Informe de Testing

Estudiante #3:
Diego Manzanos Anento
(diemanane@alum.us.es)

26-06-2024

Grupo: C1.003
Repositorio:
https://github.com/IsmaelRuizJurado/Acme-SF-D04

Escuela Técnica Superior de Ingeniería Informática

Tabla de Contenidos

Resumen Ejecutivo

En este documento se recogen los detalles del reporte tras realizar el requisito de testing y evaluar los resultados, cómo se indica en el requisito número 9 del Student 3, para cumplir con el requisito número 10 del Student 3, ambos siendo obligatorios y pertenecientes a la cuarta y última entrega.

Historial de Versiones

Versión	Fecha	Descripción de los cambios
V1.0	24/04/2024	Creación del documento
V2.0	26/06/2024	Cambios segunda convocatoria

Introducción

Tras la implementación de las funcionalidades sobre los Training Modules y Training Sessions correspondientes al anterior entregable, se ha procedido a realizar una serie de tests para comprobar su correcto funcionamiento y su comportamiento al realizar peticiones incorrectas que resulten en “panics”.

Para ello, a continuación se verán detallados los detalles encontrados sobre las pruebas, divididos por “testing funcional”, apartado en el que se describe cada prueba ejecutada, y en “testing de rendimiento”.

Contenidos

- Testing Funcional:

A continuación, se detalla cada prueba realizada, divididas por la funcionalidad.

- Training Module

Test pertenecientes a la entidad trainingModule:

  - Create

Se han realizado las pruebas positivas y negativas en create.safe, donde se ha comprobado el correcto funcionamiento del formulario de creación de un nuevo Training Module. Se ha probado con valores incorrectos para cada atributo de forma que nos aseguraba que se rechazaba, así como con las restricciones.

  - Delete

Para las pruebas .safe, se ha comprobado que un trainingModule con el draftMode a true, puede ser borrado.

  - Update

Para las pruebas .safe se han comprobado las restricciones de cada atributo y las validaciones de código y se ha comprobado que, al tener todos los atributos correctamente, se actualizaba correctamente.

  - List

En las pruebas .safe se ha validado que un developer pueda acceder correctamente al listado de sus trainingModules.

  - Show

Se ha comprobado que se muestran correctamente los formularios sobre los trainingModules ya creados.

  - Publish

En las pruebas positivas y negativas se ha comprobado que no es posible publicar un trainingModule sin trainingSessions, que creando una trainingSession no se puede publicar el trainingModule ya que estaría sin publicar y que publicando la trainingSession ya se puede publicar el trainingModule.

- Hack

Se ha optado por hacer un único archivo .hack para todas las funcionalidades, comprobando si se muestra un trainingModule de otro developer, crear cambiando la id de un Project y cambios desde el menú “inspeccionar” de los readOnly, resultando en panic o no realizando ningún cambio.

- Training Session

Test pertenecientes a la entidad trainingSession:

- Create

Se han realizado las pruebas positivas y negativas en create.safe, donde se ha comprobado el correcto funcionamiento del formulario de creación de un nuevo Training Session. Se ha probado con valores incorrectos para cada atributo de forma que nos aseguraba que se rechazaba, así como con las restricciones.

- Delete

Para las pruebas .safe, se ha comprobado que un trainingSession con el draftMode a true, puede ser borrado.

- Update

Para las pruebas .safe se han comprobado las restricciones de cada atributo y las validaciones de código y se ha comprobado que, al tener todos los atributos correctamente, se actualizaba correctamente.

- List

En las pruebas .safe se ha validado que un developer pueda acceder correctamente al listado de sus trainingSessions.

- Show

Se ha comprobado que se muestran correctamente los formularios sobre los trainingSessions ya creados.

- Publish

Para las pruebas .safe se han comprobado las restricciones de cada atributo y las validaciones de código y se ha comprobado que, al tener todos los atributos correctamente, se actualizaba correctamente.

Contenidos

- Hack

Se ha optado por hacer un único archivo .hack para todas las funcionalidades, comprobando si se muestra un trainingSession de otro developer, crear cambiando la id de un Training Module y cambios desde el menú “inspeccionar” de los readOnly, resultando en panic o no realizando ningún cambio.

Contenidos

- Testing de rendimiento.

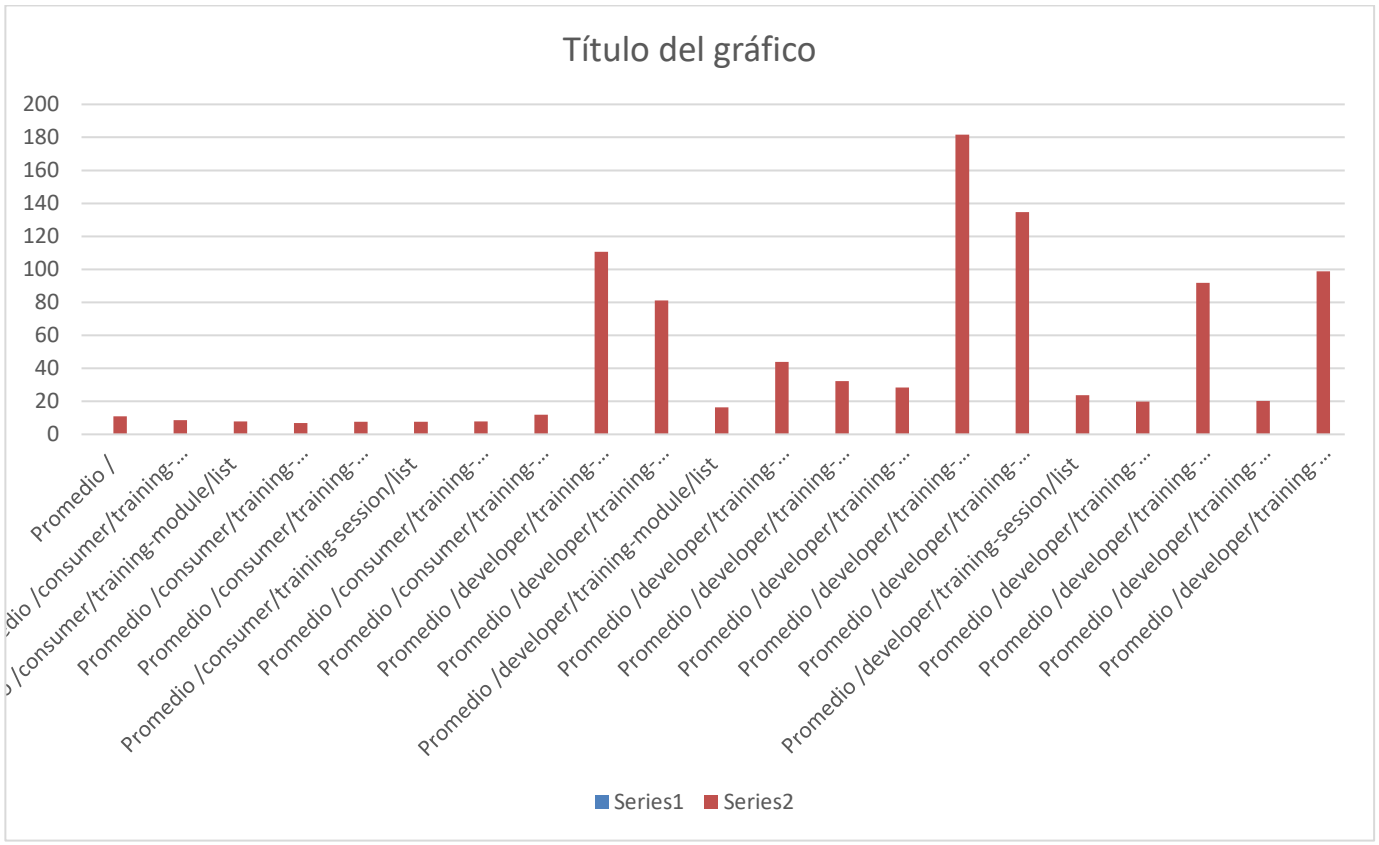
En este capítulo, se evaluará el rendimiento de nuestro proyecto mediante pruebas detalladas que analizan el tiempo de respuesta al atender las solicitudes en nuestras pruebas funcionales. Utilizaremos dos muestras, una de ellas estará compuesta por los resultados obtenidos por hacer realplayer desde coverage, sobre las entidades con índices y sin estos. La segunda muestra se hará de la misma forma, pero desde el apartado de debug.

Para realizar el análisis de estas pruebas, presentaremos gráficos que ilustran los resultados obtenidos. Además, calcularemos intervalos de confianza del 95% para los tiempos de respuesta medidos y realizaremos un contraste de hipótesis, también con un intervalo de confianza del 95%, para determinar la diferencia de hacer realplayer sobre entidades con índices o entidades sin estos.

Contenidos

Muestra sobre los resultados desde Coverage:

Graficas de entidades sin índices:



Contenidos

Grafica de entidades con índices:

Comparación de resultados:

	Before	*After*
Media	44,66860262	35,6914394
Varianza (conocida)	8407,54618	4009,84651
Observaciones	1796	1746
Diferencia hipotética de las me	0	
z	3,398429336	
P(Z<=z) una cola	0,00033887	
Valor crítico de z (una cola)	1,644853627	
Valor crítico de z (dos colas)	0,00067774	
Valor crítico de z (dos colas)	1,959963985	

Contenidos

En nuestras pruebas de rendimiento, hemos obtenido un valor crítico de z de **3.39** para un nivel de significancia (α) de **0.95**. Este valor se utiliza para determinar si las diferencias en los tiempos de respuesta entre las dos pruebas son estadísticamente significativas.

De acuerdo con nuestra metodología, si el p-valor se encuentra en el intervalo (α, 1.00], esto indica que los cambios no resultaron en mejoras relevantes; es decir, aunque los tiempos de muestra sean diferentes, globalmente son equivalentes. Dado que nuestro valor crítico de z no está en este intervalo, concluimos que hay una mejora significativa en el rendimiento al comparar los tiempos de respuesta de las pruebas.

De hecho, se podría intuir que los cambios han mejorado el rendimiento del sistema, pero no se puede asegurar con exactitud dado que los equipos en los que se han realizado las pruebas han podido encontrarse en situaciones menos óptimas tras la segunda prueba.

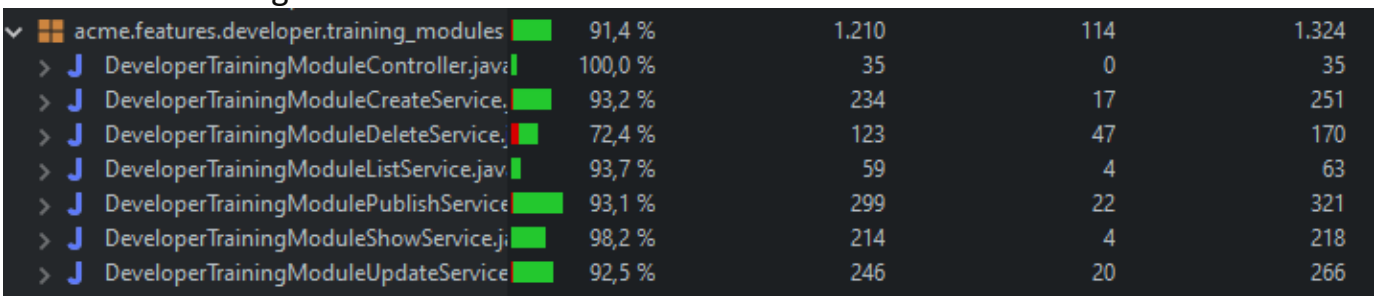
- Cobertura

En esta sección, analizaremos la cobertura de código lograda con nuestros tests. La cobertura de código indica qué porcentaje del código fuente ha sido ejecutado durante las pruebas, ayudando a identificar áreas verificadas y posibles errores no detectados.

Discutiremos los métodos de medición, presentaremos los resultados obtenidos y evaluaremos la efectividad de nuestros tests. También identificaremos lagunas en la cobertura y propondremos estrategias para mejorarla, asegurando así mayor confiabilidad y robustez del software.

Contenidos

Cobertura Training Module:



acme.features.developer.training_modules	91,4 %	1.210	114	1.324
DeveloperTrainingModuleController.java	100,0 %	35	0	35
DeveloperTrainingModuleCreateService.	93,2 %	234	17	251
DeveloperTrainingModuleDeleteService.	72,4 %	123	47	170
DeveloperTrainingModuleListService.jav	93,7 %	59	4	63
DeveloperTrainingModulePublishServic	93,1 %	299	22	321
DeveloperTrainingModuleShowService.j	98,2 %	214	4	218
DeveloperTrainingModuleUpdateServic	92,5 %	246	20	266

Como se puede apreciar, la cobertura sobre las features de la entidad es bastante buena, ya que todas las features implementadas tienen una cobertura de más del 90% excepto el delete.

Esto ocurre porque el delete cuenta con el método unbind, pero este no es nunca usado porque no es necesario para esta funcionalidad, si comentamos este código o lo borramos la cobertura ya superaría el 90% de porcentaje, es por esto por lo que no estamos preocupados porque este no llegue al 90% de cobertura.

```
    @Override
    public void unbind(final TrainingModule object) {
        assert object != null;
        Dataset dataset;
        dataset = super.unbind(object, "code", "creationTime", "details", "basicLevel");
        super.getResponse().addData(dataset);
    }
}
```

Contenidos

Cobertura Training Sessions:

acme.features.developer.training_sessions	91,2 %	1.428	137	1.565
DeveloperTrainingSessionController.java	100,0 %	41	0	41
DeveloperTrainingSessionCreateService.j	94,5 %	328	19	347
DeveloperTrainingSessionDeleteService.j	61,3 %	106	67	173
DeveloperTrainingSessionListAllService.j	94,3 %	66	4	70
DeveloperTrainingSessionListService.java	94,6 %	159	9	168
DeveloperTrainingSessionPublishService	95,4 %	310	15	325
DeveloperTrainingSessionShowService.ja	96,4 %	108	4	112
DeveloperTrainingSessionUpdateService	94,2 %	310	19	329

Ocurre exactamente lo mismo que con los TrainingModules, la cobertura es superior al 90% en todas las funcionalidades salvo para el delete.

Conclusiones

Se ha realizado un análisis profundo sobre los tests implementados para comprobar el correcto funcionamiento de las funcionalidades del Student 3.

Se han analizado las diferencias en el rendimiento del sistema al realizar cambios con los índices. También se ha visto que la cobertura de los tests ha sido bastante positiva.

Bibliografía

Intencionalmente en blanco.