

Testing report



Grado en Ingeniería Informática - Ingeniería del Software

Diseño y Pruebas II

Curso 2023 - 2024

| Código de Grupo: C1-001 | | |
|------------------------------|--|---------------|
| Autores por orden alfabético | Correo | Rol |
| Manuel Vélez López | manvellop2@alum.us.es | Desarrollador |

Repositorio: <<https://github.com/DP2-2023-2024-C1-001/Acme-SF-D04>>

Índice de Contenidos

| | |
|------------------------------|----|
| 1. Resumen Ejecutivo..... | 2 |
| 2. Control de Versiones..... | 4 |
| 3. Introducción..... | 5 |
| 4. Contenido..... | 6 |
| 5. Conclusión..... | 32 |
| 6. Bibliografía..... | 33 |

1. Resumen Ejecutivo

El presente informe de pruebas está organizado en dos capítulos, cada uno enfocado en un aspecto crucial del proceso de pruebas de nuestro proyecto: pruebas funcionales y pruebas de rendimiento.

- Capítulo sobre Pruebas Funcionales:

En este capítulo, se presenta una lista detallada de los casos de prueba implementados, agrupados por características específicas del sistema. Para cada caso de prueba, se proporciona una descripción concisa que incluye:

- Descripción del Caso de Prueba:

Un resumen breve de lo que el caso de prueba está evaluando.

- Efectividad en la Detección de Errores:

Una indicación clara de cuán efectivo fue cada caso de prueba para detectar errores en el sistema.

Este capítulo asegura que se cubran todas las funcionalidades críticas del sistema y se documente la capacidad de cada prueba para identificar fallos, proporcionando una visión clara de la robustez funcional del proyecto.

- Capítulo sobre Pruebas de Rendimiento:

El segundo capítulo está dedicado a las pruebas de rendimiento del sistema. Aquí se incluyen gráficos detallados y análisis estadísticos sobre el tiempo de respuesta (wall time) del proyecto al procesar las solicitudes en las pruebas funcionales, ejecutadas en dos computadoras diferentes. Los puntos clave de este capítulo son:

- Gráficos Adecuados:

Visualizaciones que muestran el desempeño del sistema en términos de tiempos de respuesta.

- Intervalo de Confianza del 95%:

Un intervalo de confianza del 95% para los tiempos de respuesta medidos, lo cual proporciona una estimación precisa y confiable del rendimiento.

- Contraste de Hipótesis con un 95% de Confianza:

Un análisis estadístico que compara el rendimiento entre las dos computadoras, determinando con un 95% de confianza cuál es la más potente en términos de procesamiento de las solicitudes.

Este análisis no sólo proporciona datos concretos sobre el rendimiento del sistema, sino que también permite identificar cuál de los dos entornos de hardware evaluados ofrece un mejor desempeño, guiando decisiones futuras sobre la infraestructura necesaria para optimizar el funcionamiento del proyecto.

En conjunto, estos capítulos proporcionan una visión exhaustiva y detallada de la calidad y el rendimiento del sistema probado, asegurando que se han cubierto todas las áreas críticas y que el proyecto cumple con los estándares esperados.

2. Control de Versiones

| Fecha | Versión | Descripción |
|------------|---------|--------------------------------------|
| 26/05/2024 | V1.0 | Creación del informe de testing D04. |
| | | |
| | | |
| | | |

3. Introducción

El objetivo de este informe es documentar de manera exhaustiva los resultados obtenidos a través de las pruebas funcionales y de rendimiento realizadas sobre nuestro proyecto. Las pruebas son una parte fundamental del ciclo de desarrollo de software, ya que garantizan la calidad, fiabilidad y eficiencia del producto final.

En el capítulo sobre pruebas funcionales, se detallan los diferentes casos de prueba implementados, organizados por características específicas del sistema. Cada caso de prueba incluye una descripción breve y precisa, así como una evaluación de su efectividad en la detección de errores. Este enfoque sistemático asegura que todas las funcionalidades críticas del sistema hayan sido evaluadas exhaustivamente, proporcionando una visión clara de su robustez y fiabilidad.

El capítulo sobre pruebas de rendimiento se centra en el análisis del tiempo de respuesta del sistema al procesar solicitudes, utilizando dos configuraciones de hardware diferentes. Se presentan gráficos detallados y un intervalo de confianza del 95% para los tiempos de respuesta medidos. Además, se realiza un contraste de hipótesis con un 95% de confianza para determinar cuál de las dos computadoras evaluadas ofrece un mejor rendimiento. Este análisis no sólo proporciona datos precisos sobre la eficiencia del sistema, sino que también guía futuras decisiones sobre la infraestructura tecnológica necesaria para optimizar el rendimiento.

En conjunto, este informe proporciona una evaluación completa de la calidad y el rendimiento del sistema, asegurando que cumple con los estándares esperados y que está preparado para su implementación en un entorno de producción.

4. Contenido

Capítulo sobre pruebas funcionales

En este capítulo vamos a ver como se han realizado los tests de las funcionalidades ofrecidas al rol de “**developer**”, centrándonos en las entidades “**Training Module**” y “**Training Session**”.

Al final de este capítulo se expondrá la cobertura obtenida con estos tests a través de capturas realizadas en el entorno de desarrollo Eclipse.

Descripción de los casos de prueba

Entidad Training Module:

- Casos positivos y negativos
 - **Creación de training module (Create):**este caso de prueba nos permite comprobar que la funcionalidad de crear un training module nuevo para un developer funcione correctamente, comprobando que los datos introducidos se guarden en la base de datos y que si se produce algún fallo en la introducción de cualquier dato, se muestre por pantalla.
Restricciones probadas en cada campo:
 - **code:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que cumple con el formato adecuado, para ello enviando un texto cualquiera devolviendo una alerta y otro texto que sí cumpla con el formato ([A-Z]{1,3}-[0-9]{3}).
 - Comprobación de que es único, es decir, que no pueden haber 2 training modules con código idéntico, para ello se ha enviado un código que coincide con el de otra training module para que el sistema devuelva la alerta correspondiente

- **creation moment:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que cumple con el formato adecuado, para ello enviando un texto/número cualquiera, devolviendo una alerta y otro que sí cumpla con el formato (yyyy/mm/dd hh:mm).
 - Comprobación de que es una fecha en el pasado, para ello enviando una fecha anterior a 2023/01/01 00:00, lo cual es aceptado por el sistema y también enviando una fecha posterior a esa para que el sistema devuelva la alerta correspondiente.
- **Details:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que el texto incluido no supera los 100 caracteres, para ello se ha enviado un texto que supera ese límite para que salte la alerta correspondiente y otro que cumple con el límite.
- **Difficult level:**
 - Comprobación de que no es nulo enviando el campo vacío.
- **Update moment (opcional):**
 - Comprobación de que cumple con el formato adecuado, para ello enviando un texto/número cualquiera, devolviendo una alerta y otro que sí cumpla con el formato (yyyy/mm/dd hh:mm).
 - Comprobación de que es una fecha en el pasado, para ello enviando una fecha anterior a 2023/01/01 00:00, lo cual es aceptado por el sistema y también enviando una fecha posterior a esa para que el sistema devuelva la alerta correspondiente.

- **Link (opcional):**
 - Comprobación de que el texto incluido no supera los 255 caracteres (incluido), para ello se ha enviado un texto que supera ese límite para que salte la alerta correspondiente y otro que cumple con el límite.
 - Comprobación de que el texto incluido supera los 7 caracteres (incluido), para ello se ha enviado un texto que no supera ese límite, saltando la alerta correspondiente y enviando otro que sí lo supera siendo aceptado por el sistema.
 - Comprobación de que el formato del link esté correcto, para ello se ha enviado un link con un formato erróneo para obtener la correspondiente alerta del sistema.
- **Project:**
 - Comprobación de que no es nulo enviando el campo vacío.
- **actualización de training module (update):** este caso de prueba nos permite comprobar que la funcionalidad de actualizar un training module para un developer funcione correctamente, comprobando que los datos introducidos se guarden en la base de datos y que si se produce algún fallo en la introducción de cualquier dato, se muestre por pantalla.
Restricciones probadas en cada campo:
 - **code:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que cumple con el formato adecuado, para ello enviando un texto cualquiera devolviendo una alerta y otro texto que sí cumpla con el formato ([A-Z]{1,3}-[0-9]{3}).
 - Comprobación de que es único, es decir, que no pueden haber 2 training modules con código idéntico, para ello se ha enviado un código que coincide con el de otra training module para que el sistema devuelva la alerta correspondiente

- **creation moment:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que cumple con el formato adecuado, para ello enviando un texto/número cualquiera, devolviendo una alerta y otro que sí cumpla con el formato (yyyy/mm/dd hh:mm).
 - Comprobación de que es una fecha en el pasado, para ello enviando una fecha anterior a 2023/01/01 00:00, lo cual es aceptado por el sistema y también enviando una fecha posterior a esa para que el sistema devuelva la alerta correspondiente.
- **Details:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que el texto incluido no supera los 100 caracteres, para ello se ha enviado un texto que supera ese límite para que salte la alerta correspondiente y otro que cumple con el límite.
- **Difficult level:**
 - Comprobación de que no es nulo enviando el campo vacío.
- **Update moment (opcional):**
 - Comprobación de que cumple con el formato adecuado, para ello enviando un texto/número cualquiera, devolviendo una alerta y otro que sí cumpla con el formato (yyyy/mm/dd hh:mm).
 - Comprobación de que es una fecha en el pasado, para ello enviando una fecha anterior a 2023/01/01 00:00, lo cual es aceptado por el sistema y también enviando una fecha posterior a esa para que el sistema devuelva la alerta correspondiente.

- **Link (opcional):**
 - Comprobación de que el texto incluido no supera los 255 caracteres (incluido), para ello se ha enviado un texto que supera ese límite para que salte la alerta correspondiente y otro que cumple con el límite.
 - Comprobación de que el texto incluido supera los 7 caracteres (incluido), para ello se ha enviado un texto que no supera ese límite, saltando la alerta correspondiente y enviando otro que sí lo supera siendo aceptado por el sistema.
 - Comprobación de que el formato del link esté correcto, para ello se ha enviado un link con un formato erróneo para obtener la correspondiente alerta del sistema.
 - **Project:**
 - Comprobación de que no es nulo enviando el campo vacío.
-
- **publicación de un training module:** este caso de prueba nos permite comprobar que la funcionalidad de publicar un training module para un developer funcione correctamente, comprobando que los datos introducidos se guarden en la base de datos y que si se produce algún fallo en la introducción de cualquier dato, se muestre por pantalla. Para probar correctamente esta funcionalidad previamente deben de haber Training sessions asociadas a este training module y que estas ya estén publicadas, para ello al intentar publicar un training module que no tiene ningún training session asociado, nos salta la alerta correspondiente.
Restricciones probadas en cada campo:

- **code:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que cumple con el formato adecuado, para ello enviando un texto cualquiera devolviendo una alerta y otro texto que sí cumpla con el formato ([A-Z]{1,3}-[0-9]{3}).
 - Comprobación de que es único, es decir, que no pueden haber 2 training modules con código idéntico, para ello se ha enviado un código que coincide con el de otra training module para que el sistema devuelva la alerta correspondiente

- **creation moment:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que cumple con el formato adecuado, para ello enviando un texto/número cualquiera, devolviendo una alerta y otro que sí cumpla con el formato (yyyy/mm/dd hh:mm).
 - Comprobación de que es una fecha en el pasado, para ello enviando una fecha anterior a 2023/01/01 00:00, lo cual es aceptado por el sistema y también enviando una fecha posterior a esa para que el sistema devuelva la alerta correspondiente.
- **Details:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que el texto incluido no supera los 100 caracteres, para ello se ha enviado un texto que supera ese límite para que salte la alerta correspondiente y otro que cumple con el límite.
- **Difficult level:**
 - Comprobación de que no es nulo enviando el campo vacío.
- **Update moment (opcional):**
 - Comprobación de que cumple con el formato adecuado, para ello enviando un texto/número cualquiera, devolviendo una alerta y otro que sí cumpla con el formato (yyyy/mm/dd hh:mm).
 - Comprobación de que es una fecha en el pasado, para ello enviando una fecha anterior a 2023/01/01 00:00, lo cual es aceptado por el sistema y también enviando una fecha posterior a esa para que el sistema devuelva la alerta correspondiente.

- **Link (opcional):**
 - Comprobación de que el texto incluido no supera los 255 caracteres (incluido), para ello se ha enviado un texto que supera ese límite para que salte la alerta correspondiente y otro que cumple con el límite.
 - Comprobación de que el texto incluido supera los 7 caracteres (incluido), para ello se ha enviado un texto que no supera ese límite, saltando la alerta correspondiente y enviando otro que sí lo supera siendo aceptado por el sistema.
 - Comprobación de que el formato del link esté correcto, para ello se ha enviado un link con un formato erróneo para obtener la correspondiente alerta del sistema.
- **Project:**
 - Comprobación de que no es nulo enviando el campo vacío.
- **eliminación de un training module (delete):** este caso de prueba nos permite comprobar que la funcionalidad de eliminación de un training module para un developer funcione correctamente, comprobando que si eliminamos ese training module, efectivamente se ha eliminado de la base de datos y por ende no es mostrado en la lista de training modules asociados al developer correspondiente.
- **listado de un training module (list):** este caso de prueba nos permite comprobar que la funcionalidad de listado de training modules para un developer funcione correctamente, comprobando que se listan todas los training modules asociados a ese developer y que se muestran de forma correcta.
- **mostrado de detalles de un training module (show):** este caso de prueba nos permite comprobar que la funcionalidad de mostrado de training modules para un developer funcione correctamente, comprobando que se muestran algunos de los training modules asociados a ese developer en concreto y que además se muestran de forma completa y correcta.

- **Casos hack:**

- **Creación:**

- Se ha comprobado que un usuario que no tiene un rol de developer no puede crear un training module.
- Se ha comprobado que un usuario con rol developer (developer2) si copia la url de otro developer (developer1) e intenta crear un training module con esa url, se le creará a ese mismo developer (developer2) que está intentando hacer ese ataque de hacking, lo cual es correcto debido a que este puede crear sus propios training modules al ser también un developer.

- **Actualización:**

- Se ha comprobado de que un usuario que no tiene rol no pueda editar un training module de algún developer copiando la url correspondiente.
- Se ha comprobado de que un usuario con rol developer no pueda editar un training module de otro developer copiando la url correspondiente.
- Se ha comprobado que un usuario con rol developer no puede editar un training module que ya está publicado con la url correspondiente, ya sea suyo el training module o no.

- **Publicación:**

- Se ha comprobado de que un usuario que no tiene rol no pueda publicar un training module de algún developer copiando la url correspondiente.
- Se ha comprobado de que un usuario con rol developer no pueda publicar un training module de otro developer copiando la url correspondiente.

- **Eliminación:**

- Se ha comprobado de que un usuario que no tiene rol no pueda eliminar un training module de algún developer copiando la url correspondiente.
- Se ha comprobado de que un usuario con rol developer no pueda eliminar un training module de otro developer copiando la url correspondiente
- Se ha comprobado que un usuario con rol developer no puede eliminar un training module que ya está publicado con la url correspondiente, ya sea suyo el training module o no.

- **Listado:**

- Se ha comprobado de que un usuario que no tiene rol no pueda listar los training modules de algún developer copiando la url correspondiente.
- Se ha comprobado de que un usuario con rol developer no pueda listar los training modules de otro developer copiando la url correspondiente.

- **Muestreo:**

- Se ha comprobado de que un usuario que no tiene rol no pueda mostrar los detalles de los training modules de algún developer copiando la url correspondiente.
- Se ha comprobado de que un usuario con rol developer no pueda mostrar los detalles de los training modules de otro developer copiando la url correspondiente.

(Tanto en creación y listado no se ha devuelto un error 500 debido a que la url para todos los developers es la misma, por lo que si intentas copiar la url de otro developer estarás accediendo a tus propios training modules. En todos los demás casos se devuelve un error 500 con su correspondiente mensaje)

Entidad Training session

- **Casos de prueba positivos y negativos:**

- **creación de training session (create):**este caso de prueba nos permite comprobar que la funcionalidad de crear una training session nueva para un developer funcione correctamente, comprobando que los datos introducidos se guarden en la base de datos y que si se produce algún fallo en la introducción de cualquier dato, se muestre por pantalla.

Restricciones probadas en cada campo:

- **code:**

- Comprobación de que no es nulo enviando el campo vacío.
- Comprobación de que cumple con el formato adecuado, para ello enviando un texto cualquiera devolviendo una alerta y otro texto que sí cumpla con el formato (TS-[A-Z]{1,3}-[0-9]{3}).
- Comprobación de que es único, es decir, que no pueden haber 2 training session con código idéntico, para ello se ha enviado un código que coincide con el de otra training session para que el sistema devuelva la alerta correspondiente

- **initial period:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que cumple con el formato adecuado, para ello enviando un texto/número cualquiera, devolviendo una alerta y otro que sí cumpla con el formato (yyyy/mm/dd hh:mm).
 - Comprobación de que la fecha introducida sea una semana posterior a la fecha de creación del training module al que está asociado la training session. Para ello se ha enviado una fecha que sea anterior a la fecha de creación del training module para que el sistema devuelva la alerta correspondiente. Luego se ha enviado una fecha que sí cumpla con la restricción para que sea aceptada por el sistema.
- **final period:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que cumple con el formato adecuado, para ello enviando un texto/número cualquiera, devolviendo una alerta y otro que sí cumpla con el formato (yyyy/mm/dd hh:mm).
 - Comprobación de que la fecha introducida en final period sea 1 semana posterior a la de initial period. Para ello, se ha enviado una fecha que no cumple con la restricción para que el sistema devuelva la alerta correspondiente y luego se ha enviado una fecha válida para que el sistema la acepte.
- **location:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que el texto incluido no supera los 75 caracteres (incluido), para ello se ha enviado un texto que supera ese límite para que salte la alerta correspondiente y otro que cumple con el límite.

- **instructor:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que el texto incluido no supera los 75 caracteres (incluido), para ello se ha enviado un texto que supera ese límite para que salte la alerta correspondiente y otro que cumple con el límite.
- **email:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que el texto incluido no supera los 75 caracteres (incluido), para ello se ha enviado un texto que supera ese límite para que salte la alerta correspondiente y otro que cumple con el límite.
 - Comprobación de que el texto incluido supera los 6 caracteres (incluido), para ello se ha enviado un texto que no supera ese límite para que salte la alerta correspondiente y otro que lo supere.
 - Comprobación de que el texto enviado cumpla con el formato de un correo, para ello se ha enviado un texto cualquiera para que el sistema devuelva la alerta correspondiente.
- **Link (opcional):**
 - Comprobación de que el texto incluido no supera los 255 caracteres (incluido), para ello se ha enviado un texto que supera ese límite para que salte la alerta correspondiente y otro que cumple con el límite.
 - Comprobación de que el texto incluido supera los 7 caracteres (incluido), para ello se ha enviado un texto que no supera ese límite, saltando la alerta correspondiente y enviando otro que sí lo supera siendo aceptado por el sistema.
 - Comprobación de que el formato del link esté correcto, para ello se ha enviado un link con un formato erróneo para obtener la correspondiente alerta del sistema.

- **actualización de training session (update):** este caso de prueba nos permite comprobar que la funcionalidad de actualizar una training session para un developer funcione correctamente, comprobando que los datos introducidos se guarden en la base de datos y que si se produce algún fallo en la introducción de cualquier dato, se muestre por pantalla.

Restricciones probadas en cada campo:

- **code:**

- Comprobación de que no es nulo enviando el campo vacío.
- Comprobación de que cumple con el formato adecuado, para ello enviando un texto cualquiera devolviendo una alerta y otro texto que sí cumpla con el formato (TS-[A-Z]{1,3}-[0-9]{3}).
- Comprobación de que es único, es decir, que no pueden haber 2 training session con código idéntico, para ello se ha enviado un código que coincide con el de otra training session para que el sistema devuelva la alerta correspondiente

- **initial period:**

- Comprobación de que no es nulo enviando el campo vacío.
- Comprobación de que cumple con el formato adecuado, para ello enviando un texto/número cualquiera, devolviendo una alerta y otro que sí cumpla con el formato (yyyy/mm/dd hh:mm).
- Comprobación de que la fecha introducida sea una semana posterior a la fecha de creación del training module al que está asociado la training session. Para ello se ha enviado una fecha que sea anterior a la fecha de creación del training module para que el sistema devuelva la alerta correspondiente. Luego se ha enviado una fecha que sí cumpla con la restricción para que sea aceptada por el sistema.

- **final period:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que cumple con el formato adecuado, para ello enviando un texto/número cualquiera, devolviendo una alerta y otro que sí cumpla con el formato (yyyy/mm/dd hh:mm).
 - Comprobación de que la fecha introducida en final period sea 1 semana posterior a la de initial period. Para ello, se ha enviado una fecha que no cumple con la restricción para que el sistema devuelva la alerta correspondiente y luego se ha enviado una fecha válida para que el sistema la acepte.
- **location:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que el texto incluido no supera los 75 caracteres (incluido), para ello se ha enviado un texto que supera ese límite para que salte la alerta correspondiente y otro que cumple con el límite.
- **instructor:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que el texto incluido no supera los 75 caracteres (incluido), para ello se ha enviado un texto que supera ese límite para que salte la alerta correspondiente y otro que cumple con el límite.
- **email:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que el texto incluido no supera los 75 caracteres (incluido), para ello se ha enviado un texto que supera ese límite para que salte la alerta correspondiente y otro que cumple con el límite.
 - Comprobación de que el texto incluido supera los 6 caracteres (incluido), para ello se ha enviado un texto que no supera ese límite para que salte la alerta correspondiente y otro que lo supere.
 - Comprobación de que el texto enviado cumpla con el formato de un correo, para ello se ha enviado un texto cualquiera para que el sistema devuelva la alerta correspondiente.

- **Link (opcional):**
 - Comprobación de que el texto incluido no supera los 255 caracteres (incluido), para ello se ha enviado un texto que supera ese límite para que salte la alerta correspondiente y otro que cumple con el límite.
 - Comprobación de que el texto incluido supera los 7 caracteres (incluido), para ello se ha enviado un texto que no supera ese límite, saltando la alerta correspondiente y enviando otro que sí lo supera siendo aceptado por el sistema.
 - Comprobación de que el formato del link esté correcto, para ello se ha enviado un link con un formato erróneo para obtener la correspondiente alerta del sistema.

- **publicación de training session (publish):** este caso de prueba nos permite comprobar que la funcionalidad de publicar una training session para un developer funcione correctamente, comprobando que los datos introducidos se guarden en la base de datos y que si se produce algún fallo en la introducción de cualquier dato, se muestre por pantalla.
 Restricciones probadas en cada campo:
 - **code:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que cumple con el formato adecuado, para ello enviando un texto cualquiera devolviendo una alerta y otro texto que sí cumpla con el formato (TS-[A-Z]{1,3}-[0-9]{3}).
 - Comprobación de que es único, es decir, que no pueden haber 2 training session con código idéntico, para ello se ha enviado un código que coincide con el de otra training session para que el sistema devuelva la alerta correspondiente

- **initial period:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que cumple con el formato adecuado, para ello enviando un texto/número cualquiera, devolviendo una alerta y otro que sí cumpla con el formato (yyyy/mm/dd hh:mm).
 - Comprobación de que la fecha introducida sea una semana posterior a la fecha de creación del training module al que está asociado la training session. Para ello se ha enviado una fecha que sea anterior a la fecha de creación del training module para que el sistema devuelva la alerta correspondiente. Luego se ha enviado una fecha que sí cumpla con la restricción para que sea aceptada por el sistema.
- **final period:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que cumple con el formato adecuado, para ello enviando un texto/número cualquiera, devolviendo una alerta y otro que sí cumpla con el formato (yyyy/mm/dd hh:mm).
 - Comprobación de que la fecha introducida en final period sea 1 semana posterior a la de initial period. Para ello, se ha enviado una fecha que no cumple con la restricción para que el sistema devuelva la alerta correspondiente y luego se ha enviado una fecha válida para que el sistema la acepte.
- **location:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que el texto incluido no supera los 75 caracteres (incluido), para ello se ha enviado un texto que supera ese límite para que salte la alerta correspondiente y otro que cumple con el límite.
- **instructor:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que el texto incluido no supera los 75 caracteres (incluido), para ello se ha enviado un texto que supera ese límite para que salte la alerta correspondiente y otro que cumple con el límite.

- **email:**
 - Comprobación de que no es nulo enviando el campo vacío.
 - Comprobación de que el texto incluido no supera los 75 caracteres (incluido), para ello se ha enviado un texto que supera ese límite para que salte la alerta correspondiente y otro que cumple con el límite.
 - Comprobación de que el texto incluido supera los 6 caracteres (incluido), para ello se ha enviado un texto que no supera ese límite para que salte la alerta correspondiente y otro que lo supere.
 - Comprobación de que el texto enviado cumpla con el formato de un correo, para ello se ha enviado un texto cualquiera para que el sistema devuelva la alerta correspondiente.
- **Link (opcional):**
 - Comprobación de que el texto incluido no supera los 255 caracteres (incluido), para ello se ha enviado un texto que supera ese límite para que salte la alerta correspondiente y otro que cumple con el límite.
 - Comprobación de que el texto incluido supera los 7 caracteres (incluido), para ello se ha enviado un texto que no supera ese límite, saltando la alerta correspondiente y enviando otro que sí lo supera siendo aceptado por el sistema.
 - Comprobación de que el formato del link esté correcto, para ello se ha enviado un link con un formato erróneo para obtener la correspondiente alerta del sistema.
- **eliminación de training session:**este caso de prueba nos permite comprobar que la funcionalidad de eliminación de una training session para un developer funcione correctamente, comprobando que si eliminamos esa training session, efectivamente se ha eliminado de la base de datos y por ende no es mostrado en la lista de training sessions asociadas al developer correspondiente.
- **listado de un training session(list):** este caso de prueba nos permite comprobar que la funcionalidad de listado de training sessions para un developer funcione correctamente, comprobando que se listan todas las training sessions asociadas a ese developer y que se muestran de forma correcta.

- **muestro de detalles de un training session (show):** este caso de prueba nos permite comprobar que la funcionalidad de mostrado de training sessions para un developer funcione correctamente, comprobando que se muestran algunos de las training sessions asociadas a ese developer en concreto y que además se muestran de forma completa y correcta.
- **Casos hack:**
 - **Creación:**
 - Se ha comprobado que un usuario que no tiene un rol de developer no puede crear una training session.
 - Se ha comprobado que un usuario con rol developer si copia la url de otro developer e intenta crear una training session con esa url, no se podrá crear.
 - **Actualización:**
 - Se ha comprobado de que un usuario que no tiene rol no pueda editar una training session de algún developer copiando la url correspondiente.
 - Se ha comprobado de que un usuario con rol developer no pueda editar una training session de otro developer copiando la url correspondiente.
 - Se ha comprobado que un usuario con rol developer no puede editar una training session que ya está publicada con la url correspondiente, ya sea suyo el training module o no.
 - **Publicación:**
 - Se ha comprobado de que un usuario que no tiene rol no pueda publicar una training session de algún developer copiando la url correspondiente.
 - Se ha comprobado de que un usuario con rol developer no pueda publicar una training session de otro developer copiando la url correspondiente.
 - **Eliminación:**
 - Se ha comprobado de que un usuario que no tiene rol no pueda eliminar una training session de algún developer copiando la url correspondiente.
 - Se ha comprobado de que un usuario con rol developer no pueda eliminar una training session de otro developer copiando la url correspondiente
 - Se ha comprobado que un usuario con rol developer no puede eliminar una training session que ya está publicada con la url correspondiente, ya sea suyo el training session o no.

- **Listado:**

- Se ha comprobado de que un usuario que no tiene rol no pueda listar las training session de algún developer copiando la url correspondiente.
- Se ha comprobado de que un usuario con rol developer no pueda listar las training session de otro developer copiando la url correspondiente

- **Muestreo:**

- Se ha comprobado de que un usuario que no tiene rol no pueda mostrar los detalles de las training session de algún developer copiando la url correspondiente.
- Se ha comprobado de que un usuario con rol developer no pueda mostrar los detalles de las training session de otro developer copiando la url correspondiente

































Bugs encontrados

Gracias a estos tests se ha podido encontrar un bug grave en la entidad training session:

- Al comprobar el publish, a la hora de realizar la comprobación de la duración de una semana entre los campos initial period y final period, al poner uno de los dos campos a null el sistema devolvía un error 500 significando un error garrafal para la entrega final del proyecto.

Todos los demás bugs se han ido resolviendo durante las clases de seguimiento de proyecto junto al tutor.

Cobertura de tests

| | | | | | |
|---|---|---------|-------|----|-------|
| ▼  acme.features.developer.trainingModule |  | 94,7 % | 1.465 | 82 | 1.547 |
| >  DeveloperTrainingModulePublishService.java |  | 95,2 % | 398 | 20 | 418 |
| >  DeveloperTrainingModuleDeleteService.java |  | 92,7 % | 229 | 18 | 247 |
| >  DeveloperTrainingModuleUpdateService.java |  | 94,6 % | 318 | 18 | 336 |
| >  DeveloperTrainingModulesCreateService.java |  | 94,0 % | 250 | 16 | 266 |
| >  DeveloperTrainingModuleShowService.java |  | 95,6 % | 130 | 6 | 136 |
| >  DeveloperTrainingModuleListService.java |  | 96,3 % | 105 | 4 | 109 |
| >  DeveloperTrainingModuleController.java |  | 100,0 % | 35 | 0 | 35 |
| ▼  acme.features.developer.trainingSession |  | 94,2 % | 1.287 | 79 | 1.366 |
| >  DeveloperTrainingSessionCreateService.java |  | 94,0 % | 283 | 18 | 301 |
| >  DeveloperTrainingSessionDeleteService.java |  | 91,1 % | 164 | 16 | 180 |
| >  DeveloperTrainingSessionPublishService.java |  | 94,7 % | 286 | 16 | 302 |
| >  DeveloperTrainingSessionUpdateService.java |  | 94,6 % | 283 | 16 | 299 |
| >  DeveloperTrainingSessionListService.java |  | 94,2 % | 130 | 8 | 138 |
| >  DeveloperTrainingSessionShowService.java |  | 95,5 % | 106 | 5 | 111 |
| >  DeveloperTrainingSessionController.java |  | 100,0 % | 35 | 0 | 35 |

Respecto a la cobertura obtenida en las clases training Module y training Session podemos decir que ha sido bastante buena, ya que para ambas entidades no se ha bajado más de un 90% de líneas de código sin testear, teniendo en cuenta que algunas de esas líneas de código eran imposibles de testear.

La cobertura de los métodos que más destacan de cada entidad son:

- Para Training module, **developerTrainingModuleController**, el cual tiene una cobertura del 100%.
- Para Training session, **developerTrainingSessionController**, el cual tiene una cobertura del 100%.

Líneas de código sin cubrir:

Como se ha expresado anteriormente, hay líneas de código imposibles de cubrir por la metodología seguida para la realización de los tests.

Estas líneas de código son:

```
assert object != null;
```

Esta línea de código se repite en los métodos bind, validate, perform y unbind de todas las funcionalidades proporcionadas por las entidades training session y training module. Esta línea nunca se llega a ejecutar por completo porque el objeto que llega a estos métodos nunca es nulo, ya que si lo fuera, la transacción se cancela antes de llegar a esta instrucción.

```
developer = trainingModule == null ? null : trainingModule.getDeveloper();
```

Esta línea de código se puede observar en las funcionalidades

DeveloperTrainingModulePublishService,
DeveloperTrainingModuleDeleteService,
DeveloperTrainingModuleUpdateService,
DeveloperTrainingModuleShowService.

Esta instrucción nunca se llega a ejecutar de forma total debido a que mediante las herramientas de grabación de tests no existe ninguna forma de solicitar operaciones con un trainingModule no existente.

```
status = trainingModule != null && !trainingModule.isPublished() && super.getRequest().getPrincipal().hasRole(developer);
```

Esta línea de código se puede observar en las funcionalidades

DeveloperTrainingModulePublishService,
DeveloperTrainingModuleDeleteService,
DeveloperTrainingModuleUpdateService.

Esta instrucción nunca se llega a ejecutar de forma total debido a que mediante las herramientas de grabación de tests no existe ninguna forma de solicitar operaciones con un trainingModule que esté publicado, por lo que no tenemos una forma de que esta instrucción se ejecute con un trainigModule publicado.

```
status = tm != null && super.getRequest().getPrincipal().hasRole(tm.getDeveloper()) && !tm.isPublished();
```

Esta línea de código se puede observar en las funcionalidades

DeveloperTrainingSessionCreateService,
DeveloperTrainingSessionShowService,
DeveloperTrainingSessionListService.

Este caso es idéntico al anterior pero en la entidad de trainingSession.

```
if (!super.getBuffer().getErrors().hasErrors("initialPeriod") && object.getFinalPeriod() != null && tm.getCreationMoment() != null) {
```

Esta línea de código se puede observar en las funcionalidades

DeveloperTrainingSessionPublishService,
DeveloperTrainingSessionUpdateService,
DeveloperTrainingSessionCreateService.

Esta instrucción nunca se ejecutará de forma completa debido a tm.getCreationMoment(), la cual nunca será null, debido a que para poder crear una training session primero debes crear una training module y justo ese campo es obligatorio.

```
if (!super.getBuffer().getErrors().hasErrors("link") && object.getLink() != null)
```

Esta línea de código se puede observar en las funcionalidades

DeveloperTrainingSessionPublishService,
DeveloperTrainingSessionUpdateService,
DeveloperTrainingSessionCreateService,
DeveloperTrainingModulePublishService,
DeveloperTrainingModuleCreateService,
DeveloperTrainingModuleUpdateService.

Esto ocurre debido a que la propiedad link puede ser nula y para que tenga errores ya tiene que dejar de ser nula, por esa razón esa condición no se comprobará en su completitud.

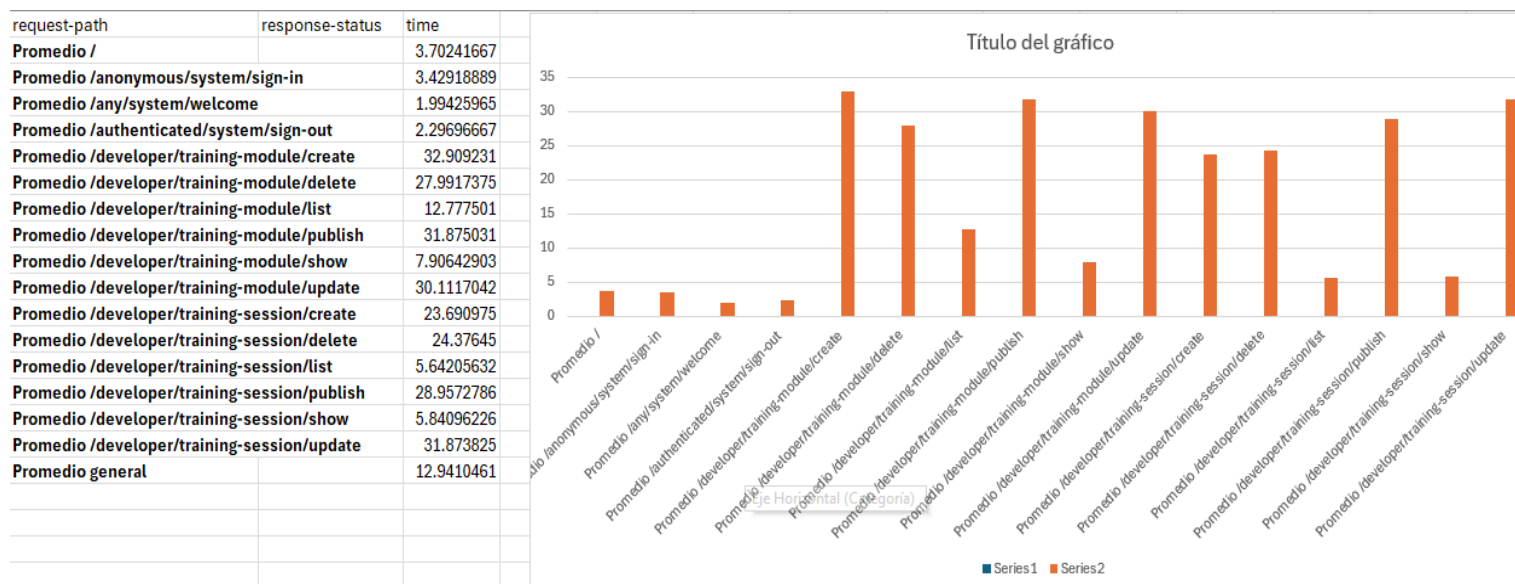
Además de esta información, también se añade las trazas obtenidas al lanzar el analyser.

(El archivo se denomina anexo 1).

Capítulo sobre Pruebas de Rendimiento

Este capítulo nos permite presentar algunas de las diferencias obtenidas al realizar varias mejoras en nuestro proyecto, como implementar índices. En sí el objetivo de este capítulo no es más que medir el impacto de los cambios realizados en las entidades correspondientes. Para esto, se han efectuado pruebas de rendimiento que miden el tiempo de respuesta de cada solicitud. Los datos recopilados incluyen promedios de tiempos de respuesta y otros estadísticos importantes. Además, se ha realizado un análisis estadístico exhaustivo, empleando el valor p para determinar si las mejoras en el rendimiento son estadísticamente significativas. Este análisis permitirá concluir si las optimizaciones han mejorado efectivamente el rendimiento del sistema.

- Promedio del tiempo de espera de las peticiones:



| request-path | response-status | time |
|--|-----------------|------------|
| Promedio / | | 3.70171481 |
| Promedio /anonymous/system/sign-in | | 3.31519441 |
| Promedio /any/system/welcome | | 2.06014735 |
| Promedio /authenticated/system/sign-out | | 2.44316667 |
| Promedio /developer/training-module/create | | 31.0829103 |
| Promedio /developer/training-module/delete | | 29.4301251 |
| Promedio /developer/training-module/list | | 12.7183695 |
| Promedio /developer/training-module/publish | | 30.5081069 |
| Promedio /developer/training-module/show | | 7.92927635 |
| Promedio /developer/training-module/update | | 28.9727875 |
| Promedio /developer/training-session/create | | 25.5491094 |
| Promedio /developer/training-session/delete | | 25.0256083 |
| Promedio /developer/training-session/list | | 5.65117471 |
| Promedio /developer/training-session/publish | | 29.6022607 |
| Promedio /developer/training-session/show | | 5.7868283 |
| Promedio /developer/training-session/update | | 27.7264042 |
| Promedio general | | 12.7706011 |

En esta captura ya podemos ver el tiempo promedio de las funcionalidades pero con los índices implementados. En este caso la funcionalidad que más ha tardado en ejecutarse ha sido el create con un total de 31 ms promedio, siendo 2 ms más rápida que antes de implementar los índices. Respecto a las otras funcionalidades podemos decir que se mantiene lo anterior, es decir las que seguían siendo las más rápidas ahora lo son un poco más y las que estaban rondando los 20-30 ms, algunas han subido y otras han bajado el tiempo promedio de ejecución. En general el promedio ha sido de 12.77 ms, 0.13 ms más rápido aproximadamente.

- **Comparación de datos estadísticos antes y después de añadir índices:**

| Antes | | | | | Después | | |
|---------------------------|------------|------------|--|--|---------------------------|------------|------------|
| Media | 12.9410461 | | | | Media | 12.7706011 | |
| Error típico | 0.51226381 | | | | Error típico | 0.4946481 | |
| Mediana | 6.8294 | | | | Mediana | 6.9466 | |
| Moda | 1.3444 | | | | Moda | #N/D | |
| Desviación estándar | 13.5532264 | | | | Desviación estándar | 13.0871585 | |
| Varianza de la muestra | 183.689947 | | | | Varianza de la muestra | 171.273717 | |
| Curtosis | 5.59516013 | | | | Curtosis | 11.19983 | |
| Coefficiente de asimetría | 2.02038284 | | | | Coefficiente de asimetría | 2.30928813 | |
| Rango | 102.8746 | | | | Rango | 131.6106 | |
| Mínimo | 1.1838 | | | | Mínimo | 1.2198 | |
| Máximo | 104.0584 | | | | Máximo | 132.8304 | |
| Suma | 9058.7323 | | | | Suma | 8939.4208 | |
| Cuenta | 700 | | | | Cuenta | 700 | |
| Nivel de confianza(95.0%) | 1.0057601 | | | | Nivel de confianza(95.0%) | 0.97117405 | |
| interval (ms) | 11.935286 | 13.9468062 | | | interval (ms) | 11.7994271 | 13.7417752 |
| interval (s) | 0.01193529 | 0.01394681 | | | interval (s) | 0.01179943 | 0.01374178 |

Con esta comparación de datos estadísticos antes y después de implementar los índices podemos ver varios datos relevantes:

- **Media:** la media podemos ver que ha mejorado 0.17 ms aproximadamente
- **Error típico:** al reducir el error podemos asumir que después de implementar los índices la precisión ha incrementado.
- **Mediana:** la mediana ha aumentado, por lo que la mitad de las consultas después de añadir índices son más lentas que antes.
- **Desviación estándar:** la desviación ha disminuido por lo que los tiempos se dispersan menos
- **Varianza:** podemos observar como la varianza ha disminuido bastante, por lo que hay más tiempos de ejecución iguales
- **Intervalo de confianza del 95%:** podemos observar una pequeña disminución del nivel de confianza, por lo que obtendríamos valores más bajos al implementar los índices.

Una mayor curtosis y coeficiente de asimetría luego de implementar los índices nos indica que tendríamos que ajustar más los índices o revisar algunas consultas para alcanzar una mayor optimización.

- **Análisis con z-test:**

Tras realizar una prueba z para poder analizar el valor p obtenemos los siguientes resultados:

| Prueba z para medias de dos muestras | | |
|--------------------------------------|------------|------------|
| | Antes | Después |
| Media | 12.9410461 | 12.7706011 |
| Varianza (conocida) | 183689947 | 171273717 |
| Observaciones | 700 | 700 |
| Diferencia hipotética de las medias | 0 | |
| z | 0.00023935 | |
| P(Z<=z) una cola | 0.49990451 | |
| Valor crítico de z (una cola) | 1.64485363 | |
| Valor crítico de z (dos colas) | 0.99980902 | |
| Valor crítico de z (dos colas) | 1.95996398 | |

El valor más importante es el de p, ya que nos indica la probabilidad de obtener resultados tan extremos como los que hemos observado en nuestros datos.

Si el valor p estuviera entre 0 y α (calculada como 1- z (dos colas)), lo que indicaría que es menor que nuestro nivel de significancia establecido, sería

necesario realizar una comprobación adicional. Esto implicaría que existe evidencia estadística para sugerir que los cambios implementados podrían haber tenido un efecto significativo en el rendimiento de las consultas.

En tal caso, sería fundamental llevar a cabo una comparación de las medias antes y después de implementar los cambios. Esto nos permitiría determinar si hay una diferencia significativa en el rendimiento de las consultas con y sin los índices agregados.

En nuestro caso, debido a que el valor de p es de 0.999809023, que está muy por encima de α (0.001) no necesitaríamos realizar esta comparación adicional. Esto es así debido a que p tener un valor muy alto nos sugiere que las mejoras al implementar los índices se podrían dar por la aleatoriedad de los datos y no por implementar esas mejoras como tal.

5. Conclusión

El análisis detallado de las pruebas funcionales y de rendimiento realizadas sobre nuestro proyecto revela resultados positivos en términos de eficiencia y fiabilidad del sistema. En el ámbito de las pruebas funcionales, se implementaron y documentaron exhaustivamente los casos de prueba, agrupados por características específicas del sistema.

En cuanto a las pruebas de rendimiento, se realizaron mediciones detalladas del tiempo de respuesta del sistema al procesar solicitudes. Los resultados incluyen gráficos y análisis estadísticos, junto con un intervalo de confianza del 95% para los tiempos de respuesta. El contraste de hipótesis realizado con un 95% de confianza permitió determinar que las acciones realizadas no han mejorado el sistema notablemente, pudiendo ser que la mejora observada sea producida por la aleatoriedad de los datos de prueba.

6. Bibliografía

Intencionadamente en blanco.

[OBJ]•[OBJ]