

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**

García Galocha, Rafael David

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


## Índice de contenido

<b>1. Resumen ejecutivo .....</b>	<b>3</b>
<b>2. Tabla de revisiones .....</b>	<b>4</b>
<b>3. Introducción .....</b>	<b>5</b>
<b>4. Functional testing .....</b>	<b>6</b>
<b>5. Performance testing .....</b>	<b>13</b>
<b>6. Conclusiones .....</b>	<b>15</b>
<b>7. Bibliografía .....</b>	<b>16</b>

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

## 2. Tabla de revisiones

Fecha	Versión	Descripción
23/05/2024	1.0	Primera versión del documento
27/05/2024	1.1	Prueba de rendimiento en otro ordenador

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

## 4. Functional testing

### Requirement #6:

Para este requisito, se han implementado las siguientes pruebas:

-  developer
  -  dashboard
  -  training-module
    -  create.safe
    -  delete.safe
    -  list.safe
    -  publish.safe
    -  publish2.safe
    -  readonly.hack
    -  show.safe
    -  update.safe
    -  wrong-action.hack
    -  wrong-role.hack
    -  wrong-user.hack

En el *list* se ha comprobado que carga bien el listado de los módulos, 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 módulo si no tiene sesiones o si tiene alguna sesión sin publicar.


Y en el *delete* se comprueba que se puede eliminar sin problemas un módulo en modo borrador.

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

















En *wrong user* se hace login como developer2 y se comprueba que no deja hacer acciones sobre un módulo perteneciente al developer1.

En *wrong action* se hace login con developer1 y se comprueba que no podemos borrar, actualizar o publicar un módulo ya publicado.


Y en *readonly* se comprueba que, al crear o actualizar, si quitamos desde el modo desarrollador del navegador el `readonly=""` al campo que lo tiene y cambiamos el dato, al hacer post no ocurre absolutamente nada y no se envía a la base de datos.

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

Tras ejecutar el replayer obtenemos la siguiente cobertura:















▼  acme.features.developer.trainingModule	 86,4 %
>  DeveloperTrainingModuleDeleteService.java	 57,3 %
>  DeveloperTrainingModuleCreateService.java	 87,9 %
>  DeveloperTrainingModulePublishService.java	 90,4 %
>  DeveloperTrainingModuleUpdateService.java	 93,5 %
>  DeveloperTrainingModuleShowService.java	 96,6 %
>  DeveloperTrainingModuleListService.java	 92,6 %
>  DeveloperTrainingModuleController.java	 100,0 %

Como vemos, la mayoría de la funcionalidad está cubierta, aunque parece que en el delete no del todo. Sin embargo, esto es lógico y se debe a que al realizar esta acción, no se está ejecutando el código de la función unbind.

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

### Requirement #7:

Para este requisito, se han implementado las siguientes pruebas:

-  developer
  -  dashboard
  -  training-module
  -  training-session
    -  create.safe
    -  delete.safe
    -  list.safe
    -  publish.safe
    -  readonly.hack
    -  show.safe
    -  update.safe
    -  wrong-action.hack
    -  wrong-role.hack
    -  wrong-user.hack

En el *list* se ha comprobado que carga bien el listado de las sesiones de un módulo, mientras que para el *show* se ha entrado en cada uno de ellas 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 en el *delete* se comprueba que se puede eliminar sin problemas una sesión en modo borrador.


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

En *wrong user* se hace login como developer2 y se comprueba que no deja hacer acciones sobre una sesión perteneciente a un módulo de developer1.









En *wrong action* se hace login con developer1 y se comprueba que no podemos borrar, actualizar o publicar una sesión ya publicada.

Y en *readonly* se comprueba que, al crear o actualizar, si quitamos desde el modo desarrollador del navegador el `readonly=""` al campo que lo tiene y cambiamos el dato, al hacer post no ocurre absolutamente nada y no se envía a la base de datos.




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

Tras ejecutar el replayer obtenemos la siguiente cobertura:





▼	acme.features.developer.trainingSession		85,2 %
>	DeveloperTrainingSessionPublishService.java		70,0 %
>	DeveloperTrainingSessionDeleteService.java		61,0 %
>	DeveloperTrainingSessionCreateService.java		93,9 %
>	DeveloperTrainingSessionUpdateService.java		94,9 %
>	DeveloperTrainingSessionShowService.java		96,5 %
>	DeveloperTrainingSessionListService.java		96,0 %
>	DeveloperTrainingSessionController.java		100,0 %

Como vemos, la mayoría de la funcionalidad está cubierta, pero ocurre lo mismo que antes. Cuando se realizan esas dos acciones que tienen menos coverage, no se está ejecutando el código de la función unbind.

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








### Requirement #8:

Para este requisito, se han implementado las siguientes pruebas:


-  > developer
  -  > dashboard
    -  show.safe
    -  wrong-role.hack

En este caso, solo hemos comprobado que se muestran bien las estadísticas correspondientes al developer logueado, y que si no eres developer no tienes acceso.

Tras ejecutar el replayer obtenemos la siguiente cobertura:






  acme.features.developer.dashboard	 99,4 %
>  DeveloperDashboardShowService.java	 99,3 %
>  DeveloperDashboardController.java	 100,0 %

Como podemos ver, la cobertura es casi perfecta, prácticamente la máxima que podemos alcanzar.

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

### Requirements #17 #18:













Para este requisito, se han implementado las siguientes pruebas:

-  **authenticated**
  -  **developer**
    -  **create.safe**
    -  **readonly.hack**
    -  **update.safe**


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.

Y en *readonly* se comprueba que si quitamos desde el modo desarrollador del navegador el `readonly=""` al campo que lo tiene y cambiamos el dato, al hacer post no ocurre absolutamente nada y no se envía a la base de datos.

Tras ejecutar el replayer obtenemos la siguiente cobertura:





  <b>acme.features.authenticated.developer</b>	 90,9 %
  <b>AuthenticatedDeveloperUpdateService.java</b>	 89,3 %
  <b>AuthenticatedDeveloperCreateService.java</b>	 91,7 %
  <b>AuthenticatedDeveloperController.java</b>	 100,0 %

Como podemos ver, la cobertura es casi perfecta, prácticamente la máxima que podemos alcanzar.

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









### Requirements #19:

Para este requisito, se han implementado las siguientes pruebas:

- ▼  any
  - ▼  training-module
    -  list.safe
    -  show.safe

Para este caso, en el *list* se ha comprobado que carga bien el listado de módulos que están publicados, y en el *show* se ha entrado en cada uno de ellos para ver que no hay problema.

Tras ejecutar el replayer obtenemos la siguiente cobertura:

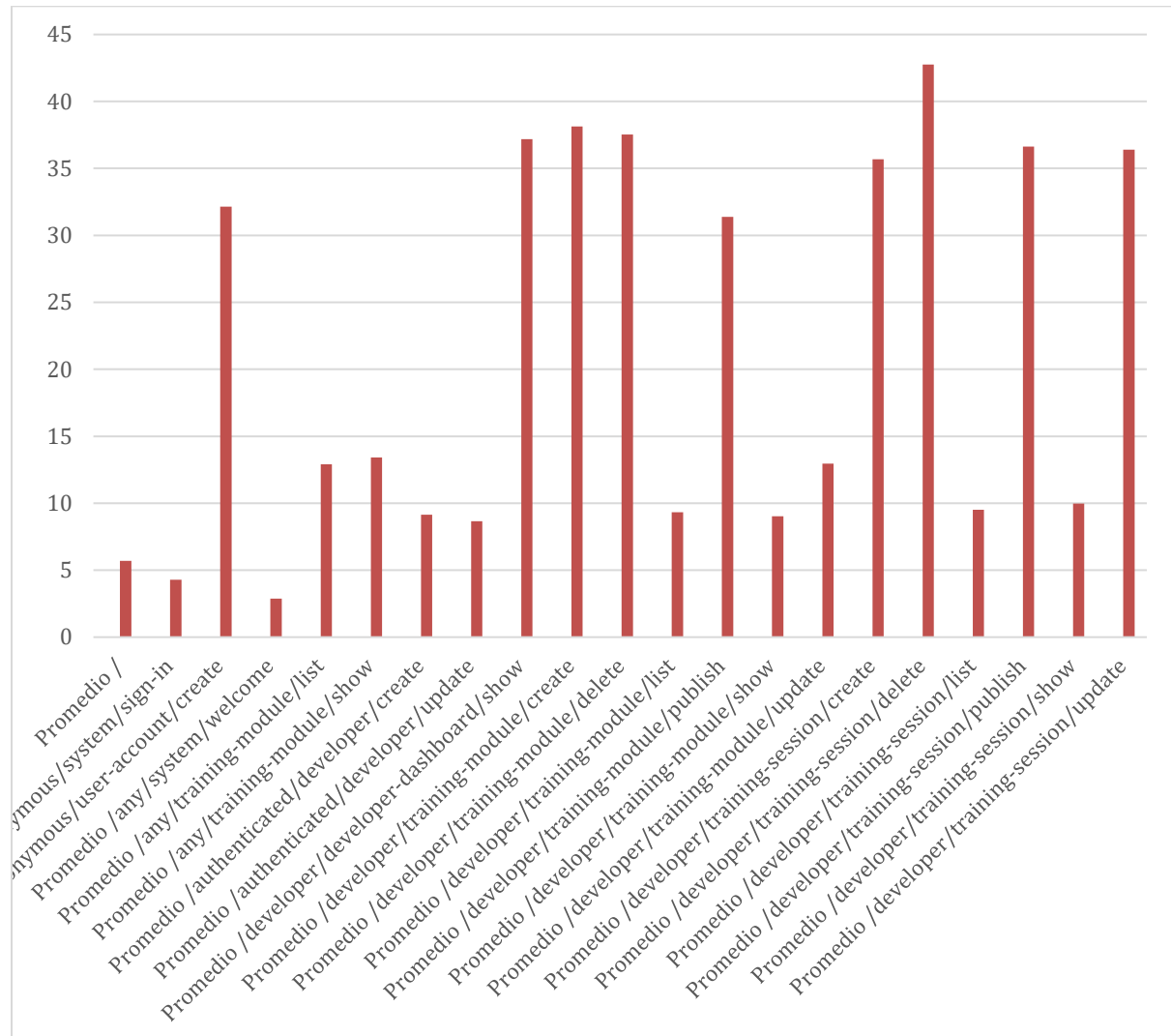
▼  acme.features.any.trainingModule	 94,9 %
>  AnyTrainingModuleShowService.java	 95,7 %
>  AnyTrainingModuleListService.java	 91,7 %
>  AnyTrainingModuleController.java	 100,0 %

Como podemos ver, la cobertura es casi perfecta, prácticamente la máxima que podemos alcanzar.



## 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)	15,7412646	17,975103
Interval (s)	0,01574126	0,0179751

Si volvemos a reproducir las pruebas, pero en el ordenador de mi compañero (student 4), tras analizar y trabajar con estos nuevos datos obtenemos que el intervalo en este caso es:

Interval (ms)	15,471976	17,4083077
Interval (s)	0,01547198	0,01740831

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


A simple vista se pueden comparar y observar cómo mi compañero y yo obtenemos intervalos muy parecidos dado a que nuestros portátiles tienen características bastante 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: 0,57884557

	Diseño y Pruebas II 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 esta funciona correctamente bajo las condiciones evaluadas. Además, tras las pruebas realizadas en diferentes ordenadores, podemos observar que los resultados son positivos, lo que nos hace pensar que nuestro proyecto está bien optimizado.

Por último, dejar claro que tanto toda la información anterior como la realización de las pruebas mostradas se ha realizado sobre la rama “*#349/Task-S03-009*” (donde se encuentran solo las pruebas relacionadas con la funcionalidad del student 3), 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
	<b>Testing report</b>

## 7. Bibliografía

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