

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: C1-009**

**Autores por orden alfabético**

León Madroñal, Juan Carlos

|   |  |
|---|--|
|  | Diseño y Pruebas II<br>Acme-Software-Factory |
|   | <b>Testing report</b>                        |

## Índice de contenido

|                             |    |
|-----------------------------|----|
| 1. Resumen ejecutivo .....  | 3  |
| 2. Tabla de revisiones..... | 4  |
| 3. Introducción .....       | 5  |
| 4. Functional testing.....  | 6  |
| 5. Performance testing..... | 10 |
| 6. Conclusiones .....       | 12 |
| 7. Bibliografía .....       | 13 |

|   |  |
|---|--|
|  | Diseño y Pruebas II<br>Acme-Software-Factory |
|   | <b>Testing report</b>                        |

## 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<br>Acme-Software-Factory |
|   | <b>Testing report</b>                        |

## 2. Tabla de revisiones

| Fecha      | Versión | Descripción                   |
|------------|---------|-------------------------------|
| 27/05/2024 | 1.0     | Primera versión del documento |

|   |  |
|---|--|
|  | Diseño y Pruebas II<br>Acme-Software-Factory |
|   | <b>Testing report</b>                        |

### 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.

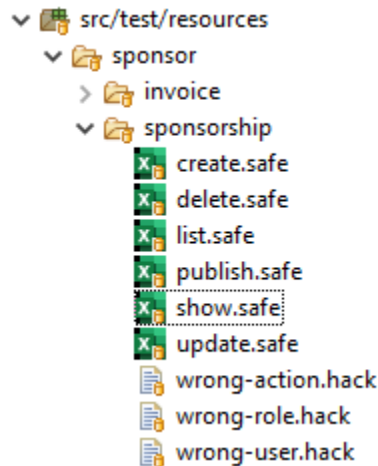
Este informe busca proporcionar una evaluación clara y precisa del sistema, asegurando su calidad y rendimiento.

|   |  |
|---|--|
|  | Diseño y Pruebas II<br>Acme-Software-Factory |
|   | Testing report                               |

## 4. Functional testing

### Requirement #6:

Para este requisito, se han implementado las siguientes pruebas:



En el *list* se ha comprobado que carga bien el listado de los patrocinios, mientras que para el *show* se ha entrado en cada uno de ellos para ver que no hay problema.

En el *create* y el *update* se ha seguido la metodología explicada en la que se comprueba todas las posibilidades existentes en cada campo teniendo en cuenta sus restricciones.









En el *publish* se comprueba que se puede publicar correctamente, y que no deja publicar un patrocinio si no tiene todas sus facturas publicadas y las cantidades totales de estas no suman la cantidad del propio patrocinio.

En el *delete* se comprueba que se puede eliminar sin problemas un patrocinio en modo borrador.

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 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     |  | 90,5 %  | 1.318 | 139 | 1.457 |
| > | SponsorSponsorshipController.java     |  | 100,0 % | 35    | 0   | 35    |
| > | SponsorSponsorshipCreateService.java  |  | 96,1 %  | 293   | 12  | 305   |
| > | SponsorSponsorshipDeleteService.java  |  | 58,1 %  | 122   | 88  | 210   |
| > | SponsorSponsorshipListService.java    |  | 94,2 %  | 65    | 4   | 69    |
| > | SponsorSponsorshipPublishService.java |  | 95,5 %  | 365   | 17  | 382   |
| > | SponsorSponsorshipShowService.java    |  | 97,2 %  | 138   | 4   | 142   |
| > | SponsorSponsorshipUpdateService.java  |  | 95,5 %  | 300   | 14  | 314   |

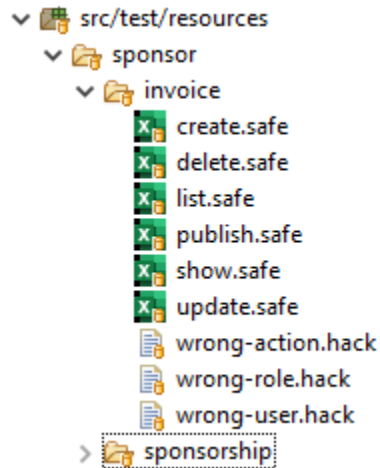
|   |  |
|---|--|
|  | Diseño y Pruebas II<br>Acme-Software-Factory |
|   | <b>Testing report</b>                        |

Como se puede ver, tenemos una cobertura muy destacable salvo en el *delete*, ya que la operación de *unbind* no se llega a ejecutar nunca.

|   |  |
|---|--|
|  | Diseño y Pruebas II<br>Acme-Software-Factory |
|   | Testing report                               |

### Requirement #7:

Para este requisito, se han implementado las siguientes pruebas:



En el *list* se ha comprobado que carga bien el listado de facturas de un patrocinio, mientras que para el *show* se ha entrado en cada una de las facturas para ver que no hay problema.

En el *create* y el *update* se ha seguido la metodología explicada en la que se comprueba todas las posibilidades existentes en cada campo teniendo en cuenta sus restricciones.









En el *publish* se comprueba que se puede publicar correctamente, es decir, a parte de que no fallen las mismas cosas que un *update*, que no se pueda publicar si la cantidad total de la factura sumada a las demás supera a la cantidad del patrocinio que tienen asociado.

En el *delete* se comprueba que se puede eliminar sin problemas una factura en modo borrador.

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    |  | 90,0 %  | 1.331 | 148 | 1.479 |
| > | SponsorInvoiceController.java     |  | 100,0 % | 35    | 0   | 35    |
| > | SponsorInvoiceCreateService.java  |  | 92,3 %  | 313   | 26  | 339   |
| > | SponsorInvoiceDeleteService.java  |  | 57,2 %  | 107   | 80  | 187   |
| > | SponsorInvoiceListService.java    |  | 96,4 %  | 107   | 4   | 111   |
| > | SponsorInvoicePublishService.java |  | 94,7 %  | 324   | 18  | 342   |
| > | SponsorInvoiceShowService.java    |  | 96,9 %  | 125   | 4   | 129   |
| > | SponsorInvoiceUpdateService.java  |  | 95,2 %  | 320   | 16  | 336   |



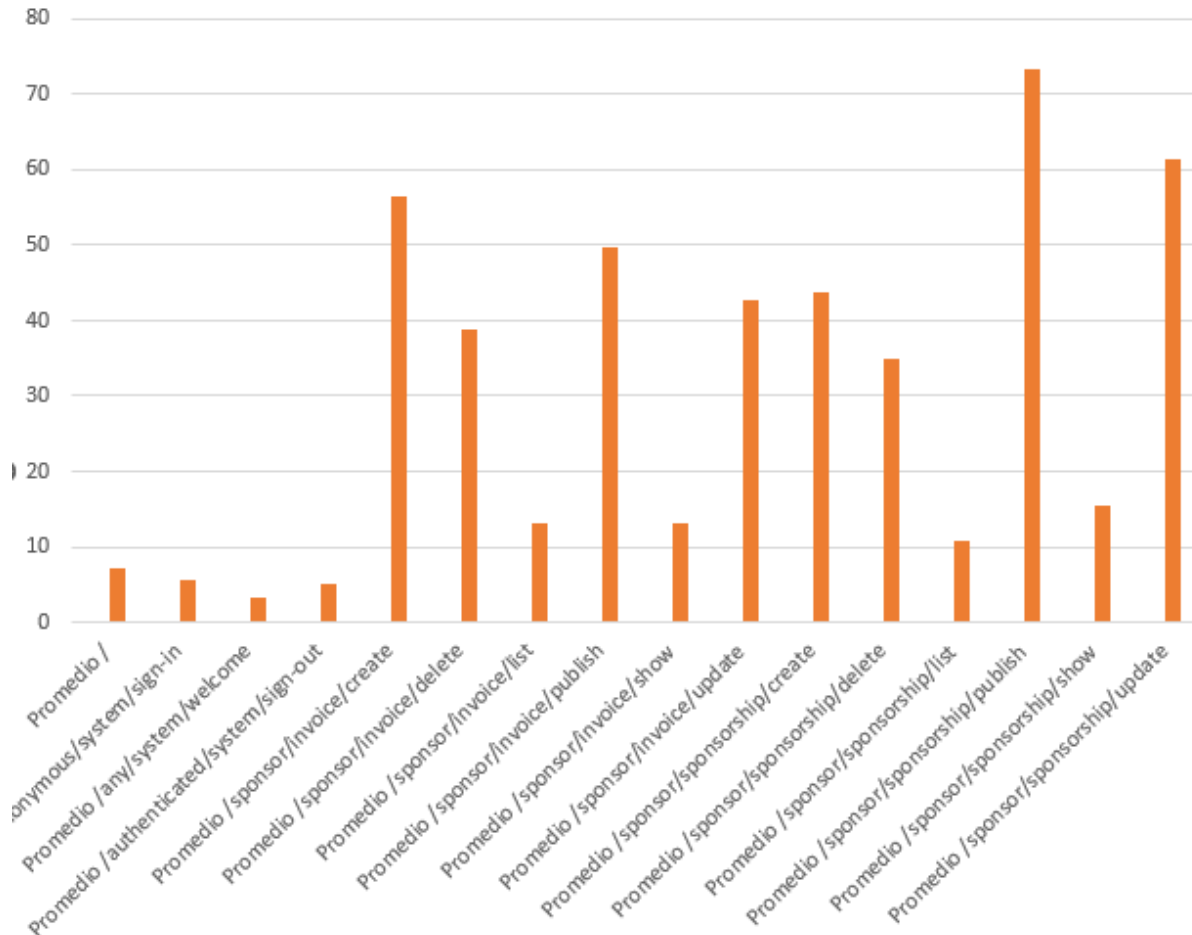
|   |  |
|---|--|
|  | Diseño y Pruebas II<br>Acme-Software-Factory |
|   | <b>Testing report</b>                        |

Como se puede ver, tenemos una cobertura muy destacable salvo en el *delete*, ya que la operación de *unbind* no se llega a ejecutar nunca.



## 5. Performance testing

Tras hacer un análisis de los resultados de reproducir las pruebas anteriores, podemos generar este gráfico que muestra cuáles son las solicitudes más ineficientes:



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) | 19,0406014 | 23,4616618 |
| Interval (s)  | 0,0190406  | 0,02346166 |

Si volvemos a reproducir las pruebas, pero en el ordenador de mi compañero, tras analizar y trabajar con estos nuevos datos obtenemos que el intervalo en este caso es:

|               |             |             |
|---------------|-------------|-------------|
| Interval (ms) | 31,01012491 | 41,1212379  |
| Interval (s)  | 0,031010125 | 0,041121238 |

|   |  |
|---|--|
|  | Diseño y Pruebas II<br>Acme-Software-Factory |
|   | <b>Testing report</b>                        |

Como se aprecia en las dos tablas, mi compañero y yo obtuvimos intervalos un tanto diferentes, lo cual puede deberse a variaciones en el rendimiento de nuestros portátiles, a pesar de que tienen características similares. No obstante, aprovechando que tenemos dos muestras de datos, se ha realizado una prueba Z para medias de dos muestras, para poder obtener más información respecto a la comparación. De esta podemos destacar el valor crítico de z (dos colas), el cual resulta ser 0,65.

Este valor de z sugiere que la diferencia entre los intervalos de tiempo medidos en ambos ordenadores no es estadísticamente significativa. En otras palabras, un valor de z tan cercano a cero indica que las variaciones observadas en los intervalos pueden atribuirse al azar más que a una diferencia real en el rendimiento de los portátiles.

|   |  |
|---|--|
|  | Diseño y Pruebas II<br>Acme-Software-Factory |
|   | <b>Testing report</b>                        |

## 6. Conclusiones

En conclusión, tras analizar toda la información anterior, podemos afirmar que tenemos bien cubierta toda la funcionalidad y que nuestro proyecto está optimizado.

Por último, dejar claro que tanto toda la información anterior como la realización de las pruebas mostradas se ha realizado sobre la rama “*#370/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<br>Acme-Software-Factory |
|   | <b>Testing report</b>                        |

## 7. Bibliografía

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