



# Testing Report

C1.017

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

Jorge Gómez de Tovar  
jorgomde@alum.us.es

# Tabla de contenidos

Tabla de contenidos	1
Resumen ejecutivo	2
Tabla de revisiones	2
Contenido	2
Conclusión	2
Bibliografía	2

# Resumen ejecutivo

Este informe tiene como objetivo evaluar las pruebas realizadas en el proyecto para analizar su rendimiento. Sin embargo, se reconoce que debido a una planificación errónea, no se han llevado a cabo pruebas exhaustivas de todos los casos posibles. Se ha priorizado obtener un cálculo del rendimiento en lugar de realizar pruebas exhaustivas. A pesar de las limitaciones, este informe busca proporcionar una evaluación objetiva del rendimiento obtenido hasta la fecha.

En resumen, aunque se reconoce la falta de exhaustividad en las pruebas realizadas sobre todo las de hacking, este informe pretende ofrecer una visión útil y perspicaz del rendimiento del proyecto hasta la fecha además del proceso al crear las pruebas.

Añadir que el conjunto de pruebas y los datos de muestra necesarios para estas se encuentra en esta rama del repositorio online:

<https://github.com/vicgrabru/Acme-SF-D04/tree/task-023-s4>

# Tabla de revisiones

Revisión	Fecha	Descripción
1.0	27-03-2024	Primera versión del documento

# Introducción









El presente documento tiene como objetivo principal proporcionar una guía integral para el proceso de testing en el contexto del desarrollo de software. El testing, o pruebas de software, desempeña un papel fundamental en garantizar la calidad y confiabilidad de los productos y sistemas de software. A través de una serie de técnicas y metodologías, el testing permite identificar y corregir errores, validar el cumplimiento de requisitos y mejorar la experiencia del usuario final.









En esta introducción, explicaremos los conceptos básicos del testing realizado y el rendimiento obtenido por el mismo.

# Contenido

## Coverage

Aquí se muestra el coverage obtenido al reproducir todas las pruebas, se ve que todas las funcionalidades superan el 80% sin problemas. Además, las líneas marcadas como “amarillas” no muestran ningún problema ya que la mayoría son “assert object != null;” y algunos cálculos del status para la comprobación de autorización.

acme.features.sponsor.sponsorship		94,6 %
> SponsorSponsorshipController.java		100,0 %
> SponsorSponsorshipCreateService.java		94,9 %
> SponsorSponsorshipDeleteService.java		90,2 %
> SponsorSponsorshipListMineService.java		94,9 %
> SponsorSponsorshipPublishService.java		95,4 %
> SponsorSponsorshipShowService.java		96,3 %
> SponsorSponsorshipUpdateService.java		94,2 %

acme.features.sponsor.invoice		93,0 %
> SponsorInvoiceController.java		100,0 %
> SponsorInvoiceCreateService.java		94,0 %
> SponsorInvoiceDeleteService.java		87,9 %
> SponsorInvoiceListService.java		95,2 %
> SponsorInvoicePublishService.java		92,7 %
> SponsorInvoiceShowService.java		95,7 %
> SponsorInvoiceUpdateService.java		91,3 %

# Casos de pruebas positivos, negativos y hackeos para Sponsorship

## List.safe

Con sponsor1 accedemos a su lista de patrocinios y observamos que todo está bien. Puntualizar que para que se haga todo el coverage se observa una lista que tenga patrocinios publicados y no publicados.

## List.hack

Sin iniciar sesión, con developer1 entramos en `localhost:8082/Acme-SF-D04/sponsor/sponsorship/list-mine` para ver que no estamos autorizados.

## Show.safe

Con sponsor1 visualizamos el patrocinio T -000 (id 659).Observando que todo está bien.

## Show.hack

Sin iniciar sesión, con developer1 y con sponsor2,, entramos en `localhost:8082/Acme-SF-D04/sponsor/sponsorship/show?id=659` para ver que no estamos autorizados (**por ser patrocinio de sponsor1**).

## Delete.safe

Con sponsor1 borramos los patrocinios T-001 (id 660), T-003 (id 662) y T-005 (id 664), los cuales tienen 0, 1 y 3 facturas respectivamente. Obviamente todos estos patrocinios están en borrador.

## Publish.safe

Con sponsor1 intentamos publicar los patrocinios T-000 (id 659), T-001 (id 661) y T-004 (id 663) estos con 0, 1 y 3 facturas respectivamente. Estos no fallan ya que cumple las restricciones de que todas sus facturas estén publicadas y que la suma de la cantidad de sus facturas sea igual a la cantidad del patrocinio.

Tras esto intentamos publicar los patrocinios T-007 (id 666), T-008(id 667) y T-009(id 668) estos con 0, 1 y 3 facturas respectivamente.No funcionarán ya que no cumplen uno o varias de las restricciones mencionadas anteriormente

## Create.safe

Con sponsor1 intentamos crear un patrocinio sin datos para que nos de fallo en los campos no opcionales.

Luego rellenamos los campos con valores correctos y vamos atributo por atributo comprobando sus casos negativos:

- **code:** Sin respetar patrón (**A-41**), código repetido (**A-000**).
- **start duration:** Con valores antes (**2022/07/29 23:59**) y en el momento de creación (**2022/07/30 00:00**)
- **end duration:** Con valores antes (**2022/08/28 23:59**) y justo cuando pasa un mes de la start duration (**2022/07/30 00:00**).
- **amount:** Un valor negativo(**EUR -9999999999.98**), por encima del valor máximo para Money (**EUR 10,000,000,000**), divisa no aceptada por el sistema (**FKQ 0.03**).
- **type:** No tiene valores negativos.
- **email:** En email se ha probado el máximo tamaño de correo y un correo no válido.
- **link:** En el link se ha probado el máximo tamaño de enlace y un enlace no válido.
- **project:** No tiene valores negativos.

Una vez comprobados los casos negativos creamos varios patrocinios con datos válidos principalmente metiendo datos límites como correos y enlaces con la máxima longitud menos uno y varios valores de enlace singular como <http://example.org/a/b>, <http://example.org/a/b?>, <http://example.org/a/b?a&b> y valores de dinero cómo EUR 9999999999.98 ,EUR 9999999999.99.

## Update.safe

Con el sponsor1 se comprueba exactamente lo mismo que en create.safe editando el patrocinio T-010 (id 669), aunque en este caso no hay que comprobar code porque no puede modificarse una vez asignado.

## Casos de pruebas positivos, negativos y hackeos para Invoices

### List.safe

Con sponsor1 entramos en el listado de invoices del patrocinio T-011 (id 670) y observamos que todo está bien.

### List.hack

Sin iniciar sesión, con developer1 y con sponsor2, entramos en *localhost:8082/Acme-SF-D04/sponsor/invoice/list?masterId=670* para ver que no estamos autorizados

### show.safe

Con sponsor1 visualizamos la factura IN-0007-0001 (id 692) Se ha podido encontrar a través de la lista de invoices de T-011 (id 670).

## show.hack

Sin iniciar sesión, con developer1 y con sponsor2 entramos en `localhost:8082/Acme-SF-D04/sponsor/invoice/show?id=692` para ver que no estamos autorizados.

## delete.safe

Con sponsor1 eliminamos la factura IN-0007-0001 (id 692) perteneciente al patrocinio T-011 (id 670). Obviamente dicha factura está en borrador.

## publish.safe

Con sponsor1 publicamos la factura IN-0007-0001 (id 692) perteneciente al patrocinio T-011 (id 670). Obviamente dicha factura está en borrador.

## create.safe

Con sponsor1 intentamos crear una factura sin datos para que nos de fallo en los campos no opcionales.

Luego rellenamos los campos con valores correctos y vamos atributo por atributo comprobando sus casos negativos:

- **code:** Sin respetar patrón (**IN-00-00**), código repetido (**IN-0001-0001**).
- **due date:** Con valores antes (**2022/08/28 23:59**) y justo cuando pasa un mes de la start duration (**2022/07/29 00:00**).
- **amount:** Un valor negativo(**EUR -9999999999.98**), por encima del valor máximo para Money (**EUR 10,000,000,000**), divisa no aceptada por el sistema (**FKQ 0.03**).
- tax:** Un valor negativo (**-1**), un valor mayor que el máximo (**101**).
- **link:** En link se ha probado el máximo tamaño de correo y un correo no válido.

Una vez comprobados los casos negativos creamos varias facturas con datos válidos principalmente metiendo datos límites como enlaces con la máxima longitud menos uno y varios valores de enlace singular como <http://example.org/a/b>, <http://example.org/a/b?>, <http://example.org/a/b?a&b> y valores de dinero cómo EUR 9999999999.98 ,EUR 9999999999.99. y valores de impuesto cerca del máximo y el máximo 99.9, 100.

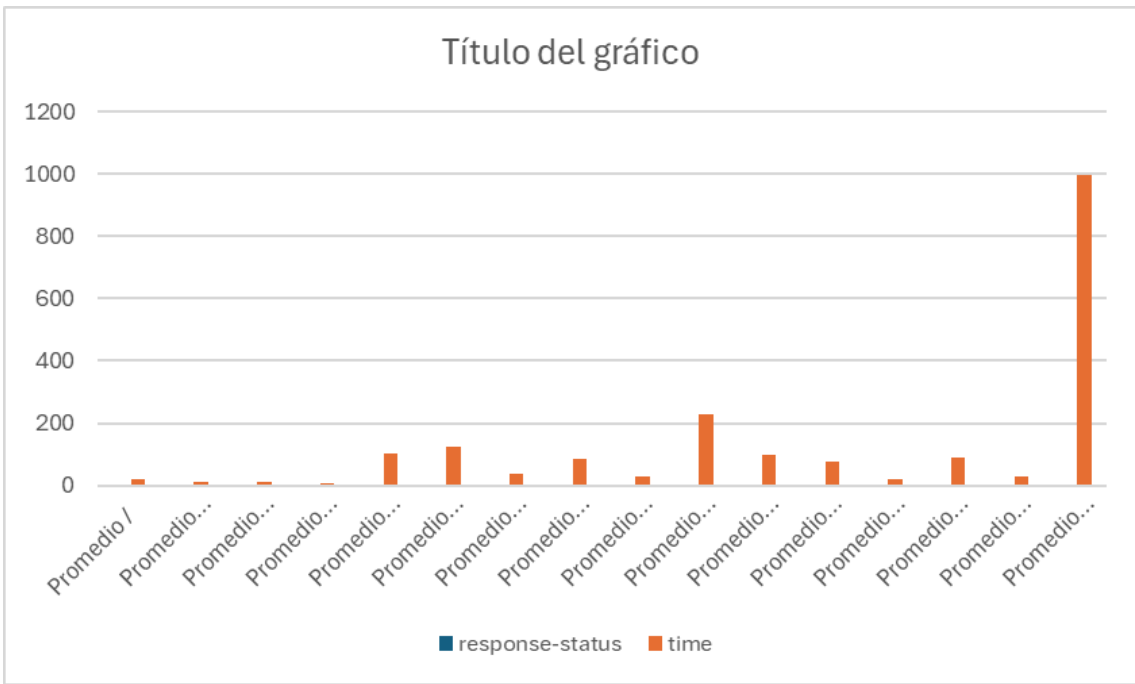
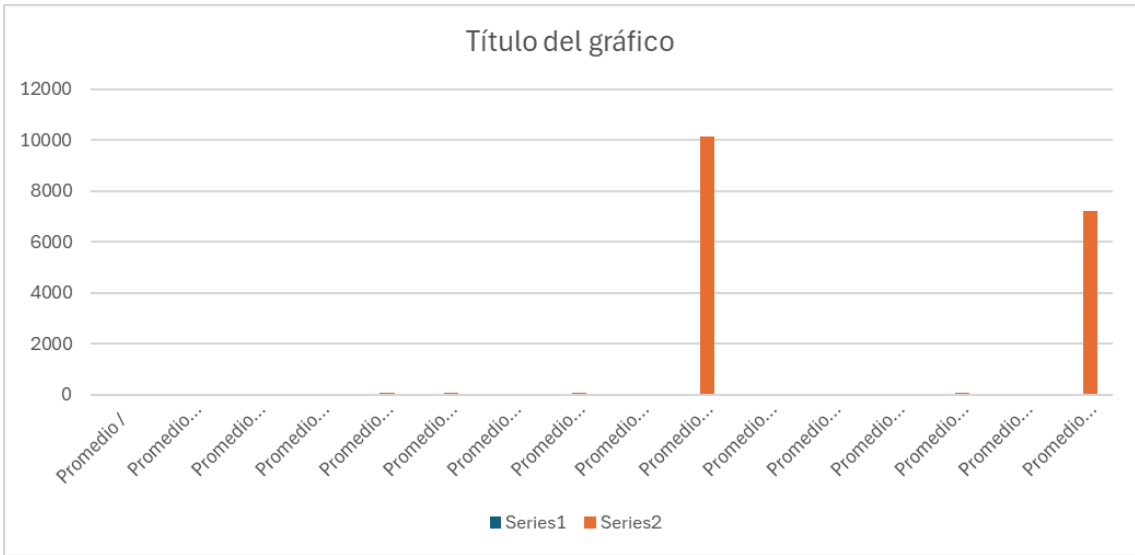
## update.safe

Con el sponsor1 se comprueba exactamente lo mismo que en create.safe editando la factura IN-0007-0001 (id 692) perteneciente al patrocinio T-011 (id 670)., aunque en este caso no hay que comprobar code porque no puede modificarse una vez asignado.

# Resultados de pruebas de rendimiento

Para probar el rendimiento del sistema respecto a las pruebas se ha lanzado el replayer en 2 ordenadores personales distintos, aplicando en el segundo caso una pequeña optimización añadiendo índices.

Los gráficos de tiempo obtenidos en el primer y segundo ordenador son los siguientes:



Se puede ver que el segundo ordenador mejora al primero aunque ambos son bastante ineficientes ya que hay valores muy altos para algunas acciones como el update o el create. Observemos los intervalos de confianza del 95% para el tiempo de respuesta de los



ordenadores:

Columna1			Columna1		
Media	737,877736		Media	88,2362777	
Error típico	503,79668		Error típico	46,3183125	
Mediana	12,97155		Mediana	18,79045	
Moda	#N/D		Moda	#N/D	
Desviación e	8460,18491		Desviación e	777,816735	
Varianza de	71574728,7		Varianza de	604998,873	
Curtosis	138,475634		Curtosis	271,295388	
Coeficiente d	11,8106113		Coeficiente d	16,3453579	
Rango	101005,019		Rango	12974,5842	
Mínimo	3,5272		Mínimo	4,3806	
Máximo	101008,547		Máximo	12978,9648	
Suma	208081,522		Suma	24882,6303	
Cuenta	282		Cuenta	282	
Nivel de cont	991,694591		Nivel de cont	91,174916	
intervalo(ms)	0	1729,57233	intervalo(ms)	0	179,411194
intervalo(s)	0	1,72957233	intervalo(s)	0	0,17941119

Después de analizar los cálculos, con la columna de la izquierda refiriéndose a los datos antes de optimizar y la de la derecha después de optimizar. Se puede ver que sigue pareciendo haber una mejora, dando la segunda ejecución una mejoría en el intervalo, así que por último observemos el contraste de hipótesis con un 95%:

	Before	After
Media	737,877736	88,2362777
Varianza (conocida)	71574728,7	604998,873
Observaciones	282	282
Diferencia hipotética de las medias	0	
z	1,28407582	
P(Z<=z) una cola	0,09955771	
Valor crítico de z (una cola)	1,64485363	
Valor crítico de z (dos colas)	0,19911543	
Valor crítico de z (dos colas)	1,95996398	

Por último, al comprobar que el p-value (0.199) es superior a alfa (0.05), podemos deducir que no ha habido realmente una mejora en la ejecución de las pruebas. Por ello el uso de los índices no ha supuesto una mejora muy evidente.

# Conclusión

En resumen, aunque no se han llevado a cabo pruebas exhaustivas en todos los casos de hacking, se ha logrado obtener una medida del rendimiento de la aplicación. Los resultados indican que el rendimiento de la aplicación en ordenadores con menor capacidad no es óptimo. Esto remarca la importancia de optimizar la aplicación para mejorar su desempeño en una variedad de entornos, especialmente en hardware menos potente.

# Bibliografía

No aplica.