

# TESTING REPORT

## Tabla de revisiones

Nº de revisión	Fecha	Descripción
001	26/05/24	Creacion y finalización del documento

## Información general

**Fecha:** 26/05/24

**Grupo:** C1.006

**Repositorio:** [Danielruizlopezcc/Acme-SF-D04](#)

**Miembro:** Daniel Ruiz López ([danruilop1@alum.us.es](mailto:danruilop1@alum.us.es))

## Tabla de contenidos

[Tabla de revisiones](#)

[Información general](#)

[Tabla de contenidos](#)

[Resumen ejecutivo](#)

[Introducción](#)

[Contenidos](#)

[Testing Funcional](#)

[Project](#)

[Cobertura Project](#)

[ProjectUserStory](#)

[Cobertura ProjectUserStory](#)

[UserStory](#)

[Cobertura UserStory](#)

[Testing de rendimiento](#)

[Análisis de rendimiento](#)

[Intervalos de confianza](#)

## Resumen ejecutivo

El proyecto Acme-SF-D04 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.

## Introducción

En el contexto del proyecto Acme-SF, se presenta el siguiente documento de testing, que tiene como objetivo presentar el informe de pruebas realizadas a los servicios de las clases Project, ProjectUserStory y UserStory.

Se mostrarán todos los resultados obtenidos en el test funcional, donde se incluirán tanto casos de prueba negativos y positivos, así como los resultados obtenidos en la cobertura de código para cada uno de los servicios de estas clases. También se incluirá una sección para mostrar los resultados obtenidos en el test de rendimiento, acompañados de gráficas y tablas que permitirán visualizar de manera más clara todos los resultados.

## Contenidos

### Testing Funcional

#### Project

##### ManagerProjectCreateService

- **Safe testing:** en este servicio fue necesario hacer iteraciones de creación, probando enviar el formulario vacío, con errores, con distinto formato de url, string, money...
- **Hacking**

- Intentar acceder con un usuario distinto a manager al formulario de creación
- **Bugs:** no encontré problemas con este servicio

### **ManagerProjectDeleteService**

- **Safe testing:** en este servicio fue necesario borrar algunos proyectos
- **Hacking**
  - Intentar borrar mediante post-hacking:
    - Un proyecto que no exista
    - Un proyecto de otro manager
    - Un proyecto publicado
- **Bugs:** no encontré problemas con este servicio

### **ManagerProjectListMine**

- **Safe testing:** en este servicio fue necesario listar los proyectos
- **Hacking**
  - Intentar acceder con un usuario distinto a manager al listado de proyectos
- **Bugs:** no encontré problemas con este servicio

### **ManagerProjectPublishService**

- **Safe testing:** en este servicio fue necesario hacer iteraciones de publicación, probando publicar el formulario vacío, con errores, con distinto formato de url, string, money, con/sin user stories asociadas
- **Hacking**
  - Intentar publicar un proyecto que ya está publicado
  - Intentar publicar un proyecto que no existe
  - Intentar publicar un proyecto de otro manager
- **Bugs:** no encontré problemas con este servicio

## ManagerProjectShowService

- **Safe testing:** en este servicio fue necesario meterse en distintos proyectos para ver su información
- **Hacking**
  - Intentar ver la información del proyecto con usuario distinto a manager
  - Intentar ver la información del proyecto con manager2
- **Bugs:** no encontré problemas con este servicio

## ManagerProjectUpdateService

- **Safe testing:** en este servicio fue necesario hacer iteraciones de actualización, probando enviar el formulario vacío, con errores, con distinto formato de url, string, money...
- **Hacking**
  - Intentar actualizar la información de un proyecto que no existe
  - Intentar actualizar la información de un proyecto publicado
  - Intentar actualizar la información de un proyecto de manager2
- **Bugs:** no encontré problemas con este servicio

## Cobertura Project

A continuación se muestra el porcentaje de cobertura de sentencias de todos los servicios de la clase Project:

▼ acme.features.manager.projects	90.3 %	1,352	146	1,498
> ManagerProjectDeleteService.java	74.5 %	239	82	321
> ManagerProjectPublishService.java	94.7 %	323	18	341
> ManagerProjectCreateService.java	93.5 %	243	17	260
> ManagerProjectUpdateService.java	94.1 %	271	17	288
> ManagerProjectShowService.java	95.4 %	146	7	153
> ManagerProjectListMineService.java	95.0 %	95	5	100
> ManagerProjectController.java	100.0 %	35	0	35

Como se puede observar en la imagen la cobertura es de un 90.3%, teniendo en cuenta que hay líneas que no se analizan tales como las cabeceras de funciones, la internacionalización de los booleans en el unbind (como el endpoint pone =en, solo he hecho las pruebas en inglés) y assert object != null;

## ProjectUserStory

### ManagerProjectUserStoryCreateService

- **Safe testing:** en este servicio fue necesario hacer iteraciones de creación, probando enviar el formulario vacío, con un proyecto pero sin user story y viceversa y creando la misma asignación
- **Hacking**
  - Intentar acceder con un usuario distinto a manager al formulario de creación
- **Bugs:** no encontré problemas con este servicio

### ManagerProjectUserStoryDeleteService

- **Safe testing:** en este servicio fue necesario borrar algunos proyectos
- **Hacking**
  - Intentar borrar mediante post-hacking:
    - Una asignación que no exista
    - Una asignación de otro manager
    - Una asignación con un proyecto publicado
- **Bugs:** al intentar borrar una asignación con proyectos publicados, me dejaba, tuve que arreglar el authorise del servicio y ya me saltaba el error 500

### ManagerProjectUserStoryListMine

- **Safe testing:** en este servicio fue necesario listar las asignaciones
- **Hacking**

- Intentar acceder con un usuario distinto a manager al listado de asignaciones
- **Bugs:** no encontré problemas con este servicio

### ManagerProjectUserStoryShowService

- **Safe testing:** en este servicio fue necesario meterse en distintas asignaciones para ver su información
- **Hacking**
  - Intentar ver la información de la asignación con usuario distinto a manager
  - Intentar ver la información de la asignación con manager2
- **Bugs:** no encontré problemas con este servicio

### Cobertura ProjectUserStory

A continuación se muestra el porcentaje de cobertura de sentencias de todos los servicios de la clase ProjectUserStory:

acme.features.manager.projectUserStories	82.5 %	570	121	691
> ManagerProjectUserStoriesDeleteService.java	55.3 %	110	89	199
> ManagerProjectUserStoriesCreateService.java	90.8 %	227	23	250
> ManagerProjectUserStoriesShowService.java	96.4 %	132	5	137
> ManagerProjectUserStoriesListMineService.java	95.1 %	77	4	81
> ManagerProjectUserStoriesController.java	100.0 %	24	0	24

Como se puede observar en la imagen la cobertura es de un 82.5%, teniendo en cuenta que hay líneas que no se analizan tales como las cabeceras de funciones, la internacionalización de los booleans en el unbind (como el endpoint pone =en, solo he hecho las pruebas en inglés) y assert object != null;

También se ve como el Delete tiene un 55%, esto es simplemente porque en el unbind tengo la funcionalidad de las SelectChoices y eso no se ha tenido en cuenta en la cobertura, porque si he intentado borrar una asignación y me ha dado error, por lo tanto saltaría el unbind.

## UserStory

### ManagerUserStoryCreateService

- **Safe testing:** en este servicio fue necesario hacer iteraciones de creación, probando enviar el formulario vacío, con errores, con distinto formato de url, string...
- **Hacking**
  - Intentar acceder con un usuario distinto a manager al formulario de creación
- **Bugs:** no encontré problemas con este servicio

### ManagerUserStoryDeleteService

- **Safe testing:** en este servicio fue necesario borrar algunas user stories
- **Hacking**
  - Intentar borrar mediante post-hacking:
    - Una user stories que no exista
    - Una user stories de otro manager
    - Una user stories publicada
- **Bugs:** no encontré problemas con este servicio

### ManagerUserStoryListMine

- **Safe testing:** en este servicio fue necesario listar las user stories
- **Hacking**

- Intentar acceder con un usuario distinto a manager al listado user stories
- **Bugs:** no encontré problemas con este servicio

### **ManagerUserStoryPublishService**

- **Safe testing:** en este servicio fue necesario hacer iteraciones de publicación, probando publicar el formulario vacío, con errores, con distinto formato de url, string...
- **Hacking**
  - Intentar publicar una user stories que ya está publicado
  - Intentar publicar una user stories que no existe
  - Intentar publicar una user stories de otro manager
- **Bugs:** no encontré problemas con este servicio

### **ManagerUserStoryShowService**

- **Safe testing:** en este servicio fue necesario meterse en distintas user stories para ver su información
- **Hacking**
  - Intentar ver la información de la user story con usuario distinto a manager
  - Intentar ver la información de la user story con manager2
- **Bugs:** no encontré problemas con este servicio

### **ManagerUserStoryUpdateService**

- **Safe testing:** en este servicio fue necesario hacer iteraciones de actualización, probando enviar el formulario vacío, con errores, con distinto formato de url, string...
- **Hacking**
  - Intentar actualizar la información de una user story que no existe
  - Intentar actualizar la información de una user story publicado



- Intentar actualizar la información de una user story de manager2
- **Bugs:** no encontré problemas con este servicio

## Cobertura UserStory

A continuación se muestra el porcentaje de cobertura de sentencias de todos los servicios de la clase UserStory:

▼ acme.features.manager.userStories	89.0 %	978	121	1,099
> ManagerUserStoryDeleteService.java	69.3 %	131	58	189
> ManagerUserStoryUpdateService.java	91.3 %	179	17	196
> ManagerUserStoryCreateService.java	90.7 %	156	16	172
> ManagerUserStoryPublishService.java	91.2 %	165	16	181
> ManagerUserStoryShowService.java	95.1 %	116	6	122
> ManagerUserStoryListByProyectService.java	96.6 %	115	4	119
> ManagerUserStoryListMineService.java	94.9 %	74	4	78
> ManagerUserStoryController.java	100.0 %	42	0	42

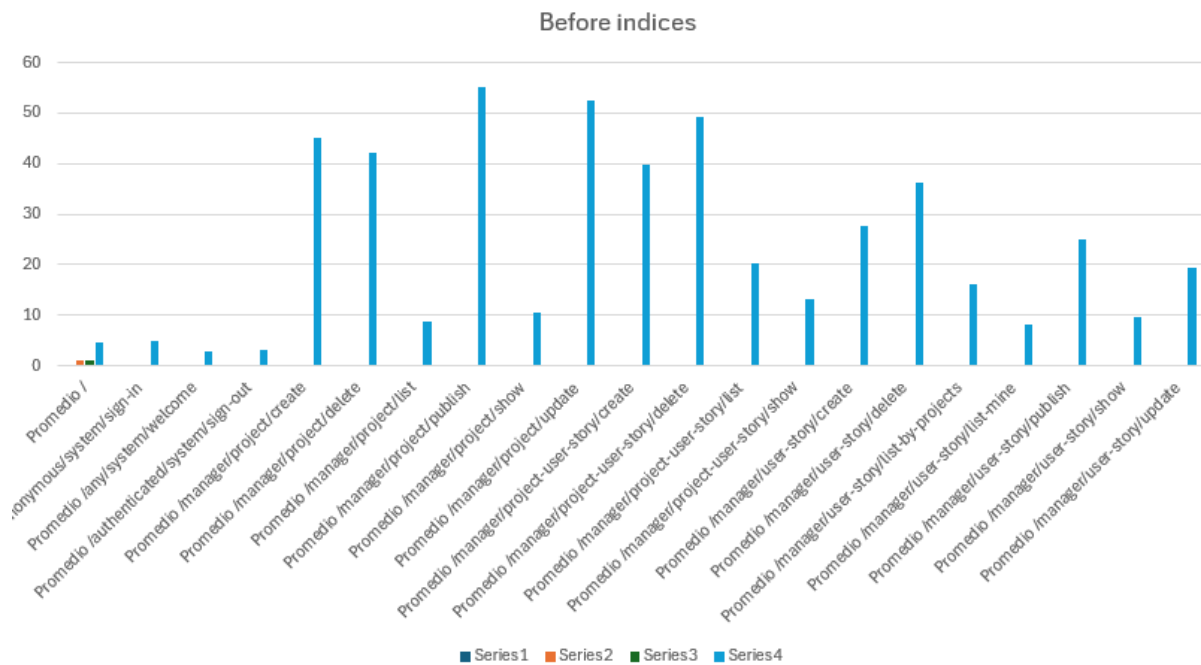
Como se puede observar en la imagen la cobertura es de un 89%, teniendo en cuenta que hay líneas que no se analizan tales como las cabeceras de funciones, la internacionalización de los booleans en el unbind (como el endpoint pone =en, solo he hecho las pruebas en inglés) y assert object != null;

## Testing de rendimiento

### Análisis de rendimiento

Tras la ejecución de los tests, y con los datos recopilados sobre las peticiones realizadas en los mismos, se llevó a cabo un análisis de los resultados obtenidos en cuanto al rendimiento de la aplicación, tomando como referencia el tiempo de respuesta de los servicios, centrándonos en los servicios de las clases Project y UserStory. A continuación se muestran los datos del promedio de cada petición, aunque para mayor claridad también se proporciona una gráfica de barras:

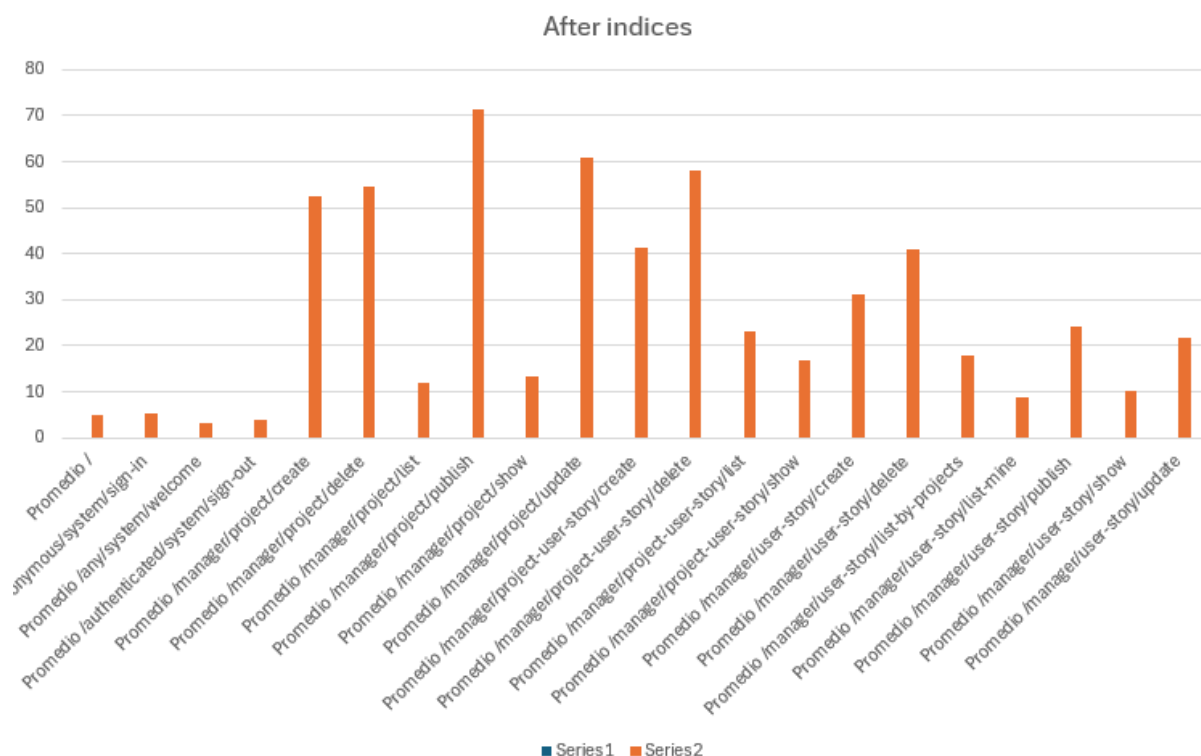
request-path	response-status	time
Promedio /		4.63291047
Promedio /anonymous/system/sign-in		5.00560094
Promedio /any/system/welcome		2.91577561
Promedio /authenticated/system/sign-out		3.10299143
Promedio /manager/project/create		45.2291057
Promedio /manager/project/delete		42.2506875
Promedio /manager/project/list		8.65991358
Promedio /manager/project/publish		55.2995824
Promedio /manager/project/show		10.44716
Promedio /manager/project/update		52.4460158
Promedio /manager/project-user-story/create		39.746912
Promedio /manager/project-user-story/delete		49.2832571
Promedio /manager/project-user-story/list		20.2274429
Promedio /manager/project-user-story/show		13.0335133
Promedio /manager/user-story/create		27.5520474
Promedio /manager/user-story/delete		36.1568
Promedio /manager/user-story/list-by-projects		16.2171
Promedio /manager/user-story/list-mine		8.00631746
Promedio /manager/user-story/publish		25.062145
Promedio /manager/user-story/show		9.64287073
Promedio /manager/user-story/update		19.3393286
Promedio general		14.1094605



Como vemos en la gráfico, existe grandes diferencia entre los tiempos de respuesta de las peticiones de inicio de sesión o bienvenido, comparado con las features de ambas entidades que estamos analizando. En concreto nos centraremos en los servicios de la clase Project, que parece tener una de las peticiones más ineficientes y lentas de la aplicación, por ejemplo publicar o actualizar un contrato superan los 50 ms de promedio, 10 veces más que por ejemplo el inicio de sesión. El promedio de todas las respuestas es de 14 ms.

Esta identificación de los servicios más lentos fueron el punto de partida para optimizar la aplicación, y se llevaron a cabo una serie de cambios en el código, en concreto se añadieron índices a las tablas de la base de datos de la entidad Contract, con el objetivo de mejorar el rendimiento de las peticiones. A continuación se muestra una gráfica de barras con los resultados obtenidos tras la optimización y los nuevos promedios de tiempo de respuesta:

request-path	response-status	time
Promedio /		5.10605814
Promedio /anonymous/system/sign-in		5.254429245
Promedio /any/system/welcome		3.18468374
Promedio /authenticated/system/sign-out		3.910822857
Promedio /manager/project/create		52.48310857
Promedio /manager/project/delete		54.6976875
Promedio /manager/project/list		11.90867901
Promedio /manager/project/publish		71.23651176
Promedio /manager/project/show		13.25429833
Promedio /manager/project/update		60.96623684
Promedio /manager/project-user-story/create		41.249852
Promedio /manager/project-user-story/delete		58.00432857
Promedio /manager/project-user-story/list		23.20676667
Promedio /manager/project-user-story/show		16.95262
Promedio /manager/user-story/create		31.26587895
Promedio /manager/user-story/delete		41.10498
Promedio /manager/user-story/list-by-projects		17.8054875
Promedio /manager/user-story/list-mine		8.735204762
Promedio /manager/user-story/publish		24.12929
Promedio /manager/user-story/show		10.36322439
Promedio /manager/user-story/update		21.79365
Promedio general		16.41239534



Observamos que aún añadiendo índices los tiempos de respuesta no han mejorado, esto se puede debe a haber añadido un índice en alguna de las 2 entidades cuando posiblemente no era necesario, o simplemente que en este replayer el ordenador ha ido mas lento.

## Intervalos 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		
Media	14.1094605		Media	16.2459864	
Error típico	0.73436977		Error típico	0.8696515	
Mediana	6.2669		Mediana	6.87705	
Moda	2.3165		Moda	8.9186	
Desviación estándar	21.2460169		Desviación estándar	25.1448118	
Varianza de la muestra	451.393234		Varianza de la muestra	632.261559	
Curtosis	13.0046035		Curtosis	13.348662	
Coefficiente de asimetría	3.21253833		Coefficiente de asimetría	3.29137485	
Rango	173.3268		Rango	194.8845	
Mínimo	1.7645		Mínimo	1.8094	
Máximo	175.0913		Máximo	196.6939	
Suma	11809.6184		Suma	13581.6446	
Cuenta	837		Cuenta	836	
Nivel de confianza(95.0%)	1.44142516		Nivel de confianza(95.0%)	1.70695986	
Intervalo (ms)	12.6680353	15.5508856	Intervalo (ms)	14.5390265	17.9529462
Intervalo(s)	0.012668035	0.01555089	Intervalo (s)	0.014539027	0.01795295

## Hypothesis contrast

Como se comentó anteriormente se ha llevado a cabo una refactorización mediante el uso de índices para aumentar el rendimiento de las queries de los repositorios asociados a las entidades Project y UserStory. Para comprobar si esta refactorización ha tenido un impacto en el rendimiento de la aplicación, se ha llevado a cabo un contraste de hipótesis, mas

concretamente, se ha realizado nuevamente en excel con el complemento Herramientas para el análisis, un Z-test (con  $\alpha = 0.05$ ) para evaluar si los nuevos tiempos de respuesta son mejores o peores y en que medida. Para poder realizar el Z-test, se ha debido de recopilar nuevamente los tiempos de respuesta tras la refactorización y obtener los datos estadísticos. A continuación se muestran los datos del análisis post-refactorización, así como los resultados del Z-test.

	<i>BEFORE</i>	<i>AFTER</i>
Media	14.10946045	16.41239534
Varianza (conocida)	451393234	6322615594
Observaciones	837	837
Diferencia hipotética de la	0	
z	-0.000809509	
P(Z<=z) una cola	0.499677053	
Valor crítico de z (una cola)	1.644853627	
Valor crítico de z (dos colas)	0.999354106	
Valor crítico de z (dos colas)	1.959963985	

Como podemos observar, el valor crítico de z (dos colas) es 0.00, para saber si los cambios han sido significativos, hemos de comparar el valor z con  $\alpha$ . En este caso, nuestro z no se encuentra entre el intervalo  $[0.00, \alpha)$ , por lo que podemos afirmar que los cambios realizados no han sido significativos y no han mejorado el rendimiento de la aplicación. Igualmente, esto se veía bastante claro en las imágenes de las gráficas de barras o en los promedios, la refactorización no ha sido exitosa.

## Conclusiones

A lo largo del documento se ha llevado a cabo un análisis de los servicios de las clases Project, ProjectUserStory y UserStory, en el que se han realizado tests funcionales y de rendimiento, con el objetivo de comprobar el correcto funcionamiento de los servicios y el rendimiento de la aplicación. En cuanto a los tests funcionales, se han realizado pruebas tanto positivas como negativas, y se ha comprobado que los servicios funcionan correctamente y no presentan

bugs. En cuanto a los tests de rendimiento, se ha llevado a cabo un análisis de los tiempos de respuesta de las peticiones, y se ha comprobado que la aplicación cumple con el requisito de rendimiento establecido. Además, se ha llevado a cabo una refactorización de los servicios de las clases Project y UserStory, añadiendo índices a las tablas de la base de datos, con el objetivo de mejorar el rendimiento de la aplicación. Tras la refactorización, se ha llevado a cabo un contraste de hipótesis, y se ha comprobado que los cambios realizados no han sido significativos y no han mejorado el rendimiento de la aplicación.