

Testing report

Información general

Autor: Rafael Pulido Cifuentes

Fecha: 23/05/2024

Grupo: C1.006

Repositorio: <https://github.com/Danielruizlopezcc/Acme-SF-D04.git>

Miembros:

- Alberto Carmona Sicre (albcarsic@alum.us.es)
- Ramón Gavira Sánchez (ramgavsan@alum.us.es)
- Marco Padilla Gómez (marpadgom1@alum.us.es)
- Rafael Pulido Cifuentes (rafpulcif@alum.us.es)
- Daniel Ruiz López (danruilop1@alum.us.es)

Tabla de contenidos

- [Testing report](#)
- [Información general](#)
- [Tabla de contenidos](#)
- [Resumen ejecutivo](#)
- [Tabla de revisión](#)
- [Introducción](#)
- [Contenidos](#)
 - [Testing funcional](#)
 - TrainingModule
 - [Cobertura](#)
 - TrainingSession
 - [Cobertura](#)
 - [Testing de rendimiento](#)
 - [Análisis del rendimiento](#)

- [Intervalo de confianza](#)
- [Hypothesis contrast](#)

Resumen ejecutivo

El proyecto Acme-SF-D01 de la asignatura Diseño y Pruebas II, es un proyecto con fines meramente educativos, con el que se busca mejorar las habilidades y trabajar como desarrolladores web. El objetivo es aprender a producir un sistema de información web típico de tamaño pequeño a mediano basándose en una especificación de requisitos informal y métodos y herramientas de potencia industrial.

Tabla de revisión

Nº de revisión	Fecha	Descripción
001	18/05/24	Creación del documento y su estructura
002	18/05/24	Introducción y testing funcional
003	20/05/24	Testing de rendimiento

Introducción

Este documento tiene como objetivo presentar el reporte de pruebas realizadas a los servicios de las clases TrainingModule y TrainingSession. Se mostrarán los resultados obtenidos en cada uno de los servicios probados, incluyendo los casos de prueba tanto positivos como negativos probados en cada uno de los servicios, así como también los resultados obtenidos en la cobertura de código de cada uno de los servicios.

Además, se tratarán los bugs y errores encontrados durante la ejecución de las pruebas y las soluciones que se les ha dado a los mismos.

También se verán reflejados en este documento los resultados obtenidos tras el análisis del rendimiento de los servicios, acompañados de gráficas y tablas que permitirán visualizar de manera más clara todos los resultados.

Por último, se presentarán las conclusiones obtenidas tras la realización de las pruebas y se propondrán posibles mejoras a los servicios probados.

Contenidos

Testing funcional

TrainingModule

DeveloperTrainingModuleListMineService

- **Safe testing:** Para el testing legal de este servicio únicamente fué necesario mostrar los módulos de entrenamiento del desarrollador correspondiente
- **Hacking:**
 - Listado de módulos de un desarrollador sin estar logueado
 - Listado de módulos de un desarrollador estando logueado como otro desarrollador.
 - Listado de módulos de un desarrollador habiendo iniciado sesión con otro rol
- **Bugs:** No se encontraron bugs

DeveloperTrainingModuleShowService

- **Safe testing:** Se mostraron los datos de módulos tanto publicados como no publicados del desarrollador con el que se había iniciado sesión
- **Hacking:**
 - Mostrar datos de un módulo no publicado de un desarrollador sin haber iniciado sesión
 - Mostrar datos de un módulo no publicado de un desarrollador habiendo iniciado sesión como otro desarrollador distinto
 - Mostrar datos de un módulo no publicado de un desarrollador habiendo iniciado sesión con otro rol
- **Bugs:** No se encontraron bugs

DeveloperTrainingModuleCreateService

- **Safe testing:**
 - Creación de módulos en los que los datos eran incorrectos: Nulos en aquellos obligatorios, formato incorrecto, fechas que no cumplían los requisitos de duración y creación, tamaño excesivo y código duplicado
 - Creación de módulos en los que los datos si eran correctos: Formato correcto, tiempos de duración permitidos, tamaños aceptados y dejando en blanco solo aquellos que eran opcionales.
- **Hacking:**

- Acceder al formulario de creación sin haber iniciado sesión
- Acceder al formulario de creación habiendo iniciado sesión con otro rol
- **Bugs:** Se encontró un bug cuando se intentó crear un módulo en el que alguna de las fechas era nula
- **Soluciones:** Para solucionar el bug, en el método validate, se añadió la comprobación a ambas fechas de que estas no podían ser nulas si se quería comprobar el número de días entre dos fechas. Esta solución se implementó también en los servicios de actualización y publicación.

DeveloperTrainingModuleUpdateService

- **Safe testing:**
 - Actualización de módulos en los que los datos eran incorrectos: Nulos en aquellos obligatorios, formato incorrecto, fechas que no cumplían los requisitos de duración y creación, tamaño excesivo y código duplicado
 - Actualización de módulos en los que los datos si eran correctos: Formato correcto, tiempos de duración permitidos, tamaños aceptados y dejando en blanco solo aquellos que eran opcionales.
- **Hacking:** Desde las herramientas del desarrollador, se modificó el id del módulo que iba a actualizar, de esta forma se probó
 - Actualización de módulo de otro desarrollador
 - Actualización de módulo inexistente
 - Actualización de módulo publicado
- **Bugs:** No se encontraron bugs

DeveloperTrainingModulePublishService

- **Safe testing:**
 - Publicación de módulos en los que los datos eran incorrectos: Nulos en aquellos obligatorios, formato incorrecto, fechas que no cumplían los requisitos de duración y creación, tamaño excesivo y código duplicado
 - Publicación de módulos en los que los datos si eran correctos: Formato correcto, tiempos de duración permitidos, tamaños aceptados y dejando en blanco solo aquellos que eran opcionales.
 - Publicación de módulos sin sesiones
 - Publicación de módulos con sesiones sin publicar
- **Hacking:** Desde las herramientas del desarrollador, se modificó el id del módulo que iba a publicar, de esta forma se probó
 - Publicación de módulo de otro desarrollador
 - Publicación de módulo inexistente

















- Publicación de módulo ya publicado
- **Bugs:** No se encontraron bugs

DeveloperTrainingModuleDeleteService

- **Safe testing:**
 - Eliminación de módulos con sesiones sin publicar
 - Eliminación de sin éxito de módulos con sesiones publicadas
- **Hacking:** Desde las herramientas del desarrollador, se modificó el id del módulo que iba a eliminar, de esta forma se probó
 - Eliminación de módulo de otro desarrollador
 - Eliminación de módulo inexistente
 - Eliminación de módulo publicado
- **Bugs:** No se encontraron bugs

Cobertura

A continuación se muestra el porcentaje de cobertura de sentencias de todos los servicios asociados a TrainingModule

▼  acme.features.developer.trainingModule		95,1 %	1.366	70	1.436
>  DeveloperTrainingModulePublishService		94,8 %	310	17	327
>  DeveloperTrainingModuleCreateService		93,5 %	232	16	248
>  DeveloperTrainingModuleDeleteService		94,0 %	251	16	267
>  DeveloperTrainingModuleUpdateService		94,3 %	266	16	282
>  DeveloperTrainingModuleShowService.java		97,3 %	146	4	150
>  DeveloperTrainingModuleListMineService		99,2 %	125	1	126
>  DeveloperTrainingModuleController.java		100,0 %	36	0	36

TrainingSession

DeveloperTrainingSessionListService

- **Safe testing:**
 - Listado de sesiones de módulos sin publicar
 - Listado de sesiones de módulos publicados
- **Hacking:**
 - Listado de sesiones de un desarrollador sin haber iniciado sesión
 - Listado de sesiones de un desarrollador estando logueado como otro desarrollador.
 - Listado de sesiones de un desarrollador estando logueado con otro rol
 - Listado de sesiones que pertenecen a otro módulo
- **Bugs:** No se encontraron bugs

DeveloperTrainingSessionShowService

- **Safe testing:**
 - Muestra de datos de sesiones sin publicar
 - Muestra de datos de sesiones publicadas
- **Hacking:**
 - Muestra de datos de una sesión de un desarrollador sin haber iniciado sesión
 - Muestra datos de un módulo de un desarrollador habiendo iniciado sesión como otro desarrollador distinto
 - Muestra de datos de un módulo no publicado de un desarrollador habiendo iniciado sesión con otro rol
 - Muestra de datos de una sesión perteneciente a otro módulo
- **Bugs:** No se encontraron bugs

DeveloperTrainingSessionCreateService

- **Safe testing:**
 - Creación de sesiones en los que los datos eran incorrectos: Nulos en aquellos obligatorios, formato incorrecto, fechas que no cumplían los requisitos de duración y creación, tamaño excesivo y código duplicado
 - Creación de sesiones en los que los datos si eran correctos: Formato correcto, tiempos de duración permitidos, tamaños aceptados y dejando en blanco solo aquellos que eran opcionales.
- **Hacking:**
 - Acceder al formulario de creación sin haber iniciado sesión
 - Acceder al formulario de creación habiendo iniciado sesión con otro rol
- **Bugs:** Al igual que con los módulos, se encontró un bug cuando se intentó crear una sesión en la que alguna de las fechas era nula
- **Soluciones:** Para solucionar el bug, en el método validate, se añadió la comprobación a ambas fechas de que estas no podían ser nulas si se quería comprobar el número de días entre dos fechas. Este solución se implementó también en los servicios de actualización y publicación.

DeveloperTrainingSessionUpdateService

- **Safe testing:**
 - Actualización de sesiones en los que los datos eran incorrectos: Nulos en aquellos obligatorios, formato incorrecto, fechas que no cumplían los requisitos de duración y creación, tamaño excesivo y código duplicado
 - Actualización de sesiones en los que los datos si eran correctos: Formato correcto, tiempos de duración permitidos, tamaños aceptados y dejando en blanco solo aquellos que eran

opcionales.

- **Hacking:** Desde las herramientas del desarrollador, se modificó el id de la sesión que iba a actualizar, de esta forma se probó
 - Actualización de sesión de otro desarrollador
 - Actualización de sesión inexistente
 - Actualización de sesión publicada
- **Bugs:** No se encontraron bugs

DeveloperTrainingSessionPublishService















- **Safe testing:**
 - Publicación de sesiones en los que los datos eran incorrectos: Nulos en aquellos obligatorios, formato incorrecto, fechas que no cumplían los requisitos de duración y creación, tamaño excesivo y código duplicado
 - Publicación de sesiones en los que los datos si eran correctos: Formato correcto, tiempos de duración permitidos, tamaños aceptados y dejando en blanco solo aquellos que eran opcionales.
- **Hacking:** Desde las herramientas del desarrollador, se modificó el id de la sesión que iba a publicar, de esta forma se probó
 - Publicación de sesión de otro desarrollador
 - Publicación de sesión inexistente
 - Publicación de sesión ya publicada
- **Bugs:** No se encontraron bugs

DeveloperTrainingSessionDeleteService

- **Safe testing:** Eliminación de sesiones no publicadas
- **Hacking:** Desde las herramientas del desarrollador, se modificó el id de la sesión que iba a eliminar, de esta forma se probó
 - Eliminación de sesión de otro desarrollador
 - Eliminación de sesión inexistente
 - Eliminación de sesión ya publicada
- **Bugs:** No se encontraron bugs

Cobertura

A continuación se muestra el porcentaje de cobertura de sentencias de todos los servicios asociados a TrainingSession

▼  acme.features.developer.trainingSession	 89,8 %	1.164	132	1.296
>  DeveloperTrainingSessionDeleteService.j	 60,0 %	105	70	175
>  DeveloperTrainingSessionPublishService	 93,5 %	245	17	262
>  DeveloperTrainingSessionCreateService,	 94,2 %	262	16	278
>  DeveloperTrainingSessionUpdateService	 94,2 %	259	16	275
>  DeveloperTrainingSessionListService.java	 94,3 %	148	9	157
>  DeveloperTrainingSessionShowService.j	 96,5 %	110	4	114

En el caso de las TrainingSessions, observamos que el coverage en el servicio de eliminación es considerablemente menor que el de el resto de los servicios. Al realizar el testing de este servicio observamos que no existen casos negativos a la hora de eliminar las sesiones ya que realizando acciones legales, siempre es posible eliminar una sesión que no este publicada. Este caso particular es el que impide que se ejecuten las siguientes lineas en ningun caso.

```

65  @Override
66  public void unbind(final TrainingSession object) {
67
68      assert object != null;
69
70      Dataset dataset;
71
72      dataset = super.unbind(object, "code", "sessionStart", "sessionEnd", "location", "instructor", "contactEmail", "link", "draftMode");
73      dataset.put("masterId", object.getTrainingModule().getId());
74
75      super.getResponse().addData(dataset);
76  }
77 }

```

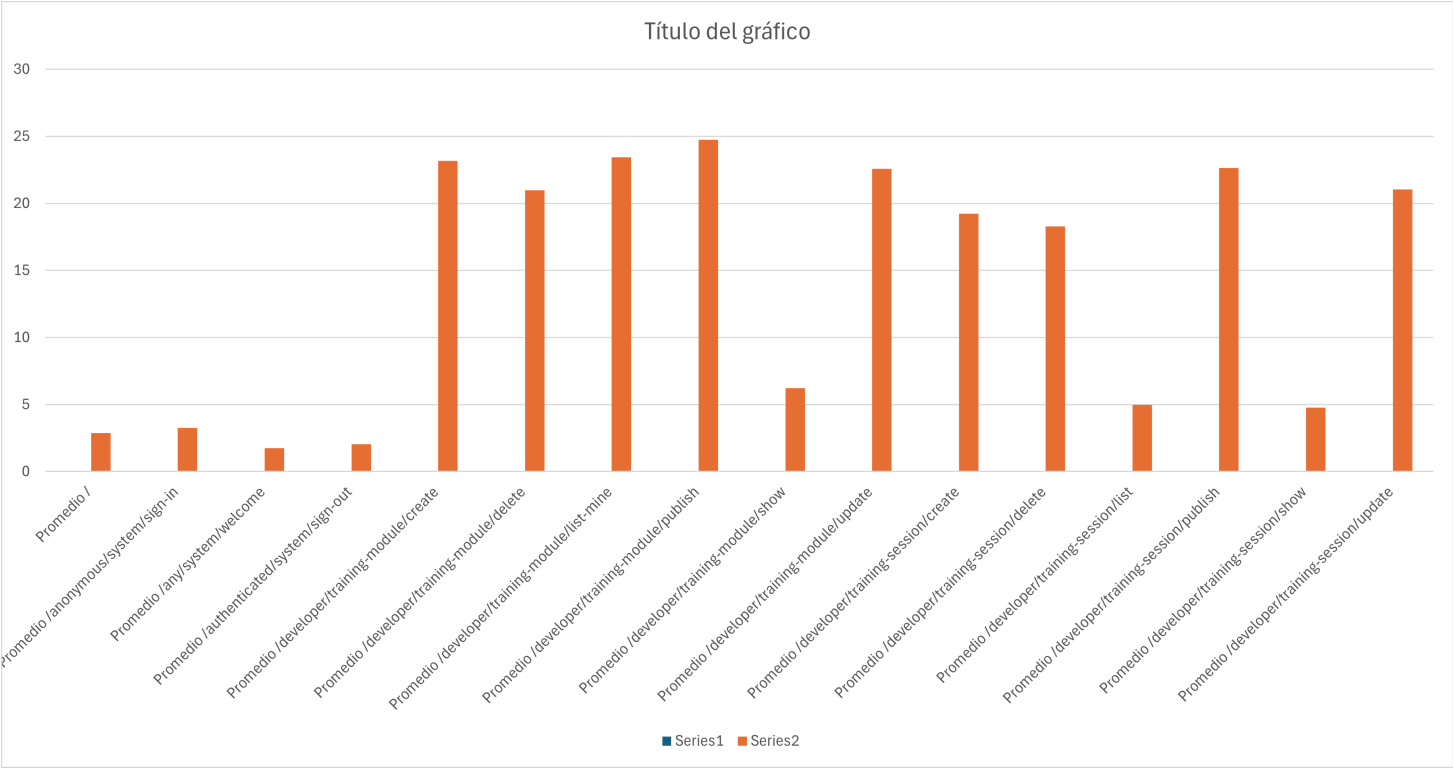
Una de las soluciones que por tanto se consideran es la eliminación del método por completo ya que con las pruebas de testing queda comprobado que no se está utilizando para nada.

Testing de rendimiento

Análisis del rendimiento

Tras la ejecución de los tests, y con los datos recopilados sobre las peticiones realizadas en los mismos, se ha llevado a cabo un análisis de los resultados obtenidos en cuanto al rendimiento la aplicación tomando como referencia el tiempo de respuesta de los servicios, en concreto, de los servicios asociados a las entidades TrainingModule y TrainingSession. A continuación, se proporcionan los datos del promedio de cada petición, aunque para una mayor claridad, también se proporcionan una gráfica de barras.

Promedio /	2,88100192
Promedio /anonymous/system/sign-in	3,25859
Promedio /any/system/welcome	1,74765119
Promedio /authenticated/system/sign-out	2,043904
Promedio /developer/training-module/create	23,1681483
Promedio /developer/training-module/delete	20,990325
Promedio /developer/training-module/list-mine	23,4486514
Promedio /developer/training-module/publish	24,7478679
Promedio /developer/training-module/show	6,22179014
Promedio /developer/training-module/update	22,5786343
Promedio /developer/training-session/create	19,2307179
Promedio /developer/training-session/delete	18,27295
Promedio /developer/training-session/list	4,98783529
Promedio /developer/training-session/publish	22,6487261
Promedio /developer/training-session/show	4,76767778
Promedio /developer/training-session/update	21,0486333
Promedio general	11,852434



Como vemos en la gráfica, existe una gran diferencia entre los tiempos de respuesta de las peticiones de inicio de sesión, bienvenida y cierre de sesión y las peticiones relacionadas con las *features* de ambas entidades. En el caso de las primeras, el tiempo de respuesta es muy bajo, mientras que en las segundas, el tiempo de respuesta es mucho mayor. Sin embargo, es destacable que aunque los tiempos de respuesta para las *features* de las entidades son mucho mayores por norma general, no es el caso de las peticiones *show* en ambas, en las que el tiempo de respuesta es significativamente menor. Esto se debe a la sencillez de las peticiones *show* en comparación con el resto de peticiones. Por último, si echamos un vistazo mas detallado al gráfico, observamos nuevamente algo a destacar,

y es nuevamente la diferencia entre los tiempos de respuesta de las peticiones *list* de TrainingSession y *list-mine* de TrainingModule, siendo el tiempo de respuesta de la primera mucho menor que el de la segunda llegando esta ultima incluso a ser de la segunda petición con mayor tiempo promedio de respuesta, a pesar de que ambas *list* y *list-mine* son peticiones que realizan acciones casi idénticas.

Intervalo de confianza

Una vez analizados los tiempos de respuesta, se ha calculado el intervalo de confianza de la media los mismos, con un nivel de confianza del 95%, con el objetivo de comprobar si la aplicación cumple con el requisito de rendimiento establecido: *que el tiempo de respuesta de las peticiones no supere el segundo de media*.

La herramienta utilizada para el análisis del rendimiento ha sido excel, que proporciona un complementento llamado *Herramientas para el análisis* que permite realizar cálculos de estadística descriptivas dando la opción de establecer un nivel de confianza del 95%, con este nivel de confianza podremos esperar que los tiempos promedios de ejecución, estén en un 95% de los casos dentro del intervalo de confianza.

Para obtener el intervalo de confianza, hemos debido de establecer el límite superior y el límite inferior (tanto en milisegundos como en segundos).

- **Límite superior:** Media + Nivel de confianza
- **Límite inferior:** max(Media - Nivel de confianza, 0)

A continuación se muestran los datos del análisis estadístico asi como del intervalo de confianza de la media de los tiempos de respuesta de las peticiones.

Columna1					
			Interval (ms)	10,7663434	12,5639804
Media	11,6651619		Interval (s)	0,01076634	0,01256398
Error típico	0,45775962				
Mediana	5,20395				
Moda	4,5769				
Desviación estándar	11,848814				
Varianza de la muestra	140,394393				
Curtosis	10,4143805				
Coefficiente de asimetría	2,01859521				
Rango	118,5655				
Mínimo	1,0321				
Máximo	119,5976				
Suma	7815,65847				
Cuenta	670				
Nivel de confianza(95,0%)	0,89881847				

Como podemos observar, gracias al calculo del intervalo de confianza, podemos afirmar que el tiempo de respuesta de las peticiones de la aplicación se encuentra dentro del rango de [0,1], en concreto [0.011, 0.012] segundos, cumpliendo así con el requisito de rendimiento establecido.

Hypothesis contrast

A pesar de haber comprobado que se está cumpliendo con el requisito de rendimiento, se ha llevado a cabo una refactorización mediante el uso de índices para aumentar el rendimiento de las *queries* de los repositorios asociados a las entidades TrainingModule y TrainingSession. Para comprobar si esta refactorización ha tenido un impacto en el rendimiento de la aplicación, se ha llevado a cabo un contraste de hipótesis, mas concretamente, se ha realizado nuevamente en excel con el complemento *Herramientas para el análisis*, un *Z-test* (con $\alpha = 0.05$) para evaluar si los nuevos tiempos de respuesta son mejores o peores y en que medida.

Para poder realizar el Z-test, se ha debido de recopilar nuevamente los tiempos de respuesta tras la refactorización y obtener los datos estadísticos.

A continuación se muestran los datos del análisis post-refactorización, así como los resultados del Z-test.

Before				After			Prueba z para medias de dos muestras		
							Before	After	
Media	11,85243403			Media	11,33743295				
Error típico	0,46027331			Error típico	0,483737746		Media	11,852434	11,337433
Mediana	5,3174			Mediana	5,0731		Varianza (conocida)	141,94052	141,33733
Moda	4,5769			Moda	4,0205		Observaciones	670	604
Desviación estándar	11,91387924			Desviación estándar	11,88853788		Diferencia hipotética de las medias	0	
Varianza de la muestra	141,9405185			Varianza de la muestra	141,3373329		z	0,7712798	
Curtosis	10,11303206			Curtosis	22,19085963		P(Z<=z) una cola	0,2202706	
Coefficiente de asimetría	1,983908158			Coefficiente de asimetría	2,947022984		Valor crítico de z (una cola)	1,6448536	
Rango	118,5655			Rango	135,6426		Valor crítico de z (dos colas)	0,4405411	
Mínimo	1,0321			Mínimo	1,2062		Valor crítico de z (dos colas)	1,959964	
Máximo	119,5976			Máximo	136,8488				
Suma	7941,1308			Suma	6847,8095				
Cuenta	670			Cuenta	604				
Nivel de confianza(95,0%)	0,903754145			Nivel de confianza(95,0%)	0,9500154				
Interval (ms)	10,94867988	12,7561882		Interval (ms)	10,38741755	12,2874483			
Interval (s)	0,01094868	0,01275619		Interval (s)	0,010387418	0,01228745			

Como podemos observar, el valor crítico de z (dos colas) es 0.44, para saber si los cambios han sido significativos, hemos de comparar el valor de z con α . Es este caso, nuestro z se encuentra en el intervalo (α , 1] por lo que podemos afirmar que los cambios realizados con la refactorización no han tenido un impacto significativo en el rendimiento de la aplicación.

De igual forma, esto se puede comprobar de una manera mas visual si observamos la media obtenida antes y después, en el primer caso, la media es aproximadamente 11,85ms mientras que en el segundo caso, la media es aproximadamente 11,34ms. Si bien es cierto que el tiempo medio de respuesta ha disminuido, la diferencia no es significativa.