



# Testing Report

C2.017

<https://github.com/vicgrabru/Acme-SF-D04>

Victor Graván Bru  
vicgrabru@alum.us.es

# Tabla de contenidos

<b>Tabla de contenidos</b>	<b>1</b>
<b>Resumen ejecutivo</b>	<b>2</b>
<b>Tabla de revisiones</b>	<b>2</b>
<b>Contenido</b>	<b>3</b>
Pruebas Funcionales	4
Casos de prueba positivos, negativos y hackeos para Project	4
List.safe	4
List.hack	4
Show.safe	4
Show.hack	4
Create.safe	5
Create.hack	7
Update.safe	7
Update.hack	8
Delete.safe	8
Delete.hack	9
Publish.safe	9
Publish.hack	10
Casos de prueba positivos, negativos y hackeos para User Story	10
Assign-buttons.safe	10
Por qué se hace esta prueba	10
La prueba en sí	11
List.safe	11
List.hack	12
List-mine.safe	12
List-mine.hack	12
Show.safe	12
Show.hack	13
Create.safe	13
Create.hack	15
Update.safe	16
Update.hack	16
Delete.safe	17
Delete.hack	17
Publish.safe	18
Publish.hack	18
Casos de prueba positivos, negativos y hackeos para User Story	19
Create.safe	19
Create.hack	19
Delete.safe	20

Delete.hack	21
Pruebas de Rendimiento	22
Efectos de índices en PC 1	22
Efectos de índices en PC 2	24
Comparación de rendimientos entre PCs	26
Comparación sin índices	26
Comparación con índices	27
<b>Conclusión</b>	<b>29</b>
<b>Bibliografía</b>	<b>29</b>

## Resumen ejecutivo

Las pruebas de un sistema software son esenciales para garantizar el funcionamiento deseado del producto, permitiendo gran variedad de beneficios como el descubrimiento temprano de bugs, el análisis y comprensión del código, y muchas otras ventajas.

Sin embargo, un testing de calidad implica un procedimiento riguroso comprobando hasta el último recorrido del código a probar mediante un alto coverage, pues sino es así es posible que queden escondidos algunos fallos entre las sombras del sistema.

## Tabla de revisiones

Revisión	Fecha	Descripción
1.0	04-05-2024	Primera versión del documento
2.0	27-05-2024	Versión final del documento
3.0	29-06-2024	Arreglos para segunda convocatoria

# Contenido

En primer lugar, se muestra la cobertura que se obtiene al reproducir todas las pruebas.

▼	acme.features.manager.project	93,7 %
>	ManagerProjectController.java	100,0 %
>	ManagerProjectCreateService.java	93,7 %
>	ManagerProjectDeleteService.java	90,4 %
>	ManagerProjectListService.java	93,7 %
>	ManagerProjectPublishService.java	92,8 %
>	ManagerProjectShowService.java	96,7 %
>	ManagerProjectUpdateService.java	94,3 %
▼	acme.features.manager.userStory	93,8 %
>	ManagerUserStoryController.java	100,0 %
>	ManagerUserStoryCreateService.java	94,3 %
>	ManagerUserStoryDeleteService.java	92,0 %
>	ManagerUserStoryListMineService.java	89,3 %
>	ManagerUserStoryListService.java	93,8 %
>	ManagerUserStoryPublishService.java	91,6 %
>	ManagerUserStoryShowService.java	97,4 %
>	ManagerUserStoryUpdateService.java	94,3 %
▼	acme.features.manager.userStoryAssign	93,5 %
>	ManagerUserStoryAssignController.java	100,0 %
>	ManagerUserStoryAssignCreateService.java	93,4 %
>	ManagerUserStoryAssignDeleteService.java	93,3 %

Se observa que todos los servicios que se han probado superan el 90% sin problemas.

Las instrucciones marcadas en amarillo por el coverage son los siguientes casos:

- **assert object != null:** Estas instrucciones son una medida preventiva para no ejecutar instrucciones que accedan a propiedades de un objeto nulo, lo que lanzaría una excepción, que llevaría a mostrar una vista de pánico.

```
@Override
public void validate(final UserStory object) {
    assert object != null;
```

- **Ramas inaccesibles de condicionales:** Esto ocurre solamente en el servicio de publicación de Project. Se debe a que compruebo que el Project no tenga errores fatales antes de publicarlo, y como el mensaje de error lo pongo en el campo "hasFatalErrors", compruebo primero que el framework no haya encontrado algún error primero. Como este campo es un checkbox, no puedo hacer un caso de prueba en el que el framework encuentre un error en ese campo (ya que el servicio es de publicar el Project, no de actualizarlo), por lo que siempre va a entrar en el condicional.

```
if (!super.getBuffer().getErrors().hasErrors("hasFatalErrors"))
    super.state(!object.isHasFatalErrors(), "hasFatalErrors", "manager.project.form.error.has-fatal-errors");
}
```

# Pruebas Funcionales

## Casos de prueba positivos, negativos y hackeos para Project

### List.safe

Con manager1, accedemos a su lista de Project, y comprobamos que todo funciona bien.

Accedemos a la lista de Projects de manager1 para maximizar la cobertura de código, ya que en el listado se muestran símbolos distintos para Projects publicados y sin publicar, y manager1 tiene ambos.

### List.hack

Entramos en la url `http://localhost:8082/Acme-SF-D04/manager/project/list`

- Sin iniciar sesión
- Con developer1, que no tiene el rol adecuado

En ambos casos, no estamos autorizados a acceder a la vista

### Show.safe

Con manager1, accedemos a la vista de detalles de los Projects de code:

- AAA-1234: está publicado
- NOP-0123: no está publicado

Esto lo hacemos para maximizar la cobertura de código, ya que los botones de actualizar, publicar, y borrar no aparecen si un Project está publicado.

### Show.hack

Entramos en la url `http://localhost:8082/Acme-SF-D04/manager/project/show?id=[id]`

Con los IDs:

- 318: es un Project del manager10 sin publicar
- 778: no es un Project

Con los usuarios:

- Sin iniciar sesión
- Con developer1: Rol incorrecto
- Con manager1: Rol correcto, Usuario incorrecto

En todos los casos, no estamos autorizados a acceder a la vista.

Para maximizar la cobertura de código, utilizamos el id 778, que cubre la rama del `authorise` en la que el objeto obtenido con el id es nulo.

Create.safe

Con manager1 intentamos crear proyectos en el siguiente orden:

- 1. Todos los campos vacíos (para comprobar que da fallo en los campos obligatorios)
- 2. Todos los casos negativos detallados en la tabla de valores
- 3. Un caso positivo genérico (con valores de longitud intermedia para todos los campos)
- 4. Casos positivos detallados en la tabla de valores

Los casos positivos de cada campo no se empiezan a probar hasta que no se hayan probado todos los casos negativos de todos los campos.

En otras palabras, en lugar de hacer un caso de prueba separado para cada límite de cada campo, en cada caso de prueba se ha utilizado el siguiente valor de la tabla, hasta que no queden valores por probar.

En los casos en los que no queden valores por probar para un campo, pero quedan valores por probar de otro campo, utilizamos un valor positivo genérico hasta que no queden valores por probar de ningún campo.

Atributo	Casos Positivos			Casos Negativos		
	Valores Probados		Nº de casos	Valores Probados		Nº de casos
Code	Siguiendo el patrón	TST-0000	1	Sin respetar el patrón	TS-001	2
				Código duplicado	AAA-1234	
Title	LL	L	8	Spam	sex	3
	LL + Δ	Lo				
	UL - Δ	74 caracteres				
	UL	75 caracteres		UL + Δ	76 caracteres	
	Caracteres de otros alfabetos	국민경제의 발전을				
		العظمى واعتلاء				
	Intento de inyección de código	<h1>!</h1>		UL + 2*Δ	77 caracteres	
		' or 'A'='A				
Abstract	LL	L	8	Spam	sex	3
	LL + Δ	Lo				
	UL - Δ	99 caracteres				

	UL	100 caracteres		UL + Δ	101 caracteres		
	Caracteres de otros alfabetos	국민경제의 발전을 العظمى واعتلاء					
		Intento de inyección de código		<h1>!</h1>	UL + 2*Δ		102 caracteres
	' or 'A'='A						
Fatal errors	Valores Booleanos	Verdadero	2	*(No hay casos negativos)		0	
Falso							
Cost	LL	EUR 0.00	5	LL - Δ	EUR -0.01	4	
	LL + Δ	EUR 0.01		UL + Δ	EUR 10000000000.00		
	UL - Δ	EUR 9999999999.98		UL + 2*Δ	EUR 10000000000.01		
	UL	EUR 9999999999.99		Más de 2 decimales	EUR 1.234		
	Otra moneda reconocida por el sistema	USD 0.00		Moneda no reconocida por el sistema	ABC 0.00		
Optional Link	Vacío (es un atributo opcional)		9	Url no válida	estonoesunaur l	3	
	LL	ftp://a					
	LL + Δ	ftp://a.b					
	UL - Δ	254 caracteres		UL + Δ	256 caracteres		
	UL	255 caracteres					
	https://www.lorem-ipsum.org						
	http://www.lorem-ipsum.org/dolor/sit.html#dolor			UL + 2*Δ	257 caracteres		
	http://example.org?a=1&b=2						
	http://example.org/a/b?a&b						

Notas: LL, UL, y  $\Delta$  son, respectivamente, los valores mínimos y máximos aceptables que cumplan los formatos del tipo de dato del atributo, y el mínimo incremento posible. En el caso de los enlaces, se prueban los enlaces de longitud mínima aunque sea un campo opcional (en cuyo caso también se prueba que acepte no introducir un valor).

El campo “Fatal errors” no tiene casos negativos porque, al ser un checkbox, solo puede estar marcado o no (valores Verdadero y Falso), por lo que los valores negativos solo pueden ocurrir inspeccionando el formulario y alterando el campo “value” a otra cosa distinta. Esto se prueba en la grabación de hacking.

### Create.hack

Entramos en la url <http://localhost:8082/Acme-SF-D04/manager/project/create>

- Sin iniciar sesión
- Con developer1: Rol incorrecto

En ambos casos, no estamos autorizados a acceder a la vista.

Con manager1, intentamos crear un Project, pero antes de hacer click en el botón de crear, alteramos los siguientes campos (inspeccionando los elementos en el navegador):

- Has any Fatal Errors: cambiamos el atributo “value” a algo aleatorio (por ejemplo, “Lorem Ipsum”)
- DraftMode: establecemos el valor a False, y comprobamos en la vista de detalles que no se ha guardado
- id (input oculto del formulario): lo cambiamos a 279 (es el id del Project con code AAA-1234, que pertenece al manager1 y está publicado)

Cambiar el id se hace para probar que el sistema es resistente a un tipo de POST hacking en el que se usurpa el id de otra entidad para hacer una operación ilegal.

Este tipo de hacking se realiza introduciendo el id de otra entidad cuando se está creando o actualizando una entidad. Como las 2 operaciones se hacen con la misma instrucción en el repositorio (save), la forma que tiene el framework de diferenciar las operaciones es mediante el id. Cuando se crea una entidad nueva, el id que se le da es 0 hasta que se guarda en la base de datos.

- Si la entidad que se intenta guardar tiene id 0, el se considera que es una entidad nueva y el framework le da el id que le corresponda
- Si la entidad ya tiene un id, se considera que es una operación de actualización

Al cambiar el id del formulario, se hace pasar la entidad nueva como otra entidad que hay que actualizar.

Para evitar este tipo de ataque, en el método perform, se establece a 0 el id de la entidad antes de guardarla en la base de datos.

### Update.safe

Accedemos a la url <http://localhost:8082/Acme-SF-D04/manager/project/update?id=332> con el manager1 (la id pertenece al Project con code ZZZ-0100, que pertenece al manager1 y no está publicado). Esto es para asegurar que se puede acceder a esa vista mediante una petición GET cuando es una operación legal.

Con manager1, editamos el Project con “code” ZZZ-0100, que está preparado para este test.

Hacemos los mismos casos de prueba que en create.safe (excepto con el campo “code”, que no se puede cambiar).



## Update.hack

Entramos en la url `http://localhost:8082/Acme-SF-D04/manager/project/update?id=[ID]`

Con las IDs:

- 279: es un Project del manager1 publicado
- 318: es un Project del manager10 sin publicar
- 778: no es un Project

Con los usuarios:

- Sin iniciar sesión
- Con developer1: Rol incorrecto
- Con manager1: Rol correcto, Usuario incorrecto

En todos los casos, no estamos autorizados a acceder a la vista.

Para maximizar la cobertura de código, utilizamos el ID 778, que cubre la rama del `authorise` en la que el objeto obtenido con el `id` es nulo.

Con manager1, intentamos editar el Project con code ZZZ-0100, pero antes de hacer click en el botón de actualizar, alteramos los siguientes campos (inspeccionando los elementos en el navegador):

- Has any Fatal Errors: cambiamos el atributo "value" a algo aleatorio (por ejemplo, "Lorem Ipsum")
- DraftMode: establecemos el valor a False, y comprobamos en la vista de detalles que no se ha guardado
- Code: cambiamos el Code a otro valor u comprobamos en la vista de detalles que no se ha modificado
- id (input oculto del formulario): lo cambiamos a los mismos valores usados al principio de la grabación en la url.

## Delete.safe

Accedemos a la url `http://localhost:8082/Acme-SF-D04/manager/project/delete?id=323` con el manager1 (la id pertenece al Project con code ZZZ-0000, que pertenece al manager1 y no está publicado). Esto es para asegurar que se puede acceder a esa vista mediante una petición GET cuando es una operación legal.

Con manager1, borramos los Projects con los siguientes codes, que no están publicados:

- ZZZ-0000: No tiene ninguna historia de usuario asignada
- ZZZ-0001: Tiene 1 historia de usuario asignada
- ZZZ-0002: Tiene 3 historias de usuario asignada

## **Delete.hack**

Entramos en la url `http://localhost:8082/Acme-SF-D04/manager/project/delete?id=[ID]`

Con las IDs:

- 279: es un Project del manager1 publicado
- 318: es un Project del manager10 sin publicar
- 778: no es un Project

Con los usuarios:

- Sin iniciar sesión
- Con developer1: Rol incorrecto
- Con manager1: Rol correcto, Usuario incorrecto

En todos los casos, no estamos autorizados a acceder a la vista.

Para maximizar la cobertura de código, utilizamos el ID 778, que cubre la rama del `authorise` en la que el objeto obtenido con el id es nulo.

Con manager1, accedemos a la vista de detalles del Project con code ZZZ-0000, y antes de pulsar el botón de borrar, cambiamos el valor del campo id (inspeccionando el formulario) a los ids usados al principio de la grabación en la url.

## **Publish.safe**

Accedemos a la url `http://localhost:8082/Acme-SF-D04/manager/project/publish?id=326` con el manager1 (la id pertenece al Project con code ZZZ-0003, que pertenece al manager1 y no está publicado). Esto es para asegurar que se puede acceder a esa vista mediante una petición GET cuando es una operación legal.

Con manager1, intentamos publicar los Projects con los siguientes codes:

- ZZZ-0500: No tiene historias de usuario asignadas
- ZZZ-0501: Tiene 1 historia de usuario sin publicar asignada
- ZZZ-0502: Tiene varias historias de usuario asignadas (2 publicadas y 1 sin publicar)
- ZZZ-0503: Tiene errores fatales

Con el mismo usuario, publicamos los Projects con los siguientes codes:

- ZZZ-0003: Tiene asignada 1 historia de usuario publicada
- ZZZ-0004: Tiene asignadas 3 historias de usuario publicadas

## **Publish.hack**

Entramos en la url `http://localhost:8082/Acme-SF-D04/manager/project/publish?id=[ID]`

Con las IDs:

- 279: es un Project del manager1 publicado
- 318: es un Project del manager10 sin publicar
- 778: no es un Project

Con los usuarios:

- Sin iniciar sesión
- Con developer1: Rol incorrecto
- Con manager1: Rol correcto, Usuario incorrecto

En todos los casos, no estamos autorizados a acceder a la vista.

Para maximizar la cobertura de código, utilizamos el ID 778, que cubre la rama del `authorise` en la que el objeto obtenido con el id es nulo.

Con manager1, accedemos a la vista de detalles del Project con code ZZZ-0003, y antes de pulsar el botón de publicar, cambiamos el valor del campo id (inspeccionando el formulario) a los mismos valores usados al principio de la grabación en la url.

## **Casos de prueba positivos, negativos y hackeos para User Story**

### **Assign-buttons.safe**

#### **Por qué se hace esta prueba**

Para maximizar la cobertura de código, he hecho también esta grabación. No prueba específicamente la funcionalidad de ningún requisito, pero aumenta la cobertura de varios servicios relacionados con UserStory.

Los botones para acceder a la función de asignar y desasignar una historia de usuario de un proyecto se encuentran en la vista de detalles de las UserStory. Para evitar complicaciones de intentar hacer estas operaciones cuando no hay opciones posibles, cuando no esté asignada a ningún Project o esté asignada a todos los Projects posibles; estos botones se ocultan en dichos casos.

Esto se realiza añadiendo un parámetro para cada botón al modelo en el `unbind` de todos los servicios que cargan esta vista: los servicios de actualizar, mostrar, publicar, y eliminar historias de usuario (no se hace en el servicio de crear porque los botones nunca se van a mostrar si el comando es `"create"`).

```
dataset.put("showAssignButton", !draftModeProjectsUnassigned.isEmpty());  
dataset.put("showUnassignButton", !draftModeProjectsAssigned.isEmpty());
```

Esto nos crea 2 ramas para cada parámetro que hay que comprobar en cada servicio. Como este método se ejecuta al hacer una petición GET (que no afecta a la base de datos y se puede hacer con la misma entidad para todos los servicios), he decidido hacer estos casos de prueba en una grabación aparte.

## La prueba en sí

Entramos en la url `http://localhost:8082/Acme-SF-D04/manager/user-story/[accion]?id=961` con `manager1`, con las siguientes acciones:

- `update`
- `delete`
- `publish`
- `show`

Esta historia de usuario no está publicada, pertenece al `manager1`, y está asociada a los Projects con code `ZZZ-0501` y `ZZZ-0502`, ambos sin publicar y del `manager1`.

Desasociamos la historia de usuario de ambos Projects (a esta función se accede mediante el botón de “Unassign from a Project” de la vista de detalles).

Cuando no se puede desasignar de más Projects (lo sabemos porque el botón deja de mostrarse), volvemos a entrar en las mismas 4 urls.

Después de entrar en las 4 urls, asignamos la historia de usuario a todos los Projects que nos permita el sistema (a esta función se accede mediante el botón de “Assign to a Project” de la vista de detalles).

Cuando no se puede asignar de más Projects (lo sabemos porque el botón deja de mostrarse), volvemos a entrar en las mismas 4 urls.

## List.safe

Con `manager1`, accedemos a la lista de historias de usuario asignadas a los Projects con los siguientes codes (se accede mediante el botón de “User Stories” de la vista de detalles del proyecto):

- `AAA-1234`: está publicado
- `ZZZ-0502`: tiene historias de usuario publicadas y sin publicar

Se accede a la lista de historias de usuario de estos 2 proyectos para maximizar la cobertura de código.

En el primer caso, al estar publicado, en el listado de sus UserStory no se muestra el botón de crear una nueva.

En el segundo caso, se accede a la lista de un proyecto que tiene asignadas historias de usuario publicadas y sin publicar, ya que se muestran símbolos distintos dependiendo del caso.

### **List.hack**

Entramos en la url `http://localhost:8082/Acme-SF-D04/manager/user-story/list?masterId=[id]`

Con los IDs:

- 318: es un Project del manager10 sin publicar
- 778: no es un Project

Con los usuarios:

- Sin iniciar sesión
- Con developer1: Rol incorrecto
- Con manager1: Rol correcto, Usuario incorrecto

En todos los casos, no estamos autorizados a acceder a la vista.

Para maximizar la cobertura de código, utilizamos el id 778, que cubre la rama del `authorise` en la que el objeto obtenido con el id es nulo.

### **List-mine.safe**

Con manager1, accedemos a su lista de UserStory, y comprobamos que todo funciona bien.

Repetimos la operación, pero con manager44.

Accedemos a la lista de UserStory de manager1 para maximizar la cobertura de código, ya que en el listado se muestran símbolos distintos para Projects publicados y sin publicar, y manager1 tiene ambos.

Accedemos a la lista de UserStory de manager44 para probar el caso del listado vacío.

### **List-mine.hack**

Entramos en la url `http://localhost:8082/Acme-SF-D04/manager/user-story/list-mine`

- Sin iniciar sesión
- Con developer1: Rol incorrecto

En todos los casos, no estamos autorizados a acceder a la vista.

### **Show.safe**

Con manager1, accedemos a la lista de historias de usuario asignadas al Project con code ZZZ-0502, y accedemos a la vista de detalles de las 2 primeras historias de usuario (una está publicada y la otra no).

Esto lo hacemos para maximizar la cobertura de código, ya que los botones de actualizar, publicar, y borrar no aparecen si un Project está publicado.

Show.hack

Entramos en la url `http://localhost:8082/Acme-SF-D04/manager/user-story/show?id=[id]`

Con los IDs:

- 927: es una UserStory del manager9 sin publicar
- 778: no es un Project

Con los usuarios:

- Sin iniciar sesión
- Con developer1: Rol incorrecto
- Con manager1: Rol correcto, Usuario incorrecto

En todos los casos, no estamos autorizados a acceder a la vista.

Para maximizar la cobertura de código, utilizamos el id 778, que cubre la rama del `authorise` en la que el objeto obtenido con el id es nulo.

Create.safe

Con manager1 intentamos crear una User Story, accediendo desde la lista de User Stories asignadas al Project ZZZ-0502, para que quede asignada al Project.

La creamos siguiendo el mismo criterio para el orden de los casos de prueba y los valores usados en los mismos que se ha seguido para la creación de historias de usuario.

Atributo	Casos Positivos			Casos Negativos		
	Valores Probados		Nº de casos	Valores Probados		Nº de casos
Title	LL	L	8	Spam	sex	3
	LL + Δ	Lo				
	UL - Δ	74 caracteres		UL + Δ	76 caracteres	
	UL	75 caracteres				
	Caracteres de otros alfabetos	국민경제의 발전을		UL + 2*Δ	77 caracteres	
		العظمى واعتلاء				
	Intento de inyección de código	<h1>!</h1>				
		' or 'A'='A				
Description	LL	L	8	Spam	sex	3
	LL + Δ	Lo				
	UL - Δ	99 caracteres		UL + Δ	101 caracteres	
	UL	100 caracteres				

	Caracteres de otros alfabetos	국민경제의 발전을				
		العظمى واعتلاء				
	Intento de inyección de código	<h1>!</h1>		UL + 2*Δ	102 caracteres	
		' or 'A'='A				
Estimated cost in hours	LL	1	2	LL - 2*Δ	0	3
	LL + Δ	2		LL - Δ	-1	
				Valor decimal	0.1	
Acceptance Criteria	LL	L	8	Spam	sex	3
	LL + Δ	Lo				
	UL - Δ	99 caracteres		UL + Δ	101 caracteres	
	UL	100 caracteres				
	Caracteres de otros alfabetos	국민경제의 발전을		UL + 2*Δ	102 caracteres	
		العظمى واعتلاء				
	Intento de inyección de código	<h1>!</h1>				
		' or 'A'='A				
Priority	Valores posibles	MUST	4	*Valor Nulo	----	1
		SHOULD				
		COULD				
		WONT				
Optional Link	Vacío (es un atributo opcional)		9	Url no válida	estonoesunaurL	3
	LL	ftp://a				
	LL + Δ	ftp://a.b		UL + Δ	256 caracteres	
	UL - Δ	254 caracteres				
	UL	255 caracteres		UL + 2*Δ	257 caracteres	
	https://www.lorem-ipsum.org					
	http://www.lorem-ipsum.org/dolor/sit.html#dolor					

	http://example.org?a=1&b=2				
	http://example.org/a/b?a&b				

Notas: En el tiempo estimado no he probado limite superior porque no tiene ninguna restricci3n de l3mite superior.

Priority es un caso similar al del campo “Fatal errors” de la creaci3n de Projects: en lugar de ser solo verdadero o falso, es un selector, por lo que el caso negativo (aparte de seleccionar la opci3n nula) es alterar el valor a uno aleatorio inspeccionando el formulario, lo que se prueba en la grabaci3n de hacking.

## Create.hack

Entramos en la url

http://localhost:8082/Acme-SF-D04/manager/user-story/create?masterId=[ID]

Con los IDs:

- 279: es un Project del manager1 publicado
- 318: es un Project del manager10 sin publicar
- 778: no es un Project

Con los usuarios:

- Sin iniciar sesi3n
- Con developer1: Rol incorrecto
- Con manager1: Rol correcto, Usuario incorrecto

En todos los casos, no estamos autorizados a acceder a la vista.

Para maximizar la cobertura de c3digo, utilizamos el ID 778, que cubre la rama del authorise en la que el objeto obtenido con el id es nulo.

Con manager1, accedemos al listado de UserStory asociadas al Project con code NOP-0123, e intentamos crear una, pero antes de hacer click en el bot3n de crear, alteramos los siguientes campos (inspeccionando los elementos en el navegador):

- Priority: cambiamos el valor del atributo “value” a “LOREM” (no es una de las opciones del selector)
- DraftMode: establecemos el valor a False, y comprobamos en la vista de detalles que no se ha guardado
- id (input oculto del formulario): lo cambiamos a 906 (es el id de una UserStory asociada al Project con code AAA-1234, pertenece al manager1 y est3 publicada)
- bot3n “Create”: cambiamos el valor del par3metro “masterId” a los ids usados al principio de la grabaci3n en la url



### **Update.safe**

Accedemos a la url `http://localhost:8082/Acme-SF-D04/manager/user-story/update?id=966` con el manager1 (la id pertenece a una UserStory del manager1 y no está publicada). Esto es para asegurar que se puede acceder a esa vista mediante una petición GET cuando es una operación legal.

Con manager1, accedemos a la lista de historias de usuario del Project con code ZZZ-0502, y editamos la primera que no está publicada.

Hacemos los mismos casos de prueba que en create.safe.

### **Update.hack**

Entramos en la url `http://localhost:8082/Acme-SF-D04/manager/user-story/update?id=[ID]`

Con las IDs:

- 906: es una UserStory del manager1 que está publicada
- 927: es una UserStory del manager9 que no está publicada
- 778: no es una UserStory

Con los usuarios:

- Sin iniciar sesión
- Con developer1: Rol incorrecto
- Con manager1: Rol correcto, Usuario incorrecto

En todos los casos, no estamos autorizados a acceder a la vista.

Para maximizar la cobertura de código, utilizamos el ID 778, que cubre la rama del `authorise` en la que el objeto obtenido con el id es nulo.

Con manager1, accedemos a la lista de historias de usuario del Project con code ZZZ-0502, y editamos la primera que no está publicada, pero antes de hacer click en el botón de actualizar, alteramos los siguientes campos (inspeccionando los elementos en el navegador):

- DraftMode: establecemos el valor a False, y comprobamos en la vista de detalles que no se ha guardado
- Priority: cambiamos el valor a "LOREM"
- id (input oculto del formulario): lo cambiamos a los mismos valores usados al principio de la grabación en la url.

### **Delete.safe**

Accedemos a la url <http://localhost:8082/Acme-SF-D04/manager/user-story/delete?id=966> con el manager1 (la id pertenece a una UserStory del manager1 y no está publicada). Esto es para asegurar que se puede acceder a esa vista mediante una petición GET cuando es una operación legal.

Con manager1, accedemos a la lista de User Stories asignadas al Project con code ZZZ-0502, y eliminamos la primera que no está publicada.

Para maximizar la cobertura de código, debemos cubrir los casos en los que se borran las siguientes UserStory:

- Una UserStory que no esté asociada a ningún Project
- Una UserStory que esté asociada a 1 Project

Para ello, con manager1 creamos 2 historias de usuario desde la lista de historias de usuario del Project con code ZZZ-0502.

Accedemos a la vista de detalles de una de ellas y la eliminamos.

Accedemos a la vista de detalles de la otra, la desvinculamos del Project al que se ha asociado, y la eliminamos.

### **Delete.hack**

Entramos en la url [http://localhost:8082/Acme-SF-D04/manager/user-story/delete?id=\[ID\]](http://localhost:8082/Acme-SF-D04/manager/user-story/delete?id=[ID])

Con las IDs:

- 906: es una UserStory del manager1 que está publicada
- 927: es una UserStory del manager9 que no está publicada
- 778: no es una UserStory

Con los usuarios:

- Sin iniciar sesión
- Con developer1: Rol incorrecto
- Con manager1: Rol correcto, Usuario incorrecto

En todos los casos, no estamos autorizados a acceder a la vista.

Para maximizar la cobertura de código, utilizamos el ID 778, que cubre la rama del `authorise` en la que el objeto obtenido con el id es nulo.

Con manager1, accedemos a la lista de historias de usuario del Project con code ZZZ-0502, entramos a la vista de detalles de la primera historia de usuario sin publicar, y antes de pulsar el botón de borrar, cambiamos el valor del campo id (inspeccionando el formulario) a los ids usados al principio de la grabación en la url.

### **Publish.safe**

Accedemos a la url `http://localhost:8082/Acme-SF-D04/manager/user-story/publish?id=966` con el manager1 (la id pertenece a una UserStory del manager1 y no está publicada). Esto es para asegurar que se puede acceder a esa vista mediante una petición GET cuando es una operación legal.

Con manager1, accedemos a la lista de User Stories asignadas al Project con code ZZZ-0502, y publicamos la primera que no está publicada.

### **Publish.hack**

Entramos en la url `http://localhost:8082/Acme-SF-D04/manager/user-story/publish?id=[ID]`

Con las IDs:

- 906: es una UserStory del manager1 que está publicada
- 927: es una UserStory del manager9 que no está publicada
- 778: no es una UserStory

Con los usuarios:

- Sin iniciar sesión
- Con developer1: Rol incorrecto
- Con manager1: Rol correcto, Usuario incorrecto

En todos los casos, no estamos autorizados a acceder a la vista.

Para maximizar la cobertura de código, utilizamos el ID 778, que cubre la rama del `authorise` en la que el objeto obtenido con el id es nulo.

Con manager1, accedemos a la lista de historias de usuario del Project con code ZZZ-0502, entramos a la vista de detalles de la primera historia de usuario sin publicar, y antes de pulsar el botón de publicar, cambiamos el valor del campo id (inspeccionando el formulario) a los ids usados al principio de la grabación en la url.

## Casos de prueba positivos, negativos y hackeos para User Story

### Create.safe

Con el manager1, accedemos a la url

<http://localhost:8082/Acme-SF-D04/manager/user-story-assign/create?userId=957> (la id pertenece a una UserStory del manager1 que está publicada y que todavía se puede asignar a más Projects). Esto es para asegurar que se puede acceder a esa vista mediante una petición GET cuando es una operación legal.

A continuación, entramos al listado de historias de usuario asignadas al Project con code NOP-0123 (no está publicado y no tiene ninguna historia de usuario asociada), y creamos 3 historias de usuario distintas.

Entramos en el formulario de asignación de historias de usuario de una de las 3, e intentamos asignarla dejando vacío el campo de "Project".

Acto seguido, asignamos una de ellas a varios Projects, y otra a todos los Projects posibles. De esta forma, acabamos con:

- 1 UserStory asignada solamente a 1 Project
- 1 UserStory asignada a varios Projects
- 1 UserStory asignada a todos los Projects posibles

### Create.hack

Entramos en la url

[http://localhost:8082/Acme-SF-D04/manager/user-story-assign/create?userId=\[ID\]](http://localhost:8082/Acme-SF-D04/manager/user-story-assign/create?userId=[ID])

Con las IDs:

- 927: es una UserStory del manager9
- 778: no es una UserStory

Con los usuarios:

- Sin iniciar sesión
- Con developer1: Rol incorrecto
- Con manager1: Rol correcto, Usuario incorrecto

En todos los casos, no estamos autorizados a acceder a la vista.

Para maximizar la cobertura de código, utilizamos el ID 778, que cubre la rama del authorise en la que el objeto obtenido con el id es nulo.

Después, accedemos con el manager1 a su lista de historias de usuario, asignamos la cuarta a todos los Projects posibles, y accedemos a la url

<http://localhost:8082/Acme-SF-D04/manager/user-story-assign/create?userId=957>

Accedemos a la lista de historias de usuario asociadas al Project AAA-1234 y comprobamos que la primera tiene de title "Test title"

Accedemos al listado de historias de usuario del manager1, entramos en la vista de asignación de la quinta historia de usuario,

Accedemos al listado de historias de usuario del manager1, e intentamos asignar la quinta a un Project, pero antes de hacer click en el botón de asignar, alteramos los siguientes campos (inspeccionando los elementos en el navegador):

- Project: cambiamos el valor del atributo “value” de una las opciones del select a los siguientes valores antes de seleccionarla:
  - 279 (Pertenece a un Project del manager1 ya publicado)
  - 330 (Ya está asignada a ese Project)
  - LOREM (para cubrir la rama en la que el framework detecta un fallo en el input)
- id (input oculto del formulario): lo cambiamos a 968 (corresponde a la relación entre el Project con code AAA-1234 y la UserStory con title “Test title”). Después de intentar esto, accedemos a la lista de historias de usuario del Project con code AAA-1234 y comprobamos que no ha cambiado.
- botón “Create”: cambiamos el valor del parámetro “userId” a los ids usados al principio de la grabación en la url

### **Delete.safe**

Con el manager1, accedemos a la url

<http://localhost:8082/Acme-SF-D04/manager/user-story-assign/delete?userId=957> (la id pertenece a una UserStory del manager1 que está publicada y que está asignada a varios Projects sin publicar). Esto es para asegurar que se puede acceder a esa vista mediante una petición GET cuando es una operación legal.

A continuación, entramos al listado de historias de usuario asignadas al Project con code NOP-0123 (no está publicado y no tiene ninguna historia de usuario asociada), y creamos 3 historias de usuario distintas.

Entramos en el formulario de desvinculación de historias de usuario de una de las 3, dejamos vacío el campo de “Project” y pulsamos el botón de desvincular.

Acto seguido, asignamos la primera a 3 Projects, y la segunda a 4 Projects.

Después, desvinculamos:

- La primera UserStory solo de 1 Project
- La segunda UserStory de varios Projects, dejándola vinculada solo a 2
- La tercera UserStory del Project al que está asociada (dejándola sin vincular a ningún Project)

## Delete.hack

Entramos en la url

`http://localhost:8082/Acme-SF-D04/manager/user-story-assign/delete?userId=[ID]`

Con las IDs:

- 927: es una UserStory del manager9
- 778: no es una UserStory

Con los usuarios:

- Sin iniciar sesión
- Con developer1: Rol incorrecto
- Con manager1: Rol correcto, Usuario incorrecto

En todos los casos, no estamos autorizados a acceder a la vista.

Para maximizar la cobertura de código, utilizamos el ID 778, que cubre la rama del `authorise` en la que el objeto obtenido con el id es nulo.

Con manager1, accedemos a la url

`http://localhost:8082/Acme-SF-D04/manager/user-story-assign/delete?userId=956.`

El id pertenece a una historia de usuario del manager1 que está publicada y solo está asignada a Projects publicados.

Después, accedemos con el manager1 a su lista de historias de usuario, asignamos la cuarta al Project con title "Project con link de test 12", y lo publicamos (tiene code NOP-0123).

Acto seguido, accedemos al listado de historias de usuario del Project con code AAA-1234 y comprobamos que tiene 2 historias de usuario.

Accedemos al listado de historias de usuario del manager1, e intentamos desvincular la cuarta de un Project, pero antes de hacer click en el botón de desvincular, alteramos los siguientes campos (inspeccionando los elementos en el navegador):

- Project: cambiamos el valor del atributo "value" de una las opciones del select a los siguientes valores antes de seleccionarla:
  - 321 (Pertenece al Project que hemos publicado)
  - 332 (No está asociada a este Project)
  - 778 (no es un Project)
  - LOREM (para cubrir la rama en la que el framework detecta un fallo en el input)
- id (input oculto del formulario): lo cambiamos a 968 (corresponde a la relación entre el Project con code AAA-1234 y una de sus historias de usuario). Después de intentar esto, accedemos a la lista de historias de usuario del Project con code AAA-1234 y comprobamos que sigue teniendo 2 historias de usuario.
- botón "Delete": cambiamos el valor del parámetro "userId" a los ids usados al principio de la grabación en la url.

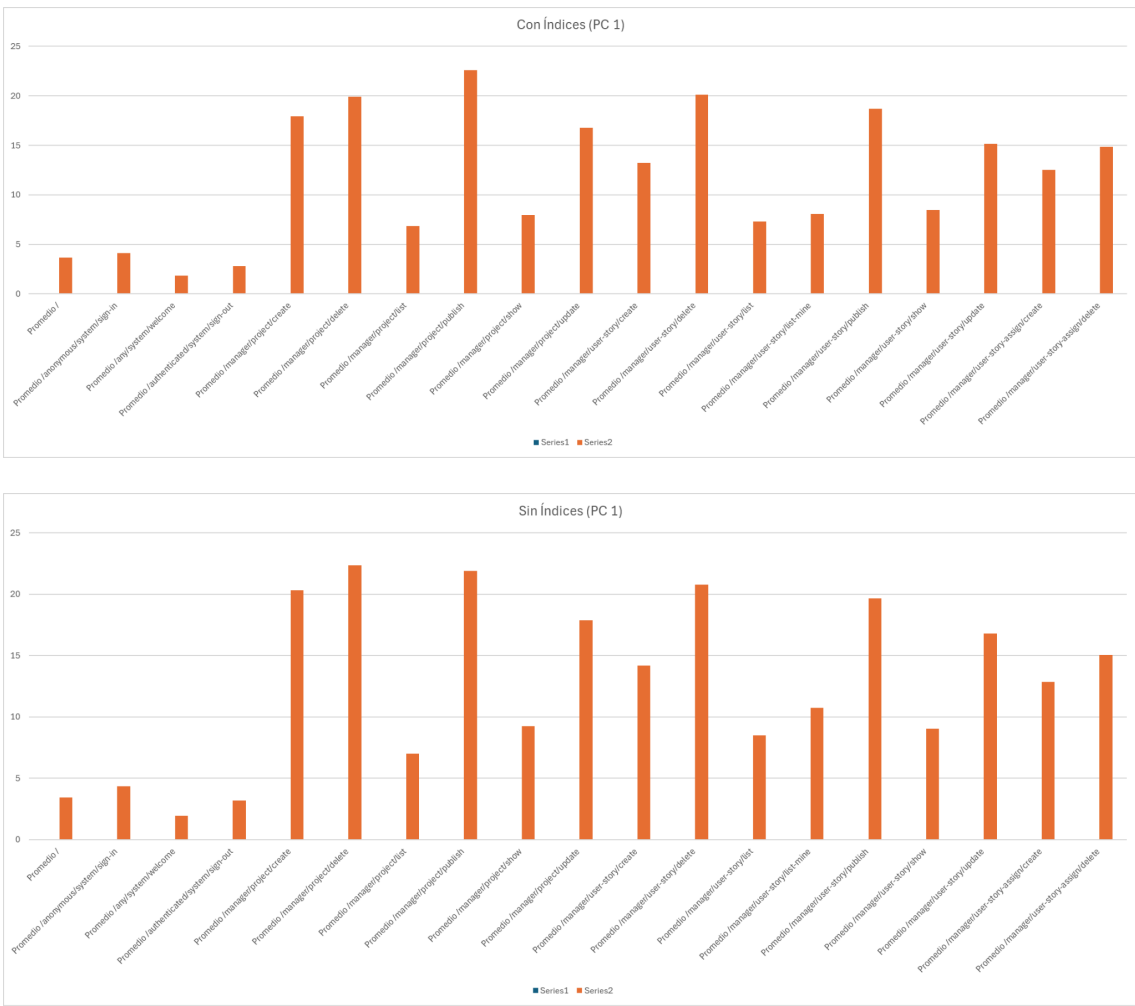
# Pruebas de Rendimiento

Para probar el rendimiento del sistema respecto a las pruebas he lanzado el replayer en 2 ordenadores personales distintos (en ambos PCs se han cerrado todos los programas que no sean necesarios para grabar las pruebas, para liberar todos los recursos posibles y maximizar el rendimiento). Para mejorar el rendimiento, he añadido índices a las entidades que se consultan en las pruebas grabadas. En ambos ordenadores he lanzado el replayer con y sin índices.

Los gráficos de tiempo obtenidos en cada ordenador son los siguientes:

## Efectos de índices en PC 1

Para el primer ordenador:



Las gráficas muestran que el tiempo de ejecución es similar en los 2 casos. Para comprobar si existe alguna diferencia real, observamos los intervalos de confianza y los resultados de la prueba z.

Las estadísticas para cada caso, y los intervalos de confianza son los siguientes:

<i>Before</i>				<i>After</i>		
Media	9,44481691			Media	10,0688265	
Error típico	0,31721973			Error típico	0,32884634	
Mediana	6,19745			Mediana	6,72205	
Moda	4,8433			Moda	1,3821	
Desviación e	9,91026105			Desviación e	10,2734881	
Varianza de	98,2132741			Varianza de	105,544559	
Curtosis	27,2272797			Curtosis	20,869766	
Coeficiente d	3,88154053			Coeficiente d	3,48464288	
Rango	114,8413			Rango	100,5545	
Mínimo	0,9165			Mínimo	0,9003	
Máximo	115,7578			Máximo	101,4548	
Suma	9218,1413			Suma	9827,1747	
Cuenta	976			Cuenta	976	
Nivel de cont	0,62251201			Nivel de cont	0,64532808	
Interval (ms)	8,8223049	10,0673289		Interval (ms)	9,42349846	10,7141546
Interval (s)	0,0088223	0,01006733		Interval (s)	0,0094235	0,01071415

El resultado de la prueba z es la tabla siguiente:

Prueba z para medias de dos muestras		
	<i>Before</i>	<i>After</i>
Media	9,44481691	10,0688265
Varianza (conocida)	98,2132741	105,544559
Observaciones	976	976
Diferencia hipotética de las medias	0	
z	-1,36571175	
P(Z<=z) una cola	0,08601473	
Valor crítico de z (una cola)	1,64485363	
Valor crítico de z (dos colas)	0,17202946	
Valor crítico de z (dos colas)	1,95996398	
	p-value	0,17202946



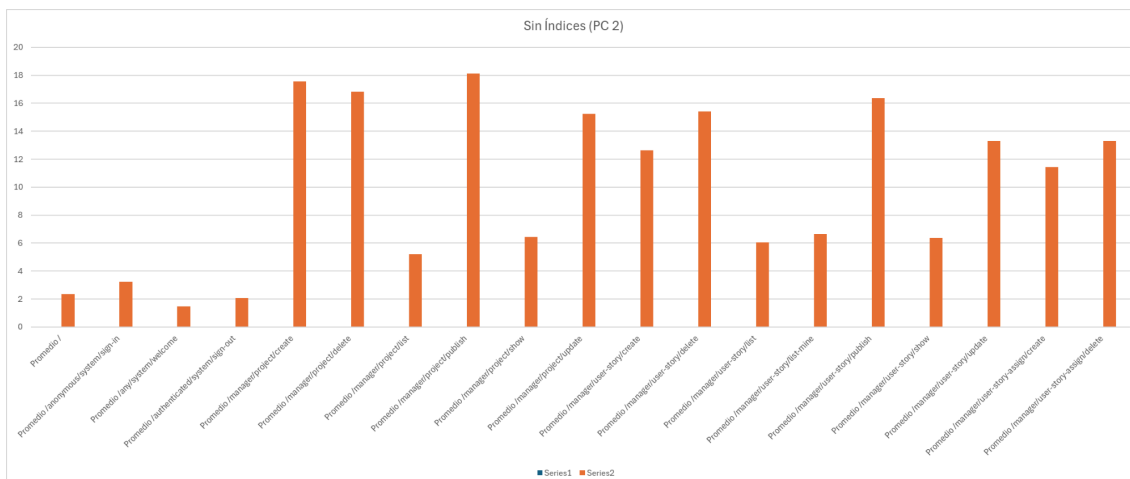
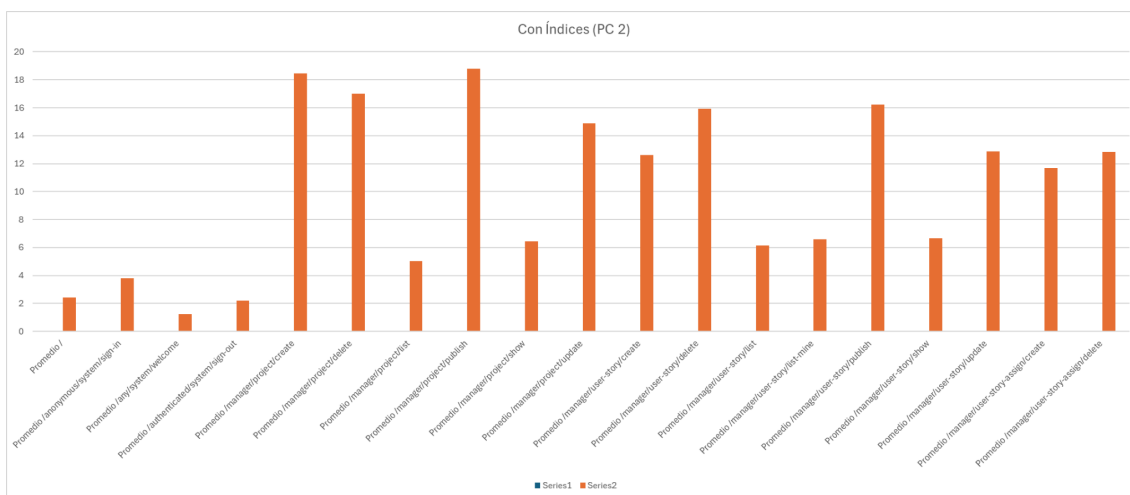
Estableciendo el contraste de hipótesis, tenemos la hipótesis nula ( $H_0$ ), y la hipótesis alternativa ( $H_1$ ) siguientes:

- $H_0$  : Introducir índices afecta al rendimiento de las pruebas en el PC 1
- $H_1$  : Introducir índices no influye en el rendimiento de las pruebas en el PC 1

Para un nivel de confianza del 95%, tenemos un nivel de significación ( $\alpha$ ) de 0.05. El p-value obtenido es de 0.17029. Como el p-value no se encuentra en el intervalo  $[0.00, \alpha]$ , no tenemos información suficiente para aceptar la hipótesis nula, por lo que podemos aceptar la hipótesis alternativa: introducir índices no influye significativamente en el rendimiento de las pruebas en el PC 1.

## Efectos de índices en PC 2

Para el segundo ordenador:



Las gráficas muestran que el tiempo de ejecución es similar en los 2 casos. Para comprobar si existe alguna diferencia real, observamos los intervalos de confianza y los resultados de la prueba z.

Las estadísticas para cada caso, y los intervalos de confianza son los siguientes:

Before				After		
Media	8,04468719			Media	8,13531137	
Error típico	0,27550337			Error típico	0,27027868	
Mediana	5,0163			Mediana	5,2008	
Moda	4,3933			Moda	1,0924	
Desviación e	8,60700046			Desviación e	8,44377576	
Varianza de	74,0804569			Varianza de	71,2973491	
Curtosis	26,18106			Curtosis	25,1104503	
Coeficiente d	3,78353016			Coeficiente d	3,6316322	
Rango	89,4215			Rango	90,7462	
Mínimo	0,738			Mínimo	0,7547	
Máximo	90,1595			Máximo	91,5009	
Suma	7851,6147			Suma	7940,0639	
Cuenta	976			Cuenta	976	
Nivel de cont	0,54064783			Nivel de cont	0,53039489	
Interval (ms)	7,50403936	8,58533502		Interval (ms)	7,60491648	8,66570626
Interval (s)	0,00750404	0,00858534		Interval (s)	0,00760492	0,00866571

El resultado de la prueba z es la tabla siguiente:

Prueba z para medias de dos muestras		
	Before	After
Media	8,04468719	8,13531137
Varianza (conocida)	74,0804569	71,2973491
Observaciones	976	976
Diferencia hipotética de las medias	0	
z	-0,23481181	
P(Z<=z) una cola	0,4071774	
Valor crítico de z (una cola)	1,64485363	
Valor crítico de z (dos colas)	0,8143548	
Valor crítico de z (dos colas)	1,95996398	
	p-value	0,8143548

Estableciendo el contraste de hipótesis, tenemos la hipótesis nula ( $H_0$ ), y la hipótesis alternativa ( $H_1$ ) siguientes:

- $H_0$  : Introducir índices afecta al rendimiento de las pruebas en el PC 2
- $H_1$  : Introducir índices no influye en el rendimiento de las pruebas en el PC 2

Para un nivel de confianza del 95%, tenemos un nivel de significación ( $\alpha$ ) de 0.05. El p-value obtenido es de 0.814. Como el p-value no se encuentra en el intervalo  $[0.00, \alpha)$ , no tenemos información suficiente para aceptar la hipótesis nula, por lo que podemos aceptar la hipótesis

alternativa: introducir índices no influye significativamente en el rendimiento de las pruebas en el PC 2.

## Comparación de rendimientos entre PCs

Para comprobar cual de los dos PCs es más potente, realizamos contrastes de hipótesis comparando los 2 PCs tanto con índices como sin índices.

### Comparación sin índices

Para comprobar si existe alguna diferencia real, observamos los intervalos de confianza y los resultados de la prueba z.

Las estadísticas para cada caso, y los intervalos de confianza son los siguientes:

PC 1				PC 2		
Media	9,44481691			Media	8,04468719	
Error típico	0,31721973			Error típico	0,27550337	
Mediana	6,19745			Mediana	5,0163	
Moda	4,8433			Moda	4,3933	
Desviación e	9,91026105			Desviación e	8,60700046	
Varianza de	98,2132741			Varianza de	74,0804569	
Curtosis	27,2272797			Curtosis	26,18106	
Coeficiente d	3,88154053			Coeficiente d	3,78353016	
Rango	114,8413			Rango	89,4215	
Mínimo	0,9165			Mínimo	0,738	
Máximo	115,7578			Máximo	90,1595	
Suma	9218,1413			Suma	7851,6147	
Cuenta	976			Cuenta	976	
Nivel de confi	0,62251201			Nivel de confi	0,54064783	
Interval (ms)	8,8223049	10,0673289		Interval (ms)	7,50403936	8,58533502
Interval (s)	0,0088223	0,01006733		Interval (s)	0,00750404	0,00858534

El resultado de la prueba z es la tabla siguiente:

Prueba z para medias de dos muestras		
	PC 1	PC2
Media	9,44481691	8,04468719
Varianza (conocida)	98,2132741	74,0804569
Observaciones	976	976
Diferencia hipotética de las medias	0	
z	3,33241011	
P(Z<=z) una cola	0,00043049	
Valor crítico de z (una cola)	1,64485363	
Valor crítico de z (dos colas)	0,00086097	
Valor crítico de z (dos colas)	1,95996398	
	p-value	0,00086097

Estableciendo el contraste de hipótesis, tenemos la hipótesis nula ( $H_0$ ), y la hipótesis alternativa ( $H_1$ ) siguientes:

- $H_0$  : Sin índices, uno de los ordenadores es más eficiente que el otro
- $H_1$  : Sin índices, los ordenadores son igual de eficientes

Para un nivel de confianza del 95%, tenemos un nivel de significación ( $\alpha$ ) de 0.05. El p-value obtenido es de 0,00086097.

Como el p-value se encuentra en el intervalo  $[0.00, \alpha)$ , podemos aceptar la hipótesis nula: uno de los ordenadores es más potente

Para ver cual de los 2 ordenadores es más eficiente sin índices, comparamos las medias de tiempo de ejecución en cada ordenador.

PC 1	PC 2	Diferencia
9,4448169057377 ms	8,04468719262295 ms	1,40012971 ms

Podemos comprobar que, de media, el ordenador 1 tarda de media 1,4 ms más en responder a las peticiones que el ordenador 2. Es decir, el primer ordenador es un 17.4% más lento cuando no se introducen índices.

### Comparación con índices

Para comprobar si existe alguna diferencia real, observamos los intervalos de confianza y los resultados de la prueba z.

Las estadísticas para cada caso, y los intervalos de confianza son los siguientes:

PC 1				PC 2		
Media	10,0688265			Media	8,13531137	
Error típico	0,32884634			Error típico	0,27027868	
Mediana	6,72205			Mediana	5,2008	
Moda	1,3821			Moda	1,0924	
Desviación e	10,2734881			Desviación e	8,44377576	
Varianza de	105,544559			Varianza de	71,2973491	
Curtosis	20,869766			Curtosis	25,1104503	
Coeficiente d	3,48464288			Coeficiente d	3,6316322	
Rango	100,5545			Rango	90,7462	
Mínimo	0,9003			Mínimo	0,7547	
Máximo	101,4548			Máximo	91,5009	
Suma	9827,1747			Suma	7940,0639	
Cuenta	976			Cuenta	976	
Nivel de confi	0,64532808			Nivel de confi	0,53039489	
Interval (ms)	9,42349846	10,7141546		Interval (ms)	7,60491648	8,66570626
Interval (s)	0,0094235	0,01071415		Interval (s)	0,00760492	0,00866571

El resultado de la prueba z es la tabla siguiente:

Prueba z para medias de dos muestras		
	PC 1	PC 2
Media	10,0688265	8,13531137
Varianza (conocida)	105,544559	71,2973491
Observaciones	976	976
Diferencia hipotética de las medias	0	
z	4,54234269	
P(Z<=z) una cola	2,7816E-06	
Valor crítico de z (una cola)	1,64485363	
Valor crítico de z (dos colas)	5,5633E-06	
Valor crítico de z (dos colas)	1,95996398	
	p-value	5,5633E-06

Estableciendo el contraste de hipótesis, tenemos la hipótesis nula ( $H_0$ ), y la hipótesis alternativa ( $H_1$ ) siguientes:

- $H_0$  : Con índices, uno de los ordenadores es más eficiente que el otro
- $H_1$  : Con índices, los ordenadores son igual de eficientes

Para un nivel de confianza del 95%, tenemos un nivel de significación ( $\alpha$ ) de 0.05. El p-value obtenido es de  $5.5633 \cdot 10^{-6}$ .

Como el p-value se encuentra en el intervalo  $[0.00, \alpha)$ , podemos aceptar la hipótesis nula: uno de los ordenadores es más potente

Para ver cual de los 2 ordenadores es más eficiente con índices, comparamos las medias de tiempo de ejecución en cada ordenador.

PC 1	PC 2	Diferencia
10,0688265368852 ms	8,13531137295081 ms	1,93351516 ms

Podemos comprobar que, de media, el ordenador 1 tarda de media 1.93 ms más en responder a las peticiones que el ordenador 2. Es decir, el primer ordenador es un 23,767% más lento cuando se introducen índices.

Si nos basamos en estas 2 últimas comparaciones, llegamos a la conclusión de que el segundo ordenador es más potente que el primero.

# Conclusión

En cuanto a los tests, puedo decir que he intentado hacerlos lo más rigurosos posibles, alcanzando un coverage muy satisfactorio.

En cuanto al rendimiento, se han demostrado 2 cosas:

- 1. Introducir índices no tiene un efecto en el rendimiento
- 2. El primer ordenador es menos eficiente para la ejecución de las pruebas grabadas

# Bibliografía

No aplica.