



Escuela Técnica Superior de  
**Ingeniería Informática**

## Testing report Do4

Group C2.020 | Diseño y Pruebas II | 21/10/2024

Fecha	Versión	Autor
16-05-2024	1.0	Gabriel Vacaro Goytia
06-07-2024	2.0 Se ha mejorado el coverage y explicado porque no alcanzan el 100% algunos elementos, así como algunas correcciones en la conclusión del rendimiento y un nuevo contraste de hipótesis relacionado a los cambios de esta segunda entrega, se rehicieron todas las pruebas debido a la nueva actualización del framework para evitar phantom requests y para cubrir el cambio de tipo en el atributo cost de Project (de Integer a Money) y otros cambios menores.	Gabriel Vacaro Goytia
21-10-2024	3.0 Se han realizado cambios en el atributo cost de Project nuevamente, puesto que el cliente deseaba que se midiera en horas, no en cantidades de dinero, adicionalmente se han rehecho todas las pruebas para ajustarse a los nuevos datos expandidos de la base de datos y los cambios menores.	Gabriel Vacaro Goytia

### **Miembros:**

- Gabriel Vacaro Goytia (gabvacgoy@alum.us.es)

**Repositorio de Github:** <https://github.com/DP2-C1-020/Acme-SF-Do4>

## **Contenido**

Resumen ejecutivo .....	3
Introducción .....	4
Pruebas funcionales: .....	5
Análisis de desempeño: .....	15
Conclusiones .....	22
Bibliografía.....	23

## Resumen ejecutivo

El propósito de este informe es ofrecer una descripción detallada de los diferentes procedimientos seguidos que deben considerarse en el ámbito del testing formal del proyecto Acme-SF, desde la generación de suficientes datos de ejemplo, hasta la meticulosa comprobación de la correcta implementación de los requisitos funcionales y las distintas herramientas estadísticas para comparar los tiempos de ejecución entre peticiones, las diferencias del rendimiento entre dos ordenadores diferentes (antes de la mejora de índices), y por último la comparación de un mismo ordenador (antes y después de la mejora de la base de datos mediante índices).

En resumen, se ha utilizado un enfoque diligente para abordar y solucionar los errores encontrados durante el proceso de testeo con la finalidad de asegurar un producto de alto nivel con el que se satisfagan las expectativas del cliente y se han generado documentos que lo avalan.

## Introducción

En este tipo de informe, se pretende realizar una explicación de como se ha realizado el testing funcional y el análisis de desempeño, con el objetivo de entregar un proyecto de alto nivel con garantías. Para lograrlo, se lleva a cabo una serie de pruebas en relación con los requisitos funcionales 6 y 7.

Este escrito se ha organizado según el documento de anexo proporcionado en enseñanza virtual, en primer lugar, con una portada con las credenciales del autor del reporte, una tabla de versiones en la que se especifican las modificaciones realizadas en este documento clasificadas por número y con sus fechas respectivas; seguidamente un resumen ejecutivo el cual pretende poner en contexto al lector sobre el contenido del documento, una introducción al documento donde se describe el contenido de forma sucinta y se trata la estructura del documento en este último párrafo introductorio.

El trabajo presenta dos apartados clave: uno sobre pruebas funcionales y otro sobre pruebas de rendimiento. En el primero, se detallan los casos de prueba implementados, organizados por característica, con evaluaciones sobre su efectividad para detectar errores. En el segundo, se proporcionan gráficos pertinentes y un intervalo de confianza del 95% para el tiempo de respuesta del proyecto ante las solicitudes funcionales y la comparación pertinente. Estos capítulos constituyen una evaluación exhaustiva de la calidad y eficiencia del proyecto en desarrollo, por último, una conclusión sobre el proceso de testing formal y que ha aportado al proyecto y la bibliografía utilizada.

## Contenido

### Pruebas funcionales:

Para cada característica se han realizado tanto pruebas positivas y negativas (X.safe) como intentos de hacking (Y.hack), para la realización de estas pruebas, se ha seguido las recomendaciones dictadas en el documento “L04-S01 – Formal testing”:

Para no caer en redundancias, listaré las pruebas realizadas según el tipo de operación, pues esto será lo que determinará el modo de actuar de la prueba:

### OPERACIONES "LIST-MINE"

#### *Pruebas Seguras:*

- Listar todos los proyectos o historias de usuario de varios gerentes.
  - Se probaron tanto mánager1 como mánager2

#### *Pruebas de Hacking:*

- Intentar acceder con todos los roles incorrectos o sin rol a las siguientes URLs:
  - <http://localhost:8082/Acme-SF-D04/manager/project/list-mine>
  - <http://localhost:8082/Acme-SF-D04/manager/user-story/list-mine>

### OPERACIONES "LIST-FOR-PROJECT"

#### *Pruebas Seguras:*

- Listar las historias de usuario en 5 proyectos diferentes de cada mánager, cubriendo los casos en los que existen 1 historias de usuario, varias historias o ninguna, independientemente de si está o no publicado.
- Manager1:
  - Proyecto con id 59 y código VWX-2340 ( no publicado y con varias historias)
  - Proyecto con id 64 y código KLM-7890 ( no publicado y con varias historias)
  - Proyecto con id 61 y código BCD-8900 ( publicado y con 2 historias)
  - Proyecto con id 62, con código EFG-1230 ( publicado y con 1 historia)
  - Proyecto con id 63, con código HIJ-4560 ( no publicado y sin historias)
- Manager2:
  - Proyecto con id 86 y código AAJ-0123 ( no publicado y con varias historias)
  - Proyecto con id 87 y código AAK-0123 ( publicado y con varias historias)
  - Proyecto con id 88 y código AAL-0123 (no publicado y con 2 historias)
  - Proyecto con id 89 con código AAM-0123 ( publicado y con 1 historia)
  - Proyecto con id 93, con código AAP-0123 ( no publicado y sin historias)

*Pruebas de Hacking:*

- Intentar acceder con todos los roles incorrectos, sin rol, o con un manager diferente a historias de usuario de un proyecto en concreto (GET hacking)., se probaron las siguientes URLs:
  - <http://localhost:8082/Acme-SF-D04/manager/user-story/list-for-project?projectId=XX>

Donde XX representa la id del proyecto, se utilizaron los proyectos mencionados anteriormente en la etapa segura, en concreto los que tienen id: 59, 61, 86, 89, también se probaron id de proyectos inexistentes.

## OPERACIONES "SHOW"

*Pruebas Seguras:*

- Seleccionar varios proyectos o historias de usuario de una lista para ver sus detalles (10 proyectos distintos de cada manager y 10 historias de usuario distintas de cada manager).
  - Se probaron los proyectos con id 75-85 para manager 1 y los proyectos con id 86-96 para manager 2.
  - Se probaron las historias de usuario con id 515-525 para manager 1 y las historias de usuario con id 545-555 para manager 2.

*Pruebas de Hacking:*

- Intentar acceder a los detalles de un proyecto o historia de usuario (tanto publicados como no) con todos los roles incorrectos, con un gerente diferente o sin rol (GET hacking). Se probaron las siguientes URLs:
  - <http://localhost:8082/Acme-SF-D04/manager/project/show?id=XX>
  - <http://localhost:8082/Acme-SF-D04/manager/user-story/show?id=XXX>

Donde XX representa la id del proyecto y XXX representa la historia de usuario, también se probaron id inexistentes.

Se prueba a entrar en los proyectos con ids 87 (AAK-0123) y 104 (ABZ-0123) de manager 2 ; 62 (EFG-1230) y 65 (NOP-0120) de manager 1 entre otros.

Se prueba a entrar en las historias de usuario con id 593 y 594 de manager 1; 575 y 576 de manager 2 entre otros.

## OPERACIONES "CREATE"

*Pruebas Seguras:*

- Intentar crear un nuevo proyecto o historia de usuario.
- Verificar que el sistema rechace datos inválidos para cada atributo de los proyectos:
  - Código en diferentes formatos incorrectos (en minúsculas, al revés, demasiado largo), vacío o repetidos (de ambos manager).

- título demasiado largo (76 o más), vacío.
  - Abstracto demasiado largo (101 o más), vacío.
  - coste: negativo, texto, Money, más de 1.000.000.000.
  - Link: "null", link no bien formado, 251 caracteres o más.
- Verificar que el sistema acepte datos válidos para cada atributo de los proyectos:
  - Código en formatos válidos.
  - Título con emoticonos, letras de distintos alfabetos y límites en el tamaño (1 y 75 caracteres)
  - Abstracto con emoticonos, letras de distintos alfabetos y límites en el tamaño (1 y 100 caracteres)
  - Coste menor o igual a 1.000.000.000 y mayor o igual que 0.
  - Link: vacío y menor de 251 caracteres con formato link, con emoticonos y en distintos alfabetos.
- Verificar que el sistema acepte datos válidos para cada atributo de las historias de usuario:
  - Título con emoticonos, letras de distintos alfabetos y límites en el tamaño (1 y 75 caracteres)
  - Descripción con emoticonos, letras de distintos alfabetos y límites en el tamaño (1 y 100 caracteres)
  - Criterios de aceptación con emoticonos, letras de distintos alfabetos y límites en el tamaño (1 y 100 caracteres)
  - Coste estimado en horas menor o igual a 10.000.000 y mayor o igual que 0.
  - Prioridad no nula.
  - Link: vacío y menor de 251 caracteres con formato link, con emoticonos.
- Verificar que el sistema rechace datos inválidos para cada atributo de las historias de usuario:
  - Título demasiado largo(76 caracteres o más), vacío.
  - Abstracto demasiado largo(101 caracteres o más), vacío.
  - Criterio de aceptación demasiado largo ( 101 o más), vacío.
  - Coste estimado en horas negativo, texto, Money, más de 10.000.000.
  - Prioridad nula.
  - Link: "null", link no bien formado, 251 caracteres o más.

#### *Pruebas de Hacking:*

- Intentar acceder con todos los roles incorrectos o sin rol a la pantalla de creación de proyectos e historias de usuario (GET hacking)., se probaron las siguientes URLs:
  - <http://localhost:8082/Acme-SF-D04/manager/project/create>
  - <http://localhost:8082/Acme-SF-D04/manager/user-story/create>

## OPERACIONES "UPDATE"

### *Pruebas Seguras:*

- Actualizar un proyecto o historia de usuario existente siguiendo los mismos procedimientos (que rechace datos inválidos y acepte los válidos) que, en el apartado anterior de creación de proyectos e historias de usuario, además de actualizar proyectos con una, varias o ninguna historia de usuario.
  - Proyecto probado con varias historias: id 59 (VWX-2340)
  - Proyecto probado con una historia: id 80 (AAD-0123)
  - Proyecto probado sin historias: id 72 (GHI-2345)

### *Pruebas de Hacking:*

- Intentar acceder a la característica de actualización desde todos los roles incorrectos o sin rol.
- Usar el inspeccionador de elementos del navegador para intentar actualizar proyectos e historias de usuario ya publicados (POST hacking):
  - Desde un usuario no autenticado o con rol incorrecto a un proyecto publicado desde la vista any (todos probados).
  - Desde mánager 1 actualizar proyectos publicados tanto de mánager 2 como suyos propios (desde la vista de any), y viceversa (desde mánager 2 a mánager 1).
  - Desde la vista de mis proyectos con el mánager adecuado.
  - Desde la vista de mis historias de usuario con el mánager adecuado.
  - URL utilizada:
    - <http://localhost:8082/Acme-SF-D04/manager/project/update>
    - <http://localhost:8082/Acme-SF-D04/manager/user-story/update>

## OPERACIONES "DELETE"

### *Pruebas Seguras:*

- Intentar eliminar proyectos, considerando restricciones como elementos dependientes.
  - Proyectos 59 (VWX-2340), 60 (YZA-5670) no pueden eliminarse debido a contratos y training modules dependientes.
  - Proyectos 63 (HIJ-4560), 64 (KLM-7890), 65 (NOP-0120), 66 (QRS-3450), 86 (AAJ-0123) entre otros no pueden eliminarse debido a contratos dependientes.
- Eliminar correctamente un proyecto o historia de usuario cuando no haya elementos dependientes.
  - Proyectos con distinto número de historias de usuario (entre otros):
    - Id 94 (AAQ-0123) varias historias
    - Id 90 (AAN-0123) proyecto con 2 historias de usuario
    - Id 91 (ABA-0123) 1 historia
    - Id 93 (AAP-0123) sin historias
  - Historias de usuario con distinto número de proyectos vinculados (entre otras):
    - Id 522 varios proyectos vinculados
    - Id 517, 1 proyecto vinculado



- Id 519, 0 proyectos vinculados

#### *Pruebas de Hacking:*

- Intentar acceder al borrado de un proyecto con un usuario no autenticado o con un rol incorrecto mediante URLs (POST hacking)
- Intentar borrar proyectos publicados con el mánager adecuado (tanto desde la vista mis proyectos como las de any)
- Intentar borrar proyectos publicados de otro mánager desde la vista any.
- Intentar borrar historias de usuario publicadas.
  - URLs probadas:
    - <http://localhost:8082/Acme-SF-D04/manager/project/delete>
    - <http://localhost:8082/Acme-SF-D04/manager/user-story/delete>

## OPERACIONES "PUBLISH"

#### *Pruebas Seguras:*

- Publicar un proyecto o historia de usuario existente siguiendo los mismos procedimientos (que rechace datos inválidos y acepte los válidos) que, en el apartado anterior de creación y actualización de proyectos e historias de usuario, además de asegurarse de que no se pueden publicar proyectos sin historias de usuario o con errores fatales.
- Se utiliza el proyecto con id 77 (AAA-0123), debido a que tiene varias historias de usuario, de las cuales solo 1 no está publicada, esto nos ayudará a probar la validación.
- Se utiliza el proyecto con id 83 (AAG-0123), debido a que no tiene historias de usuario.
- Se utiliza el proyecto con id 65 (NOP-0120), debido a que tiene una historia y esta publicada, por tanto, este será el proyecto usado para testear validación de los atributos del proyecto.
- Se crearon proyectos para publicar con formatos válidos.

#### *Pruebas de Hacking:*

- Intentar republicar proyectos e historias de usuario ya publicadas (POST hacking) haciendo uso del inspeccionador de elementos del navegador para cambiar parámetros:
  - Desde un usuario no autenticado.
  - Desde cualquier rol incorrecto.
  - Desde manager1 republicar proyectos tanto de manager2 como suyos propios (desde la vista de any) y viceversa (desde manager2 a manager1).
  - Desde la vista de mis proyectos con el mánager adecuado.
  - Desde la vista de mis historias de usuario con el mánager adecuado
  - URL utilizada:
    - <http://localhost:8082/Acme-SF-D04/manager/project/publish>
    - <http://localhost:8082/Acme-SF-D04/manager/user-story/publish>

## OPERACIONES EN LA TABLA INTERMEDIA "PROJECTUSERSTORY"

### Pruebas Seguras:

- Vincular y desvincular proyectos a historias de usuario.
- Verificar que se rechacen asociaciones o desvinculaciones inválidas (sin proyecto elegido).
- Asegurar que las asociaciones y desvinculaciones se manejen correctamente según el estado de publicación, si un proyecto está publicado no se podrán vincular o desvincular más historias de usuario (al resto de proyectos si será posible vincular/desvincular historias), en este caso son los proyectos con id:
  - 61 (BCD-8900), 62 (EFG-1230) y 75 (MNA-0123) para manager 1
  - 87 (AAK-0123) y 89 (AAM-0123) para manager 2.

### Pruebas de Hacking:

- Intentar acceder a las páginas de vinculación y desvinculación con un rol incorrecto o con un gerente diferente al propietario de la historia de usuario (GET hacking).
- URLs accedidas:
  - <http://localhost:8082/Acme-SF-D04/manager/project-user-story/create?userId=XXX>
  - <http://localhost:8082/Acme-SF-D04/manager/project-user-story/delete?userId=XXX>

Donde XXX significa el id de la historia de usuario, también se probaron ids inexistentes.

Por último, véase aquí las pruebas implementadas y una categorización sobre su efectividad en esta última fase del proyecto y el coverage de cada característica (explicados en profundidad tras la tabla resumen).

Ruta de la feature	X.safe	Y.hack	Efectividad	Coverage
/manager/Dashboard/..-	Show	Show	1.0 Baja, No se encontraron fallos ni vulnerabilidades 2.0 Baja, no se encontraron fallos ni vulnerabilidades 3.0 Baja, no se encontraron fallos ni vulnerabilidades	98.4%
/manager/Project/...	List-mine Show Créate Update Publish Delete	List-mine Show Créate Update Publish Delete	1.0 Muy alta, se depuraron algunas líneas de impresión por pantalla pensadas para el debug que no deberían estar y un bug que, bajo ciertas	92.6%

			<p>condiciones provocaba que al borrar un proyecto no se borrarán algunas historias de usuario.</p> <p>2.0 Alta, Solución de errores al meter una cantidad muy grande de cualquier divisa.</p> <p>3.0 Alta, se descubrió que el atributo fatalErrors permitía el valor null.</p>	
/manager/Project-user-story/...	Link Unlink	Link Unlink	<p>1.0 Alta, encontrado bug que bajo ciertos datos de prueba las relaciones en las que el proyecto era nulo ocasionaban un panic.</p> <p>2.0 Alta, se refactorizó el código de validaciones debido a restricciones innecesarias</p> <p>3.0 Baja, no se detectaron nuevos errores</p>	92.2%
/manager/User-stories/...	List-mine List-for-project Show Créate Update Publish Delete	List-mine List-for-project Show Créate Update Publish Delete	<p>1.0 Media, bug menor encontrado que hacía comprobaciones redundantes al borrar una historia de usuario bajo ciertas condiciones.</p> <p>2.0 Baja, no se detectaron nuevos errores.</p> <p>3.0 Baja, no se detectaron nuevos errores</p>	87.4%

## COVERAGE EN LOS REQUISITOS 6 Y 7

Solo se estudiarán en profundidad los requisitos 6 y 7 tal como se pide, el resto de las características solo serán mencionadas.

Recubrimiento en los servicios relacionados con la entidad Project (requisito 6):

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
acme.features.manager.project	92,6 %	1.038	83	1.121
> ManagerProjectController.java	100,0 %	36	0	36
> ManagerProjectShowService.java	96,2 %	101	4	105
> ManagerProjectListMineService.java	94,9 %	112	6	118
> ManagerProjectDeleteService.java	92,2 %	226	19	245
> ManagerProjectPublishService.java	92,1 %	223	19	242
> ManagerProjectUpdateService.java	91,2 %	197	19	216
> ManagerProjectCreateService.java	89,9 %	143	16	159

Como se puede observar en la imagen la cobertura es de un 92.6%, teniendo en cuenta que hay líneas que no se analizan tales como las cabeceras de funciones, la internacionalización de los booleanos en el unbind (al haber realizado las pruebas solo en inglés y por tanto el endpoint ser “=en”, solo vemos reflejado una de las dos posibilidades):

```
if (object.isDraftMode()) {
    final Locale local = super.getRequest().getLocale();

    dataset.put("draftMode", local.equals(Locale.ENGLISH) ? "Yes" : "Sí");
} else
    dataset.put("draftMode", "No");
```

Tampoco se comprueban completamente las líneas “assert object != null” y por ultimo expresiones booleanas/lógicas compuestas donde explorar todas las opciones no siempre es posible si la aplicación funciona como es debido.

En referencia a este último aspecto, la posibilidad de línea no comprobada completamente existe, por ejemplo, en el caso del servicio de actualización de proyectos:

```
42 manager = project == null ? null : project.getManager();
43 status = project != null && project.isDraftMode() && super.getRequest().getPrincipal().hasRole(manager);
44
```

Las líneas 42 y 43 no podrán ser corroboradas como revisadas en todos sus aspectos, pues si todo va según lo esperado, en ambas líneas el proyecto nunca será nulo (y por tanto no se comprobará completamente, lo que no implica que este mal) y por lo tanto el mánager tampoco, pues si así fuera estaríamos incurriendo en un fallo de lógica, pues si no se es mánager ni siquiera se podría acceder a la actualización de proyectos.

Recubrimiento en los servicios relacionados con la entidad User Story (requisito 7):

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
acme.features.manager.userStory	87,4 %	895	129	1.024
> ManagerUserStoryController.java	100,0 %	42	0	42
> ManagerUserStoryShowService.java	96,5 %	111	4	115
> ManagerUserStoryListForProjectService.java	95,6 %	108	5	113
> ManagerUserStoryListMineService.java	93,6 %	73	5	78
> ManagerUserStoryPublishService.java	89,4 %	160	19	179
> ManagerUserStoryUpdateService.java	89,2 %	157	19	176
> ManagerUserStoryCreateService.java	89,0 %	130	16	146
> ManagerUserStoryDeleteService.java	65,1 %	114	61	175

Como se puede observar en la imagen la cobertura es de un 87.4%, visiblemente peor que en Project, teniendo en cuenta que, al igual que en Project, hay líneas que no se analizan

tales como las cabeceras de funciones, la internacionalización de los booleanos en el unbind (al haber realizado las pruebas solo en inglés y por tanto el endpoint ser “=en”, solo vemos reflejado una de las dos posibilidades)

```

51     if (object.isPublished()) {
52         final Locale local = super.getRequest().getLocale();
53
54         dataset.put("published", local.equals(Locale.ENGLISH) ? "Yes" : "Sí");
55     } else
56         dataset.put("published", "No");

```

Tampoco se evalúan completamete `assert object != null` y expresiones booleanas/lógicas compuestas donde explorar todas las opciones no siempre es posible si la aplicación funciona como es debido (como se explicó anteriormente).

La diferencia reside en el servicio de borrado, el cual tiene un 65.1%, el método `unbind` se usa en `ManagerProjectDeleteService` pero no en `ManagerUserStoryDeleteService` debido a la complejidad y múltiples relaciones de un `Project`. La eliminación de un `Project` requiere compilar y enviar datos de varias entidades relacionadas, actualizando así la interfaz de usuario de manera integral. En cambio, una `UserStory` es un componente más simple dentro de un `Project`, con menos relaciones y menor necesidad de datos complejos post-eliminación, lo que hace innecesario el uso de `unbind` en su eliminación.

Recubrimiento en los servicios relacionados relación `ProjectUser Story` (relación entre los requisitos 6 y 7):

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
acme.features.manager.projectUserStory	92,2 %	376	32	408
> ManagerProjectUserStoryController.java	100,0 %	14	0	14
> ManagerProjectUserStoryDeleteService.java	91,9 %	182	16	198
> ManagerProjectUserStoryCreateService.java	91,8 %	180	16	196

Como se puede observar en la imagen la cobertura es de un 92.2%, teniendo en cuenta que hay líneas que no se analizan tales como las cabeceras de funciones, `assert object != null` y expresiones booleanas/lógicas compuestas donde explorar todas las opciones no siempre es posible si la aplicación funciona como es debido.

Aquí podemos ver otro caso, donde si bien se han comprobado los casos donde no hay error con el `projectId`, no podemos comprobar el caso donde si lo haya, porque no es algo posible de hacer desde el cliente (necesitaríamos generar desde nuestra base de datos un proyecto con una id inválida), y al no comprobarse, la contraparte a la condición se pondría como no completada, la razón de que siga existiendo ese `if` es para tener otra capa de seguridad adicional:

```

72
73     if (!super.getBuffer().getErrors().hasErrors("projectId"))
74         super.state(object.getProject() != null, "projectId", "manager.project-user-story.form.error.no-proje
75     }





```

Recubrimiento en los servicios relacionados con el dashboard:

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
acme.features.manager.managerDashboard	98,4 %	250	4	254
> ManagerDashboardController.java	100,0 %	9	0	9
> ManagerDashboardRepository.java	100,0 %	56	0	56
> ManagerDashboardShowService.java	97,9 %	185	4	189







100% coverage (en este caso solo hay un `assert!=null`, ninguna internalización ni condición que sabemos no va a comprobarse nunca en caso de que la aplicación funcione bien, el coverage es casi completo)

Recubrimiento en los servicios relacionados con los proyectos para cualquier usuario:

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
▼ acme.features.any.project	 94,4 %	152	9	161
> AnyProjectController.java	 100,0 %	14	0	14
> AnyProjectShowService.java	 94,7 %	90	5	95
> AnyProjectListService.java	 92,3 %	48	4	52

94.4% coverage, no llegando al 100% por los motivos anteriormente mencionados.

Recubrimiento de las entidades Project y userStory:

▼ acme.entities.project	 100,0 %	76	0	76
> Project.java	 100,0 %	59	0	59
> ProjectUserStory.java	 100,0 %	17	0	17
▼ acme.entities.userstory	 100,0 %	103	0	103
> Priority.java	 100,0 %	44	0	44
> UserStory.java	 100,0 %	59	0	59

100% de coverage en todos los casos.

## Análisis de desempeño:

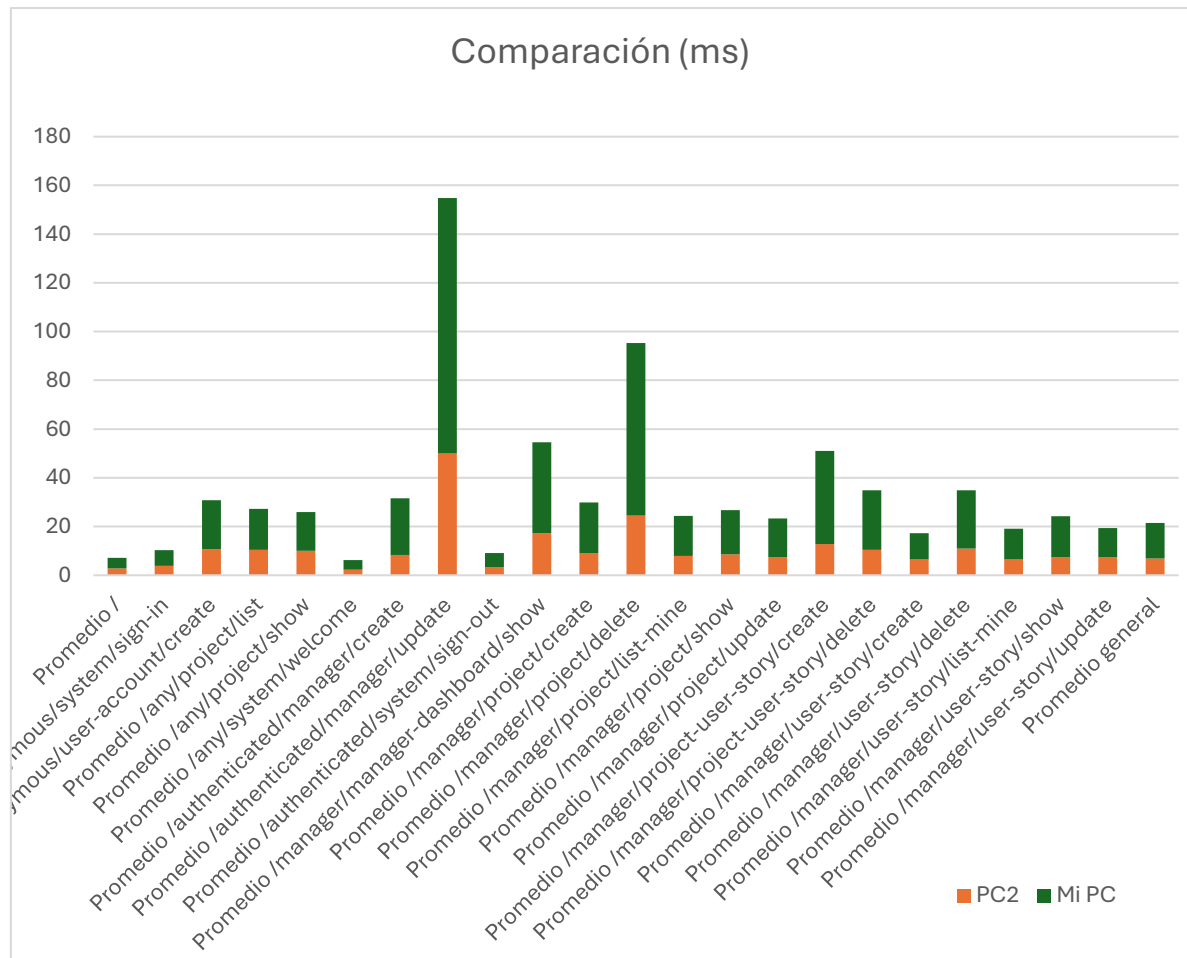
En este apartado veremos las estadísticas propias de la aplicación en su version 1.0 del entregable, para tener una referencia de los valores de respuesta de la aplicación, y después veremos los propios de la aplicación final en esta version 3.0 del entregable, y por último comprobaremos que los dos contrastes de hipótesis coinciden:

- 1) Version 1.0 Sin optimizar índices en dos ordenadores distintos:

STATISTICS-PC2					time-pc2	time-myPc	STATISTICS-MYPC				
Media	7,032846065	Interval(ms)	6,205195166	7,860496964	4,3535	7,5932	Media	15,05672832	Interval(ms)	12,81553318	17,29792346
Error típico	0,421209404	Interval(s)	0,006205195	0,007860497	3,052101	5,1372	Error típico	1,140622545	Interval(s)	0,012815533	0,017297923
Mediana	5,677401				3,8435	4,8268	Mediana	9,55035			
Moda	5,6025				3,232201	5,7233	Moda	3,7192			
Desviación estándar	9,218617915				2,650201	4,5814	Desviación estándar	25,09369599			
Varianza de la muestra	84,98291626				4,6674	5,0114	Varianza de la muestra	629,6935786			
Curtosis	150,9159598				2,9709	4,2208	Curtosis	60,79906946			
Coefficiente de asimetría	10,23075059				3,2419	5,1148	Coefficiente de asimetría	6,815479595			
1 Rango	155,122699				3,323801	6,1338	1 Rango	291,9664			
2 Mínimo	1,6883				16,823899	6,5744	2 Mínimo	2,6699			
3 Máximo	156,810999				3,345701	4,2615	3 Máximo	294,6363			
4 Suma	3368,733265				2,3352	3,9968	4 Suma	7287,456506			
5 Cuenta	479				3,1075	4,1644	5 Cuenta	484			
6 Nivel de confianza(95,0%)	0,827650899				3,4948	7,0456	6 Nivel de confianza(95,0%)	2,241195142			
					3,068899	3,9241					

Ilustración 1 Tabla comparativa entre el pc2 y mi pc

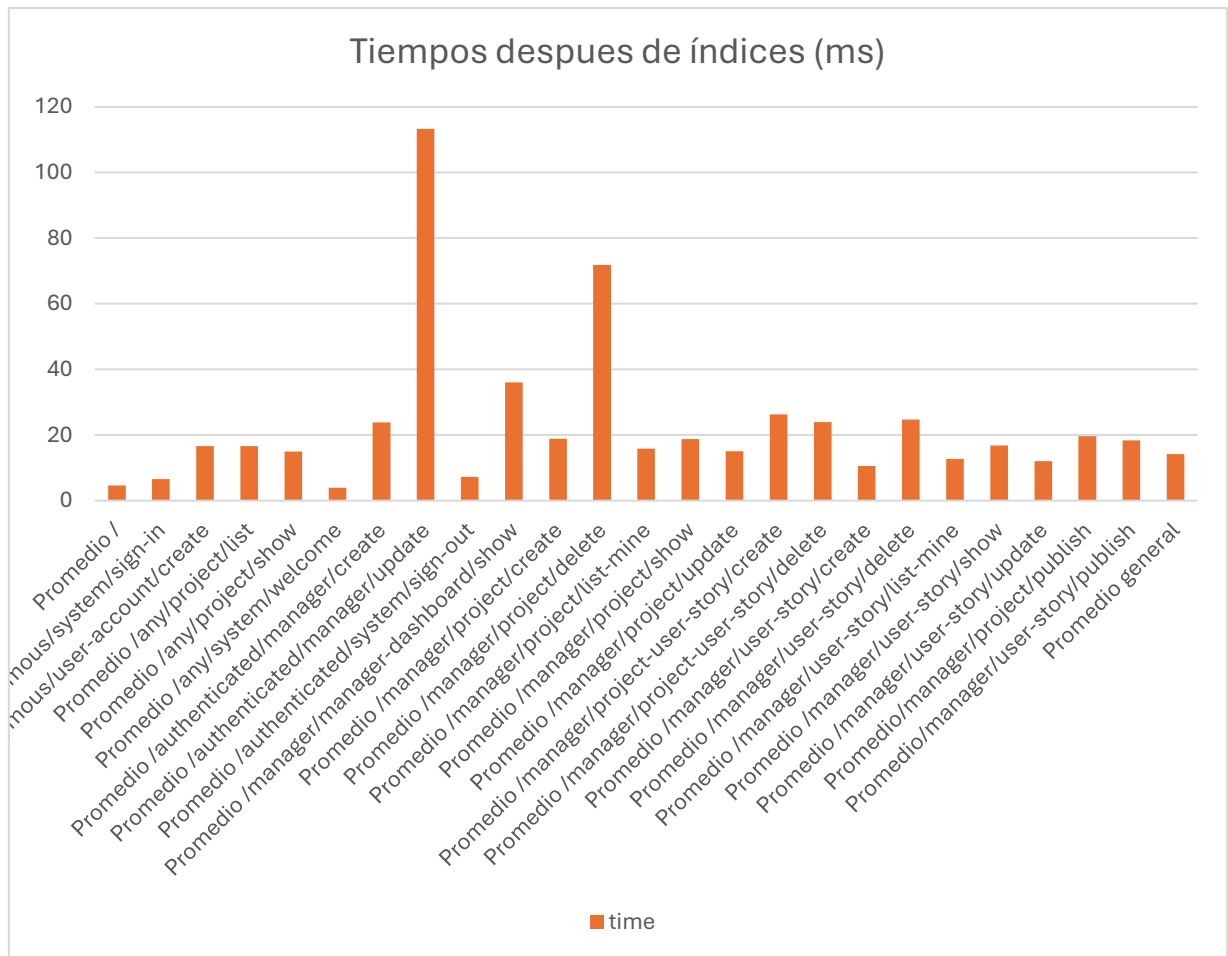
En este caso podemos observar que el pc2 (de mi compañero Guillermo), tiene unas prestaciones mucho mayores a las de mi pc, siendo mucho más constante a la hora de servir las peticiones y superándolo por un “speed up” de 2,1409 en la media, además de tener un intervalo de confianza del 95% mucho más bajo, aunque ambos sean tiempos muy decentes.(en esta versión 1.0 los test publish tanto de Project como de user story solo están en la gráfica después de implementar índices)



Como podemos observar, su ordenador claramente tiene unas prestaciones mayores, esto se puede observar sobre todo en el MIR, donde el peor caso se ve aún más acentuado.

- 2) Version 1.0 Tras realizar la modificación en base de datos para agilizar búsquedas mediante los índices, obtenemos este nuevo resultado:





ESTADÍSTICAS MI PC DESPUES DE INDICES				
Media	14,61002707	interval(ms)	12,470293	16,749761
Error típico	1,088985392	interval(s)	0,0124703	0,0167498
Mediana	9,3568			
Moda	4,1188			
Desviación estándar	23,95767862			
Varianza de la muestra	573,9703649			
Curtosis	78,62945671			
Coefficiente de asimetría	7,464295193			
Rango	327,3614			
Mínimo	2,5326			
Máximo	329,894			
Suma	7071,2531			
Cuenta	484			
Nivel de confianza(95,0%)	2,139733937			

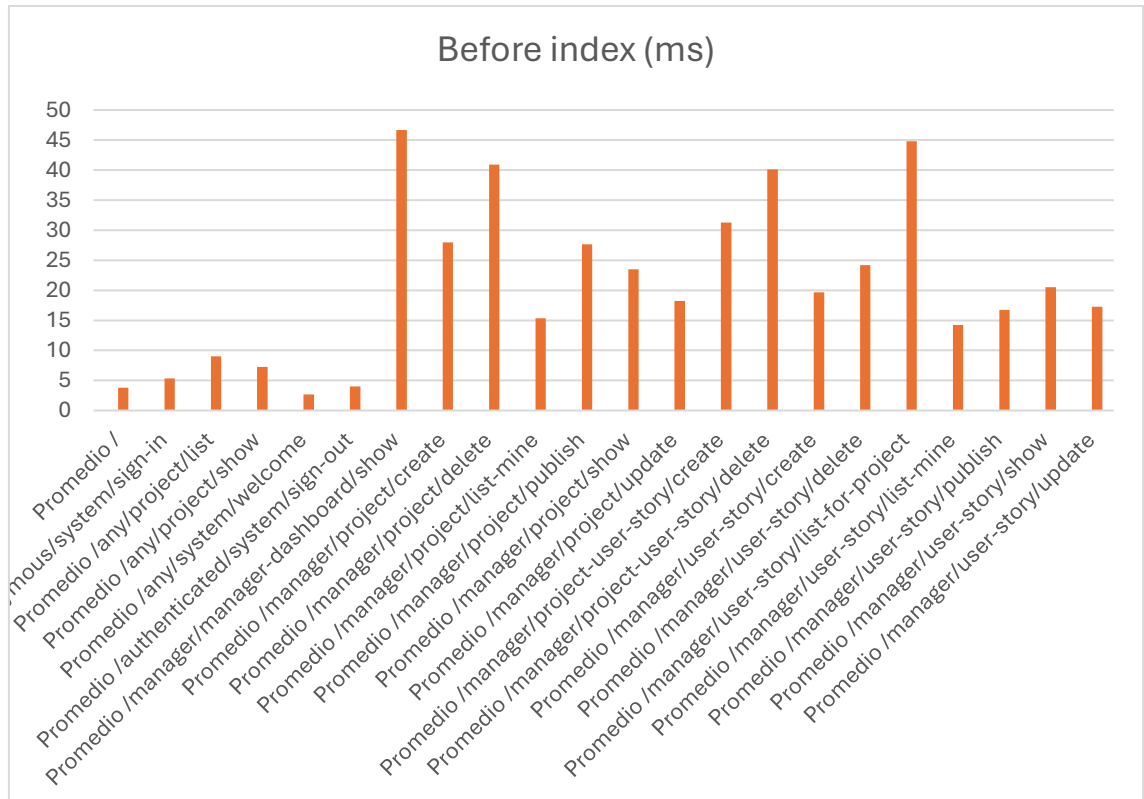
De acuerdo con los intervalos de confianza obtenidos antes y después de crear los índices la diferencia es mínima, de 0,017s a 0,016s.

Por último, en el contraste de hipótesis, nos damos cuenta de que el p-value (0.7435), se sitúa entre 0.05 y 1, por lo tanto, concluyo que no han sido cambios que se hayan traducido en una mejora de gran significancia, aunque podamos ver que la media es débilmente menor.

Prueba z para medias de dos muestras		
	294,6363	329,894
Media	14,0425449	13,514336
Varianza (conocida)	631,48215	570,18472
Observaciones	461	461
Diferencia hipotética de las medias	0	
z	0,327162959	
P(Z<=z) una cola	0,371772317	
Valor crítico de z (una cola)	1,644853627	
Valor crítico de z (dos colas)	0,743544635	
Valor crítico de z (dos colas)	1,959963985	

Ahora procederemos con este nuevo análisis de desempeño de la version 3.0 del entregable, cabe destacar que la comparación entre estas dos versiones tiene dos diferencias, en esta tercera versión los test de autenticación de manager no han sido realizados para priorizar los requisitos obligatorios que debemos satisfacer completamente y el framework ha sido actualizado a una versión mayor:

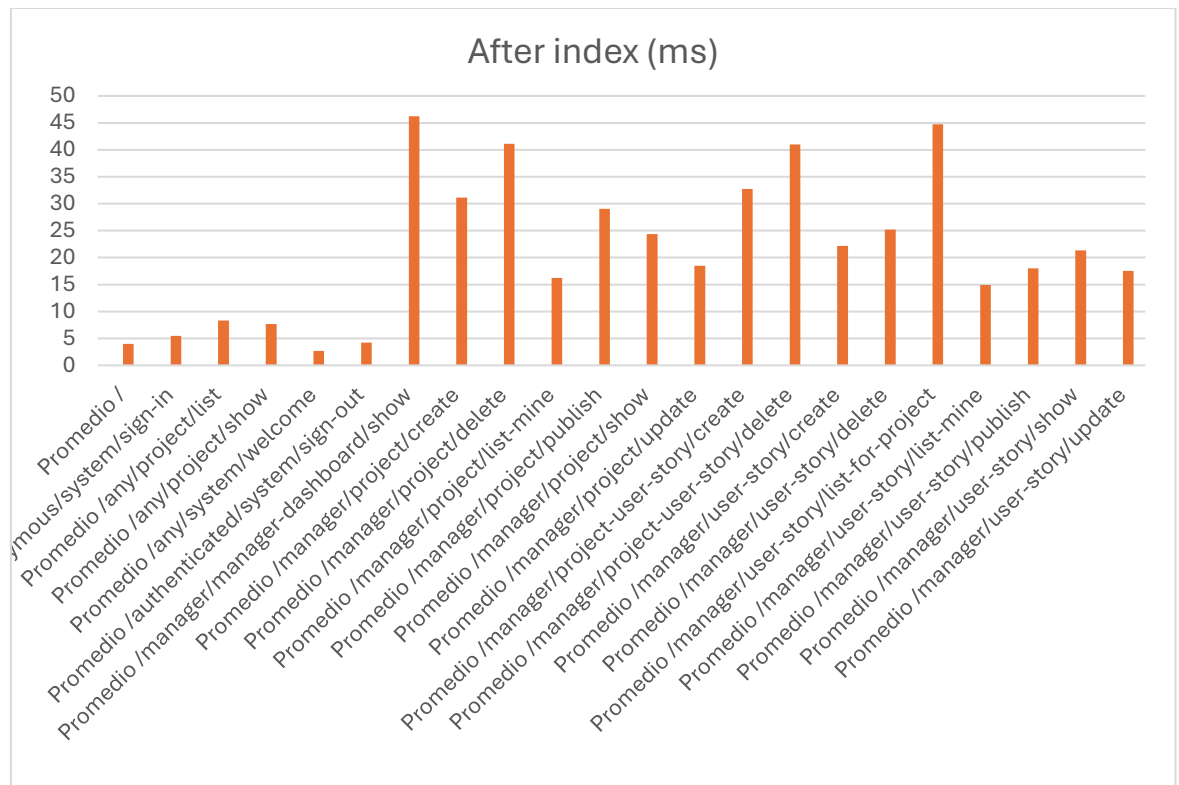
1) Version 3.0 Sin optimizar índices:



BEFORE					
Media	14,65414213	Interval(ms)	13,79063617	15,5176481	
Error típico	0,440311478	Interval(s)	0,013790636	0,01551765	
Mediana	7,2714				
Moda	2,9357				
Desviación estándar	19,91162308				
Varianza de la muestra	396,4727337				
Curtosis	13,61524903				
Coeficiente de asimetría	2,777610821				
Rango	248,4084				
Mínimo	1,7946				
Máximo	250,203				
Suma	29967,72066				
Cuenta	2045				
Nivel de confianza(95,0%)	0,863505962				

Como podemos observar los valores del intervalo de confianza son similares a la prueba 1.0 antes de los índices, esto se debe a que no se han realizado cambios significativos en la lógica de la aplicación, la cota superior del intervalo sigue siendo casi la misma (aproximadamente 2 milisegundos más baja), y la media se mantiene entorno a los 15 milisegundos.

- 2) Version 3.0 Tras realizar la modificación en la aplicación para agilizar búsquedas mediante los índices en la base de datos, obtenemos este nuevo resultado:



AFTER					
Media	15,23424126	Interval(ms)	14,3570951	16,1113874	
Error típico	0,447266781	Interval(s)	0,0143571	0,01611139	
Mediana	7,4469				
Moda	3,1798				
Desviación estándar	20,22615353				
Varianza de la muestra	409,0972868				
Curtosis	15,00228215				
Coefficiente de asimetría	2,798346633				
Rango	262,996				
Mínimo	1,7882				
Máximo	264,7842				
Suma	31154,02338				
Cuenta	2045				
Nivel de confianza(95,0%)	0,877146183				

De acuerdo con los intervalos de confianza obtenidos antes y después de crear los índices la diferencia es mínima, de 0,016s a 0,015s, que son prácticamente los mismos que en la versión 1.0 (aproximadamente 0.017s).

Por último, en el contraste de hipótesis, nos damos cuenta de que el p-value (0.355112807), se sitúa entre 0.05 y 1, por lo tanto, concluyo que no han sido cambios que se hayan traducido en una mejora de gran significancia, aunque podamos ver que la media es débilmente mayor.

	<i>BEFORE</i>	<i>AFTER</i>
Media	14,65410858	15,2342071
Varianza (conocida)	396,2789	408,8972
Observaciones	2046	2046
Diferencia hipotética de las medias	0	
z	-0,924717628	
P(Z<=z) una cola	0,177556404	
Valor crítico de z (una cola)	1,644853627	
Valor crítico de z (dos colas)	0,355112807	
Valor crítico de z (dos colas)	1,959963985	

## Conclusiones

Sin esta fase de testing formal, no hubiera sido capaz de aseverar con una rigurosidad de nivel profesional que la calidad de mi producto era alta. Esta etapa ha sido esencial no solo para identificar y corregir errores, sino también para asegurarme de que cada aspecto de la aplicación funciona según lo esperado bajo diversas condiciones. Este proceso meticuloso me ha permitido detectar fallos críticos que podrían haber pasado desapercibidos en una revisión más superficial, que, de no haber sido corregidos, podrían haberse traducido en problemas graves y potenciales fallos durante el uso por parte del cliente, afectando su experiencia y la reputación del producto.

Gracias al testing formal, he podido asegurarme de que el producto final es robusto, fiable y capaz de cumplir con las expectativas y necesidades del usuario. Este nivel de escrutinio no solo mejora la calidad del producto, sino que también genera una mayor confianza tanto entre los usuarios y clientes como entre el propio desarrollador y su aplicación, quienes pueden estar seguros de que están adquiriendo un producto bien desarrollado y probado rigurosamente.

Finalmente, el uso de comprobaciones en distintos ordenadores, así como el elemento estadístico utilizado sobre los datos del producto, como el Z-Test y el intervalo de confianza, han sido fundamentales para validar el rendimiento y la estabilidad de la aplicación. Estas herramientas estadísticas nos han permitido evaluar de manera objetiva si los cambios realizados han tenido un impacto significativo. El Z-Test, en particular, ha proporcionado una forma precisa de comparar los tiempos de ejecución antes y después de las modificaciones, mientras que el intervalo de confianza nos ha ofrecido una medida de la precisión de nuestras estimaciones. Estas metodologías han asegurado que los resultados obtenidos no son producto del azar, sino reflejan mejoras o consistencias reales en el rendimiento del producto.

En resumen, el testing formal no es solo una fase más del desarrollo, sino una piedra angular que asegura la calidad del producto protege la satisfacción del cliente y fortalece la confianza en los productos que desarrollo.

## Bibliografía

Web de la universidad de Sevilla - <https://ev.us.es>