

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática



Grado en Ingeniería Informática – Ingeniería del Software

Diseño y Pruebas 2

Curso 2023 – 2024

## Testing Report Student 1

**David Fuentelsaz Rodríguez**

**davfuerod@alum.us.es**

Fecha	Versión
26/05/2024	v1.0

# ÍNDICE

<b>1. Introducción</b>	<b>3</b>
<b>2. Testeo funcional</b>	<b>3</b>
2.1 Listado de Casos de Prueba por funcionalidad	3
2.2.1 List	3
2.2.2 List (Hacking)	4
2.2.2 List All	4
2.2.3 Show	5
2.2.4 Show (Hacking)	5
2.2.5 Create	6
2.2.6 Create (Hacking)	7
2.2.7 Publish	7
2.2.8 Publish (Hacking)	8
2.2.9 Update	8
2.2.10 Update (Hacking)	9
2.2.11 Delete	9
2.2.12 Delete (Hacking)	10
<b>3. Testeo de rendimiento</b>	<b>11</b>
3.1 Metodología	11
3.2 Resultados	11
3.2.1 Entidades sin índice	11
3.2.2 Entidades con índice	12
3.2.3 Z-Test	13
<b>4. Cobertura</b>	<b>14</b>
4.1 Cobertura de Project	14
4.2 Cobertura de User Story	14
4.3 Cobertura de Project User Story	15
<b>5. Conclusión</b>	<b>15</b>
<b>6. Bibliografía</b>	<b>15</b>

# 1. Introducción

Este informe tiene como objetivo documentar los resultados de las pruebas funcionales y de rendimiento realizadas en el proyecto para garantizar que las funcionalidades del software operen correctamente y que el rendimiento del sistema sea adecuado.

Las pruebas están divididas en dos secciones: pruebas funcionales y pruebas de rendimiento. Las pruebas funcionales utilizan una lista de casos de prueba organizados por funcionalidad, con una descripción breve de cada caso y una evaluación de su eficacia en la detección de errores. Esto permite verificar que cada característica del software cumpla con los requisitos especificados.

Las pruebas de rendimiento, realizadas en dos muestras diferentes, miden el tiempo de respuesta del sistema. Se proporcionan gráficos ilustrativos, intervalos de confianza del 95% para los tiempos de respuesta y un contraste de hipótesis para evaluar los resultados.

## 2. Testeo funcional

Para llevar a cabo las pruebas funcionales, se empleó una metodología sistemática que incluye la identificación y creación de casos de prueba basados en los requisitos del sistema. Cada caso de prueba fue diseñado para verificar una funcionalidad específica del software, asegurando que todos los aspectos críticos fueran evaluados.

### 2.1 Listado de Casos de Prueba por funcionalidad

A continuación se presenta una lista de los casos de prueba agrupados por funcionalidad. Cada caso de prueba incluye una descripción breve y una evaluación de su resultado en la detección de errores.

#### 2.2.1 List

Entidad	Descripción	Resultado
Project	Verificar que un manager puede listar sus proyectos.	La funcionalidad se validó correctamente, el sistema mostró los proyectos del manager logueado.
User story	Verificar que un manager puede listar las historias de usuario asociadas a sus proyectos.	La funcionalidad se validó correctamente, el sistema mostró las historias de usuario de los proyectos del manager logueado.

Assignment	Verificar que un manager puede listar las asignaciones entre sus proyectos e historias de usuario.	La funcionalidad se validó correctamente, el sistema mostró las asignaciones entre los proyectos e historias de usuario del manager logueado.
------------	--	---

### 2.2.2 List (Hacking)

Entidad	Descripción	Resultado
Project	Verificar que un rol distinto a manager no puede listar proyectos.	La funcionalidad se validó correctamente, el sistema mostró un error 500 (uso no autorizado) por cada rol distinto a manager que quisiera listar proyectos.
User story	Verificar que un rol distinto a manager no puede listar las historias de usuario asociadas a un proyecto determinado.	La funcionalidad se validó correctamente, el sistema mostró un error 500 (uso no autorizado) por cada rol distinto a manager que quisiera listar las historias de usuario asociadas a un proyecto determinado.
Assignment	Verificar que un rol distinto al manager no puede listar las asignaciones entre proyectos e historias de usuario.	La funcionalidad se validó correctamente, el sistema mostró un error 500 (uso no autorizado) por cada rol distinto al manager que quisiera listar las asignaciones entre proyectos e historias de usuario.

### 2.2.2 List All

Entidad	Descripción	Resultado
---------	-------------	-----------

User story	Verificar que un manager puede listar sus historias de usuario creadas.	La funcionalidad se validó correctamente, el sistema mostró las historias de usuario creadas del manager logueado.
------------	---	--

### 2.2.3 Show

Entidad	Descripción	Resultado
Project	Verificar que un manager puede ver todos los detalles de un proyecto asociado a dicho manager.	La funcionalidad se validó correctamente, el sistema mostró todos los detalles de un proyecto seleccionado por el manager logueado .
User story	Verificar que un manager puede ver todos los detalles de una historia de usuario de dicho manager	La funcionalidad se validó correctamente, el sistema mostró todos los detalles de una historia de usuario perteneciente al manager logueado.
Assignment	Verificar que un manager puede ver todos los detalles de una asignación entre un proyecto y una historia de usuario de dicho manager.	La funcionalidad se validó correctamente, el sistema mostró todos los detalles de una asignación entre un proyecto y una historia de usuario perteneciente al manager logueado.

### 2.2.4 Show (Hacking)

Entidad	Descripción	Resultado
Project	Verificar que un rol distinto al manager no puede ver todos los detalles de un proyecto de un manager. De igual forma, verificar que sólo sean visibles para el manager que haya creado el proyecto.	La funcionalidad se validó correctamente, el sistema mostró un error 500 (uso no autorizado) por cada rol distinto al manager que quisiera ver los detalles de un proyecto. También mostró un error 500 cuando

		se intentó acceder desde otro manager distinto.
User story	Verificar que un rol distinto al manager no puede ver todos los detalles de una historia de usuario de un manager. De igual forma, verificar que sólo sean visibles para el manager que haya creado la historia de usuario.	La funcionalidad se validó correctamente, el sistema mostró un error 500 (uso no autorizado) por cada rol distinto al manager que quisiera ver los detalles de una historia de usuario. También mostró un error 500 cuando se intentó acceder desde otro manager distinto.
Assignment	Verificar que un rol distinto al manager no puede ver todos los detalles de una asignación. De igual forma, verificar que sólo sean visibles para el manager que haya creado la asignación.	La funcionalidad se validó correctamente, el sistema mostró un error 500 (uso no autorizado) por cada rol distinto al manager que quisiera ver los detalles de una asignación. También mostró un error 500 cuando se intentó acceder desde otro manager distinto.

## 2.2.5 Create

Entidad	Descripción	Resultado
Project	Verificar que un manager puede crear proyectos de forma adecuada.	La funcionalidad se validó correctamente, el sistema permitió la creación de proyectos válidos (cumpliendo con todas las restricciones).
User story	Verificar que un manager puede crear historias de usuario de forma adecuada, ya sea desde un proyecto o de manera independiente.	La funcionalidad se validó correctamente, el sistema permitió la creación de historias de usuario válidas (cumpliendo con todas las restricciones) tanto desde un proyecto como de manera aislada.
Assignment	Verificar que un manager puede asignar historias de usuario a proyectos no	La funcionalidad se validó correctamente, el sistema permitió la asignación de

	publicados.	historias de usuario pertenecientes a un manager a proyectos no publicados por el mismo.
--	-------------	--

### 2.2.6 Create (Hacking)

Entidad	Descripción	Resultado
Project	Verificar que un rol distinto al manager no puede crear proyectos.	La funcionalidad se validó correctamente, el sistema mostró un error 500 (uso no autorizado) por cada rol distinto al manager que quisiera crear un proyecto.
User story	Verificar que un rol distinto al manager no puede crear historias de usuario.	La funcionalidad se validó correctamente, el sistema mostró un error 500 (uso no autorizado) por cada rol distinto al manager que quisiera crear una historia de usuario.
Assignment	Verificar que un rol distinto al manager no puede crear una asignación. De igual manera, verificar que un manager no pueda crear una asignación a otro manager.	En primera instancia, se pudo realizar una asignación a un manager desde otro distinto (post hacking). Este error fue interceptado y corregido, de manera que ahora el sistema lanza un error 500 (uso no autorizado) siempre que se quiera crear una asignación desde un rol o usuario incorrectos.

### 2.2.7 Publish

Entidad	Descripción	Resultado
Project	Verificar que un manager puede publicar proyectos de	La funcionalidad se validó correctamente, el sistema

	forma adecuada.	permitió la publicación de proyectos válidos (cumpliendo con todas las restricciones)
User story	Verificar que un manager puede publicar historias de usuario de forma adecuada.	La funcionalidad se validó correctamente, el sistema permitió la publicación de historias de usuario válidas (cumpliendo con todas las restricciones) tanto desde un proyecto como de manera aislada.

### 2.2.8 Publish (Hacking)

Entidad	Descripción	Resultado
Project	<p>Verificar que un rol distinto al manager no puede publicar proyectos de forma adecuada.</p> <p>Verificar que un manager no puede publicar proyectos a otro manager.</p>	<p>La funcionalidad se validó correctamente, el sistema mostró un error 500 (uso no autorizado) por cada rol distinto al manager que quisiera publicar un proyecto.</p> <p>Por otro lado, se comprobó cómo era posible publicar proyectos de un manager desde otro distinto. El error fue corregido satisfactoriamente.</p>
User story	<p>Verificar que un rol distinto a manager no puede publicar historias de usuario de forma adecuada, ya sea desde un proyecto o de manera independiente.</p> <p>Verificar que un manager no puede publicar historias de usuario pertenecientes a otro manager.</p>	<p>La funcionalidad se validó correctamente, el sistema mostró un error 500 (uso no autorizado) por cada rol distinto al manager que quisiera publicar una historia de usuario tanto desde un proyecto como de manera aislada.</p> <p>También se verificó que el sistema lanzaba un error 500 cuando se pretendía publicar una historia de usuario de un manager</p>



		desde otro distinto.
--	--	----------------------

### 2.2.9 Update

Entidad	Descripción	Resultado
Project	Verificar que un manager puede actualizar proyectos no publicados de forma adecuada.	La funcionalidad se validó correctamente, el sistema permitió la actualización de proyectos válidos no publicados (cumpliendo con todas las restricciones).
User story	Verificar que un manager puede actualizar historias de usuario no publicadas de forma adecuada.	La funcionalidad se validó correctamente, el sistema permitió la actualización de historias de usuario válidas no publicadas (cumpliendo con todas las restricciones).

### 2.2.10 Update (Hacking)

Project	<p>Verificar que un rol distinto al manager no puede actualizar proyectos de forma adecuada.</p> <p>Verificar que un manager no puede actualizar proyectos a otro manager.</p>	<p>La funcionalidad se validó correctamente, el sistema mostró un error 500 (uso no autorizado) por cada rol distinto al manager que quisiera actualizar un proyecto.</p> <p>Por otro lado, se comprobó cómo era posible actualizar proyectos de un manager desde otro distinto. El error fue corregido satisfactoriamente</p>
User story	<p>Verificar que un rol distinto a manager no puede actualizar historias de usuario de forma adecuada, ya sea desde un proyecto o de manera independiente.</p> <p>Verificar que un manager no</p>	<p>La funcionalidad se validó correctamente, el sistema mostró un error 500 (uso no autorizado) por cada rol distinto al manager que quisiera actualizar una historia de usuario tanto desde un proyecto como de</p>

	puede actualizar historias de usuario pertenecientes a otro manager	manera aislada.  También se verificó que el sistema lanzaba un error 500 cuando se pretendía actualizar una historia de usuario de un manager desde otro distinto.
--	---	--

## 2.2.11 Delete

Entidad	Descripción	Resultado
Project	Verificar que un manager pueda eliminar proyectos no publicados de forma adecuada.	La funcionalidad se validó correctamente, el sistema permitió la eliminación de proyectos no publicados.
User story	Verificar que un manager pueda eliminar historias de usuario no publicadas de forma adecuada, ya sea desde un proyecto o de manera independiente.	La funcionalidad se validó correctamente, el sistema permitió la eliminación de historias de usuario no publicadas tanto desde un proyecto como de manera aislada.
Assignment	Verificar que un manager pueda eliminar historias de usuario de un proyecto no publicado.	La funcionalidad se validó correctamente, el sistema permitió la eliminación de historias de usuario pertenecientes a un proyecto no publicado.

## 2.2.12 Delete (Hacking)

Entidad	Descripción	Resultado
Project	Verificar que un rol distinto al manager no puede eliminar proyectos.  Verificar que un manager no	La funcionalidad se validó correctamente, el sistema mostró un error 500 (uso no autorizado) por cada rol distinto al manager que

	<p>puede eliminar proyectos a otro manager.</p>	<p>quisiera eliminar un proyecto.</p> <p>También se verificó que el sistema lanzaba un error 500 cuando se pretendía eliminar un proyecto de un manager desde otro distinto.</p>
User story	<p>Verificar que un rol distinto al manager no puede eliminar historias de usuario.</p> <p>Verificar que un manager no puede eliminar historias de usuario a otro manager.</p>	<p>La funcionalidad se validó correctamente, el sistema mostró un error 500 (uso no autorizado) por cada rol distinto al manager que quisiera eliminar una historia de usuario.</p> <p>También se verificó que el sistema lanzaba un error 500 cuando se pretendía eliminar una historia de usuario de un manager desde otro distinto.</p>
Assignment	<p>Verificar que un rol distinto al manager no pueda eliminar una asignación.</p> <p>Verificar que un manager no puede eliminar asignaciones a otro manager.</p>	<p>La funcionalidad se validó correctamente, el sistema mostró un error 500 (uso no autorizado) por cada rol distinto al manager que quisiera eliminar una historia de usuario de un proyecto.</p> <p>También se verificó que el sistema lanzaba un error 500 cuando se pretendía eliminar una asignación de un manager desde otro distinto.</p>

## 3. Testeo de rendimiento

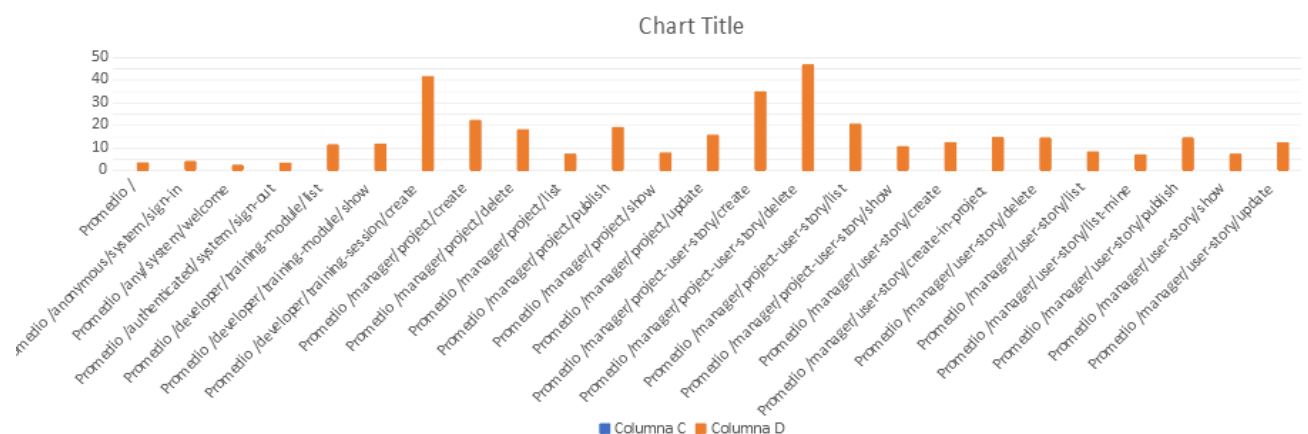
### 3.1 Metodología

Para evaluar el rendimiento del sistema, se realizaron pruebas de tiempo de respuesta sobre los tests que fueron grabados. Para ello se utilizó el launcher realplayer, con el que se ejecutaron todas las peticiones que fueron grabadas en el recorder.

Fueron analizados los tiempos de respuesta comparando entre las entidades sin índices, y por otro lado con la inclusión de un índice sobre la entidad Project para consultas en las que se utilizara el parámetro code.

## 3.2 Resultados

### 3.2.1 Entidades sin índice



En el gráfico el eje Y mide el tiempo en milisegundos que toma, en promedio, cada tipo de petición del eje X.

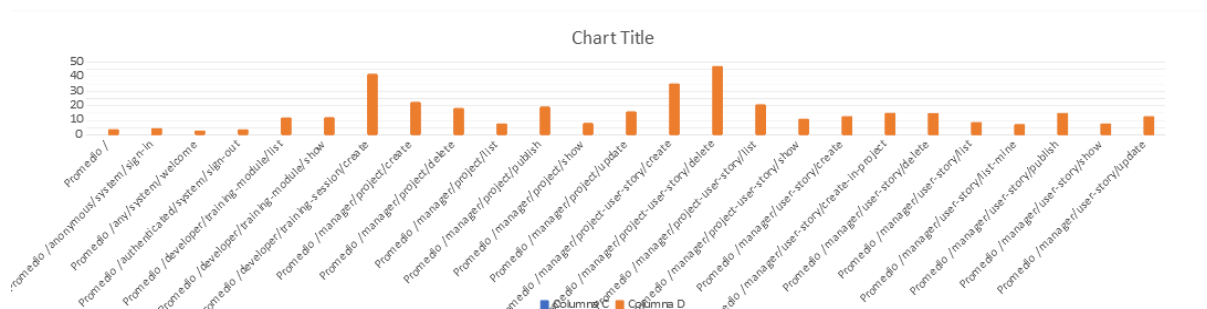
Como podemos comprobar, las dos peticiones que mayor tiempo toman son la creación de un training session (los tests del rol de developer no corresponden al estudiante 1, pero en el momento de hacer el realplayer se introdujeron por error) y el borrado de una asignación entre un proyecto y una entidad de usuario.

<u>Before</u>				
		<u>Interval(ms)</u>	6,4715336	7,75935577
Media	7,1154447	<u>Interval(s)</u>	0,0064715	0,00775936
Error típico	0,3281077			
Mediana	3,978			
Moda	1,8815			
Desviación	10,048885			
Varianza de	100,9801			
<u>Curtosis</u>	31,862607			
Coeficiente	4,6930213			
Rango	107,7603			
Mínimo	1,4322			
Máximo	109,1925			
Suma	6674,2871			
Cuenta	938			
Nivel de cor	0,6439111			

De estos datos, cabe destacar el intervalo de confianza al 95% (6.4715335738310, 7.7593557651881) en milisegundos. También vemos que la media es de 7,1154446695096 milisegundos.

### 3.2.2 Entidades con índice

Tras haber visto los resultados de las entidades sin índice, ahora visualizamos los obtenidos tras haber incluido un índice en la entidad Project.



<u>After</u>					
Media	7,2718007		<u>Interval (ms)</u>	6,5790297	7,9645718
Error típico	0,3530031		<u>Interval (s)</u>	0,006579	0,0079646
Mediana	3,9924				
Moda	2,3716				
Desviación es	10,794047				
Varianza de la	116,51145				
<u>Curtosis</u>	36,004045				
Coeficiente de	5,0465517				
Rango	121,4251				
Mínimo	1,3699				
Máximo	122,795				
Suma	6799,1337				
Cuenta	935				
Nivel de confi	0,6927711				

Aquí tenemos los resultados de este segundo análisis tras incluir un sólo índice. No obstante, para comparar los dos rendimientos de forma más objetiva nos apoyamos en la prueba Z.

### 3.2.3 Z-Test

Prueba <u>z</u> para medias de dos muestras		
	<u>before</u>	<u>after</u>
Media	7,1154447	7,2718007
Varianza (conocida)	100,9801	116,51145
Observaciones	938	935
Diferencia hipotética de las medias	0	
<u>z</u>	-0,3244308	
P( $Z \leq z$ ) una cola	0,372806	
Valor crítico de <u>z</u> (una cola)	1,6448536	
Valor crítico de <u>z</u> (dos colas)	0,7456119	
Valor crítico de <u>z</u> (dos colas)	1,959964	









Podemos ver que el valor de P en este caso está contenido en el rango [0.00, 0.37], es decir, es una diferencia superior que 1-0.95. Esto sugiere que la introducción de un sólo índice no ha resultado en ninguna mejora significativa. Los tiempos de prueba son diferentes, pero globalmente son los mismos.

## 4. Cobertura

Por último, vamos a analizar las coberturas obtenidas para las entidades Project, User Story y Project User Story.

**ACLARACIÓN:** Esta cobertura fue obtenida tras ejecutar el replayer desde la rama D04/Task035/Produce-testing-report-s1-features. No obstante, tras hacer pull de la rama master antes de subir mis cambios, debió de haber alguna modificación en alguno de los ficheros csv que provocan que muchos tests den fallos, concretamente los correspondientes a las historias de usuario. No obstante, la traza de los tests grabados es idéntica a la que produjeron estos resultados.

### 4.1 Cobertura de Project

▼	acme.features.manager.project		89,4 %	1.100	131	1.231
>	ManagerProjectDeleteService.java		64,5 %	107	59	166
>	ManagerProjectCreateService.java		88,8 %	214	27	241
>	ManagerProjectPublishService.java		94,5 %	291	17	308
>	ManagerProjectUpdateService.java		93,7 %	251	17	268
>	ManagerProjectShowService.java		95,0 %	134	7	141
>	ManagerProjectListService.java		94,4 %	68	4	72
>	ManagerProjectController.java		100,0 %	35	0	35

Como se puede apreciar, la cobertura de los servicios relacionados con la entidad Project es bastante buena, debido a que la mayoría de clases superan el 93% de cobertura. No obstante, en el servicio de creación no se ha llegado al 90% porque se obtuvo el siguiente error: "ERROR Got null response for request 11b3073d-4834-e03d-746c-45c98982e08e". Aún así, durante la grabación de la prueba de creación de proyecto, se pudieron crear correctamente.

Por otro lado, en todos los servicios de borrado, la cobertura es baja porque en ninguno de ellos se han conseguido hacer que se ejecuten las líneas que aparecen en el método unbind, ya que aunque se introdujeran datos incorrectos antes del borrado, la entidad se eliminaba perfectamente sin volver a mostrar el formulario.

### 4.2 Cobertura de User Story

acme.features.manager.userstory	89,9 %	1.385	156	1.541
> ManagerUserStoryDeleteService.java	63,0 %	104	61	165
> ManagerUserStoryPublishService.java	92,2 %	226	19	245
> ManagerUserStoryUpdateService.java	91,7 %	210	19	229
> ManagerUserStoryCreateInProjectService.java	94,0 %	281	18	299
> ManagerUserStoryCreateService.java	92,8 %	205	16	221
> ManagerUserStoryShowService.java	90,7 %	107	11	118
> ManagerUserStoryListService.java	94,4 %	135	8	143
> ManagerUserStoryListMineService.java	94,6 %	70	4	74
> ManagerUserStoryController.java	100,0 %	47	0	47

En este caso, el análisis es idéntico al de la cobertura de los proyectos. En líneas generales los resultados son buenos, pero en el servicio de borrado de historias de usuario no se pudo conseguir una cobertura mayor ante la imposibilidad de que se volviera a renderizar el formulario de la historia de usuario tras darle al botón de borrar.

### 4.3 Cobertura de Project User Story

acme.features.manager.projectuserstory	92,5 %	653	53	706
> ManagerProjectUserStoryCreateService.java	91,3 %	232	22	254
> ManagerProjectUserStoryDeleteService.java	90,3 %	176	19	195
> ManagerProjectUserStoryListService.java	91,7 %	88	8	96
> ManagerProjectUserStoryShowService.java	97,1 %	133	4	137
> ManagerProjectUserStoryController.java	100,0 %	24	0	24

Como vemos aquí, la cobertura es bastante aceptable. En este caso, el servicio de borrado sí tiene una buena cobertura porque sí se han ejecutado las líneas del método unbind. Esto es debido a que se intentaron quitar historias de usuario de proyectos que estaban publicados, lo que resultó en error y, por tanto, se ejecutaron las líneas del método unbind.

## 5. Conclusión

Tras realizar pruebas para cada funcionalidad, se encontraron pocos errores, gracias en parte a la utilidad de las sesiones de seguimiento del proyecto previas a la realización de las pruebas formales. No obstante, todo el proceso de ejecución y análisis de las pruebas ha subrayado la importancia fundamental de realizar pruebas exhaustivas en cualquier proyecto de software.



## 6. Bibliografía

Intencionadamente en blanco.