



# INFORME DE TESTING

Ignacio Blanquero Blanco

**Grupo:** C2.026

- Ignacio Blanquero Blanco ([ignblabla@alum.us.es](mailto:ignblabla@alum.us.es))
- Adrián Cabello Martín ([adrcabmar@alum.us.es](mailto:adrcabmar@alum.us.es))
- María de la Salud Carrera Talaverón ([marcartal1@alum.us.es](mailto:marcartal1@alum.us.es))
- Joaquín González Ganfornina ([joagongan@alum.us.es](mailto:joagongan@alum.us.es))
- Natalia Olmo Villegas ([natolmvil@alum.us.es](mailto:natolmvil@alum.us.es))

**Repositorio:** <https://github.com/maryycarrera/Acme-SF-D04>

**Fecha:** 07 de julio de 2024

## Tabla de contenido

Resumen ejecutivo .....	3
Historial de versiones.....	4
Introducción .....	5
Contenido .....	6
Pruebas funcionales .....	6
TrainingModule .....	6
Cobertura TrainingModule .....	9
TrainingSession .....	10
Cobertura TrainingSession .....	15
Testing de rendimiento .....	16
Análisis del rendimiento .....	16
Contraste de hipótesis .....	20
Conclusiones .....	21
Bibliografía .....	22

## Resumen ejecutivo

El objetivo de este informe es realizar un análisis exhaustivo de las pruebas efectuadas para evaluar el funcionamiento de dos de los requisitos obligatorios establecidos en la cuarta entrega del proyecto Acme-SF-D04, correspondiente a la asignatura Diseño y Pruebas II. Este proyecto tiene como finalidad mejorar las competencias en el desarrollo web, con un enfoque particular en la implementación de un sistema de información de tamaño moderado.

## Historial de versiones

Versión	Fecha	Descripción
v1.0	03/07/2024	Estructura inicial Informe de Testing.
v1.1	04/07/2024	Redacción pruebas .safe para los módulos de entrenamiento.
V1.2	05/07/2024	Redacción pruebas .safe para las sesiones de entrenamiento.
v1.3	06/07/2024	Redacción pruebas .hack para los módulos y las sesiones de entrenamiento.
v1.4	07/07/2024	Redacción testing de rendimiento y conclusiones.

## Introducción

Este informe tiene como propósito proporcionar un análisis detallado de las pruebas realizadas en dos de los módulos requeridos en la cuarta entrega del proyecto Acme-SF-D04 para la asignatura Diseño y Pruebas II. El objetivo principal de este proyecto es mejorar las competencias de los estudiantes en el desarrollo web, enfocándose en la implementación y prueba de un sistema de información de tamaño moderado. Para ello, se han diseñado y ejecutado una serie de pruebas tanto funcionales como de rendimiento, con el fin de evaluar la robustez y fiabilidad del sistema desarrollado.

El análisis de los resultados obtenidos en las pruebas de rendimiento permite, además, evaluar la eficiencia del sistema bajo diversas cargas de trabajo, asegurando que pueda soportar adecuadamente el uso previsto sin degradar su rendimiento. Este aspecto es crucial para garantizar una experiencia de usuario satisfactoria y para asegurar que el sistema puede escalar adecuadamente con el aumento en la demanda.

El presente documento se divide en las siguientes secciones: **Resumen ejecutivo**, el cual proporciona una visión general del proyecto y sus objetivos; **Historial de versiones**, que incluye un registro detallado de los cambios y versiones del documento; **Introducción**, que ofrece una descripción general del contenido y la estructura del documento; **Pruebas funcionales**, donde se detallan los diferentes casos de prueba diseñados y ejecutados para evaluar la funcionalidad de los módulos desarrollados, describiendo tanto los casos positivos como negativos, los intentos de hackeo y los bugs encontrados; **Cobertura del código**, que presenta un análisis gráfico de la cobertura del código alcanzada durante las pruebas, desglosado por cada uno de los servicios implementados en los módulos de entrenamiento y sesiones de entrenamiento; **Testing de rendimiento**, donde se analizan los resultados de las pruebas de rendimiento, incluyendo el análisis de los tiempos de respuesta y la capacidad del sistema para manejar diferentes niveles de carga; **Análisis del rendimiento**, en el que se realiza un contraste de hipótesis para validar los resultados obtenidos y asegurar que el sistema cumple con los requisitos de rendimiento especificados; **Conclusiones**, donde se presentan las conclusiones del informe, destacando los aspectos más relevantes de las pruebas realizadas y proponiendo posibles mejoras para futuras iteraciones del proyecto; y **Bibliografía**, que incluye una lista de las fuentes y referencias consultadas para la realización de este informe.

## Contenido

### Pruebas funcionales

A continuación, se presenta un listado estructurado de casos de prueba, clasificados según características específicas. Cada caso de prueba contiene una descripción concisa de la funcionalidad evaluada, junto con una evaluación precisa de su capacidad para detectar posibles errores en el sistema implementado.

#### TrainingModule

##### *DeveloperTrainingModuleCreateService*

- **Create.safe:**
  - **Casos de pruebas positivos:** Crear múltiples módulos de entrenamiento autenticados utilizando los roles de 'developer1' y 'developer2', asegurando que los datos sean correctos y no violen ninguna restricción del sistema.
  - **Casos de pruebas negativos:**
    - Formulario vacío.
    - Crear módulos de entrenamiento que prueben exhaustivamente todas las restricciones aplicables a los campos. Esto incluye: Valores nulos en los campos que son obligatorios, código duplicado, tamaño de cadena excesivo, fechas que no cumplen con los requisitos de creación y actualización, valores negativos y formato incorrecto para los enlaces.
- **Create.hack:**
  - **Rol incorrecto:** Acceder al formulario de creación de módulos de entrenamiento sin estar logueado (rol: Anonymous) y con un rol distinto al de 'developer'. En este último caso se ha probado con los roles 'administrator' y 'client1'.
- **Bugs:** No se han encontrado bugs en este servicio.

##### *DeveloperTrainingModuleDeleteService*

- **Delete.safe:**
  - **Casos de pruebas positivos:** Eliminar los módulos de entrenamiento que no contengan sesiones de entrenamientos y aquellos cuyas sesiones estén todas publicadas, con los roles de 'developer1' y 'developer2'.
  - **Casos de pruebas negativas:** No hay acciones negativas que puedan llevarse a cabo y que resulten en errores de lógica de negocio.
- **Delete.hack:**
  - **Rol incorrecto:** Copiar la URL para la eliminación de un módulo de entrenamiento y ejecutarla con un rol diferente al de 'developer'. En este caso, se ha probado sin iniciar sesión (rol: Anonymous) y con los roles 'administrator' y 'client1'; lo que ha resultado en una vista de pánico.
  - **Rol correcto:**
    - Eliminar un módulo de entrenamiento ya publicado (acción ilegal).
    - Eliminar un módulo de entrenamiento de otro desarrollador (acción ilegal).
    - Eliminar un módulo de entrenamiento que no existe.
- **Bugs:** No se han encontrado bugs en este servicio.

#### *DeveloperTrainingModuleListMineService*

- **List-mine.safe:**
  - **Casos de pruebas positivos:** Listar los módulos de entrenamiento autenticados con los roles 'developer1' y 'developer2'.
  - **Casos de pruebas negativos:** No hay acciones negativas que puedan llevarse a cabo y que resulten en errores de lógica de negocio.
- **List-mine.hack:**
  - **Rol incorrecto:** Listar módulos de entrenamiento sin estar logueado (rol: Anonymous) y con los roles 'administrator' y 'client1'.
- **Bugs:** No se han encontrado bugs en este servicio.

#### *DeveloperTrainingModulePublishService*

- **Publish.safe:**
  - **Casos de pruebas positivos:** Actualizar los módulos de entrenamiento que contengan todas sus sesiones publicadas, utilizando datos correctos y autenticados mediante los roles 'developer1' y 'developer2'.
  - **Casos de pruebas negativos:**
    - Formulario vacío.
    - Crear módulos de entrenamiento que prueben exhaustivamente todas las restricciones aplicables a los campos. Esto incluye: Valores nulos en los campos que son obligatorios, código duplicado, tamaño de cadena excesivo, fechas que no cumplen con los requisitos de creación y actualización, valores negativos y formato incorrecto para los enlaces.
- **Publish.hack:**
  - **Rol incorrecto:** Copiar la URL para la publicación de un módulo de entrenamiento y ejecutarla con un rol diferente al de 'developer'. En este caso, se ha probado sin iniciar sesión (rol: Anonymous) y con los roles 'administrator' y 'client1'; lo que ha resultado en una vista de pánico.
  - **Rol correcto:**
    - Publicar un módulo de entrenamiento ya publicado (acción ilegal).
    - Publicar un módulo de entrenamiento de otro desarrollador (acción ilegal).
    - Publicar un módulo de entrenamiento que no existe.
- **Bugs:** No se han encontrado bugs en este servicio.

#### *DeveloperTrainingModuleShowService*

- **Show.safe:**
  - **Casos de pruebas positivos:** Mostrar los módulos de entrenamiento autenticados con los roles de 'developer1' y 'developer2', sin importar su estado de publicación.
  - **Casos de pruebas negativos:** No hay acciones negativas que puedan llevarse a cabo y que resulten en errores de lógica de negocio.
- **Show.hack:**
  - **Rol incorrecto:** Copiar la URL para la visualización de un módulo de entrenamiento y ejecutarla con un rol diferente al de 'developer'. En este caso, se ha probado sin iniciar sesión (rol: Anonymous) y con los roles 'administrator' y 'client1'; lo que ha resultado en una vista de pánico.
  - **Rol correcto:**

- Mostrar un módulo de entrenamiento ya publicado de otro desarrollador.
  - Mostrar un módulo de entrenamiento no publicado de otro desarrollador.
  - Mostrar un módulo de entrenamiento que no existe.
- **Bugs:** No se han encontrado bugs en este servicio.



#### *DeveloperTrainingModuleUpdateService*

- **Update.safe:**
  - **Casos de pruebas positivos:** Actualizar los módulos de entrenamiento utilizando datos correctos y autenticados mediante los roles 'developer1' y 'developer2'.
  - **Casos de pruebas negativos:**
    - Formulario vacío.
    - Actualizar módulos de entrenamiento que prueben exhaustivamente todas las restricciones aplicables a los campos. Esto incluye: Valores nulos en los campos que son obligatorios, código duplicado, tamaño de cadena excesivo, fechas que no cumplen con los requisitos de creación y actualización, valores negativos y formato incorrecto para los enlaces.
- **Publish.hack:**
  - **Rol incorrecto:** Copiar la URL para la publicación de un módulo de entrenamiento y ejecutarla con un rol diferente al de 'developer'. En este caso, se ha probado sin iniciar sesión (rol: Anonymous) y con los roles 'administrator' y 'client1'; lo que ha resultado en una vista de pánico.
  - **Rol correcto:**
    - Publicar un módulo de entrenamiento ya publicado (acción ilegal).
    - Publicar un módulo de entrenamiento de otro desarrollador (acción ilegal).
    - Publicar un módulo de entrenamiento que no existe.
- **Bugs:** No se han encontrado bugs en este servicio.



















## Cobertura TrainingModule

A continuación, se muestra un resumen gráfico de la cobertura del paquete "acme.features.developer.trainingModule":

 acme.features.developer.trainingModule	 95,0 %	1.357	72	1.429
--	--	-------	----	-------

*Ilustración 1. Cobertura total TrainingModule*

▼  acme.features.developer.trainingModule	 95,0 %	1.357	72	1.429
>  DeveloperTrainingModuleController.java	 100,0 %	36	0	36
>  DeveloperTrainingModuleCreateService.java	 93,5 %	232	16	248
>  DeveloperTrainingModuleDeleteService.java	 93,9 %	245	16	261
>  DeveloperTrainingModuleListMineService.java	 97,1 %	136	4	140
>  DeveloperTrainingModulePublishService.java	 95,1 %	312	16	328
>  DeveloperTrainingModuleShowService.java	 97,2 %	138	4	142
>  DeveloperTrainingModuleUpdateService.java	 94,2 %	258	16	274

*Ilustración 2. Cobertura servicios TrainingModule*

Las imágenes proporcionadas presentan la cobertura de código del paquete "acme.features.developer.trainingModule" y sus clases asociadas. La cobertura global es del 95%, con un total de 1357 líneas cubiertas de 1429 líneas totales, lo cual es bastante satisfactorio. A continuación, se desglosa la cobertura de las clases individuales:

- **DeveloperTrainingModuleController.java:** 100.0% de cobertura, con 36 de 36 líneas cubiertas.
- **DeveloperTrainingModuleCreateService.java:** 93.5% de cobertura, con 232 de 248 líneas cubiertas.
- **DeveloperTrainingModuleDeleteService.java:** 93.9% de cobertura, con 245 de 261 líneas cubiertas.
- **DeveloperTrainingModuleListMineService.java:** 97.1% de cobertura, con 136 de 140 líneas cubiertas.
- **DeveloperTrainingModulePublishService.java:** 95.1% de cobertura, con 312 de 328 líneas cubiertas.
- **DeveloperTrainingModuleShowService.java:** 97.2% de cobertura, con 138 de 142 líneas cubiertas.
- **DeveloperTrainingModuleUpdateService.java:** 94.2% de cobertura, con 258 de 274 líneas cubiertas.

## TrainingSession

### DeveloperTrainingModuleCreateService

- **Create.safe:**
  - **Casos de pruebas positivos:** Crear múltiples sesiones de entrenamiento autenticados utilizando los roles de 'developer1' y 'developer2', asegurando que los datos sean correctos y no violen ninguna restricción del sistema.
  - **Casos de pruebas negativos:**
    - Formulario vacío.
    - Crear sesiones de entrenamiento que prueben exhaustivamente todas las restricciones aplicables a los campos. Esto incluye: Valores nulos en los campos que son obligatorios, código duplicado, tamaño de cadena excesivo, fechas que no cumplen con los requisitos de creación y duración y formato incorrecto para los enlaces y las direcciones de correo electrónico.
- **Create.hack:**
  - **Rol incorrecto:** Acceder al formulario de creación de módulos de entrenamiento sin estar logueado (rol: Anonymous) y con un rol distinto al de 'developer'. En este último caso se ha probado con los roles 'administrator' y 'client1'.
  - **Rol correcto:**
    - Crear sesiones de entrenamiento en un módulo ya publicado (acción ilegal).
    - Crear sesiones de entrenamiento en un módulo no publicado perteneciente a otro desarrollador (acción ilegal).
    - Crear sesiones de entrenamientos pertenecientes a un módulo de entrenamiento inexistente.
- **Bugs:** Se ha identificado un error que permite la creación de una sesión de entrenamiento perteneciente a otro desarrollador. Este problema ocurre al inspeccionar el código HTML y modificar el valor del identificador (id) del módulo de entrenamiento correspondiente.
- **Solución:** Añadir al método `authorise()` de la clase "DeveloperTrainingSessionCreateService" una restricción que verifique que el módulo de entrenamiento no sea nulo, que sea editable y que el ID del desarrollador autenticado coincida con el ID del desarrollador propietario de dicho módulo de entrenamiento.

```
status = trainingModule != null && trainingModule.isDraftMode() && developerId == trainingModule.getDeveloper().getId();
```

*Ilustración 3. Solución método `authorise()` - Create service*

### DeveloperTrainingSessionDeleteService

- **Delete.safe:**
  - **Casos de pruebas positivos:** Eliminar las sesiones de entrenamiento pertenecientes a un módulo que no esté publicado, con los roles de 'developer1' y 'developer2'.
  - **Casos de pruebas negativas:** No hay acciones negativas que puedan llevarse a cabo y que resulten en errores de lógica de negocio.
- **Delete.hack:**
  - **Rol incorrecto:** Copiar la URL para la eliminación de una sesión de entrenamiento y ejecutarla con un rol diferente al de 'developer'. En este caso,

se ha probado sin iniciar sesión (rol: Anonymous) y con los roles 'administrator' y 'client1'; lo que ha resultado en una vista de pánico.

- **Rol correcto:**
  - Eliminar una sesión de entrenamiento ya publicado (acción ilegal).
  - Eliminar una sesión de entrenamiento de otro desarrollador (acción ilegal).
  - Eliminar una sesión de entrenamiento que no existe.
- **Bugs:** Se ha identificado un error que permite la eliminación de una sesión de entrenamiento perteneciente a otro desarrollador que no está publicada. Este problema ocurre al inspeccionar el código HTML y modificar el valor del identificador (id) de la sesión de entrenamiento correspondiente.
- **Solución:** Añadir al método `authorise()` de la clase "DeveloperTrainingSessionDeleteService" una restricción que verifique que tanto el módulo como la sesión no sean nulos, que sean editables, y que el ID del desarrollador autenticado coincida con el ID del desarrollador propietario tanto del módulo como de la sesión de entrenamiento.

```
boolean moduleValid = module != null && module.isDraftMode() && developerId == module.getDeveloper().getId();
boolean sessionValid = session != null && session.isDraftMode() && developerId == session.getTrainingModule().getDeveloper().getId();
```

*Ilustración 4. Solución método `authorise()` - Delete service*

#### *DeveloperTrainingSessionListService*

- **List.safe:**
  - **Casos de pruebas positivos:** Listar las sesiones de entrenamiento autenticados con los roles 'developer1' y 'developer2'.
  - **Casos de pruebas negativos:** No hay acciones negativas que puedan llevarse a cabo y que resulten en errores de lógica de negocio.
- **List.hack:**
  - **Rol incorrecto:** Listar módulos de entrenamiento sin estar logueado (rol: Anonymous) y con los roles 'administrator' y 'client1'.
  - **Rol correcto:**
    - Listar sesiones de entrenamiento pertenecientes a otro desarrollador, independientemente de si están publicadas o no.
    - Listar sesiones de entrenamiento inexistentes.
- **Bugs:** Se ha identificado un error que permite visualizar el listado de sesiones de entrenamiento pertenecientes a otro desarrollador, independientemente de su estado de publicación. Este problema ocurre al inspeccionar el código HTML y modificar el valor del identificador (id) del módulo de entrenamiento del que se quieren listar sus sesiones.
- **Solución:** Añadir al método `authorise()` de la clase "DeveloperTrainingSessionListService" una restricción que verifique que el módulo no sea nulo y que el ID del desarrollador autenticado coincida con el ID del desarrollador propietario de dicho módulo de entrenamiento.

```
status = module != null && developerId == module.getDeveloper().getId();
```

*Ilustración 5. Solución método `authorise()` - List service*

### *DeveloperTrainingSessionPublishService*

- **Publish.safe:**
  - **Casos de pruebas positivos:** Publicar sesiones de entrenamiento cuyo módulo de entrenamiento no haya sido publicado, utilizando datos correctos y autenticados mediante los roles 'developer1' y 'developer2'.
  - **Casos de pruebas negativos:**
    - Formulario vacío.
    - Publicar sesiones de entrenamiento que prueben exhaustivamente todas las restricciones aplicables a los campos. Esto incluye: Valores nulos en los campos que son obligatorios, código duplicado, tamaño de cadena excesivo, fechas que no cumplen con los requisitos de creación y duración y formato incorrecto para los enlaces y las direcciones de correo electrónico.
- **Publish.hack:**
  - **Rol incorrecto:** Copiar la URL para la publicación de una sesión de entrenamiento y ejecutarla con un rol diferente al de 'developer'. En este caso, se ha probado sin iniciar sesión (rol: Anonymous) y con los roles 'administrator' y 'client1'; lo que ha resultado en una vista de pánico.
  - **Rol correcto:**
    - Publicar una sesión de entrenamiento ya publicada (acción ilegal).
    - Publicar una sesión de entrenamiento de otro desarrollador (acción ilegal).
    - Publicar una sesión de entrenamiento que no existe.
- **Bugs:** Se ha identificado un error que permite publicar una sesión de entrenamiento pertenecientes a otro desarrollador. Este problema ocurre al inspeccionar el código HTML y modificar el valor del identificador (id) de la sesión de entrenamiento que se quiere publicar.
- **Solución:** Añadir al método `authorise()` de la clase "DeveloperTrainingSessionPublishService" una restricción que verifique que tanto el módulo como la sesión no sean nulos, que sean editables, y que el ID del desarrollador autenticado coincida con el ID del desarrollador propietario tanto del módulo como de la sesión de entrenamiento.

```
boolean moduleValid = module != null && module.isDraftMode() && developerId == module.getDeveloper().getId();
boolean sessionValid = session != null && session.isDraftMode() && developerId == session.getTrainingModule().getDeveloper().getId();
```

*Ilustración 6. Solución método authorise() - Publish service*

### *DeveloperTrainingSessionShowService*

- **Show.safe:**
  - **Casos de pruebas positivos:** Mostrar sesiones de entrenamiento autenticados con los roles de 'developer1' y 'developer2', sin importar su estado de publicación.
  - **Casos de pruebas negativos:** No hay acciones negativas que puedan llevarse a cabo y que resulten en errores de lógica de negocio.
- **Show.hack:**
  - **Rol incorrecto:** Copiar la URL para la visualización de una sesión de entrenamiento y ejecutarla con un rol diferente al de 'developer'. En este caso, se ha probado sin iniciar sesión (rol: Anonymous) y con los roles 'administrator' y 'client1'; lo que ha resultado en una vista de pánico.

- **Rol correcto:**
  - Mostrar una sesión de entrenamiento de otro desarrollador, independientemente de su estado de publicación.
  - Mostrar una sesión de entrenamiento que no existe.
- **Bugs:** Se ha identificado un error que permite mostrar información de una sesión de entrenamiento perteneciente a otro desarrollador. Este problema ocurre al cambiar el ID de la sesión de entrenamiento que se desea mostrar por otro que no pertenece al desarrollador autenticado.
- **Solución:** Añadir al método `authorise()` de la clase “DeveloperTrainingSessionShowService” una restricción que verifique que la sesión no sea nula y que el ID del desarrollador autenticado coincida con el ID del desarrollador propietario de dicha sesión de entrenamiento.

```
status = session != null && developerId == session.getTrainingModule().getDeveloper().getId();
```

*Ilustración 7. Solución método `authorise()` - Show service*

#### *DeveloperTrainingSessionUpdateService*

- **Update.safe:**
  - **Casos de pruebas positivos:** Actualizar las sesiones de entrenamiento no publicadas cuyo módulo de entrenamiento tampoco haya sido publicado, utilizando datos correctos y autenticados mediante los roles ‘developer1’ y ‘developer2’.
  - **Casos de pruebas negativos:**
    - Formulario vacío.
    - Publicar sesiones de entrenamiento que prueben exhaustivamente todas las restricciones aplicables a los campos. Esto incluye: Valores nulos en los campos que son obligatorios, código duplicado, tamaño de cadena excesivo, fechas que no cumplen con los requisitos de creación y duración y formato incorrecto para los enlaces y las direcciones de correo electrónico.
- **Publish.hack:**
  - **Rol incorrecto:** Copiar la URL para la actualización de una sesión de entrenamiento y ejecutarla con un rol diferente al de ‘developer’. En este caso, se ha probado sin iniciar sesión (rol: Anonymous) y con los roles ‘administrator’ y ‘client1’; lo que ha resultado en una vista de pánico.
  - **Rol correcto:**
    - Actualizar una sesión de entrenamiento ya publicada (acción ilegal).
    - Actualizar una sesión de entrenamiento de otro desarrollador (acción ilegal).
    - Actualizar una sesión de entrenamiento que no existe.
- **Bugs:** Se ha identificado un error que permite actualizar una sesión de entrenamiento perteneciente a otro desarrollador. Este problema ocurre al inspeccionar el código HTML y modificar el valor del identificador (id) de la sesión de entrenamiento que se quiere actualizar.
- **Solución:** Añadir al método `authorise()` de la clase “DeveloperTrainingSessionUpdateService” una restricción que verifique que tanto el módulo como la sesión no sean nulos, que sean editables, y que el ID del desarrollador

<https://github.com/maryycarrera/Acme-SF-D04>

autenticado coincida con el ID del desarrollador propietario tanto del módulo como de la sesión de entrenamiento.

```
boolean moduleValid = module != null && module.isDraftMode() && developerId == module.getDeveloper().getId();  
boolean sessionValid = session != null && session.isDraftMode() && developerId == session.getTrainingModule().getDeveloper().getId();
```

*Ilustración 8. Solución método `authorise()` - Update service*

## Cobertura TrainingSession

A continuación, se muestra un resumen gráfico de la cobertura del paquete "acme.features.developer.trainingSession":

acme.features.developer.trainingSession	90,8 %	1.286	130	1.416
---	--------	-------	-----	-------

Ilustración 9. Cobertura total TrainingSession

acme.features.developer.trainingSession	90,8 %	1.286	130	1.416
> DeveloperTrainingSessionController.java	100,0 %	35	0	35
> DeveloperTrainingSessionCreateService.java	94,3 %	266	16	282
> DeveloperTrainingSessionDeleteService.java	66,0 %	136	70	206
> DeveloperTrainingSessionListService.java	95,5 %	169	8	177
> DeveloperTrainingSessionPublishService.java	94,6 %	282	16	298
> DeveloperTrainingSessionShowService.java	96,5 %	111	4	115
> DeveloperTrainingSessionUpdateService.java	94,7 %	287	16	303

Ilustración 10. Cobertura servicios TrainingSession

Las imágenes proporcionadas presentan la cobertura de código del paquete "acme.features.developer.trainingSession" y sus clases asociadas. La cobertura global es del 90.8%, con un total de 1286 líneas cubiertas de 1416 líneas totales, lo cual es bastante satisfactorio. A continuación, se desglosa la cobertura de las clases individuales:

- **DeveloperTrainingSessionController.java:** 100.0% de cobertura, con 35 de 35 líneas cubiertas.
- **DeveloperTrainingSessionCreateService.java:** 94.3% de cobertura, con 266 de 282 líneas cubiertas.
- **DeveloperTrainingSessionDeleteService.java:** 66.0% de cobertura, con 136 de 206 líneas cubiertas.
- **DeveloperTrainingSessionListService.java:** 95.5% de cobertura, con 169 de 177 líneas cubiertas.
- **DeveloperTrainingSessionPublishService.java:** 94.6% de cobertura, con 282 de 298 líneas cubiertas.
- **DeveloperTrainingSessionShowService.java:** 96.5% de cobertura, con 111 de 115 líneas cubiertas.
- **DeveloperTrainingSessionUpdateService.java:** 94.7% de cobertura, con 287 de 303 líneas cubiertas.

En el contexto de las sesiones de entrenamiento (TrainingSessions), hemos identificado que la cobertura en el servicio de eliminación es notablemente inferior en comparación con el resto de los servicios. Durante la realización de pruebas para este servicio, se observó la ausencia de casos negativos al intentar eliminar sesiones. Esto se debe a que, siempre que se sigan los procedimientos legales, es posible eliminar una sesión no publicada. Esta particularidad impide que se ejecuten las siguientes líneas de código en ningún caso.

## Testing de rendimiento

### Análisis del rendimiento

#### Primer análisis (sin índices)

Después de realizar los tests y recopilar datos sobre las peticiones ejecutadas, se llevó a cabo un análisis exhaustivo de los resultados en términos de rendimiento de la aplicación, específicamente considerando el tiempo de respuesta de los servicios asociados a las entidades TrainingModule y TrainingSession. A continuación, se presentan los datos del promedio de cada petición. Además, para una mejor comprensión, también se incluye un gráfico de barras ilustrativo.

request-path	response-status	time
Promedio /		3,732782569
Promedio /anonymous/system/sign-in		4,096111194
Promedio /any/system/welcome		2,340888068
Promedio /authenticated/system/sign-out		3,078523881
Promedio /developer/training-module/create		64,05638066
Promedio /developer/training-module/delete		46,57218
Promedio /developer/training-module/list-mine		943,7512045
Promedio /developer/training-module/publish		55,2972
Promedio /developer/training-module/show		11,19927
Promedio /developer/training-module/update		51,71798311
Promedio /developer/training-session/create		32,67739237
Promedio /developer/training-session/delete		28,80358333
Promedio /developer/training-session/list		9,963573288
Promedio /developer/training-session/publish		31,64209828
Promedio /developer/training-session/show		10,55267732
Promedio /developer/training-session/update		30,30202208
Promedio general		86,64834146

Ilustración 11. Tiempo medio de respuesta primer análisis

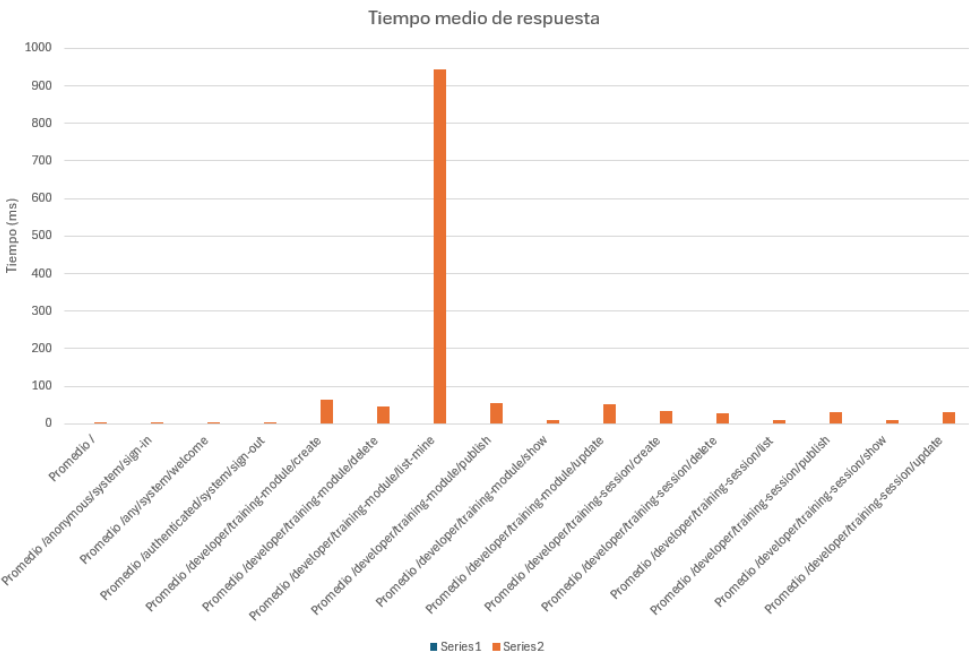


Ilustración 12. Gráfico tiempo medio de respuesta primer análisis



La razón por la cual la petición más costosa del sistema es la de listar módulos de entrenamiento puede explicarse a través de varios factores clave, dados el contexto y la estructura del sistema. En primer lugar, los módulos de entrenamiento actúan como la entidad principal o nodo padre. Para cualquier operación en las entidades hijas (trainingSession) o para realizar acciones como mostrar, actualizar, eliminar, publicar y crear módulos de entrenamiento, es necesario primero acceder a la lista de módulos de entrenamiento. Esta necesidad de acceso previo implica que esta petición se realice con mucha frecuencia, incrementando así su carga y costo.

En segundo lugar, las relaciones y la carga asociada también contribuyen al alto costo de operación. Si los módulos de entrenamiento tienen múltiples sesiones de entrenamiento asociadas, la consulta para listar los módulos puede incluir la necesidad de cargar información relacionada. Esto puede implicar la ejecución de múltiples subconsultas o uniones en la base de datos, lo cual incrementa significativamente el costo computacional.

En resumen, podemos afirmar que esta petición es la más costosa del sistema, dado que debe ejecutarse antes de cualquier otra interacción con los módulos o sus sesiones asociadas.

A continuación, se presenta una serie de variables estadísticas relativas a los tiempos de todas las peticiones realizadas:

<i>Before</i>		
Media	86,64834146	
Error típico	4,614202538	
Mediana	26,78955	
Moda	2,6109	
Desviación estándar	228,2048552	
Varianza de la muestra	52077,45594	
Curtosis	12,44745421	
Coeficiente de asimetría	3,715669439	
Rango	1530,6746	
Mínimo	1,5078	
Máximo	1532,1824	
Suma	211941,8432	
Cuenta	2446	
Nivel de confianza(95,0%)	9,04814992	
Interval (ms)	77,60019154	95,6964914
Interval (s)	0,077600192	0,09569649

*Ilustración 19. Variables estadísticas primer análisis*

Como se puede observar, gracias al cálculo del intervalo de confianza, se puede afirmar que el tiempo de respuesta de las peticiones de la aplicación se encuentra dentro del rango [0,1], específicamente entre [0.077, 0.095] segundos, cumpliendo así con el requisito de rendimiento establecido. Este intervalo de confianza es muy favorable, ya que se encuentra significativamente por debajo de la sugerencia de referencia de 1 segundo como tiempo máximo, recomendada por la asignatura.

## Segundo análisis (con índices)

Después de revisar las consultas realizadas en los repositorios de las entidades, se han incorporado los siguientes índices en las clases Java correspondientes:

- **TrainingModule:** @Index(columnList = "code"), @Index(columnList = "draftMode"), @Index(columnList = "developer\_id").
- **TrainingSession:** @Index(columnList = "code"), @Index(columnList = "training\_module\_id")

Los resultados obtenidos tras realizar dichas modificaciones han sido:

request-path	response-status	time
Promedio /		3,37233211
Promedio /anonymous/system/sign-in		4,074650746
Promedio /any/system/welcome		2,283185227
Promedio /authenticated/system/sign-out		3,02181791
Promedio /developer/training-module/create		65,00199387
Promedio /developer/training-module/delete		41,458692
Promedio /developer/training-module/list-mine		936,1424955
Promedio /developer/training-module/publish		54,11139766
Promedio /developer/training-module/show		11,35013647
Promedio /developer/training-module/update		50,21151187
Promedio /developer/training-session/create		33,10231737
Promedio /developer/training-session/delete		31,85215
Promedio /developer/training-session/list		10,23649315
Promedio /developer/training-session/publish		31,69007457
Promedio /developer/training-session/show		10,46147526
Promedio /developer/training-session/update		28,96507229
Promedio general		85,90978173

Ilustración 10. Tiempo medio de respuesta segundo análisis

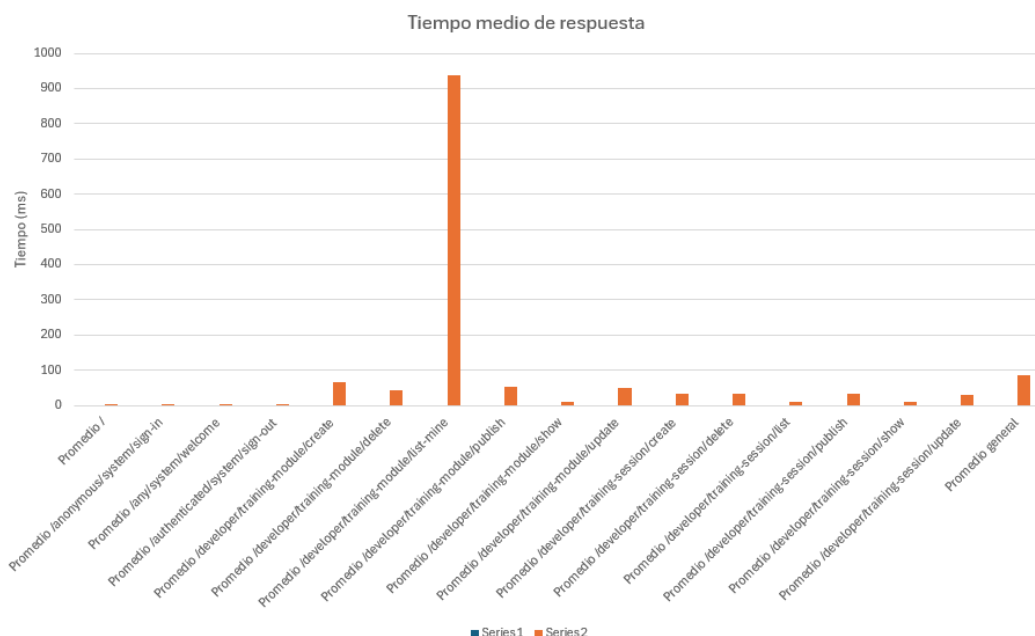


Ilustración 15. Gráfico tiempo medio de respuesta segundo análisis

Se observa que la petición más costosa sigue siendo la de listar los módulos de entrenamiento. Las demás peticiones no parecen haber experimentado cambios significativos en comparación con el análisis inicial realizado.

A continuación, se presenta una serie de variables estadísticas relativas a los tiempos de todas las peticiones realizadas:

<i>After</i>		
Media	85,90978173	
Error típico	4,565215601	
Mediana	26,9436	
Moda	29,617	
Desviación estándar	225,7821057	
Varianza de la muestra	50977,55926	
Curtosis	11,98008256	
Coefficiente de asimetría	3,673922444	
Rango	1350,1742	
Mínimo	1,4404	
Máximo	1351,6146	
Suma	210135,3261	
Cuenta	2446	
Nivel de confianza(95,0%)	8,952089735	
Interval (ms)	76,95769199	94,8618715
Interval (s)	0,076957692	0,09486187

*Ilustración 16. Variables estadísticas segundo análisis*

El análisis del intervalo de confianza indica que el tiempo de respuesta de las peticiones de la aplicación se encuentra en el intervalo  $[0,1]$ , específicamente entre  $[0.076, 0.094]$  segundos. Este intervalo de confianza es significativamente favorable y representa una ligera mejora con respecto al análisis anterior, ya que está considerablemente por debajo de la recomendación de referencia de 1 segundo como tiempo máximo sugerido por la asignatura.

## Contraste de hipótesis

Para comprobar si la refactorización llevada a cabo mediante la inserción de índices en las entidades *TrainingModule* y *TrainingSession* ha tenido un impacto en el rendimiento de la aplicación, se ha llevado a cabo un contraste de hipótesis. Utilizando el complemento de Excel *Herramientas para el análisis*, se ha realizado un Z-test (con  $\alpha = 0.05$ ) para evaluar si el nuevo tiempo medio de respuesta es distinto del original.

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

Before		After		Prueba z para medias de dos muestras	
Media	86,64834146	Media	85,90978173		
Error típico	4,614202538	Error típico	4,565215601		
Mediana	26,78955	Mediana	26,9436		
Moda	2,6109	Moda	29,617		
Desviación estándar	228,2048552	Desviación estándar	225,7821057		
Varianza de la muestra	52077,45594	Varianza de la muestra	50977,55926		
Curtosis	12,44745421	Curtosis	11,98008256		
Coefficiente de asimetría	3,715669439	Coefficiente de asimetría	3,673922444		
Rango	1530,6746	Rango	1350,1742		
Mínimo	1,5078	Mínimo	1,4404		
Máximo	1532,1824	Máximo	1351,6146		
Suma	211941,8432	Suma	210135,3261		
Cuenta	2446	Cuenta	2446		
Nivel de confianza(95,0%)	9,04814992	Nivel de confianza(95,0%)	8,952089735		
Interval (ms)	77,60019154	Interval (ms)	76,95769199		
Interval (s)	0,077600192	Interval (s)	0,076957692		

	Before	After
Media	86,6483415	85,9097817
Varianza (conocida)	52077,4559	50977,5593
Observaciones	2446	2446
Diferencia hipotética de las medias	0	
z	0,1137835	
P(Z<=z) una cola	0,45470471	
Valor crítico de z (una cola)	1,64485363	
Valor crítico de z (dos colas)	0,90940942	
Valor crítico de z (dos colas)	1,95996398	

Ilustración 17. Comparación análisis y Z-Test

Como se puede observar, el valor crítico de z (dos colas) es 0.909. Para determinar si el cambio en el tiempo medio de respuesta es significativo, es necesario comparar este valor de z con  $\alpha$ . En este caso, el valor de z se encuentra dentro del intervalo  $(\alpha, 1]$ , lo cual indica que los cambios realizados mediante la refactorización no han tenido un impacto significativo en el rendimiento medio de la aplicación. Es decir, desde un punto de vista estadístico, no existen evidencias significativas de que los tiempos medios de respuesta sean distintos.

## Conclusiones

En conclusión, el análisis exhaustivo de las pruebas funcionales y de rendimiento llevadas a cabo en los módulos de `TrainingModule` y `TrainingSession` ha proporcionado información valiosa sobre la robustez y eficiencia del sistema desarrollado en el proyecto Acme-SF D04. Las pruebas funcionales confirmaron que los módulos funcionan correctamente bajo diversas condiciones, incluyendo tanto casos positivos como negativos, y que se han identificado diversos bugs.

En cuanto a las pruebas de rendimiento, los resultados indicaron que el tiempo de respuesta de la aplicación se encuentra dentro del rango aceptable, cumpliendo con los requisitos establecidos. El análisis de contraste de hipótesis realizado mediante el Z-test mostró que la refactorización del código, que incluyó la inserción de índices en las entidades, no tuvo impacto significativo en el rendimiento. Aunque se observó una ligera menora en el tiempo medio de respuesta, esta no fue estadísticamente significativa.

En general, el sistema demostró ser robusto y eficiente, capaz de manejar adecuadamente diversas cargas de trabajo sin degradar su rendimiento.

## Bibliografía

- Documentación de la asignatura (08 Annexes).
- Diapositivas teóricas: S01 – Formal testing y S02 – Performance testing.
- Excel proporcionado por los profesores de la asignatura, titulado “Sample-Data”.