo	97				
Suma	4959				
Cuenta	435				
Nivel de confianza(95,0%)	0,631609465				

4.2.4 Hypothesis contrast

Prueba z para medias de dos muestras		
	PC1	PC2
Media	12,7931034	11,4
Varianza (conocida)	50,9663118	44,9225806
Observaciones	435	435
Diferencia hipotética de las medias	0	
z	2,96717967	
P(Z<=z) una cola	0,00150273	
Valor crítico de z (una cola)	1,64485363	
P(Z<=z) dos colas	0,00300545	
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 '0030. 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 menor a la de PC1, podemos determinar que el PC2 ha conseguido un mayor rendimiento al ejecutar los tests.

5. Conclusiones

Durante el sprint 4, el grupo 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 grupo 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.