Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática

**Testing Report**



Grado en Ingeniería Informática – Ingeniería del Software  
Diseño y Pruebas 2.

Curso 2023 – 2024

|  |  |
| --- | --- |
| **Fecha** | **Versión** |
| 02/07/2024 | v1.2.0 |

|  |  |  |
| --- | --- | --- |
| **Grupo: C1.011** | | |
| **Autores por orden alfabético** | **Rol** | **Descripción del rol** |
| Abouri, Mohamed - Y7156458E | Developer | Persona encargada de desarrollar el código. |
| Cabello Ranea, Rafael -32094496C | Operador | Encargado de las tareas de campo, de las instalaciones y del mantenimiento de los sistemas de la empresa. |
| Calderón Rodríguez, Francisco Javier - 78233219F | Project Manager | Persona encargada de tomar decisiones de diseño y vigilar el correcto desarrollo |
| Delgado Pallares, David - 29519510E | Tester | Persona encargada de realizar pruebas sobre el código. |
| Ramírez Lara, Iván- 77852382G | Developer | Persona encargada de desarrollar el código. |

**Control de Versiones**

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** |
| 11/05/2024 | v1.0.0 | Creado el reporte de testing funcional |
| 25/05/2024 | v1.1.0 | Añadido testing de rendimiento |
| 02/07/2024 | v1.2.0 | Actualización para segunda convocatoria |
|  |  |  |

**Índice de contenido**

Contenido

[1. Resumen ejecutivo 2](#_Toc166317901)

[2. Introducción 3](#_Toc166317902)

[3. Contenido 4](#_Toc166317903)

[4. Conclusiones 13](#_Toc166317904)

# 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.

# Introducción

Tras la implementación de las funcionalidades sobre Training Modules y Training Sessions durante el sprint anterior, se ha procedido a realizar un conjunto de tests para comprobar su correcto funcionamiento y su comportamiento al realizar peticiones incorrectas que resulten en panics.

Para ello, a continuación se detallan los detalles encontrados sobre las pruebas, dividiéndose en un apartado sobre el testing funcional, donde se describe cada prueba ejecutada; y en un segundo apartado del testing de rendimiento.

# Contenido

* 1. Testing Funcional

A continuación se detalla cada prueba realizada, divididas por funcionalidad

* TrainingModule

Tests 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 para asegurar que se rechazaban, y se ha comprobado que las restricciones funcionaban adecuadamente, como probando a que el updateMoment sea anterior al creationMoment.

En el .hack se ha comprobado las validaciones de acceso. Se ha comprobado que usuarios anónimos y de otros roles no pueden crear trainingModule.

* + Delete

Para las pruebas .safe se ha comprobado que un trainingModule con draftMode verdadero puede ser borrado.

En los tests .hack se ha intentado borrar un trainingModule publicado, y que esto resultaba en un panic. También se ha intentado acceder a la ruta con otros roles y anónimo para comprobar que esto no es posible. Y además se ha intentado borrar trainingModules ya publicados, resultando en panic.

* + List

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

Para las pruebas .hack se ha intentado acceder al listado con usuarios con otros roles para comprobar que no es posible.

* + Publish

En las pruebas positivas y negativas se ha comprobado que no es posible publicar un trainingModule sin trainingSessions, que creando una trainingSession tampoco permite publicar el trainingModule ya que tiene trainingSessions sin publicar, y que publicando la trainingSession ya si es válido publicar el trainingModule.

Para los tests .hack se ha intentado acceder a la ruta de publicación con otros roles, y se ha intentado publicar un trainingModule ya publicado, resultando en un panic.

* + Show

Se ha comprobado que se muestra correctamente los formularios sobre los trainingModule.

En los tests .hack se ha intentado acceder a la ruta con otros roles, se ha intentado acceder al show con un id nulo y con un id que no pertenece a ningún trainingModule, y también se ha intentado acceder al show de un trainingModule con un developer el cual no es dueño de ese trainingModule, resultando en un panic.

* + Update

Para las prubas .safe se ha comprobado las restricciones de cada atributo y las validaciones como que no se puede tener un código duplicado, y se ha comprobado que al tener todos los atributos correctamente se actualizaba correctamente.

En los test .hack se ha intentado acceder a la ruta con otros roles, se ha intentado actualizar un trainingModule ya publicado, y se ha intentado realizar un update sin ningún trainingModule que actualizar.

* TrainingSession

A continuación se detalla cada prueba realizada, divididas por funcionalidad

* + Create

En las pruebas .safe se ha comprobado que las validaciones de cada atributo funcionaban correctamente, especialmente que el finishMoment sea al menos 1 semana posterior al startMoment, y que ambos momentos deben ser posterior al startMoment del trainingModule, probando con diferentes valores. Por último se ha comprobado que al introducir valores correctos se creaba correctamente.

En los .hack se ha comprobado que no se podía acceder a la ruta con otros roles, que no se podían crear trainingSessions en trainingModules ya publicados, y que otros developers no podían crear trainingSessions de trainingModules que no fueran suyos.

* + Delete

En los tests .safe se ha comprobado que se borraban correctamente las trainingSessions que no estubieran aún publicadas.

En los .hack se ha intentado acceder a la ruta con otros roles y se ha intentado borrar trainingSessions ya publicadas.

* + List

En las prubas .safe se ha comprobado que se mostraba correctamente los listados de trainingSessions de los trainingModules.

En los .hack se ha intentado acceder a la ruta con otros roles, se ha intentado acceder al listado de trainingSessions de un trainingModule de otro developer, y se ha intentado acceder al listado con trainingModule nulo y valores inválidos.

* + Publish

Para los tests .safe se ha publicado una trainingSession aún por publicar para comprobar que se publicaba correctamente.

En los .hack se ha intentado acceder a la ruta con otros roles y se ha intentado publicar trainingSessions ya publicadas.

* + Show

Para las prubas .safe se ha comprobado que se mostraba correctamente la información de diferentes trainingSessions.

En los .hack se ha intentado acceder a las rutas con otros roles, se ha intentado acceder al show de una trainingSession con un developer al cual no pertenecía, y se ha intentado acceder con un id nulo e inválido.

* + Update

Para los tests .safe se ha probado cada validación para comprobar que funcionara correctamente. Tras introducir valores válidos se actualizaba correctamente.

En las pruebas .hack se ha intentado acceder a la ruta con otros roles, se ha intentado actualizar trainingSessions ya publicadas, y se ha intentado actualizar sin ningún trainingSession.

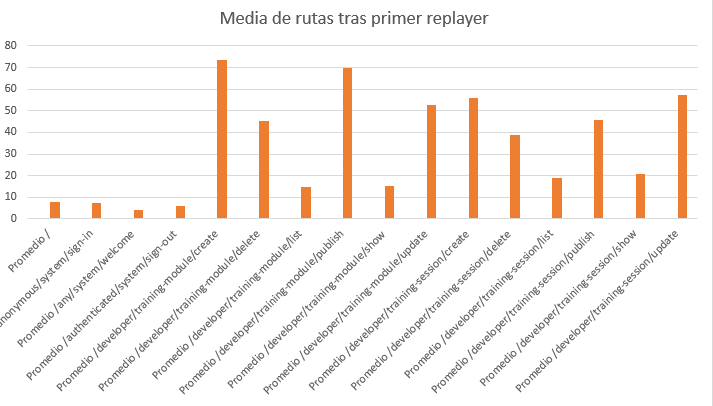
* 1. Testing de Rendimiento

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.

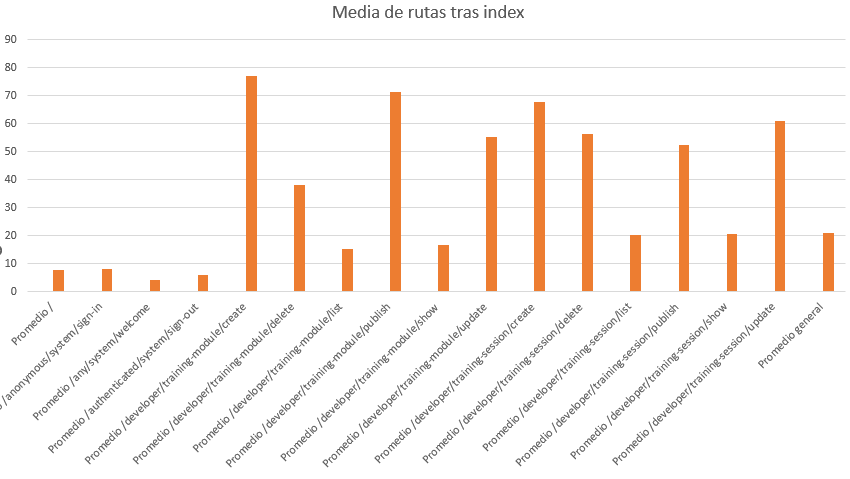
Para realizar el análisis de estas pruebas, presentaremos gráficos que ilustran los resultados obtenidos. Además, calcularemos intervalos de confianza del 95% para los tiempos de respuesta medidos y realizaremos un contraste de hipótesis, también con un intervalo de confianza del 95%, para determinar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

**Muestra sobre resultados desde Coverage.**

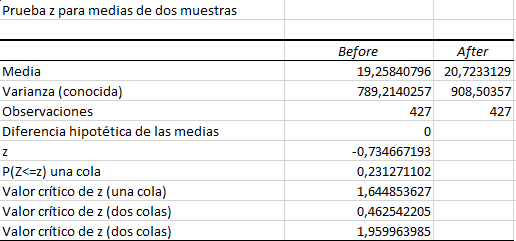
**Gráficas de entidades sin índices:**



**Gráficas de entidades con índices:**



**Comparación de resultados:**



En nuestras pruebas de rendimiento, hemos obtenido un valor crítico de z de **-0.7346** 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 está en este intervalo, concluimos que no hay una mejora significativa en el rendimiento al comparar los tiempos de respuesta de las pruebas.

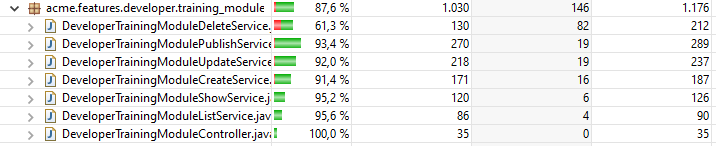
De hecho, se podría intuir que los cambios han empeorado el rendimiento del sistema, pero no se puede asegurar con exactitud dado que los equipos en los que se han realizado las pruebas han podido encontrarse en situaciones menos óptimas tras la segunda prueba.

**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.

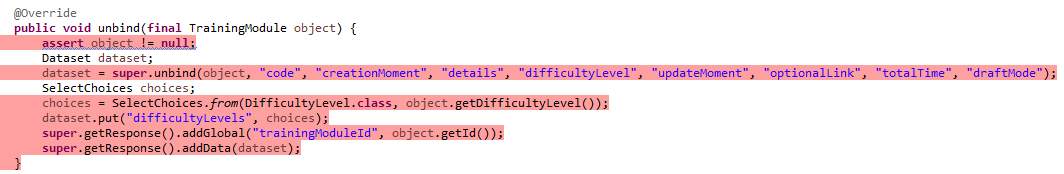
Discutiremos los métodos de medición, presentaremos los resultados obtenidos y evaluaremos la efectividad de nuestros tests. También identificaremos lagunas en la cobertura y propondremos estrategias para mejorarla, asegurando así mayor confiabilidad y robustez del software.

**Cobertura TrainingModule**



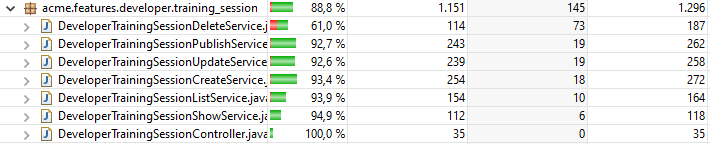
Como se puede apreciar, la cobertura sobre las features de la entidad es bastante buena, ya que todas las features implementadas tienen una cobertura de más del 90% excepto el delete.

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 superaria el 90% de porcentaje, es por esto que no estamos preocupados por que este no llegue al 90% de cobertura.



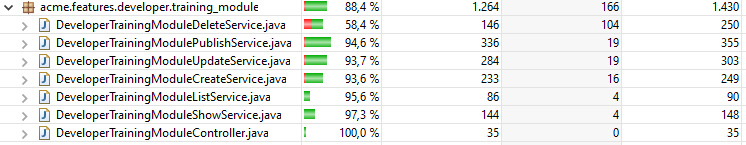
**Cobertura de TrainingSession**

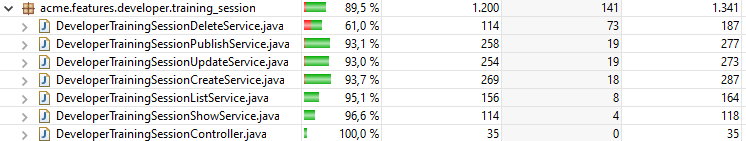
Con respecto a esta entidad, ocurre una cosa parecida a la comentada anteriormente, ya que la cobertura de todas las features es de mayor que el 90% exceptuando a delete que por el unbind no supera este porcentaje.



**Actualización tras corrección**

Al haber modificado el código para realizar las correspondientes correcciones para la segunda convocatoria, se han tenido que repetir las pruebas para comprobar el correcto funcionamiento de las funcionalidades. Tras ejecutar el replayer se han obtenido los siguientes resultados de cobertura:





Se observa que son unos resultados un poco mejores que los de la primera convocatoria debido a que hay unas pocas de líneas de código nuevas, por lo que el porcentaje de cobertura ha aumentado al ejecutar estas líneas, aunque son resultados muy similares.

# [Conclusiones](#_heading=h.1fob9te)

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.