Informe de pruebas

1. Información general

Fecha: 27/05/2024

Grupo: C1.006

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

Autor: Alberto Carmona Sicre (albcarsic@alum.us.es)

2. Tabla de contenidos

Contenido

3. Resumen ejecutivo

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

4. Tabla de revisiones

Número de revisión	Descripción	Fecha
Revisión 1	Creación del documento y adición de todos los apartados	27/05/2024

5. Introducción

En el contexto del proyecto Acme-SF, se presenta el siguiente documento de pruebas, cuyo propósito es ofrecer el informe de los tests realizados a los servicios de las clases Sponsorship e Invoice. Se detallarán todos los resultados obtenidos en las pruebas funcionales, abarcando tanto casos de prueba positivos como negativos, así como los resultados de la cobertura de código para cada uno de los servicios de estas clases. Además, se incluirá una sección dedicada a los resultados del test de rendimiento, acompañados de gráficos y tablas que facilitarán la visualización clara de todos los resultados.

6. Contenidos

6.1 Testing funcional

6.1.1 Sponsorship

SponsorSponsorshipListMineService

Safe testing: para el testing de este servicio solo era necesario listar los patrocinios de un patrocinador. No había forma de realizar un test negativo.

Hacking: se realizaron dos pruebas:

- Listado de patrocinios de un patrocinador sin haber iniciado sesión (Role: Anonymous).

- Listado de patrocinios de un patrocinador con rol incorrecto (Role: Manager).

Bugs: sin bugs.

SponsorSponsorshipShowService

Safe testing: para el testing de este servicio se muestra un único patrocinio de un patrocinador como caso de prueba positivo. No era posible realizar un caso de prueba negativo.

Hacking: se realizaron cuatro pruebas:

- Mostrar el patrocinio de un patrocinador que no ha iniciado sesión (Role: Anonymous).

- Mostrar el patrocinio de un patrocinador con un rol incorrecto (Role: Manager).

- Intentar mostrar un patrocinio que no existe.

- Mostrar un patrocinio de un patrocinador que no es poseedor de dicho patrocinio (Role: Sponsor).

Bugs: sin bugs.

SponsorSponsorshipCreateService

Safe testing:

- **Casos de prueba positivos**: crear patrocinios correctos con distintos casos por cada uno de los atributos.

- **Casos de prueba negativos**: creación de patrocinios probando todas las restricciones de los campos:

- Formulario vacío.

- Campos con valores incorrectos, intentando probar todas las restricciones especificadas tanto por las anotaciones como por la lógica del método validate (por ejemplo, códigos duplicados, divisas no permitidas, presupuestos negativos, fechas incorrectas, etc.)

Hacking: la única forma de hackear este servicio era intentar entrar al formulario de creación de patrocinios sin haber iniciado sesión (Role: Anonymous) o con un rol incorrecto (Role distinto a Sponsor).

Bugs: sin bugs.

SponsorSponsorshipUpdateService

Safe testing:

- **Casos de prueba positivos**: actualizar patrocinios correctos con distintos casos por cada uno de los atributos.

- **Casos de prueba negativos**: actualización de patrocinios probando todas las restricciones de los campos:

- Formulario vacío

- Campos con valores incorrectos, intentando probar todas las restricciones especificadas tanto por las anotaciones como por la lógica del método validate (se siguieron los mismos casos que en el servicio de creación de patrocinios).

Hacking: la única forma de hackear este servicio con un rol incorrecto era copiar la URL de actualización de un patrocinio e intentar ejecutarla, lo que resultaba en una vista de pánico.

En cambio, con rol correcto, pero ejecutando acciones ilegales se probó lo siguiente:

- Actualizar un patrocinio publicado (acción ilegal).

- Actualizar un patrocinio de otro patrocinador (acción ilegal).

- Actualizar un patrocinio que no existe.

Bugs: sin bugs.

SponsorSponsorshipDeleteService

Safe testing:

- **Casos de prueba positivos**: eliminar patrocinios.

- **Casos de prueba negativos**: no existen casos de prueba negativos.

Hacking: la única forma de hackear este servicio con un rol incorrecto era copiar la URL de eliminación de un patrocinio e intentar ejecutarla, lo que resultaba en una vista de pánico. En cambio, con rol correcto, pero ejecutando acciones ilegales se probó lo siguiente:

- Eliminar un patrocinio publicado (acción ilegal).

- Eliminar un patrocinio de otro patrocinador (acción ilegal).

- Eliminar un patrocinio que no existe.

Bugs: sin bugs.

SponsorSponsorshipPublishService

Safe testing:

- **Casos de prueba positivos**: publicar patrocinios válidos con distintos casos por cada uno de los atributos.

- **Casos de prueba negativos**: publicar contratos probando las restricciones de los campos:

- Formulario vacío.

- Campos con valores incorrectos, intentando probar todas las restricciones especificadas tanto por las anotaciones como por la lógica del método validate. Se siguieron los mismos 3 casos que en el servicio de creación de patrocinios y actualización de patrocinios, pero se añadió una restricción adicional al valor del patrocinio: todas las facturas relacionadas con el patrocinador deben estar publicadas, y la suma del valor de todas esas facturas debe ser igual al valor del patrocinio.

Hacking: la única forma de hackear este servicio con un rol incorrecto era copiar la URL de publicación de un patrocinio e intentar ejecutarla, lo que resultaba en una vista de pánico. En cambio, con rol correcto, pero ejecutando acciones ilegales se probó lo siguiente:

- Publicar un patrocinio ya publicado (acción ilegal).

- Publicar un patrocinio de otro patrocinador, tanto publicado como sin publicar (acción ilegal).

- Publicar un patrocinio que no existe.

Bugs: se encontró un fallo al tratar de publicar un patrocinio donde el valor del patrocinio se dejaba nulo al intentar publicarlo. El caso es que como se comparan el valor del patrocinio y la suma de las facturas publicadas, al no haber ningún valor de patrocinio, el sistema devolvía un error 500. La solución fue incluir en el servicio un caso por defecto si el valor del patrocinio se pasa como nulo, el cual contaba con los siguientes valores: EUR 0.00. Aunque no se pueda publicar un patrocinio con valor de 0 euros, al realizarse solo una comparación con dicho valor en un caso específico, la aplicación permite comparar los valores de suma y del patrocinio sin que de un error 500 (simplemente aparece un error en el form donde se implica que el valor del patrocinio no puede ser nulo) y, además, si se trata de publicar un patrocinio con valor de 0.00 euros, no permite realizar la acción.

Cobertura Sponsorship

A continuación, se muestra el porcentaje de cobertura de sentencias de todos los servicios de la entidad Sponsorship:

SponsorSponsorshipListMineService: 96.4%

SponsorSponsorshipShowService: 96.1%

SponsorSponsorshipCreateService: 95.3%

SponsorSponsorshipUpdateService: 95.7%

SponsorSponsorshipDeleteService: 58.7%

SponsorSponsorshipPublishService: 96.5%

Siendo la cobertura total del paquete acmé.features.sponsor.sponsorship un 90.7%.

En términos generales, la cobertura de código de los servicios de la entidad Sponsorship es bastante satisfactoria. Las únicas líneas de código no cubiertas se deben a restricciones quizás demasiado estrictas en la autorización de los métodos, las cuales consideran casos que no

pueden probarse sin el uso de herramientas externas, como Postman. Debido a esto, algunas ramas del código no se alcanzan, al igual que ciertas líneas encargadas de la internacionalización al español.

También una línea de código que encontramos en todos los métodos:

```
assert object != null
```

Esta línea de código no se llega a cubrir del todo en ningún método, pero está presente en todas las plantillas de los servicios proporcionados en la metodología de la asignatura y se consultó con el cliente y no se consideró necesario cubrirla del todo.

Caso especial: SponsorSponsorshipDeleteService. Todo el método unbind de este servicio no se llega a cubrir, al no poder realizar pruebas negativas que resulten en errores de lógica de negocio, como se explicó anteriormente. Esto se debe a que el método unbind se encargaría de reconstruir la vista en caso de que se produzca un error, pero al no poder producirse, no se llega a ejecutar.

6.1.2 Invoice

SponsorInvoiceListMineService

Safe testing: para el testing de este servicio solo era necesario listar las facturas de un patrocinio de un patrocinador. No había forma de realizar un test negativo.

Hacking: se realizaron las siguientes pruebas:

- Listado de facturas de un patrocinio de un patrocinador sin haber iniciado sesión (Role: Anonymous).

- Listado de facturas de un patrocinio de un patrocinador con rol incorrecto (Role: Manager).

- Listado de facturas de un patrocinio que no existe.

- Listado de facturas de un patrocinio de un patrocinador que no es poseedor de tal patrocinio.

Bugs: sin bugs.

SponsorInvoiceShowService

Safe testing: para el testing de este servicio se muestra una factura de unpatrocinio de un patrocinador como caso de prueba positivo. No era posible realizar un caso de prueba negativo.

Hacking: se realizaron las siguientes pruebas:

- Mostrar una factura de un patrocinio de un patrocinador que no ha iniciado sesión (Role: Anonymous).

- Mostrar una factura de un patrocinio de un patrocinador con un rol incorrecto (Role: Manager).

- Intentar mostrar un patrocinio que no existe.

- Mostrar una factura de un patrocinio de un patrocinador que no es poseedor de dicho patrocinio (Role: Sponsor).

Bugs: sin bugs.

SponsorInvoiceCreateService

Safe testing:

- **Casos de prueba positivos**: crear facturas válidas con distintos casos por cada uno de los atributos.

- **Casos de prueba negativos**: creación de facturas probando todas las restricciones de los campos:

    - Formulario vacío.

    - Campos con valores incorrectos, intentando probar todas las restricciones especificadas tanto por las anotaciones como por la lógica del método validate (por ejemplo, códigos duplicados, divisas no permitidas, presupuestos negativos, fechas incorrectas, etc.)

Hacking: la única forma de hackear este servicio era intentar entrar al formulario de creación de patrocinios sin haber iniciado sesión (Role: Anonymous), con un rol incorrecto (Role distinto a Sponsor) o con rol correcto pero con creación de una factura en un patrocinio de un patrocinador que no es poseedor de dicho patrocinio.

Bugs: sin bugs.

SponsorInvoiceUpdateService

Safe testing:

- **Casos de prueba positivos**: actualizar facturas con distintos casos válidos por cada uno de los atributos.

- **Casos de prueba negativos**: actualización de patrocinios probando todas las restricciones de los campos:

- Formulario vacío

- Campos con valores incorrectos, intentando probar todas las restricciones especificadas tanto por las anotaciones como por la lógica del método validate (se siguieron los mismos casos que en el servicio de creación de facturas).

Hacking: la única forma de hackear este servicio con un rol incorrecto era copiar la URL de actualización de una factura e intentar ejecutarla, lo que resultaba en una vista de pánico.

En cambio, con rol correcto, pero ejecutando acciones ilegales se probó lo siguiente:

- Actualizar una factura publicado (acción ilegal).

- Actualizar una factura de otro patrocinador (acción ilegal).

- Actualizar una factura que no existe.

Bugs: sin bugs.

SponsorInvoiceDeleteService

Safe testing:

- **Casos de prueba positivos**: eliminar facturas.

- **Casos de prueba negativos**: no existen casos de prueba negativos.

Hacking: la única forma de hackear este servicio con un rol incorrecto era copiar la URL de eliminación de una factura e intentar ejecutarla, lo que resultaba en una vista de pánico. En cambio, con rol correcto, pero ejecutando acciones ilegales se probó lo siguiente:

- Eliminar una factura publicada (acción ilegal).

- Eliminar una factura de otro patrocinador (acción ilegal).

- Eliminar una factura que no existe.

Bugs: sin bugs.

SponsorInvoicePublishService

Safe testing:

- **Casos de prueba positivos**: publicar facturas válidas con distintos casos por cada uno de los atributos.

- **Casos de prueba negativos**: publicar facturas probando las restricciones de los campos:

- Formulario vacío.

- Campos con valores incorrectos, intentando probar todas las restricciones especificadas tanto por las anotaciones como por la lógica del método validate. Se siguieron los mismos casos que en el servicio de creación y actualización de patrocinios.

Hacking: la única forma de hackear este servicio con un rol incorrecto era copiar la URL de publicación de un patrocinio e intentar ejecutarla, lo que resultaba en una vista de pánico. En cambio, con rol correcto, pero ejecutando acciones ilegales se probó lo siguiente:

- Publicar una factura ya publicada (acción ilegal).

- Publicar una factura de otro patrocinador, tanto publicada como sin publicar (acción ilegal).

- Publicar una factura que no existe.

Bugs: sin bugs.

Cobertura Invoice

A continuación, se muestra el porcentaje de cobertura de sentencias de todos los servicios de la entidad Sponsorship:

SponsorInvoiceListMineService: 95.1%

SponsorInvoiceShowService: 96.4%

SponsorInvoiceCreateService: 94.2%

SponsorInvoiceUpdateService: 94.5%

SponsorInvoiceDeleteService: 90.4%

SponsorInvoicePublishService: 94.2%

Siendo la cobertura total del paquete acmé.features.sponsor.invoice un 94.3%.

En términos generales, la cobertura de código de los servicios de la entidad Invoice es bastante satisfactoria. Las únicas líneas de código no cubiertas se deben a restricciones quizás demasiado estrictas en la autorización de los métodos, las cuales consideran casos que no pueden probarse sin el uso de herramientas externas, como Postman. Debido a esto, algunas ramas del código no se alcanzan, al igual que ciertas líneas encargadas de la internacionalización al español.

También una línea de código que encontramos en todos los métodos:

assert object != null

Esta línea de código no se llega a cubrir del todo en ningún método, pero está presente en todas las plantillas de los servicios proporcionados en la metodología de la asignatura y se consultó con el cliente y no se consideró necesario cubrirla del todo.

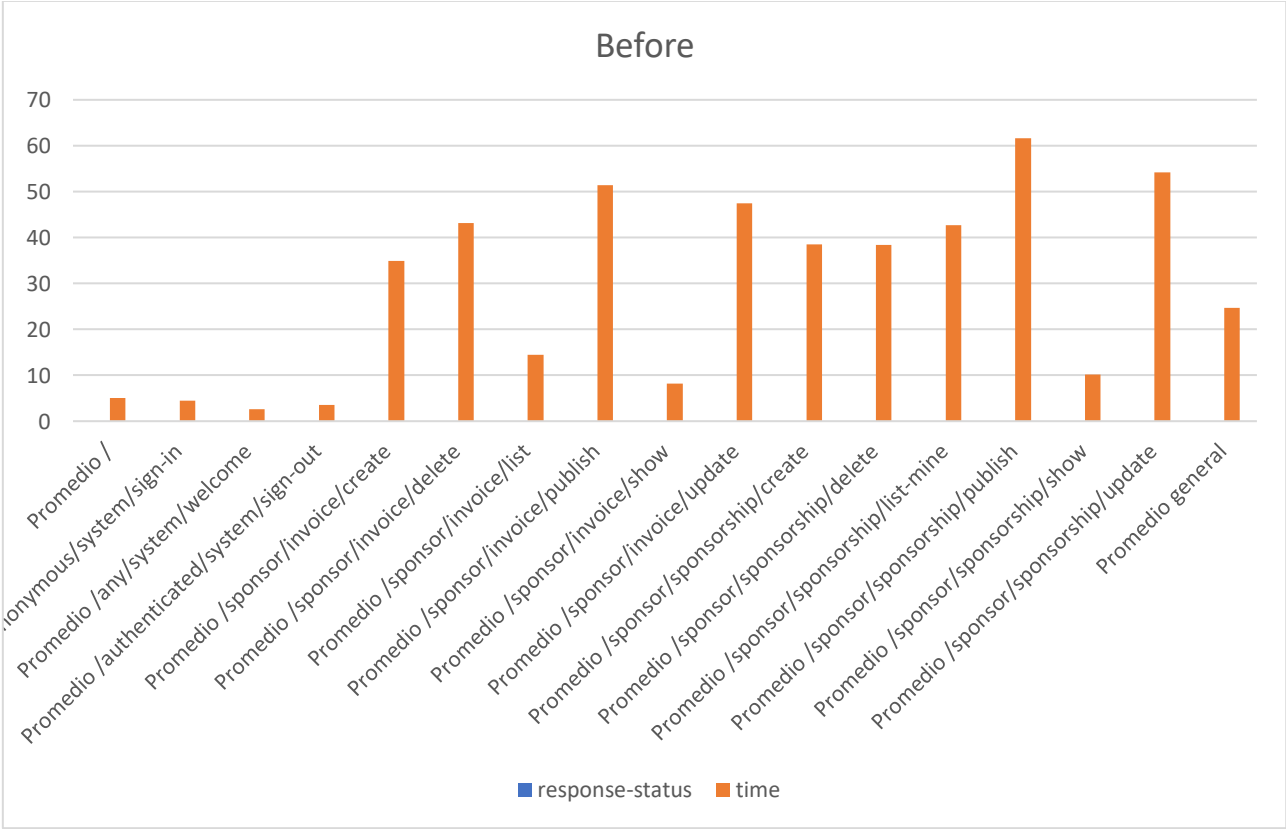
Caso especial: SponsorSponsorshipInvoiceService. Todo el método unbind de este servicio no se llega a cubrir, al no poder realizar pruebas negativas que resulten en errores de lógica de negocio, como se explicó anteriormente. Esto se debe a que el método unbind se encargaría de reconstruir la vista en caso de que se produzca un error, pero al no poder producirse, no se llega a ejecutar.

6.2 Testing de rendimiento

6.2.1 Análisis de rendimiento

Después de ejecutar las pruebas y recopilar los datos sobre las solicitudes realizadas, se realizó un análisis de los resultados obtenidos en términos de rendimiento de la aplicación, tomando como referencia el tiempo de respuesta de los servicios. Nos enfocamos en los servicios de las clases Sponsorship e Invoice. A continuación, se presentan los datos promedio de cada solicitud; además, para mayor claridad, también se incluye un gráfico de barras.

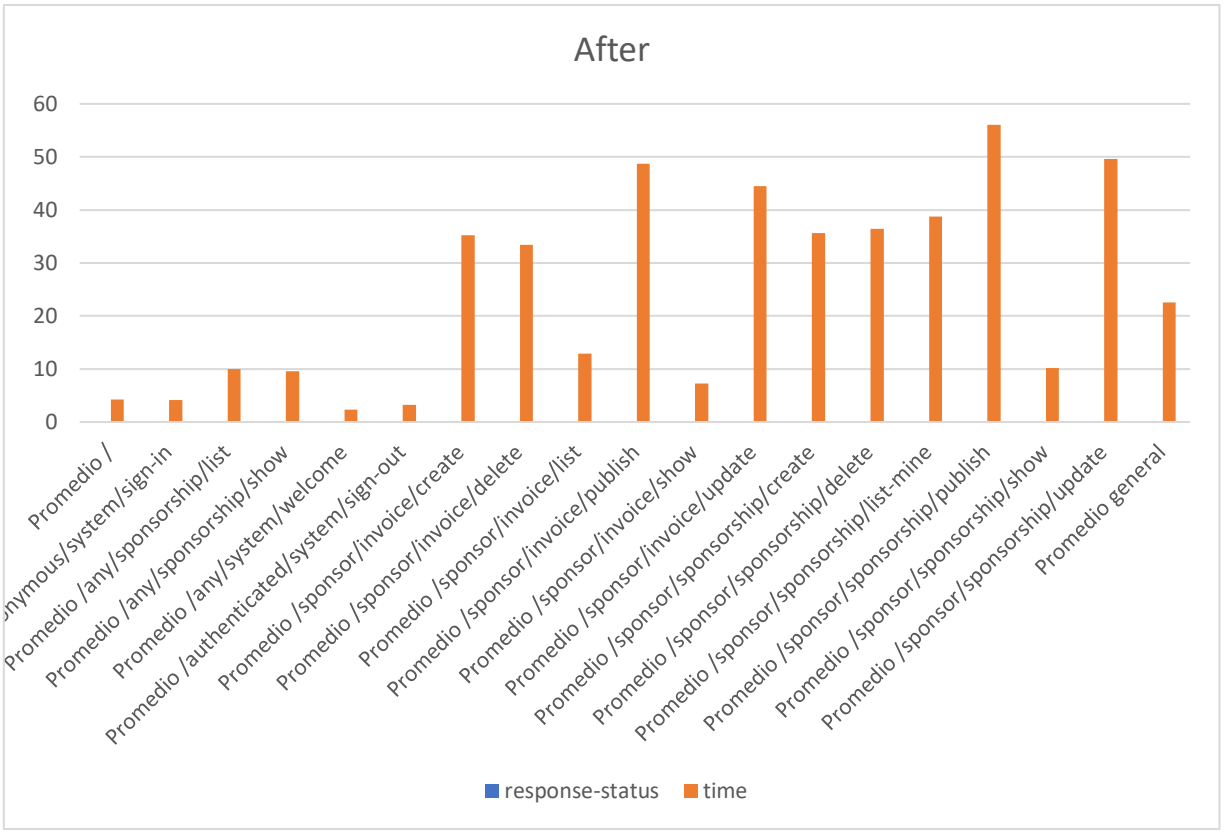
request-path	response-status	time
Promedio /		5.061386486
Promedio /anonymous/system/sign-in		4.432507609
Promedio /any/system/welcome		2.594646154
Promedio /authenticated/system/sign-out		3.590709302
Promedio /sponsor/invoice/create		34.93650476
Promedio /sponsor/invoice/delete		43.10689
Promedio /sponsor/invoice/list		14.50037032
Promedio /sponsor/invoice/publish		51.36633086
Promedio /sponsor/invoice/show		8.237005747
Promedio /sponsor/invoice/update		47.44906327
Promedio /sponsor/sponsorship/create		38.48380167
Promedio /sponsor/sponsorship/delete		38.42009091
Promedio /sponsor/sponsorship/list-mine		42.67265283
Promedio /sponsor/sponsorship/publish		61.64627903
Promedio /sponsor/sponsorship/show		10.1770602
Promedio /sponsor/sponsorship/update		54.20517586
Promedio general		24.66231525



Como vemos en la gráfica, existen grandes diferencias entre los tiempos de respuesta de las peticiones de inicio de sesión o bienvenido, comparado con las features de ambas entidades que estamos analizando. También es destacable que la *feature* de mayor tiempo y, por tanto, de mayor ineficiencia, es la de publicar de la clase Sponsorship, seguida de la de actualizar de la misma clase, que es unas 6 veces mayor que por ejemplo la de mostrar patrocinios, siendo esta una de las más pequeñas. El promedio general es de 24.66 ms.

La identificación de los servicios más lentos sirvió como punto de partida para optimizar la aplicación. Se implementaron una serie de cambios en el código, específicamente añadiendo índices a las tablas de la base de datos de la entidad Sponsorship, con el objetivo de mejorar el rendimiento de las solicitudes. A continuación, se presenta un gráfico de barras con los resultados obtenidos después de la optimización y los nuevos promedios de tiempo de respuesta.

request-path	response-status	time
Promedio /		4.260617105
Promedio /anonymous/system/sign-in		4.149446875
Promedio /any/sponsorship/list		10.01695
Promedio /any/sponsorship/show		9.61865
Promedio /any/system/welcome		2.3480525
Promedio /authenticated/system/sign-out		3.233802273
Promedio /sponsor/invoice/create		35.25260595
Promedio /sponsor/invoice/delete		33.41287
Promedio /sponsor/invoice/list		12.91703032
Promedio /sponsor/invoice/publish		48.70357901
Promedio /sponsor/invoice/show		7.214037931
Promedio /sponsor/invoice/update		44.45392041
Promedio /sponsor/sponsorship/create		35.61115
Promedio /sponsor/sponsorship/delete		36.43669091
Promedio /sponsor/sponsorship/list-mine		38.75725
Promedio /sponsor/sponsorship/publish		56.09447742
Promedio /sponsor/sponsorship/show		10.16606542
Promedio /sponsor/sponsorship/update		49.62847759
Promedio general		22.5815195



Aunque es cierto que la diferencia no es destacable, la mejora de rendimiento existe, ya que la mayoría de tiempos de las peticiones se reducen, sin rebasar los 60 ms de los tiempos anteriores. El nuevo promedio general es de 22.58 ms, habiéndose reducido en prácticamente un 10%.

6.2.2 Intervalo de confianza

Se analizó el intervalo de confianza de los tiempos de respuesta de las peticiones de la aplicación, con el objetivo de comprobar si la aplicación cumple con el requisito del rendimiento establecido: que el tiempo de respuesta de las peticiones no supere el segundo de media. Para ello se realizó un análisis de los datos obtenidos en los tests, y se calculó el intervalo de confianza de los tiempos de respuesta de las peticiones, con un nivel de confianza del 95%. La herramienta utilizada para el análisis ha sido excel, que proporciona un complemento llamado Herramientas para el análisis. A continuación, se muestra el resultado obtenido:

Before	After							
169.5832	130.4283		*After*					
5.7436	5.3792				Interval(ms)	21.12570693	24.03733208	
4.9662	4.2087		Media	22.5815195	Interval(s)	0.021125707	0.024037332	
4.1976	3.2686		Error típico	0.742031565				
8.3861	6.4272		Mediana	8.6827				
3.0964	2.6805		Moda	2.0979				
3.1196	3.2967		Desviación estándar	25.81160826				
3.1348	3.7866		Varianza de la muestra	666.2391208				
2.6437	2.2741		Curtosis	12.76802998				
2.7698	2.4051		Coeficiente de asimetría	2.462765602				
2.4793	2.2526		Rango	257.2808				
4.1323	3.8913		Mínimo	1.3689				
2.5107	2.2454		Máximo	258.6497				
2.493	2.185		Suma	27323.6386				
2.4264	2.2642		Cuenta	1210				
4.5003	4.006		Nivel de confianza(95.0%)	1.455812574				
4.1332	4.4664							
2.2185	2.0887							
2.4949	1.9986		*Before*					
2.3882	2.1242				Interval(ms)	23.19087556	26.13375494	
4.2323	10.5679		Media	24.66231525	Interval(s)	0.023190876	0.026133755	
2.9557	2.0837		Error típico	0.749982173				
2.2996	2.1355		Mediana	9.7229				
2.2596	2.1724		Moda	9.4836				
3.9923	3.6747		Desviación estándar	25.83903558				
2.3014	2.8301		Varianza de la muestra	667.6557595				
2.5652	4.7553		Curtosis	2.873361206				
3.9891	3.9054		Coeficiente de asimetría	1.430783355				
2.1987	2.2077		Rango	193.8108				
2.4293	2.0822		Mínimo	1.5356				
4.0988	3.5748		Máximo	195.3464				
2.1575	3.4354		Suma	29274.1682				
2.1317	1.9336		Cuenta	1187				
4.0883	3.9985		Nivel de confianza(95.0%)	1.47143969				

6.2.3 Hypothesis contrast

Prueba z para medias de dos muestras		
	Before	*After*
Media	24.66231525	22.5815195
Varianza (conocida)	667.6557595	666.2391208
Observaciones	1187	1210
Diferencia hipotética de las me	0	
z	1.972265876	
P(Z<=z) una cola	0.024289631	
Valor crítico de z (una cola)	1.644853627	
Valor crítico de z (dos colas)	0.048579263	
Valor crítico de z (dos colas)	1.959963985	

Como podemos observar, el valor crítico de z (dos colas) es 0.04875…, para saber si los cambios han sido significativos, hemos de comparar el valor z con α. En este caso, nuestro z se encuentra entre el intervalo [0.00, α), por lo que podemos afirmar que los cambios realizados han mejorado el rendimiento de la aplicación.

7. Conclusiones

A lo largo del documento, se ha realizado un análisis exhaustivo de los servicios de las clases Sponsorship e Invoice, que incluyó pruebas funcionales y de rendimiento para verificar el correcto funcionamiento de los servicios y el rendimiento de la aplicación. En los tests funcionales, se llevaron a cabo pruebas tanto positivas como negativas, confirmando que los servicios funcionan adecuadamente y no presentan errores. En los tests de rendimiento, se analizó el tiempo de respuesta de las solicitudes y se comprobó que la aplicación cumple con los requisitos de rendimiento establecidos.

Además, se efectuó una refactorización de los servicios de la clase Sponsorship, añadiendo índices a las tablas de la base de datos para mejorar el rendimiento de la aplicación. Posteriormente, se realizó un contraste de hipótesis que confirmó que los cambios implementados fueron significativos y mejoraron el rendimiento de la aplicación.

En resumen, he adquirido valiosas lecciones sobre el testing de aplicaciones web utilizando el Acme-Framework, que ha facilitado enormemente este proceso. También he aprendido a utilizar Excel para un análisis más profundo de los datos obtenidos en las pruebas, algo que no había hecho anteriormente.

8. Bibliografía

- 08 Annexes