

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

D04 – Testing Report



Grado en Ingeniería Informática – Ingeniería del Software

Diseño y Pruebas II

Curso 2023 – 2024

Fecha	Versión
27/05/2024	v1r1

Grupo de prácticas: C1.002	
Autores	Correo Corporativo
José Ramón Baños Botón	josbanbot@alum.us.es

Link repositorio: <https://github.com/Joserra24/Acme-SF-D04-24.4.0>



Índice

1. Tabla de versiones	3
2. Resumen ejecutivo	4
3. Introducción	5
4. Functional testing.....	6
5. Performance testing.....	10
6. Conclusiones	13
7. Bibliografía	13



1. *Tabla de versiones*

Fecha	Versión	Descripción
26/05/2024	v1r0	Creación del documento
27/05/2024	v1r1	Entrega 1



2. Resumen ejecutivo

Para este documento de testing se han desarrollado y explicado cada una de las implementaciones presentes en los requisitos obligatorios de la entrega D04.

Se han evaluado tanto el rendimiento del sistema como el desempeño funcional de todas y cada una de las funciones solicitadas, en este caso, las relativas a “training module” y a “training session”. Para ello se siguió la metodología proporcionada en “S01 - Formal testing” y “S02 - Performance testing”.

El sistema muestra un comportamiento robusto en términos de funcionalidad pese a haber áreas que podrían precisar de una ligera toma de atención.



3. Introducción

Este documento se divide en dos secciones distintas:

1. Functional testing: se presentará un listado con los casos de prueba implementados, agrupados por funcionalidad. Por cada uno se dará una descripción y una indicación de cuan efectivo es detectando errores. Para la efectividad, se usará el coverage del código para comprobar que se han probado todas las decisiones posibles durante la ejecución del programa y así evitar la existencia de errores.
2. Performance testing: se proporcionarán los gráficos adecuados y un intervalo de confianza del 95% para el tiempo tomado para las solicitudes en las pruebas en dos ordenadores distintos. Además, tras las pruebas en los diferentes ordenadores, se indicará cual de estos es el más potente y ofrece mejor rendimiento.



4. Functional testing

Casos de pruebas relativos a training module:

- Create training module:
 - Descripción: Se prueban las restricciones de todos los campos del formulario de creación de un training module con valores relativos a casos positivos, negativos y de hacking.
 - Coverage: 93.4%
 - Efectividad: Entendemos que un coverage alrededor del 95% es un valor alto de efectividad. Vemos que lo que baja el coverage son las líneas de “assert object != null” que están en amarillo. Esto nos indica que, en ningún caso al realizar la función en la que se encuentra esta línea de código, se ha obtenido un objeto nulo.
- Delete training module:
 - Descripción: Se prueba la eliminación de un training module teniendo en cuenta restricciones como que no se podrá eliminar un training module si tiene asociado algún training session que esté publicado. Se valida que un developer que no es propietario de un training module ajeno no tenga acceso a esta funcionalidad.
 - Coverage: 93.9%
 - Efectividad: Alta. Misma situación con “assert object != null”. Junto a esto, en la autorización, observamos en amarillo la línea “manager = object == null ? null : object.getManager()”, la cual se encuentra en este estado porque en las pruebas siempre se obtiene que el objeto cliente es distinto de nulo. También encontramos que la línea “status = object != null && ...” está en amarillo. Esto se debe a que en las pruebas siempre se obtiene un proyecto, por lo que nunca es nula esta variable y por ende esta primera condición siempre es verdadera.
- List my training modules:
 - Descripción: Se prueba el listado de training modules pertenecientes al cliente que ha iniciado sesión en el sistema. Se valida que un developer que no es propietario de un training module ajeno no tenga acceso a esta funcionalidad.
 - Coverage: 94.9%
 - Efectividad: Alta. Misma situación con “assert object != null”.
- Publish training module:
 - Descripción: Se prueba la publicación de un training module teniendo en cuenta restricciones como que no se podrá publicar un training module si tiene al menos un training session sin publicar o si no tiene training sessions asociados. Se valida que un developer que no es propietario de un training module ajeno no tenga acceso a esta funcionalidad.
 - Coverage: 93.5%
 - Efectividad: Alta. Misma situación con “assert object != null”, “developer= object == null ? null : object.getDeveloper()” y “status = object != null && ...”. Además de esto,



el if de la línea de la validación que comprueba que no existan training session sin publicar está en amarillo. Esto puede ser debido a un problema de Eclipse, ya que están probados tanto los casos en los que se quiere publicar un contrato, tanto con registros de progreso sin publicar, como con y sin ellos.

- Show training module details
 - Descripción: Se prueba la muestra de detalles de un training module del que el developer es propietario. Se valida que un developer que no es propietario de un training module ajeno no tenga acceso a esta funcionalidad.
 - Coverage: 96.0%
 - Efectividad: Alta. Misma situación con “assert object != null” y “status = object != null && ...”.
- Update training module:
 - Descripción: Se prueban las restricciones de todos los campos del formulario de actualización de un training module con valores relativos a casos positivos, negativos y de hacking. Se valida que un developer que no es propietario de un training module ajeno no tenga acceso a esta funcionalidad.
 - Coverage: 93.2%
 - Efectividad: Alta. Misma situación con “assert object != null” y “status = object != null && ...”.

▼	acme.features.developer.trainingModule	94,0 %
>	DeveloperTrainingModuleController.java	100,0 %
>	DeveloperTrainingModuleListMineService.java	94,9 %
>	DeveloperTrainingModuleShowService.java	96,0 %
>	DeveloperTrainingModuleCreateService.java	93,4 %
>	DeveloperTrainingModuleUpdateService.java	93,2 %
>	DeveloperTrainingModulePublishService.java	93,5 %
>	DeveloperTrainingModuleDeleteService.java	93,9 %

Casos de pruebas relativos a registros de training session:

- Create training session:
 - Descripción: Se prueban las restricciones todos los campos del formulario de creación de un training session con valores relativos a casos positivos, negativos y de hacking.
 - Coverage: 94.5%
 - Efectividad: Alta. Misma situación con “assert object != null” y “status = object != null && ...”.



- Delete training session:
 - Descripción: Se prueba la eliminación de un training session. Se valida que un developer que no es propietario de un training session ajeno no tenga acceso a esta funcionalidad.
 - Coverage: 57.3%
 - Efectividad: Alta. Misma situación con “assert object != null” y “status = object != null && ...”.
- List my training sessions:
 - Descripción: Se prueba el listado de training sessions pertenecientes a un training module que pertenece a su vez al developer que ha iniciado sesión en el sistema. Se valida que un developer que no es propietario de un training session ajeno no tenga acceso a esta funcionalidad.
 - Coverage: 95.7%
 - Efectividad: Alta. Misma situación con “assert object != null” y “status = object != null && ...”.
- Publish training session:
 - Descripción: Se prueba la publicación de un training session. Se valida que un developer que no es propietario de un training session ajeno no tenga acceso a esta funcionalidad.
 - Coverage: 57.0%
 - Efectividad: Alta. Misma situación con “assert object != null” y “status = object != null && ...”.
- Show training session details:
 - Descripción: Se prueba la muestra de detalles de un training session del que el developer es propietario. Se valida que un developer que no es propietario de un training session ajeno no tenga acceso a esta funcionalidad.
 - Coverage: 95.5%
 - Efectividad: Alta. Misma situación con “assert object != null” y “status = object != null && ...”.
- Update training session:
 - Descripción: Se prueban las restricciones de todos los campos del formulario de actualización de un training session con valores relativos a casos positivos, negativos y de hacking. Se valida que un developer que no es propietario de un training session ajeno no tenga acceso a esta funcionalidad.
 - Coverage: 94.1%
 - Efectividad: Alta. Misma situación con “assert object != null” y “status = object != null && ...”.



> acme.features.developer.trainingSession	84,5 %
> DeveloperTrainingSessionController.java	100,0 %
> DeveloperTrainingSessionListAllService.java	95,2 %
> DeveloperTrainingSessionListMineService.java	95,7 %
> DeveloperTrainingSessionShowService.java	95,5 %
> DeveloperTrainingSessionDeleteService.java	57,3 %
> DeveloperTrainingSessionPublishService.java	57,0 %
> DeveloperTrainingSessionCreateService.java	94,5 %
> DeveloperTrainingSessionUpdateService.java	94,1 %
< acme.features.sponsor.sponsorship	2,7 %

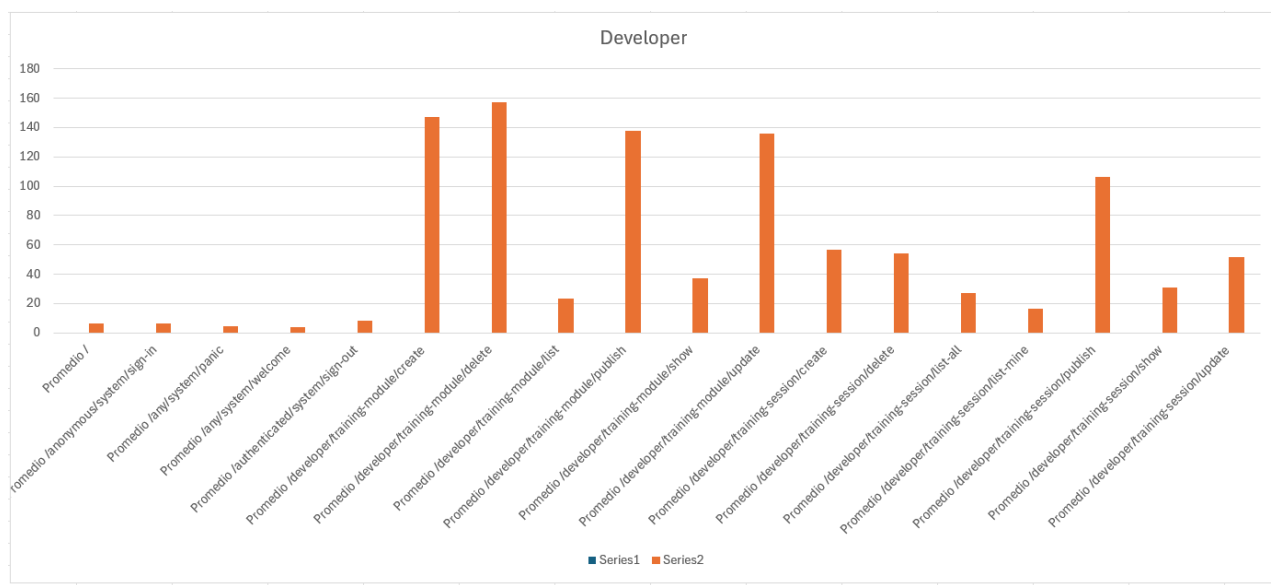
En resumen, lo único rojo que se puede apreciar es el método unbind de algunas clases, ya que éste no se acaba ejecutando en ningún momento, puesto que es imposible entrar en esa condición. Por otra parte, las líneas en amarillo ya se han explicado el motivo de su causa. Por todo esto y por tener un coverage del 94.0% de media en training module y de 84.5% de media en training session, se considera que la existencia de potenciales fallos o bugs es ínfima.



5. Performance testing

Tras realizar el conjunto de tests para las funcionalidades oportunas, se han realizado todos los pasos que se muestran en "S02 - Performance testing", obteniendo los siguientes resultados:

Promedio /	6.31419053
Promedio /anonymous/system/sign-in	6.71483723
Promedio /any/system/panic	4.70468333
Promedio /any/system/welcome	3.84573824
Promedio /authenticated/system/sign-out	8.04291429
Promedio /developer/training-module/create	147.393979
Promedio /developer/training-module/delete	157.439814
Promedio /developer/training-module/list	23.3209563
Promedio /developer/training-module/publish	137.8864
Promedio /developer/training-module/show	37.1071188
Promedio /developer/training-module/update	135.628724
Promedio /developer/training-session/create	56.8289568
Promedio /developer/training-session/delete	54.4246667
Promedio /developer/training-session/list-all	27.1791
Promedio /developer/training-session/list-mine	16.35995
Promedio /developer/training-session/publish	106.6224
Promedio /developer/training-session/show	30.7439182
Promedio /developer/training-session/update	51.7344879
Promedio general	33.8968493





Como se observa en las imágenes anteriores, el tiempo promedio que tarda el sistema en realizar una petición es de aproximadamente 33,9 ms, es decir, 0,033 segundos, lo cual es bastante rápido.

Además, el gráfico de barras muestra claramente que las peticiones que requieren más tiempo son aquellas que manejan una mayor cantidad de datos y validaciones, específicamente las relacionadas con la creación, edición, eliminación y publicación de un training module. Esto se debe, en parte, a que se deben verificar varias validaciones tanto de la clase training module como de la clase asociada TrainingSession.

Posteriormente, se añadieron los índices necesarios para mejorar el rendimiento de las consultas SQL. Se realizan las comparativas oportunas de los distintos valores obtenidos en cada prueba y obtenemos:

<i>before</i>			<i>after</i>		
Media	34.9787068		Media	17.4426404	
Error típico	2.37789433		Error típico	1.06713859	
Mediana	7.1621		Mediana	4.4433	
Moda	#N/D		Moda	1.5147	
Desviación estándar	53.96301		Desviación estándar	24.2172285	
Varianza de la muestra	2912.00645		Varianza de la muestra	586.474158	
Curtosis	7.79447253		Curtosis	5.33751623	
Coeficiente de asimetría	2.4322139		Coeficiente de asimetría	1.98099715	
Rango	429.4756		Rango	181.8133	
Mínimo	1.4053		Mínimo	1.1822	
Máximo	430.8809		Máximo	182.9955	
Suma	18014.034		Suma	8982.9598	
Cuenta	515		Cuenta	515	
Nivel de confianza(95.0%)	4.67158743		Nivel de confianza(95.0%)	2.0964898	
interval(ms)	30.3071194	39.6502942	interval(ms)	15.3461506	19.5391302
interval (s)	0.03030712	0.03965029	interval (s)	0.01534615	0.01953913

Para determinar que los promedios de los tiempos antes y después de los cambios puedan ser considerados los mismos o no, se ha realizado un z-test con los siguientes resultados:



Prueba z para medias de dos muestras		
	<i>before</i>	<i>after</i>
Media	34.9787068	17.4426404
Varianza (conocida)	2912	586
Observaciones	515	515
Diferencia hipotética de las medias	0	
z	6.72861832	
P(Z<=z) una cola	8.564E-12	
Valor crítico de z (una cola)	1.64485363	
Valor crítico de z (dos colas)	1.7128E-11	
Valor crítico de z (dos colas)	1.95996398	

Ante este resultado del two-tail p-value (0), podemos concluir que, al estar en el intervalo que se encuentra entre [0, 0.05), los cambios realizados han sido fructíferos y han ayudado a mejorar el rendimiento.

Ahora se comparará el rendimiento del sistema en dos computadoras distintas. La primera computadora será en la que se han realizado todas las pruebas anteriores y la segunda será la computadora de otro miembro del equipo. Estos son los resultados:

			Computador 2		
Computador 1					
Media	17.4426404		Media	353.942602	
Error típico	1.06713859		Error típico	14.9956934	
Mediana	4.4433		Mediana	358.038	
Moda	1.5147		Moda	#N/D	
Desviación estándar	24.2172285		Desviación estándar	217.825029	
Varianza de la muestra	586.474158		Varianza de la muestra	47447.7434	
Curtosis	5.33751623		Curtosis	10.3791617	
Coefficiente de asimetría	1.98099715		Coefficiente de asimetría	1.83591944	
Rango	181.8133		Rango	1820.621	
Mínimo	1.1822		Mínimo	67.535	
Máximo	182.9955		Máximo	1888.156	
Suma	8982.9598		Suma	74681.889	
Cuenta	515		Cuenta	211	
Nivel de confianza(95.0%)	2.0964898		Nivel de confianza(95.0%)	29.5613823	
2 interval(ms)	15.3461506	19.5391302	interval(ms)	324.38122	383.503984
3 interval(s)	0.01534615	0.01953913	interval(s)	0.32438122	0.38350398

Podemos observar la diferencia de resultados.



Prueba z para medias de dos muestras		
	67.4952	394.166
Media	17.34526187	353.751062
Varianza (con	586	47447
Observacione	514	210
Diferencia hip	0	
z	-22.32419355	
P(Z<=z) una co	0	
Valor crítico d	1.644853627	
Valor crítico d	0	
Valor crítico d	1.959963985	

Como se ve, el valor del two-tail p-value es 0, un valor muy alejado de alpha.

6. Conclusiones

Después de crear este documento sobre pruebas, se ha concluido que esta fase del ciclo de vida de un proyecto es vital. Verificar que todas las funciones desarrolladas funcionen correctamente y se revisen minuciosamente para minimizar errores o fallos, además de asegurar que el rendimiento esté optimizado al máximo, son aspectos fundamentales para el cliente. Un sistema exhaustivamente probado permite que el usuario final lo utilice de manera rápida e intuitiva, evitando problemas que puedan perjudicar su experiencia. Además, un proceso de pruebas meticuloso contribuye a la satisfacción del cliente y a la buena reputación del producto, garantizando que las expectativas de calidad y eficiencia se cumplan consistentemente.

7. Bibliografía

- 08 Annexes – Material proporcionado en la asignatura *Diseño y Pruebas II* por la Universidad de Sevilla.
- L04 - S01 - Formal testing - Material proporcionado en la asignatura *Diseño y Pruebas II* por la Universidad de Sevilla.
- L04 - S02 - Performance testing - Material proporcionado en la asignatura *Diseño y Pruebas II* por la Universidad de Sevilla.



Diseño y Pruebas II

C1.002

Testing report - José Ramón Baños Botón



Diseño y Pruebas II

C1.002

Testing report - José Ramón Baños Botón