Escuela Técnica Superior de Ingeniería Informática

Testing Report D04

Group C2.020 | Diseño y Pruebas II | 07/07/2024

Fecha	Versión	Autor
07/07/2024	1.0.0	Martín Vergara, Jesús

Miembros:

- Student #3: Martín Vergara, Jesús jesmarver@alum.us.es

Repositorio de Github: https://github.com/DP2-C1-020/Acme-SF-D04

Contenido

Resumen ejecutivo

En este documento vamos a proporcionar una vista de lo realizado en el último entregable de la asignatura el cual engloba las pruebas del proyecto. Para esto se han realizado varias pruebas para los requisitos #6 y #7 para los cuales se han tenido que generar más datos de prueba que cubran por completo los rangos de las propiedades de los objetos involucrados en estas características. Esto nos proporciona una idea de la efectividad del proyecto y que aspectos hay que mejorar para que esta efectividad se vea aumentada y consigamos una solución más escalable con un mayor rendimiento.

Introducción

En este informe vamos a detallar las pruebas realizadas para cada funcionalidad y un análisis del desempeño.

Las pruebas han sido realizadas gracias a la nueva versión del framework 24.4.0 el cual cuenta con un launcher que permite guardar las consultas realizadas en cada paso y poder reproducirlas como se explica en el documento S01 - Formal testing. Estos nuevos launchers nos permiten analizar la cobertura de nuestro código pudiendo así mejorarlo para obtener mejores resultados. Gracias a los datos guardados podemos hacer un estudio estadístico de los tiempos empleados en las funcionalidades específicas. Con esto podemos comparar los resultados cuando optimicemos el código y también nos ayuda a comprobar que todas las funcionalidades se desempeñan correctamente.

Contenido

Pruebas funcionales

Para cada característica implementada en el D03 se ha creado un archivo .safe y un archivo .hack. Estos se encuentran dentro del directorio src\test\resources y dentro de la carpeta asociada a la entidad que se prueba. Los archivos .safe contienen las pruebas realizadas en las que se ejecuta las funcionalidades de forma segura, esto es pruebas positivas y negativas del caso de uso. Los archivos .hack contienen las pruebas con intención de utilizar estas funcionalidades de forma ilegal como puede ser acceder a un módulo de entrenamiento sin siquiera estar autenticado o con algún rol que no lo permita.

Estas pruebas son las siguientes:

Módulos de entrenamiento

Operación "create"

Pruebas seguras:

- Se intenta crear un módulo de entrenamiento con todos los campos vacíos.
- Se intenta crear un módulo de entrenamiento con todos los campos completados menos 1 (se van intercambiando en cada intento).
- Se intenta crear un módulo de entrenamiento con valores inválidos.
- Se crea un módulo de entrenamiento con todos los campos correctos.

Pruebas de hacking:

- No se han encontrado formas de probar esta funcionalidad.

Operación "delete"

Pruebas seguras:

- Se elimina un módulo de entrenamiento correctamente.

Pruebas de hacking:

- Se intenta eliminar un módulo de entrenamiento publicado. Para esto, se añade un botón de eliminar desde el inspeccionar elemento de Firefox.

Operación "list-mine"

Pruebas seguras:

- Se accede al listado de módulos de entrenamiento correctamente.

Pruebas de hacking:

- Se intenta acceder al listado de módulos de entrenamiento con un rol no válido.

Operación "publish"

Pruebas seguras:

- Se intenta publicar un módulo de entrenamiento que no tiene sesiones de entrenamiento.
- Se intenta publicar un módulo de entrenamiento que no tiene sesiones de entrenamiento publicadas.
- Se intenta publicar un módulo de entrenamiento con un código ya existente.
- Se publica un módulo de entrenamiento correctamente.

Pruebas de hacking:

- Se intenta publicar un módulo de entrenamiento que ya está publicado. Para esto, se añade un botón de publicar desde el inspeccionar elemento de Firefox.

Operación "show"

Pruebas seguras:

- Se accede a los detalles de un módulo de entrenamiento correctamente.

Pruebas de hacking:

- Se intenta acceder a los detalles de un módulo de entrenamiento con un rol no válido.
- Se intenta acceder a los detalles de un módulo de entrenamiento que no es del usuario.

Operación "update"

Pruebas seguras:

- Se intenta actualizar un módulo de entrenamiento con un código ya existente.
- Se intenta actualizar un módulo de entrenamiento con todos los campos completados menos 1 (se van intercambiando en cada intento).
- Se intenta actualizar un módulo de entrenamiento con todos los campos vacíos.
- Se intenta actualizar un módulo de entrenamiento con campos inválidos.
- Se actualiza un módulo de entrenamiento correctamente.

Pruebas de hacking:

- Se intenta actualizar un módulo de entrenamiento publicado. Para esto, se añade un botón de eliminar desde el inspeccionar elemento de Firefox.

Sesiones de entrenamiento

Operación "create"

Pruebas seguras:

- Se intenta crear una sesión de entrenamiento con todos los campos vacíos.
- Se intenta crear una sesión de entrenamiento con todos los campos completados menos 1 (se van intercambiando en cada intento).
- Se intenta crear una sesión de entrenamiento con valores inválidos.
- Se crea una sesión de entrenamiento con todos los campos correctos.

Pruebas de hacking:

- No se han encontrado formas de probar esta funcionalidad.

Operación "delete"

Pruebas seguras:

- Se elimina una sesión de entrenamiento correctamente.

Pruebas de hacking:

- Se intenta eliminar una sesión de entrenamiento publicada. Para esto, se añade un botón de eliminar desde el inspeccionar elemento de Firefox.

Operación "list-mine"

Pruebas seguras:

- Se accede al listado de sesiones de entrenamiento correctamente.

Pruebas de hacking:

- Se intenta acceder al listado de sesiones de entrenamiento con un rol no válido.
- Se intenta acceder al listado de sesiones de entrenamiento de otro usuario.

Operación "publish"

Pruebas seguras:

- Se intenta publicar una sesión de entrenamiento con un código ya existente.
- Se publica una sesión de entrenamiento correctamente.

Pruebas de hacking:

- Se intenta publicar una sesión de entrenamiento que ya está publicada. Para esto, se añade un botón de publicar desde el inspeccionar elemento de Firefox.

Operación "show"

Pruebas seguras:

- Se accede a los detalles de una sesión de entrenamiento correctamente.

Pruebas de hacking:

- Se intenta acceder a los detalles de una sesión de entrenamiento con un rol no válido.
- Se intenta acceder a los detalles de una sesión de entrenamiento que no es del usuario.

Operación "update"

Pruebas seguras:

- Se intenta actualizar una sesión de entrenamiento con un código ya existente.
- Se intenta actualizar una sesión de entrenamiento con valores inválidos.
- Se intenta actualizar una sesión de entrenamiento con todos los campos completados menos 1 (se van intercambiando en cada intento).
- Se intenta actualizar una sesión de entrenamiento con todos los campos vacíos.
- Se actualiza una sesión de entrenamiento correctamente.

Pruebas de hacking:

- Se intenta actualizar una sesión de entrenamiento publicada. Para esto, se añade un botón de eliminar desde el inspeccionar elemento de Firefox.

Vamos a mostrar una tabla para que se vea los archivos que implica cada característica y como se efectivo han sido los test aplicados. También se va a mostrar la cobertura de código de cada característica.

Ruta	.safe	.hack	Efectividad	Cobertura
src\test\resources\developer\training-modules	Create Delete Show Publish Update	Delete Show Publish Update	Alta, se encontraron vulnerabilidades en las cuales usuarios con el rol de desarrollador podía ejecutar acciones sobre módulos de entrenamiento de otros usuarios.	94,0%
src\test\resources\developer\training-sessions	Create Delete Show Publish Update	Delete Show Publish Update	Alta, se encontraron vulnerabilidades en las cuales usuarios con el rol de desarrollador podía ejecutar acciones sobre sesiones de entrenamiento de otros usuarios.	93,9%

Análisis de desempeño

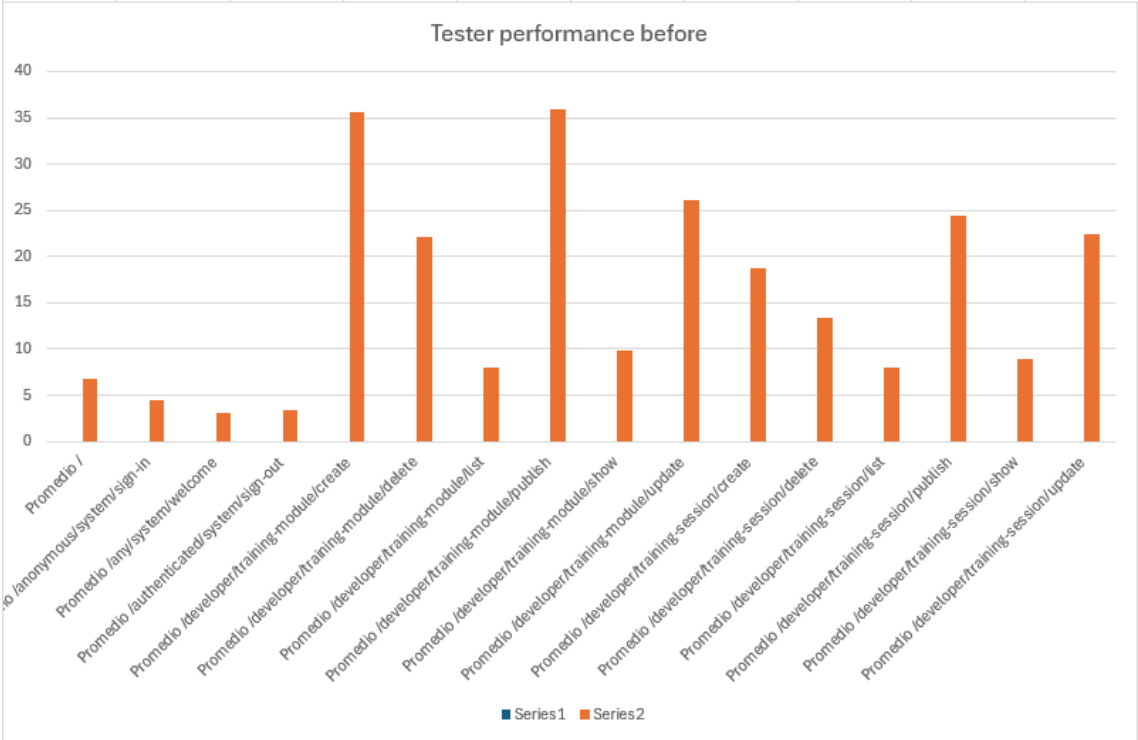
Aquí vamos a analizar la efectividad de los tests realizados, esto es, medimos el tiempo que tarda cada consulta de los tests en ejecutarse y se hace un análisis de los datos extraídos con la ayuda de Excel. Para esto se han seguido los pasos indicados por el documento S02 - Performance testing.

1. Sin optimizer índices

En este caso, ejecutamos todas las pruebas que hemos realizado anteriormente y con los datos generados en el archivo tester.trace podemos hacer un estudio de la efectividad de estos tests.

request-method	request-path	response-status	time
	Promedio /		6,78410882
	Promedio /anonymous/system/sign-in		4,469202
	Promedio /any/system/welcome		3,02453191
	Promedio /authenticated/system/sign-out		3,33013333
	Promedio /developer/training-module/create		35,5871
	Promedio /developer/training-module/delete		22,0836667
	Promedio /developer/training-module/list		7,968775
	Promedio /developer/training-module/publish		35,8793667
	Promedio /developer/training-module/show		9,8372425
	Promedio /developer/training-module/update		26,1041375
	Promedio /developer/training-session/create		18,8141778
	Promedio /developer/training-session/delete		13,292975
	Promedio /developer/training-session/list		7,9736087
	Promedio /developer/training-session/publish		24,3907273
	Promedio /developer/training-session/show		8,95380667
	Promedio /developer/training-session/update		22,3966083
	Promedio general		10,6414595

Como podemos observar, el promedio general es de 10,64 el cual no está mal pero es mejorable optimizando los índices. Esta tabla se puede ver mejor con el siguiente gráfico.



Aquí podemos observar como hay 2 picos los cuales superan el valor 35.
Lo siguiente es analizar los datos e identificar el nivel de confianza que nos da los resultados. Para esto hemos utilizado el complemento de Herramienta para análisis que nos proporciona Excel.

Before		
Media	10,6414595	
Error típico	0,727442577	
Mediana	6,3804	
Moda	3,1313	
Desviación estándar	13,0332052	
Varianza de la muestra	169,8644378	
Curtosis	32,92644923	
Coeficiente de asimetría	4,434276011	
Rango	134,353	
Mínimo	1,299	
Máximo	135,652	
Suma	3415,9085	
Cuenta	321	
Nivel de confianza(95,0%)	1,431174145	
Interval (ms)	9,210285357	12,0726336
Interval (s)	0,009210285	0,01207263

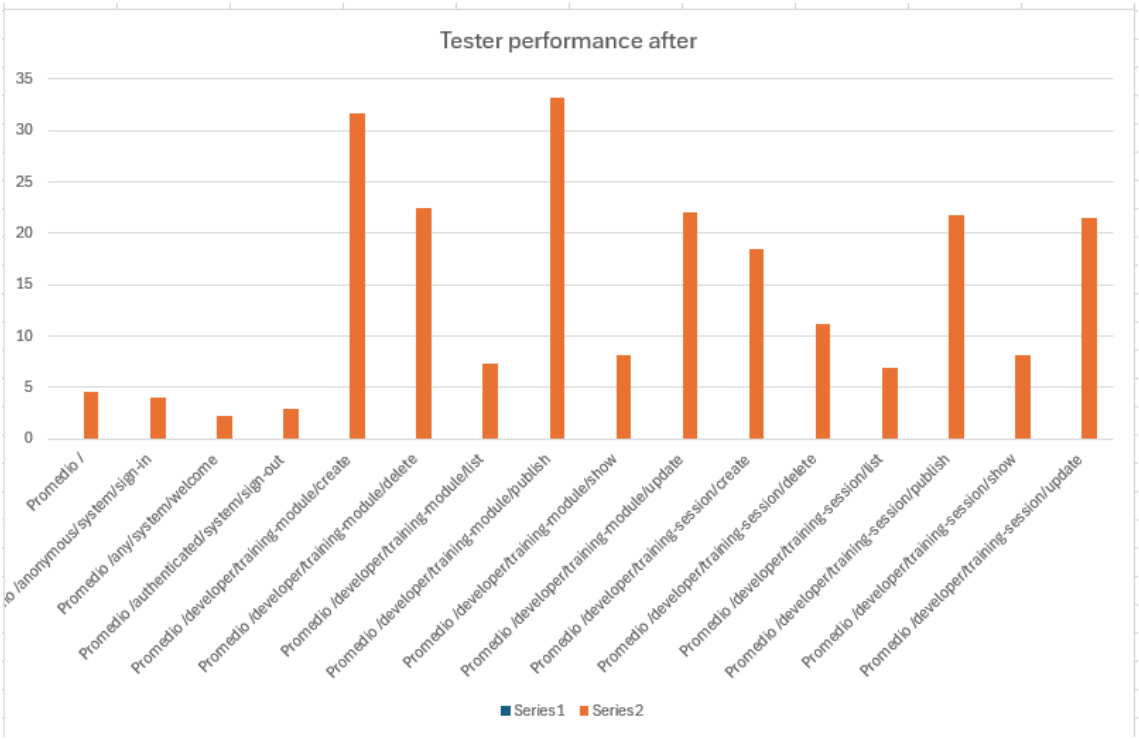
Podemos observar un nivel de confianza de 1,43 y los intervalos de tiempos calculados por la media y el nivel de confianza.

2. Optimizando índices

En este caso hemos identificado las entidades que son potencialmente mejorables mediante la implementación de índices a nivel de base de datos. Aplicando estos índices vamos a conseguir una mayor efectividad a la hora de hacer las consulatas a la base de datos y poder mejorar los tiempos que hemos visto anteriormente.

request-method	request-path	response-status	time
	Promedio /		4,509635294
	Promedio /anonymous/system/sign-in		3,973046
	Promedio /any/system/welcome		2,200191489
	Promedio /authenticated/system/sign-out		2,867
	Promedio /developer/training-module/create		31,656625
	Promedio /developer/training-module/delete		22,5313
	Promedio /developer/training-module/list		7,269741667
	Promedio /developer/training-module/publish		33,20331111
	Promedio /developer/training-module/show		8,1945725
	Promedio /developer/training-module/update		22,0009125
	Promedio /developer/training-session/create		18,45777222
	Promedio /developer/training-session/delete		11,123275
	Promedio /developer/training-session/list		6,871647826
	Promedio /developer/training-session/publish		21,79658182
	Promedio /developer/training-session/show		8,17984
	Promedio /developer/training-session/update		21,53673333
	Promedio general		9,360900935

Nos encontramos con una mejora en las consultas viendo el promedio general el cual es notoriamente inferior al anterior.



Aquí podemos observar como ninguna funcionalidad supera el valor 35 de la gráfica. Esto es debido a la mejora en los tiempos de ejecución de la aplicación.

After		
Media	9,360900935	
Error típico	0,597517391	
Mediana	5,7137	
Moda	#N/D	
Desviación estándar	10,70540412	
Varianza de la muestra	114,6056775	
Curtosis	14,82467326	
Coeficiente de asimetría	3,053773031	
Rango	86,0251	
Mínimo	1,1829	
Máximo	87,208	
Suma	3004,8492	
Cuenta	321	
Nivel de confianza(95,0%)	1,175558687	
Interval (ms)	8,185342248	10,5364596
Interval (s)	0,008185342	0,01053646

También podemos ver mejoría en el análisis donde hemos conseguido tener un nivel de confianza menor al anterior y decrementar los intervalos de tiempos.

Prueba z para medias de dos muestras		
	Before	*After*
Media	10,6414595	9,360900935
Varianza (conocida)	169,8644378	114,6056775
Observaciones	321	321
Diferencia hipotética de las medias	0	
z	1,360296679	
P(Z<=z) una cola	0,086868029	
Valor crítico de z (una cola)	1,644853627	
Valor crítico de z (dos colas)	0,173736059	
Valor crítico de z (dos colas)	1,959963985	

En esta última imagen podemos observar el análisis de la prueba z para las dos pruebas y observamos como el p-value toma un valor de 0,17 el cual está dentro de nuestro rango esperado [0.5, 1].

Conclusiones

Gracias a la elaboración de estas pruebas hemos podido comprobar que todo funcione correctamente y también hemos podido encontrar vulnerabilidades las cuales se han solucionado correctamente. Esto nos asegura que nuestro producto es rápido y seguro cosa que no podemos asegurar si no se ejecutan este tipo de pruebas. También hemos podido apreciar la mejora que nos ofrece el establecer unos índices a nivel de base de datos para poder hacer más efectivas nuestras consultas.

Este documento nos demuestra como los cambios, después de seguir las recomendaciones de la asignatura, han sido satisfactorios. Por ello a la vista se ve un proceso crucial en cualquier proyecto.

En cuanto a la asignatura, esta nos ha aportado los conocimientos tanto teóricos como prácticos a seguir a la hora de elaborar un proyecto y mantenerlo con el tiempo por lo que es de agradecer a los profesores que nos hayan impartido sus conocimientos acerca de esta metodología y el funcionamiento del framework.

Bibliografía

Enseñanza virtual: Diseño y Pruebas 2/L04 - Formal testing