

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática



Grado en Ingeniería Informática – Ingeniería del Software

Diseño y Pruebas 2

Curso 2023 – 2024

Plantilla Reporte general

Grupo C1.013

Miembros	Información de contacto
David Fuentelsaz Rodríguez	davfuerod@alum.us.es
Miguel Galán Lerate	miggaller@alum.us.es
Antonio Jiménez Ortega	antjimort@alum.us.es
Josué Rodríguez López	josrodlop19@alum.us.es
Óscar Zurita Urpina	osczururp@alum.us.es

Repositorio: <https://github.com/DP2-C1-013/Acme-SF-D01-24.1.0>

Fecha	Versión
14/2/2024	v1.0

1. Resumen.....	2
2. Control de versiones.....	3
3. Introducción.....	3
4. Contenidos.....	3
4.1. Punto 1 del contenido.....	3
5. Conclusión.....	3
6. Bibliografía.....	3

1. Resumen.

Este reporte tiene como objetivo realizar un análisis de los resultados obtenidos por el estudiante #3 en la ejecución de los tests de la aplicación.

2. Control de versiones.

Nº de revisión	Fecha	Descripción
1	24/05/2024	Desarrollo del reporte de testing

3. Introducción.

El reporte de testing está centrado en proporcionar una descripción lo más detallada posible acerca de los resultados obtenidos en la ejecución de los tests de cada una de las funcionalidades implementadas por el estudiante #3 en la aplicación.

Al estar este reporte clasificado por funcionalidades, lo dividiremos principalmente en dos partes:

- Tests en TrainingModule: incluye una descripción sobre los tests realizados sobre las funcionalidades de *list*, *show*, *create*, *update*, *publish* y *delete* de TrainingModule.
- Tests en TrainingSession: incluye una descripción sobre los tests realizados sobre las funcionalidades de *list*, *show*, *create*, *update*, *publish* y *delete* de TrainingSession.

4. Contenidos.

Todos los tests serán realizados por el *developer1*, usuario registrado en la aplicación. El uso de cualquier otro usuario (*developer2*, *sponsor1*) solo servirá para comprobar el acceso mediante usuarios con diferente rol, o con el mismo rol pero sin acceso a datos de otro usuario.

4.1. TrainingModule

4.1.1. List

La funcionalidad de list ha sido fácil de probar debido a su sencillez, ya que simplemente había que acceder al endpoint */developer/training-module/list* como *developer1*, lo que muestra todos los TrainingModule paginados, mostrando algunos de sus atributos (código, momento de creación, nivel de dificultad, proyecto asociado, y si está o no publicado).

Se ha hecho también un intento de GET Hacking desde un usuario con rol de sponsor al endpoint mencionado anteriormente, a lo cual el sistema ha respondido con un código 500.

4.1.2. Show

Para probar la funcionalidad show, he accedido mediante el endpoint */developer/training-module/show?id={trainingModuleId}* a uno de los objetos mostrado por el list, a lo que el sistema muestra los detalles del TrainingModule en cuestión con todos sus atributos (código, momento de creación, detalles, nivel de dificultad, momento de actualización, enlace opcional, proyecto asociado, si está o no publicado, y el tiempo total estimado de todas las TrainingSession que tiene asociadas), así como los botones de Training sessions, para acceder al listado de sus TrainingSession; delete, para borrar; update, para actualizar; y publish, para publicar. Cabe mencionar que estos tres últimos no aparecen si el TrainingModule mostrado ya está publicado.

Se ha hecho también un intento de GET Hacking al endpoint mencionado de un TrainingModule tanto desde un usuario con rol de sponsor, como un usuario con rol de developer pero sin acceso a ese objeto por pertenecer a otro developer (el sistema niega el acceso a ambos casos con un código 500).

4.1.3. Create

Para probar la creación, he accedido al endpoint `/developer/training-module/create` para crear un nuevo TrainingModule, lo cual me muestra un formulario vacío a rellenar con datos de un nuevo objeto. Para empezar, mando el formulario vacío, a lo que el sistema responde con un mensaje de error por cada campo obligatorio no rellenado. Después lo he rellenado todo, y he probado poniendo valores incorrectos a los atributos para comprobar que el sistema gestiona bien los errores. Para el código, al poner XXa-111, por ejemplo, el sistema responde diciendo “El código debe tener de 1 a 3 letras mayúsculas, seguidas de un guión, y luego tres números.”; al poner un código ya existente, el sistema responde diciendo “Código duplicado”. Para el momento de creación, se puede poner cualquier fecha. Para el momento de actualización, hay que poner una fecha al menos 24 horas (inclusive) posterior al momento de creación. Los detalles no pueden estar en blanco. El nivel de dificultad no puede ser nulo. El enlace, de haberlo, tiene que estar bien formado, siendo este del tipo “protocol://name.domain”. El proyecto asociado debe ser nulo, además de que el sistema solo deja escoger por aquellos proyectos publicados. El tiempo total estimado no se puede rellenar, ya que se calcula automáticamente cuando detecta que tiene TrainingSessions asociadas. Tras poner valores en los campos que cumplan con lo comentado, el sistema crea el nuevo objeto satisfactoriamente.

Se ha hecho también un intento de POST Hacking al endpoint mencionado de un TrainingModule desde un usuario con rol de sponsor, a lo cual el sistema niega la operación mostrando código 500.

4.1.4. Update

Para probar la actualización, he accedido a los detalles de un TrainingModule y, teniendo en cuenta las restricciones descritas para cada atributo en el Create, he probado de la misma forma, con todos los valores, además de con un formulario vacío (también ha tenido la misma respuesta que en el Create). El código, al haber sido considerado un identificador del objeto para el usuario, no se puede modificar en el update, por tanto solo se puede leer. El tiempo total estimado no se puede modificar, ya que se calcula automáticamente cuando detecta que tiene TrainingSessions asociadas. Cabe mencionar que antes de hacer todo esto, he creado, asociado a este TrainingModule, un TrainingSession, para asegurarme de que si cambio su momento de creación, el sistema muestre un error si entre ese momento de creación y la fecha de inicio del TrainingSession hay menos de una semana de separación, regla de negocio que se ha debido reflejar en TrainingSession. Tras modificar los campos cumpliendo con el criterio establecido en la funcionalidad create para cada atributo, el objeto es correctamente actualizado si pulsamos en el botón Update, lo que hace una petición al endpoint `/developer/training-module/update`.

Además, el botón Update no se muestra para aquellos objetos ya publicados, lo mismo ocurre con aquellos TrainingModule con alguna TrainingSession ya publicada. Para intentar forzar esto, he probado a intentar actualizar un TrainingModule ya publicado accediendo a sus detalles y haciendo una petición al endpoint `/developer/training-module/update`, a lo cual el sistema ha respondido con error 500 en ambos casos, ya que no se puede actualizar un TrainingModule ya publicado.

Se ha hecho también un intento de POST Hacking al endpoint `/developer/training-module/update` de un TrainingModule desde un usuario con rol de sponsor, a lo cual el sistema niega la operación mostrando código 500.

4.1.5. Publish

Para probar la publicación, he accedido a los detalles de un `TrainingModule` y, teniendo en cuenta las restricciones descritas para cada atributo en el `Create`, he probado de la misma forma, con todos los valores y cumpliendo la regla de negocio de la fecha de creación y de la fecha de inicio de todas las `TrainingSession` asociadas a este `TrainingModule`, además de con un formulario vacío (también ha tenido la misma respuesta que en el `Create`). Si algún atributo no cumple con esas restricciones, se muestra un error descriptivo en el campo correspondiente. Con todos los datos correctos, el sistema permite publicar el `TrainingModule` si pulsamos en el botón `Publish`, lo que hace una petición al endpoint `/developer/training-module/publish`.

Además, el botón `Publish` no se muestra para aquellos objetos ya publicados. Para intentar forzar esto, he probado a intentar publicar un `TrainingModule` ya publicado accediendo a sus detalles y haciendo una petición al endpoint `/developer/training-module/publish`, a lo cual el sistema ha respondido con error 500, ya que no se puede publicar un `TrainingModule` ya publicado.

Se ha hecho también un intento de POST Hacking al endpoint `/developer/training-module/publish` de un `TrainingModule` desde un usuario con rol de `sponsor`, a lo cual el sistema niega la operación mostrando código 500.

4.1.6. Delete

Para probar la eliminación, he accedido a los detalles de un `TrainingModule` y, sin necesidad de tener en cuenta ninguna de las restricciones descritas en el `Create`, se puede eliminar sin problemas un `TrainingModule` siempre que se cumplan las siguientes condiciones: que el `TrainingModule` no esté publicado, y/o que ninguna de las `TrainingSession` que tiene asociadas esté publicada. Al probar borrar un `TrainingModule` no publicado, éste se borra exitosamente. En los detalles de los que ya están publicados, el botón para eliminar está deshabilitado, por tanto, para forzar la operación, me he metido en uno ya publicado y he hecho una petición a su endpoint, `/developer/training-module/publish`, lo que devuelve código 500, denegando la petición; lo mismo ocurre con aquellos objetos con `TrainingSession` ya publicadas.

Se ha hecho también un intento de POST Hacking al endpoint `/developer/training-module/delete` de un `TrainingModule` desde un usuario con rol de `sponsor`, a lo cual el sistema niega la operación mostrando código 500.

4.2. TrainingSession

4.2.1. List

La funcionalidad de `list` ha sido fácil de probar debido a su sencillez, ya que simplemente había que acceder al endpoint `/developer/training-session/list?trainingModuleId=${id}`, siendo `id` el id de un `TrainingModule` cualquiera, como `developer1`, lo que muestra todos los `TrainingSession` paginados de dicho `TrainingModule`, mostrando algunos de sus atributos (código, fecha de inicio, fecha de fin, instructor, y si está o no publicado).

Se ha hecho también un intento de GET Hacking desde un usuario con rol de `sponsor` al endpoint mencionado anteriormente, a lo cual el sistema ha respondido con un código 500. Lo mismo ha pasado si accedemos al listado de `TrainingSessions` asociados a un `TrainingModule` que no nos pertenece, por ejemplo, habiendo iniciado sesión como `developer2`.

4.2.2. Show

Para probar la funcionalidad show, he accedido mediante el endpoint `/developer/training-session/show?id={trainingSessionId}` a uno de los objetos mostrado por el list, a lo que el sistema muestra los detalles del TrainingSession en cuestión con todos sus atributos (código, fecha de inicio, fecha de fin, ubicación, instructor, email de contacto, enlace opcional, si está o no publicado, y el código del TrainingModule que tiene asociado), así como los botones de delete, para borrar el objeto; update, para actualizarlo; y publish, para publicarlo. Cabe mencionar que estos botones sólo se muestran cuando el TrainingSession no está publicado.

Se ha hecho también un intento de GET Hacking al endpoint mencionado de un TrainingSession tanto desde un usuario con rol de sponsor, como un usuario con rol de developer pero sin acceso a ese objeto por pertenecer a otro developer (el sistema niega el acceso a ambos casos con un código 500).

4.2.3. Create

Para probar la creación, he accedido al endpoint `/developer/training-session/create?trainingModuleId=${trainingModuleId}` para crear un nuevo TrainingModule, lo cual me muestra un formulario vacío a rellenar con datos de un nuevo objeto. Para empezar, mando el formulario vacío, a lo que el sistema responde con un mensaje de error por cada campo obligatorio no rellenado. Después lo he rellenado todo, y he probado poniendo valores incorrectos a los atributos para comprobar que el sistema gestiona bien los errores. Para el código, he probado con TS-XXa-111, a lo que el sistema responde “El código debe tener “TS-” seguido de 1, 2 o 3 letras mayúsculas, un guión y tres números”; al poner un código ya existente, el sistema responde diciendo “Código duplicado”. En cuanto a la fecha de comienzo, ésta tiene que ser posterior al menos una semana del momento de creación de su TrainingModule, de lo contrario salta el respectivo mensaje de error. Para la fecha de fin, ésta tiene que ser posterior al menos una semana de la fecha de comienzo. El lugar no puede estar en blanco. El instructor no puede estar en blanco. El email de contacto tiene que tener el formato adecuado de un email, de lo contrario salta el error que lo describe. El enlace opcional, de haberlo, tiene que tener el formato “protocol://name.domain”. El código del TrainingModule mostrado no se puede modificar, ya que solo se muestra para indicar al usuario para qué TrainingModule se creará el TrainingSession en cuestión. Al crear el TrainingSession con datos que cumplen estas restricciones, la operación es realizada de forma satisfactoria.

Se ha hecho también un intento de POST Hacking al endpoint mencionado de un TrainingSession desde un usuario con rol de sponsor, a lo cual el sistema niega la operación mostrando código 500.

4.2.4. Update

Para probar la actualización, he accedido a los detalles de un TrainingSession y, teniendo en cuenta las restricciones descritas para cada atributo en el Create, he probado de la misma forma, con todos los valores, además de con un formulario vacío (también ha tenido la misma respuesta que en el Create). El código, al haber sido considerado un identificador del objeto para el usuario, no se puede modificar en el update, por tanto solo se puede leer. El código del TrainingModule, al igual que en Create, sólo se puede leer. Para probar la actualización, he procedido al igual que en Create, enviando primero el formulario vacío, y luego rellenando los campos de igual forma; el resultado obtenido ha sido exactamente el mismo que el comentado en el Create.

Además, el botón Update no se muestra para aquellos TrainingSession ya publicados. Para intentar forzar esto, he probado a intentar actualizar un TrainingSession ya publicado accediendo a sus

detalles y haciendo una petición al endpoint */developer/training-session/update*, a lo cual el sistema ha respondido con error 500, ya que no se puede actualizar un TrainingSession ya publicado.

Se ha hecho también un intento de POST Hacking al endpoint */developer/training-session/update* de un TrainingSession desde un usuario con rol de sponsor, a lo cual el sistema niega la operación mostrando código 500.

4.2.5. Publish

Para probar la publicación, he accedido a los detalles de un TrainingSession y, teniendo en cuenta las restricciones descritas para cada atributo en el Create, he probado de la misma forma, con todos los valores y cumpliendo la regla de negocio de la fecha de comienzo y de la fecha de fin, además de con un formulario vacío (también ha tenido la misma respuesta que en el Create). Si algún atributo no cumple con esas restricciones, se muestra un error descriptivo en el campo correspondiente. Con todos los datos correctos, el sistema permite publicar el TrainingModule si pulsamos en el botón Publish, lo que hace una petición al endpoint */developer/training-session/publish*.

Además, el botón Publish no se muestra para aquellos objetos ya publicados. Para intentar forzar esto, he probado a intentar publicar un TrainingSession ya publicado accediendo a sus detalles y haciendo una petición al endpoint */developer/training-session/publish*, a lo cual el sistema ha respondido con error 500, ya que no se puede publicar un TrainingSession ya publicado.

Se ha hecho también un intento de POST Hacking al endpoint */developer/training-session/publish* de un TrainingModule desde un usuario con rol de sponsor, a lo cual el sistema niega la operación mostrando código 500.

4.2.6. Delete

















Para probar la eliminación, he accedido a los detalles de un TrainingSession y, sin necesidad de tener en cuenta ninguna de las restricciones descritas en el Create, se puede eliminar sin problemas una TrainingSession siempre que no esté publicada. Al probar borrar un TrainingSession no publicado, éste se borra exitosamente. En los detalles de los que ya están publicados, el botón para eliminar está deshabilitado, por tanto, para forzar la operación, me he metido en uno ya publicado y he hecho una petición a su endpoint, */developer/training-session/publish*, lo que devuelve código 500, denegando la petición.

Se ha hecho también un intento de POST Hacking al endpoint */developer/training-session/delete* de un TrainingModule desde un usuario con rol de sponsor, a lo cual el sistema niega la operación mostrando código 500.

















4.3. Cobertura

En esta sección se analiza la cobertura del código lograda tras la ejecución de los tests de los requisitos grupales.

4.3.1. Cobertura de TrainingModule

▼  acme.features.developer.trainingmodule	 88,4 %	1.403	185	1.588
>  DeveloperTrainingModuleDeleteService.java	 64,6 %	168	92	260
>  DeveloperTrainingModuleUpdateService.java	 92,2 %	343	29	372
>  DeveloperTrainingModulePublishService.java	 91,5 %	303	28	331
>  DeveloperTrainingModuleCreateService.java	 91,3 %	283	27	310
>  DeveloperTrainingModuleShowService.java	 97,6 %	200	5	205
>  DeveloperTrainingModuleListService.java	 94,7 %	71	4	75
>  DeveloperTrainingModuleController.java	 100,0 %	35	0	35

4.3.2. Cobertura de TrainingSession

▼  acme.features.developer.trainingsession	 89,7 %	1.338	153	1.491
>  DeveloperTrainingSessionDeleteService.java	 64,0 %	155	87	242
>  DeveloperTrainingSessionUpdateService.java	 93,8 %	288	19	307
>  DeveloperTrainingSessionPublishService.java	 93,8 %	270	18	288
>  DeveloperTrainingSessionCreateService.java	 95,0 %	302	16	318
>  DeveloperTrainingSessionListService.java	 95,4 %	165	8	173
>  DeveloperTrainingSessionShowService.java	 96,1 %	123	5	128
>  DeveloperTrainingSessionController.java	 100,0 %	35	0	35

4.3.3. Análisis de la cobertura

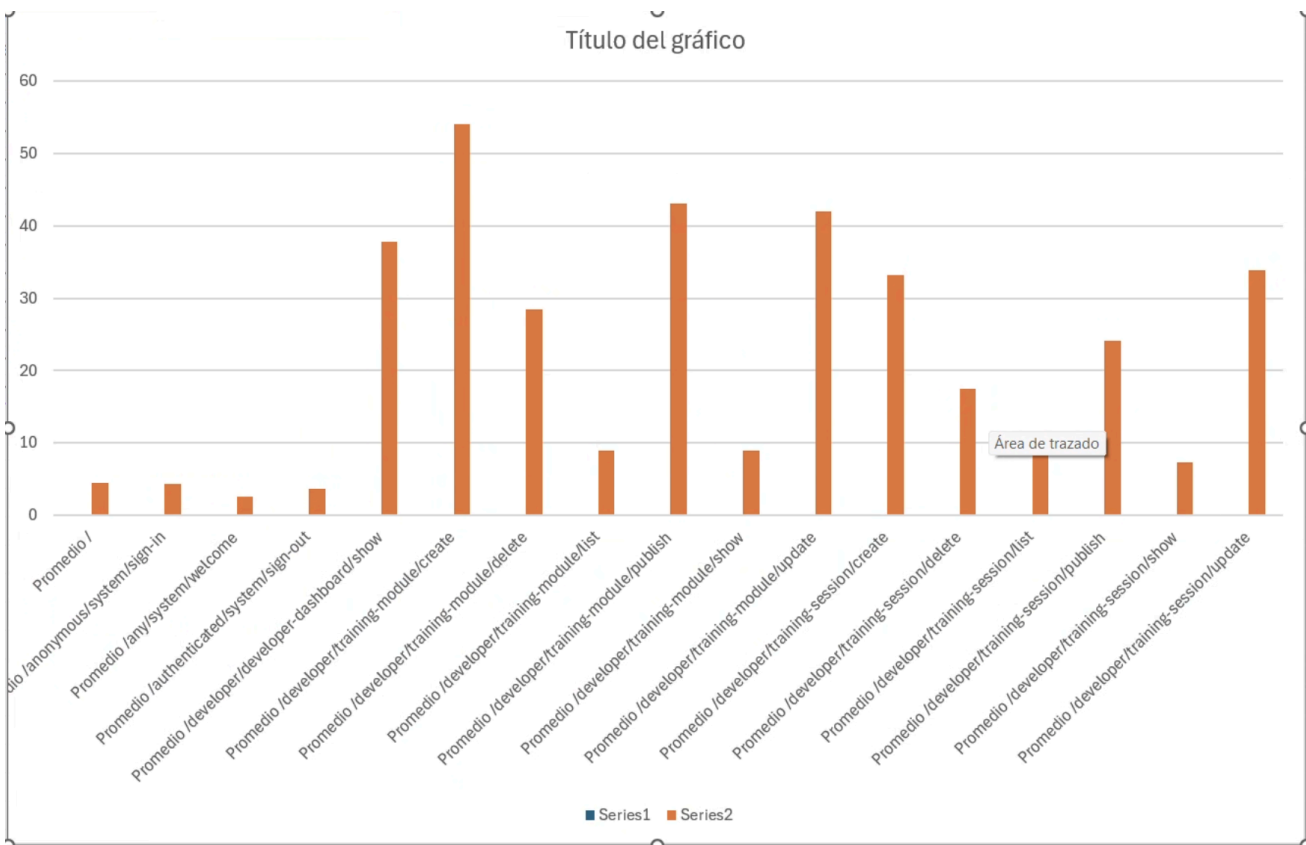
En ambos casos la cobertura general de las entidades supera el 90%. Lo mismo ocurre con los distintos servicios, controladores y repositorios a excepción de los servicios de borrado, cuyo método *unbind* no ha podido ser probado puesto que por la construcción del sistema, no ha sido posible ejecutar el borrado de una TrainingModule o TrainingSession y que dé error. El resto de cobertura que falta se debe también a la construcción del sistema, siendo imposible en la mayoría de los casos poder probar las funcionalidades sobre un objeto nulo (`assert object != null`).

4.4. Test de rendimiento

En este capítulo se evalúa el rendimiento de las funcionalidades implementadas . Para ello se analizarán los tiempos de respuesta a las peticiones realizadas en las pruebas funcionales. Para comparar resultados y evaluar el rendimiento en dos entornos distintos se utilizarán dos pruebas con la diferencia de que una de ellas implementa índices en las entidades para intentar optimizar las consultas a la base de datos.

Para realizar el análisis de estas pruebas, se presentan gráficos que ilustran los resultados obtenidos. Además, se han calculado intervalos de confianza del 95% para los tiempos de respuesta medidos y se realiza un contraste de hipótesis, también con un intervalo de confianza del 95%.

4.4.1. Gráficas de entidades sin índices



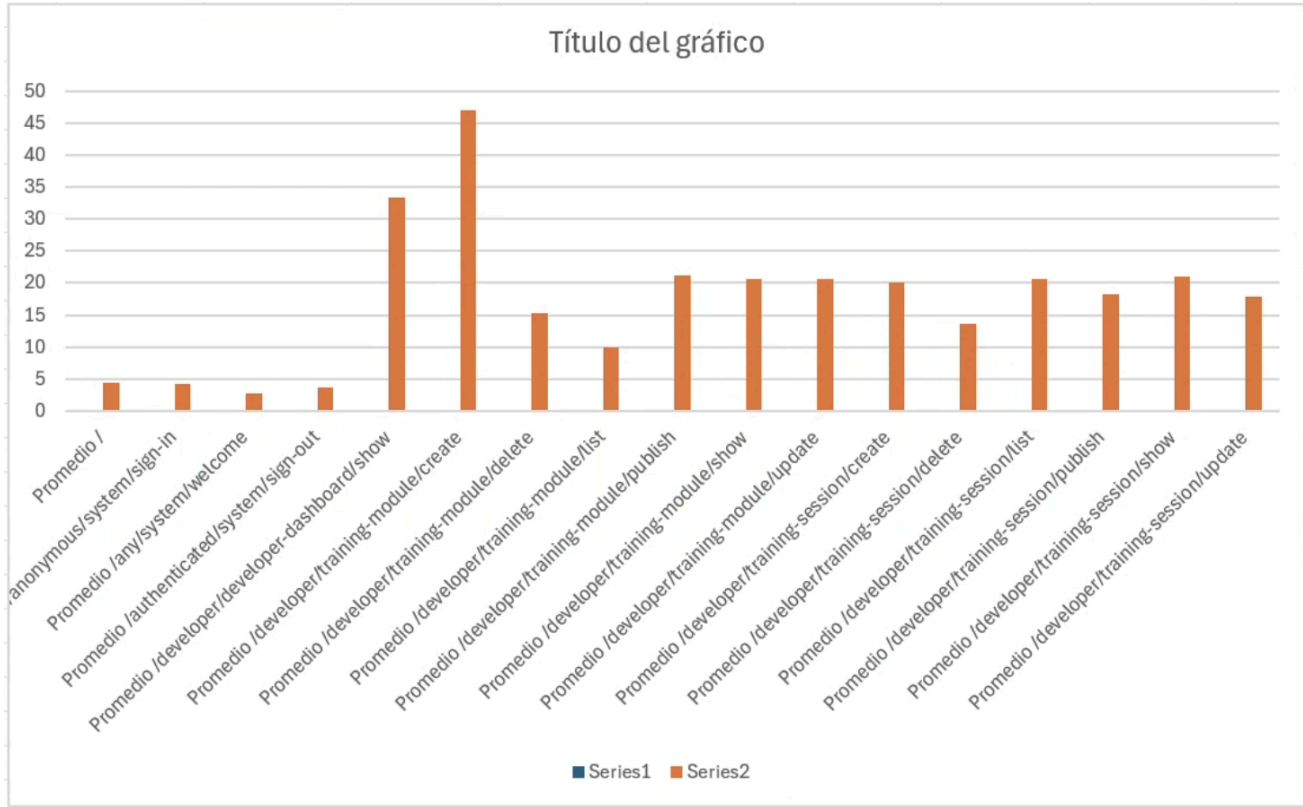
En el gráfico el eje Y mide el tiempo en milisegundos que toma, en promedio, cada tipo de petición del eje X.

Antes					
			Interval (ms)	10,5749546	14,3148205
Media	12,44488756		Interval (s)	0,01057495	0,01431482
Error típico	0,950784555				
Mediana	5,0594				
Moda	37,7963				
Desviación estándar	17,86361997				
Varianza de la muestra	319,1089186				
Curtosis	9,575695038				
Coeficiente de asimetría	2,770338878				
Rango	131,6118				
Mínimo	0				
Máximo	131,6118				
Suma	4393,04531				
Cuenta	353				
Nivel de confianza(95,0%)	1,869932924				

De estos datos, cabe destacar el intervalo de confianza al 95% (10,57495464, 14,31482049) en milisegundos. También vemos que la media es de 12,44488756 milisegundos.

4.4.2. Gráficas de entidades con índices

Tras haber visto los resultados de las entidades sin índice, ahora visualizamos los obtenidos tras haber incluido dos índices en la entidad TrainingModule y uno en TrainingSession.



<i>Después</i>					
			Interval (ms)	10,2474855	13,0779753
Media	11,6627304		Interval (s)	0,01024749	0,01307798
Error típico	0,71951415				
Mediana	5,5035				
Moda	20,9599				
Desviación es	13,3061494				
Varianza de la	177,053611				
Curtosis	23,2173354				
Coeficiente de	3,64993754				
Rango	125,3302				
Mínimo	1,452				
Máximo	126,7822				
Suma	3988,6538				
Cuenta	342				
Nivel de confi	1,41524486				

En la segunda muestra (con índices) se observa un intervalo de confianza de [10,24748555; 13,07797527].

4.4.3. Comparación de resultados: intervalos de confianza

Antes				
			Interval (ms)	10,5749546
Media	12,4448876		Interval (s)	0,01057495
Error típico	0,95078456			14,3148205
Mediana	5,0594			0,01431482
Moda	37,7963			
Desviación es	17,86362			
Varianza de la	319,108919			
Curtosis	9,57569504			
Coeficiente de	2,77033888			
Rango	131,6118			
Mínimo	0			
Máximo	131,6118			
Suma	4393,04531			
Cuenta	353			
Nivel de confi:	1,86993292			
Después				
			Interval (ms)	10,2474855
Media	11,6627304		Interval (s)	13,0779753
Error típico	0,71951415			0,01024749
Mediana	5,5035			0,01307798
Moda	20,9599			
Desviación es	13,3061494			
Varianza de la	177,053611			
Curtosis	23,2173354			
Coeficiente de	3,64993754			
Rango	125,3302			
Mínimo	1,452			
Máximo	126,7822			
Suma	3988,6538			
Cuenta	342			
Nivel de confi:	1,41524486			

4.4.4. Comparación de resultados: hipótesis de contraste

Prueba z para medias de dos muestras		
	Antes	Después
Media	11,9886295	11,6627304
Varianza (conocida)	319,108919	177,053611
Observaciones	336	342
Diferencia hipotética de las medias	0	
z	0,26903233	
P(Z<=z) una cola	0,3939524	
Valor crítico de z (una cola)	1,64485363	
Valor crítico de z (dos colas)	0,7879048	
Valor crítico de z (dos colas)	1,95996398	

5. Conclusión.

En conclusión, este documento describe los tests realizados en el sistema para las funcionalidades implementadas para las entidades TrainingModule y TrainingSession, así como un breve análisis de la cobertura de las pruebas para sus servicios.

6. Bibliografía.

Intencionadamente en blanco.