

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

Testing report



ACME SOFTWARE FACTORY

OUR FIRST PROJECT IN D&T

Grado en Ingeniería Informática – Ingeniería del Software

Diseño y Pruebas 2

Curso 2023 – 2024

Grupo de prácticas: C2-009

Autores por orden alfabético

León Madroñal, Juan Carlos

	Diseño y Pruebas II Acme-Software-Factory
	Testing report

Índice de contenido

1. Resumen ejecutivo	3
2. Tabla de revisiones.....	4
3. Introducción	5
4. Functional testing.....	6
5. Performance testing.....	11
6. Conclusiones	16
7. Bibliografía	17

	Diseño y Pruebas II Acme-Software-Factory
	Testing report

1. Resumen ejecutivo

Este documento proporciona una visión completa de las pruebas funcionales y de rendimiento que se le han realizado a nuestro proyecto. Analizar la siguiente información es crucial para saber si vamos por buen camino y si nuestro proyecto tiene una correcta optimización.

	Diseño y Pruebas II Acme-Software-Factory
	Testing report

2. Tabla de revisiones

Fecha	Versión	Descripción
27/05/2024	1.0	Primera versión del documento
07/05/2024	1.1	Actualización y nueva suite para julio

	Diseño y Pruebas II Acme-Software-Factory
	Testing report

3. Introducción

Este informe de pruebas se organiza en dos capítulos principales: *functional testing* y *performance testing*.

En *functional testing*, se lista los casos de prueba implementados, organizados por requisitos. Cada caso de prueba incluye una descripción breve y una evaluación de su efectividad en la detección de errores.

En *performance testing*, se muestran gráficos y un intervalo de confianza del 95% para los tiempos de respuesta del sistema en dos ordenadores distintos. Esto se realiza para hacer una contrastación de hipótesis para determinar cuál de los dos es más potente.

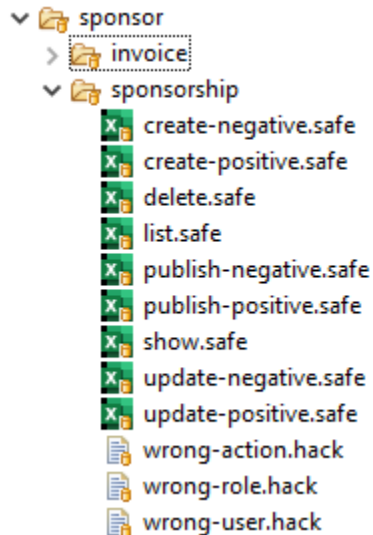
Este informe busca proporcionar una evaluación clara y precisa del sistema, asegurando su calidad y rendimiento.

	Diseño y Pruebas II Acme-Software-Factory
	Testing report

4. Functional testing

Requirement #6:


Para este requisito, se han implementado las siguientes pruebas:



















- **List:** se comprueba que un patrocinador puede listar todos sus patrocinios.
- **Show:** se comprueba que un patrocinador puede ver correctamente un patrocinio de su propiedad en concreto.
- **Delete:** se comprueba que un patrocinador puede borrar un patrocinio suyo siempre que este no se encuentre publicado.
- **Create:** en *create-negative* se prueban todos los datos límite para los que no se debería crear un patrocinio, mientras que en *create-positive* se fuerzan estos límites para datos que si debiesen permitir crear un patrocinio.
- **Update:** al igual que en el caso anterior, forzamos los límites para casos positivos (*update-positive*) y para casos negativos (*update-negative*) de actualización de un patrocinio.
- **Publish:** ya que el *publish* actúa también como un *update*, hemos seguido la misma metodología que para el testing del *update*, añadiendo las restricciones correspondientes indicadas en los requisitos correspondientes a la publicación de patrocinios

Hacking:

- En *wrong role* se comprueba que siendo un usuario anónimo/admin no podemos hacer ninguna de las acciones anteriores.

	Diseño y Pruebas II Acme-Software-Factory
	Testing report

- En *wrong user* se hace login como sponsor2 y se comprueba que no deja hacer acciones sobre un patrocinio perteneciente al sponsor1.
- En *wrong action* se hace login con sponsor1 y se comprueba que no podemos borrar, actualizar o publicar un patrocinio ya publicado.

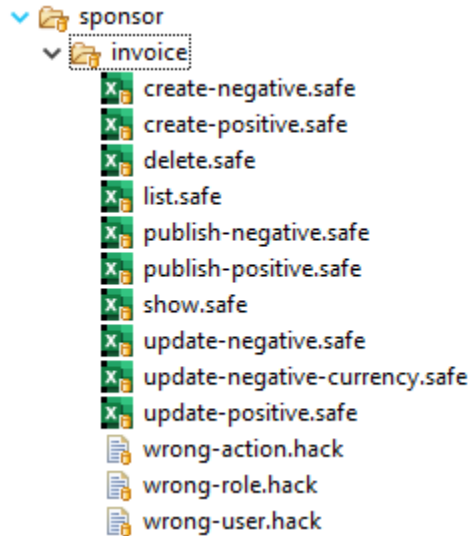
▼  acme.features.sponsor.sponsorship		90,7 %	1.342	137	1.479
>  SponsorSponsorshipController.java		100,0 %	35	0	35
>  SponsorSponsorshipCreateService.java		95,9 %	284	12	296
>  SponsorSponsorshipDeleteService.java		57,3 %	118	88	206
>  SponsorSponsorshipListService.java		94,2 %	65	4	69
>  SponsorSponsorshipPublishService.java		95,9 %	400	17	417
>  SponsorSponsorshipShowService.java		97,2 %	138	4	142
>  SponsorSponsorshipUpdateService.java		96,2 %	302	12	314

Como apreciamos en la imagen, nuestra suite de tests para *sponsor.sponsorship* está notablemente cerca del 100% salvo en el servicio que corresponde al *delete*, debiéndose al método unbind de la misma que, aunque está implementado no llega a ejecutarse.

	Diseño y Pruebas II Acme-Software-Factory
	Testing report

Requirement #7:

Para este requisito, se han implementado las siguientes pruebas:











- **List:** se comprueba que un patrocinador puede listar todas las facturas para un patrocinio de su propiedad.
- **Show:** se comprueba que un patrocinador puede ver correctamente una factura correspondiente a un patrocinio suyo.
- **Delete:** se comprueba que un patrocinador puede borrar una factura de su propiedad siempre que esta no se encuentre publicada.
- **Create:** en *create-negative* se prueban todos los datos límites para los que no se debería crear una factura, mientras que en *create-positive* se fuerzan estos límites para datos que si deberían permitir crear una factura.
- **Update:** al igual que en el caso anterior, forzamos los límites para casos positivos (*update-positive*) y para casos negativos (*update-negative*) de actualización de un patrocinio. Además, *update-negative-currency* comprueba los casos en los que no se debe actualizar una factura si ya hay una divisa definida para un patrocinio y sus facturas (esta revisión ya va incluida en las pruebas *create-negative* y *publish-negative*).
- **Publish:** como se mencionó anteriormente, ya que el *publish* actúa también como un *update* hemos seguido la misma metodología que para el testing del *update*, añadiendo las restricciones correspondientes indicadas en los requisitos correspondientes a la publicación de facturas

	Diseño y Pruebas II Acme-Software-Factory
	Testing report

Hacking:

- En *wrong role* se comprueba que siendo un usuario anónimo/admin no podemos hacer ninguna de las acciones anteriores.
- En *wrong user* se hace login como sponsor2 y se comprueba que no deja hacer acciones sobre una factura asociada a un patrocinio del sponsor1.
- En *wrong action* se hace login con sponsor1 y se comprueba que no podemos borrar, actualizar o publicar una factura ya publicada.

▼	acme.features.sponsor.invoices		90,8 %	1.308	133	1.441
>	SponsorInvoiceController.java		100,0 %	35	0	35
>	SponsorInvoiceCreateService.java		94,8 %	308	17	325
>	SponsorInvoiceDeleteService.java		56,3 %	103	80	183
>	SponsorInvoiceListService.java		96,4 %	107	4	111
>	SponsorInvoicePublishService.java		95,2 %	316	16	332
>	SponsorInvoiceShowService.java		96,9 %	125	4	129
>	SponsorInvoiceUpdateService.java		96,3 %	314	12	326

Como apreciamos en la imagen, nuestra suite de tests para *sponsor.invoice* es similar a la de los patrocinios y ocurre el mismo problema en el servicio *delete*, ya que el método *unbind* está implementado, pero no llega a ejecutarse.

	Diseño y Pruebas II Acme-Software-Factory
	Testing report

Eficacia en la detección de errores

Aun siendo la segunda vez que se desarrolla la suite de tests para el S04, se ha encontrado el siguiente error mientras se elaboraba la misma:

1. Error mostrado sin necesidad en *sponsorship.publish*:
 - **Problema:** Al publicar un patrocinio se mostraba el error de que la divisa no coincidía con la de sus facturas, aunque todas estas estuvieran todavía en draftMode, algo que no es correcto teniendo en cuenta el análisis de los requisitos hecho.
 - **Solución:** Se ha tenido que añadir el condicional correspondiente antes de comprobar si se tiene que mostrar o no el error, es decir, que al menos una de sus facturas debe estar publicada para que ese error pueda o no mostrarse.



5. Performance testing

En este apartado se van a mostrar las pruebas de rendimiento realizadas. Se ha ejecutado la suite de prueba en:

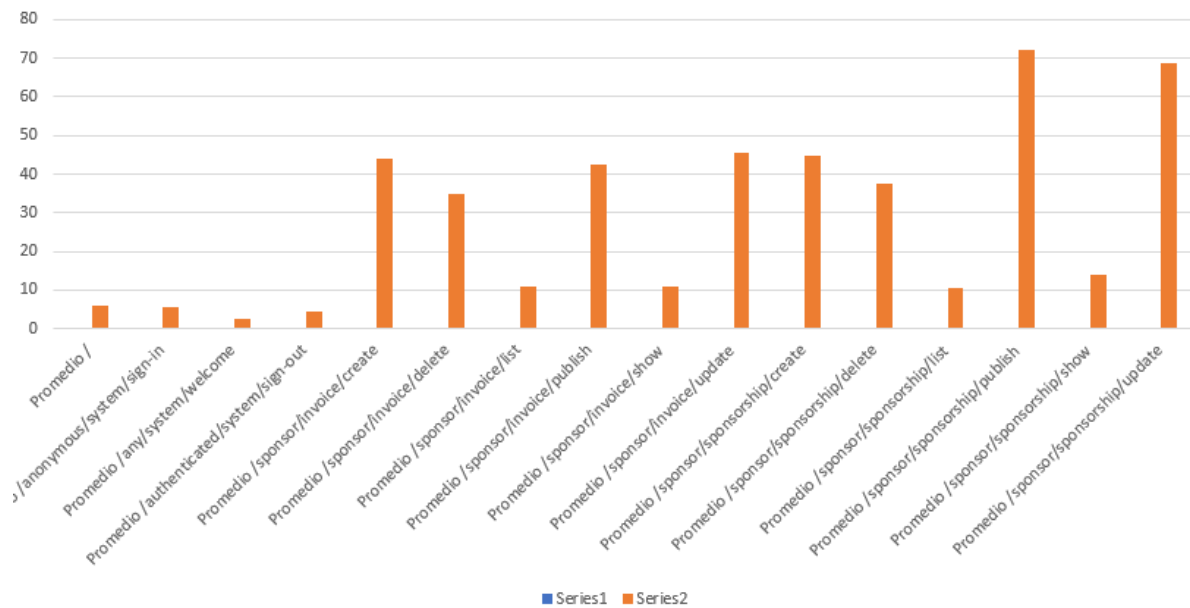
- Caso 1. Mi portátil personal: Lenovo IdeaPad Gaming 3 15IAH7 (Intel Core i7-12650H / 16GB RAM / SSD)
- Caso 2. Portátil de mi compañero: HP Pavilion Gaming 15 (Intel Core i7-9750H / 8GB RAM / SSD)

Métricas para el Caso 1:

request-path	response-status	time
Promedio /		5,95507
Promedio /anonymous/system/sign-in		5,35636596
Promedio /any/system/welcome		2,658610855
Promedio /authenticated/system/sign-out		4,53424975
Promedio /sponsor/invoice/create		43,90755552
Promedio /sponsor/invoice/delete		34,99923333
Promedio /sponsor/invoice/list		10,95426222
Promedio /sponsor/invoice/publish		42,45820476
Promedio /sponsor/invoice/show		10,86491978
Promedio /sponsor/invoice/update		45,66275308
Promedio /sponsor/sponsorship/create		44,90766458
Promedio /sponsor/sponsorship/delete		37,4363335
Promedio /sponsor/sponsorship/list		10,59003796
Promedio /sponsor/sponsorship/publish		72,14877566
Promedio /sponsor/sponsorship/show		13,76766083
Promedio /sponsor/sponsorship/update		68,93346236
Promedio general		24,66049027



Caso 1



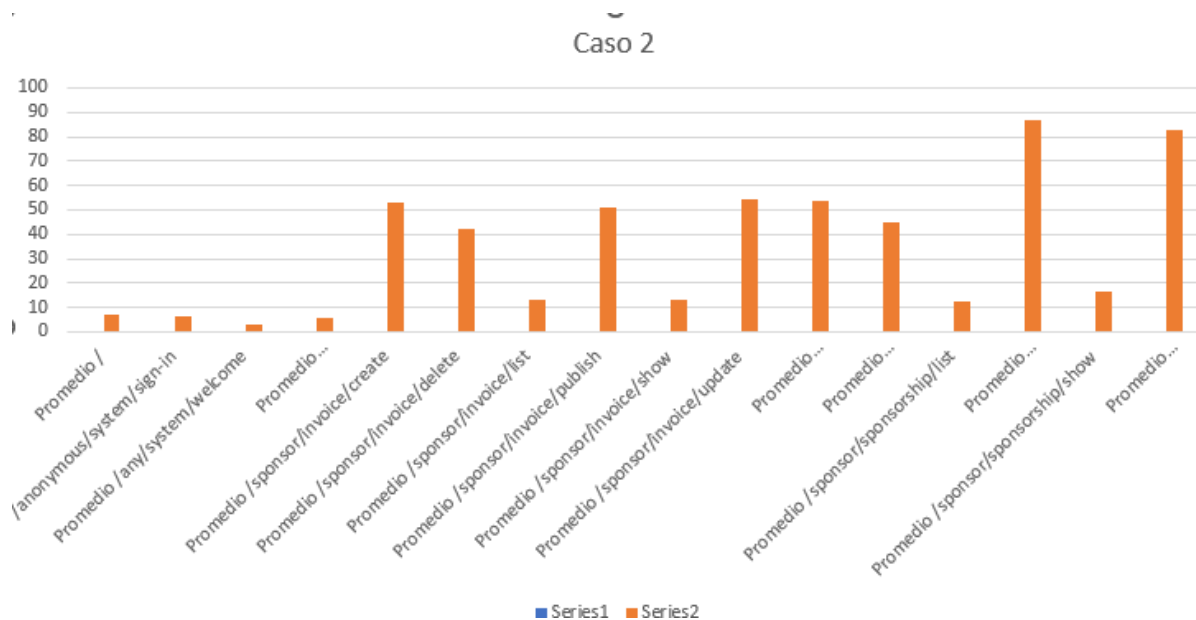
También se ha calculado el intervalo de confianza del 95 % para el tiempo que tarda en atender las solicitudes anteriores mi portátil. Nos quedaría:

Interval (ms)	3,16887788	3,4744908
Interval (s)	0,00316888	0,00347449



Métricas para el Caso 2:

request-path	response-status	time
Promedio /		7,282119443
Promedio /anonymous/system/sign-in		6,439721631
Promedio /any/system/welcome		3,183471905
Promedio /authenticated/system/sign-out		5,533542014
Promedio /sponsor/invoice/create		52,9141835
Promedio /sponsor/invoice/delete		42,04277604
Promedio /sponsor/invoice/list		13,09894469
Promedio /sponsor/invoice/publish		50,85955614
Promedio /sponsor/invoice/show		13,02504702
Promedio /sponsor/invoice/update		54,59588787
Promedio /sponsor/sponsorship/create		53,95081398
Promedio /sponsor/sponsorship/delete		44,61132392
Promedio /sponsor/sponsorship/list		12,68100658
Promedio /sponsor/sponsorship/publish		86,69515257
Promedio /sponsor/sponsorship/show		16,5208449
Promedio /sponsor/sponsorship/update		82,63638826
Promedio general		29,59841686



	Diseño y Pruebas II Acme-Software-Factory
	Testing report

También se ha calculado el intervalo de confianza del 95 % para el tiempo que tarda en atender las solicitudes anteriores el portátil de mi compañero. Nos quedaría:

Interval (ms)	3,80295062	4,16933838
Interval (s)	0,00380295	0,00416934

Análisis de los resultados obtenidos:

Viendo los tiempos de la ejecución de la suite de tests se puede apreciar como el equipo del Caso 1 tiene ligeramente más rendimiento que el del Caso 2.

Como se aprecia en los gráficos, las peticiones más livianas son las que se graban primero en cada caso de test, es decir, las que tienen que ver con la petición al *welcome* y al inicio de sesión. Por otra parte, las peticiones más demandantes son las que tienen que ver con la publicación y actualización de facturas, ya que estas peticiones realizan validaciones y consultas a la base de datos no solo ya sobre facturas, sino también sobre sus patrocinios.

Análisis estadístico:

Estas observaciones son echas a simple vista y debemos realizar un análisis estadístico para comparar los rendimientos. Para ello se calculó un intervalo de confianza del 95% para los tiempos medios de respuesta de uno y otro equipo. Una vez calculado esto, podemos realizar un contraste de hipótesis al 95% de confianza (prueba Z):

<i>Caso 1</i>	
Media	3,32168434
Error típico	0,07795695
Mediana	1,5462995
Moda	1,4725
Desviación estándar	9,07319732
Varianza de la muestra	82,3229096
Curtosis	65,6072689
Coeficiente de asimetría	7,43954566
Rango	165,39
Mínimo	1,2394
Máximo	166,6294
Suma	44995,5361
Cuenta	13546
Nivel de confianza(95,0%)	0,15280646

	Diseño y Pruebas II Acme-Software-Factory
	Testing report

<i>Caso 2</i>	
Media	3,9861445
Error típico	0,09345963
Mediana	1,86093044
Moda	1,7237
Desviación estándar	10,8775126
Varianza de la muestra	118,320281
Curtosis	64,8884228
Coefficiente de asimetría	7,41136286
Rango	197,690935
Mínimo	1,45866567
Máximo	199,149601
Suma	53996,3134
Cuenta	13546
Nivel de confianza(95,0%)	0,18319388

Contraste de hipótesis al 95% de confianza:

Prueba z para medias de dos muestras		
	<i>Caso 1</i>	<i>Caso 2</i>
Media	3,31988585	3,9861445
Varianza (conocida)	82,3229096	118,320281
Observaciones	13560	13560
Diferencia hipotética de las medias	0	
z	-5,49940654	
P(Z<=z) una cola	1,9054E-08	
Valor crítico de z (una cola)	1,64485363	
Valor crítico de z (dos colas)	3,8107E-08	
Valor crítico de z (dos colas)	1,95996398	

El **valor crítico de z (dos colas)** es extremadamente pequeño, lo que indica una diferencia estadísticamente significativa entre los tiempos de respuesta de los dos ordenadores. Además, esto nos dice que la diferencia no es tanto fruto del azar como de la diferencia de rendimiento de ambos equipos. También, si miramos las estadísticas de cada caso, vemos que el intervalo de confianza es menor en el Caso 1, por lo que dicho equipo tendría unos resultados más consistentes y menos variables que el equipo del Caso 2.

	Diseño y Pruebas II Acme-Software-Factory
	Testing report

6. Conclusiones

En conclusión, tras analizar toda la información anterior, podemos afirmar que tenemos bien cubierta toda la funcionalidad de los requisitos S04-006 y S04-007.

Por último, dejar claro que tanto toda la información anterior como la realización de las pruebas mostradas se ha realizado sobre la rama “*#411/R/Task-S04-009*”, por si desea reproducir o verificar lo que vea necesario. En la bibliografía puede encontrar el enlace al repositorio del proyecto.

	Diseño y Pruebas II Acme-Software-Factory
	Testing report

7. Bibliografía

- <https://github.com/DP2-C1-009/Acme-SF-D04>