

INFORME DE PRUEBAS



**Grado en Ingeniería Informática – Ingeniería del
Software**

Diseño y Pruebas 2
Curso 2023-2024

Antonio Daniel Porcar Aragón

Índice

1. Información general del proyecto	3
2. Resumen del ejecutivo	4
3. Tabla de revisión	5
4. Introducción	6
5. Contenidos	7
5.1 Pruebas funcionales	7
5.2 Pruebas de rendimiento	14
6. Conclusiones	17
7. Bibliografía	18

1. Información general del proyecto

NOMBRE DEL PROYECTO			Acme-SF	
PARTICIPANTES				
Nombre	Email	Rol	Nombre de usuario	Foto
Antonio Daniel Porcar Aragón	antporara@alum.us.es	Project Manager, Desarrollador	antporara	
Francisco Miguel Jiménez Morales	frajimmor2@alum.us.es	Tester, Desarrollador	frajimmor2	
Javier Santos Martín	javsanmar5@alum.us.es	Desarrollador, Secretario	javsanmar5	
Javier Ruiz Garrido	javruigar2@alum.us.es	Analista, Desarrollador	Javiruizg	
José García de Tejada Delgado	josgardel8@alum.us.es	Operador, Desarrollador	JoseGTD	
Interesados				
Francisco Miguel Jiménez Morales, Javier Ruiz Garrido, José García de Tejada Delgado, Javier Santos Martín, Antonio Daniel Porcar Aragón and José González Enriquez (the professor).				
Fecha de inicio	Fecha esperada de completado	Entregables	Fecha del documento	
12/02/2024	27/05/2024	4	24/05/2024	

2. Resumen del ejecutivo

Las pruebas son un elemento fundamental en el desarrollo y mantenimiento software, ayudando a los miembros del equipo de desarrollo a comprobar el adecuado funcionamiento del sistema y la detección de errores.

En este documento se refleja un informe sobre el que ha realizado el estudiante 1, mediante los capítulos de pruebas funcionales y pruebas de rendimiento; además se analizará la cobertura de dichas pruebas.

3. Tabla de revisión

Versión	Descripción	Fecha
v1.0	Versión inicial	24/05/2024
v2.0	Versión final	25/05/2024

4. Introducción

En el primer apartado, sobre pruebas funcionales se tratarán los casos de prueba implementados agrupados por implementación, incluyéndose su efectividad a la hora de detectar errores.

En el segundo apartado se trata el rendimiento del sistema, mediante gráficos y análisis estadístico de los tiempos en que se sirven las peticiones, tomando intervalos de confianza del 95%, previos y posteriores a la refactorización con índices; junto con un contraste de hipótesis resultante al análisis.

5. Contenidos

5.1 Pruebas funcionales

A continuación, se indicarán de manera resumida los casos de prueba implementados, indicándose una explicación adicional en las pruebas que permitieron detectar errores en la implementación de las funcionalidades. Considérese que en aquellos casos en que la prueba falló por un error encontrado o se encontró un resultado inesperado durante la realización de la misma, se arregló dicho error y se repitió la prueba, asegurándonos así de su adecuado funcionamiento.

Project

- **list-mine.safe:**
Se inicia sesión como manager1 y se accede su listado de proyectos. No se encontraron errores.
- **list-mine.hack:**
Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede al listado. En la otra se inicia sesión como otro rol y se intenta acceder a la url en la que se encuentra el manager1, obteniendo error 500 al no estar autorizado. No se encontraron errores.
- **create.safe:**
Se inicia sesión como manager1 y se accede al formulario de creación de proyecto. Se prueban todas las casuísticas posibles para garantizar el máximo de cobertura, comprobándose además que al mandar formulario vacío o datos erróneos se obtiene un mensaje de error. Se detectó un error en la comprobación de la longitud máxima del link.
- **create2.safe:**
Se realiza el mismo proceso que el detallado en create.safe pero evaluando solo la casuística del link, tanto en cobertura como datos erróneos.
- **create.hack:**
Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede al formulario de creación de proyecto. En la otra se inicia sesión como otro rol y se intenta acceder a la url en la que se encuentra el manager1, obteniendo error 500 al no estar autorizado. No se encontraron errores.
- **update.safe:**
Se inicia sesión como manager 1 y se accede al formulario de un proyecto no publicado. Se prueban todas las casuísticas posibles para garantizar el máximo de cobertura, comprobándose además que al mandar formulario vacío o datos erróneos se obtiene un mensaje de error. No se encontraron errores.

- **update-right-role-wrong-user.hack:**
Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede al formulario de edición de proyecto. En la otra se inicia sesión como manager2 y se intenta acceder a la url en la que se encuentra el manager1, obteniendo error 500 al no estar autorizado. No se encontraron errores.
- **update-wrong-action.hack:**
Se inicia sesión como manager1 y se intenta acceder al formulario de edición de un proyecto ya publicado introduciéndose en su url: "manager/project/update?id=id_proyecto_publicado", obteniendo error 500 al no estar autorizado. No se encontraron errores.
- **update-wrong-role.hack:**
Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede al formulario de edición de proyecto. En la otra se inicia sesión como otro rol y se intenta acceder a la url en la que se encuentra el manager1, obteniendo error 500 al no estar autorizado. No se encontraron errores.
- **delete.safe:**
Se inicia sesión como manager1, se accede a un proyecto no publicado y se borra mediante el botón que lo facilita. No se encontraron errores.
- **delete-right-role-wrong-user.hack:**
Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede a los detalles de un proyecto, para conocer su id. En la otra se inicia sesión como manager2 y se intenta acceder a la url: "manager/project/delete?id=id_proyecto_manager1", obteniendo error 500 al no estar autorizado. No se encontraron errores.
- **delete-wrong-action.hack:**
Se inicia sesión como manager1 y se intenta borrar un proyecto ya publicado introduciéndose en su url: "manager/project/delete?id=id_proyecto_publicado", obteniendo error 500 al no estar autorizado. No se encontraron errores.
- **delete-wrong-role.hack:**
Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede a los detalles de un proyecto, para conocer su id. En la otra se inicia sesión como otro rol y se intenta acceder a la url: "manager/project/delete?id=id_proyecto", obteniendo error 500 al no estar autorizado. No se encontraron errores.
- **show.safe:**
Se inicia sesión como manager1 y se accede a los detalles de un proyecto. No se encontraron errores.
- **show-right-role-wrong-user.hack:**
Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede a los detalles de un proyecto, para conocer su id. En la otra se inicia sesión como manager2 y se intenta acceder a la url en la que se encuentra el

manager1, obteniendo error 500 al no estar autorizado. No se encontraron errores.

- **show-wrong-role.hack:**

Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede a los detalles de un proyecto. En la otra se inicia sesión como otro rol y se intenta acceder a la url en la que se encuentra el manager1, obteniendo error 500 al no estar autorizado. No se encontraron errores.

- **publish.safe:**

Se inicia sesión como manager1 y se accede a los detalles de un proyecto, se verifica que contenga mínimo una historia de usuario y que todas las que contenga estén publicadas, además de que no contenga la indicación de errores fatales; y se publica utilizando el botón proporcionado, tras esto accedemos a los detalles de dicho proyecto y su formulario solo ha de permitir visualizar. No se encontraron errores.

- **publish-right-role-wrong-user.hack:**

Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede a los detalles de un proyecto, para conocer el id.

En la otra se inicia sesión como manager2 y se intenta acceder a la url: "manager/project/publish?id=id_proyecto", obteniendo error 500 al no estar autorizado. No se encontraron errores.

- **publish-wrong-role.hack:**

Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede a los detalles de un proyecto, para conocer el id.

En la otra se inicia sesión como otro rol y se intenta acceder a la url: "manager/project/publish?id=id_proyecto", obteniendo error 500 al no estar autorizado. No se encontraron errores.

UserStory

- **list.safe:**

Se inicia sesión como manager1 y se accede su listado de historias de usuario. No se encontraron errores.

- **list.hack:**

Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede al listado. En la otra se inicia sesión como otro rol y se intenta acceder a la url en la que se encuentra el manager1, obteniendo error 500 al no estar autorizado. No se encontraron errores.

- **list-by-project.safe:**

Se inicia sesión como manager1 y se accede su listado de historias de usuario desde dentro de los detalles de un proyecto. No se encontraron errores.

- **list-by-project-right-role-wrong-user.hack:**

Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede al listado. En la otra se inicia sesión como manager2 y se intenta acceder a la url en la que se encuentra el manager1, obteniendo error 500 al no estar autorizado. No se encontraron errores.

- **list-by-project-wrong-role.hack:**

Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede al listado. En la otra se inicia sesión como otro rol y se intenta acceder a la url en la que se encuentra el manager1, obteniendo error 500 al no estar autorizado. No se encontraron errores.

- **create.safe:**

Se inicia sesión como manager1 y se accede al formulario de creación de historia de usuario. Se prueban todas las casuísticas posibles para garantizar el máximo de cobertura, comprobándose además que al mandar formulario vacío o datos erróneos se obtiene un mensaje de error. Se detectó un error en la comprobación de la longitud máxima del link.

- **create2.safe:**

Se realiza el mismo proceso que el detallado en create.safe pero evaluando solo la casuística del link, tanto en cobertura como datos erróneos.

- **create.hack:**

Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede al formulario de creación de historia de usuario. En la otra se inicia sesión como otro rol y se intenta acceder a la url en la que se encuentra el manager1, obteniendo error 500 al no estar autorizado. No se encontraron errores.

- **update.safe:**

Se inicia sesión como manager 1 y se accede al formulario de una historia de usuario no publicada. Se prueban todas las casuísticas posibles para garantizar el máximo de cobertura, comprobándose además que al mandar formulario vacío o datos erróneos se obtiene un mensaje de error. No se encontraron errores.

- **update-right-role-wrong-user.hack:**

Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede al formulario de edición de historia de usuario. En la otra se inicia sesión como manager2 y se intenta acceder a la url en la que se encuentra el manager1, obteniendo error 500 al no estar autorizado. No se encontraron errores.

- **update-wrong-action.hack:**

Se inicia sesión como manager1 y se intenta acceder al formulario de edición de un proyecto ya publicado introduciéndose en su url: "manager/user-story/update?id=id_historia_de_usuario_publicada", obteniendo error 500 al no estar autorizado. No se encontraron errores.

- **update-wrong-role.hack:**

Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede al formulario de edición de historia de usuario. En la otra se inicia sesión como otro rol y se intenta acceder a la url en la que se encuentra el manager1, obteniendo error 500 al no estar autorizado. No se encontraron errores.

- **delete.safe:**

Se inicia sesión como manager1, se accede a una historia de usuario no publicada y se borra mediante el botón que lo facilita. No se encontraron errores.

- **delete-right-role-wrong-user.hack:**

Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede a los detalles de una historia de usuario, para conocer su id.

En la otra se inicia sesión como manager2 y se intenta acceder a la url: "manager/user-story/delete?id=id_historia_de_usuario_manager1", obteniendo error 500 al no estar autorizado. No se encontraron errores.

- **delete-wrong-action.hack:**

Se inicia sesión como manager1 y se intenta borrar una historia de usuario ya publicada introduciéndose en su url:

"manager/user-story/delete?id=id_historia_de_usuario_publicada", obteniendo error 500 al no estar autorizado. No se encontraron errores

- **delete-wrong-role.hack:**

Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede a los detalles de una historia de usuario, para conocer su id.

En la otra se inicia sesión como otro rol y se intenta acceder a la url: "manager/user-story/delete?id=id_proyecto", obteniendo error 500 al no estar autorizado. No se encontraron errores.

- **show.safe:**

Se inicia sesión como manager1 y se accede a los detalles de una historia de usuario. No se encontraron errores.

- **show-right-role-wrong-user.hack:**

Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede a los detalles de una historia de usuario, para conocer su id. En la otra se inicia sesión como manager2 y se intenta acceder a la url en la que se encuentra el manager1, obteniendo error 500 al no estar autorizado. No se encontraron errores.

- **show-wrong-role.hack:**

Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede a los detalles de una historia de usuario. En la otra se inicia sesión como otro rol y se intenta acceder a la url en la que se encuentra el manager1, obteniendo error 500 al no estar autorizado. No se encontraron errores.

- **publish.safe:**

Se inicia sesión como manager1, se accede a los detalles de una historia de usuario y se publica con el botón que se facilita para ello. No se encontraron errores.

- **publish-right-role-wrong-user.hack:**

Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede a los detalles de una historia de usuario, para conocer el id.

En la otra se inicia sesión como manager2 y se intenta acceder a la url: "manager/user-story/publish?id=id_historia_de_usuario", obteniendo error 500 al no estar autorizado. No se encontraron errores.

- **publish-wrong-role.hack:**

Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede a los detalles de una historia de usuario, para conocer el id.

En la otra se inicia sesión como otro rol y se intenta acceder a la url: "manager/user-story/publish?id=id_historia_de_usuario", obteniendo error 500 al no estar autorizado. No se encontraron errores.

ProjectUserStoryLink

- **create.safe:**

Se inicia sesión como manager1 y se accede al formulario de creación de la asignación. Se verifica que saltan errores al enviar formulario vacío o uno de los campos vacíos, además de tratar de repetir una asignación existente, y se crean casos válidos. Se encontró un error a la hora de mandar el formulario con proyecto pero sin historia de usuario y viceversa.

- **create.hack:**

Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede al formulario de creación de la asignación. En la otra se inicia sesión como otro rol y se intenta acceder a la url en la que se encuentra el manager1, obteniendo error 500 al no estar autorizado. No se encontraron errores.

- **delete.safe:**

Se inicia sesión como manager1, se accede a una asignación con proyecto no publicado y se borra mediante el botón que lo facilita. No se encontraron errores.

- **delete-right-role-wrong-user.hack:**

Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede a los detalles de una asignación, para conocer su id.

En la otra se inicia sesión como manager2 y se intenta acceder a la url: "manager/project-user-story-link/delete?id=id_asignacion_manager1", obteniendo error 500 al no estar autorizado. No se encontraron errores.

- **delete-wrong-role.hack:**

Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede a los detalles de una asignación, para conocer su id.

En la otra se inicia sesión como otro rol y se intenta acceder a la url: "manager/project-user-story-link/delete?id=id_asignacion_manager1", obteniendo error 500 al no estar autorizado. No se encontraron errores.

- **delete-wrong-action:**

Se inicia sesión como manager1 y se intenta borrar una asignación con un proyecto publicado introduciéndose en su url:

"manager/project-user-story-link/delete?id=id_asignacion_con_proyecto_publicado", obteniendo error 500 al no estar autorizado. No se encontraron errores.

- **list.safe:**

Se inicia sesión como manager1 y se accede su listado de asignaciones. No se encontraron errores.

- **list.hack:**

Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede al listado. En la otra se inicia sesión como otro rol y se intenta acceder a la url en la que se encuentra el manager1, obteniendo error 500 al no estar autorizado. No se encontraron errores.

- **show.safe:**

Se inicia sesión como manager1 y se accede a los detalles de una asignación. No se encontraron errores.

- **show-right-role-wrong-user.hack:**

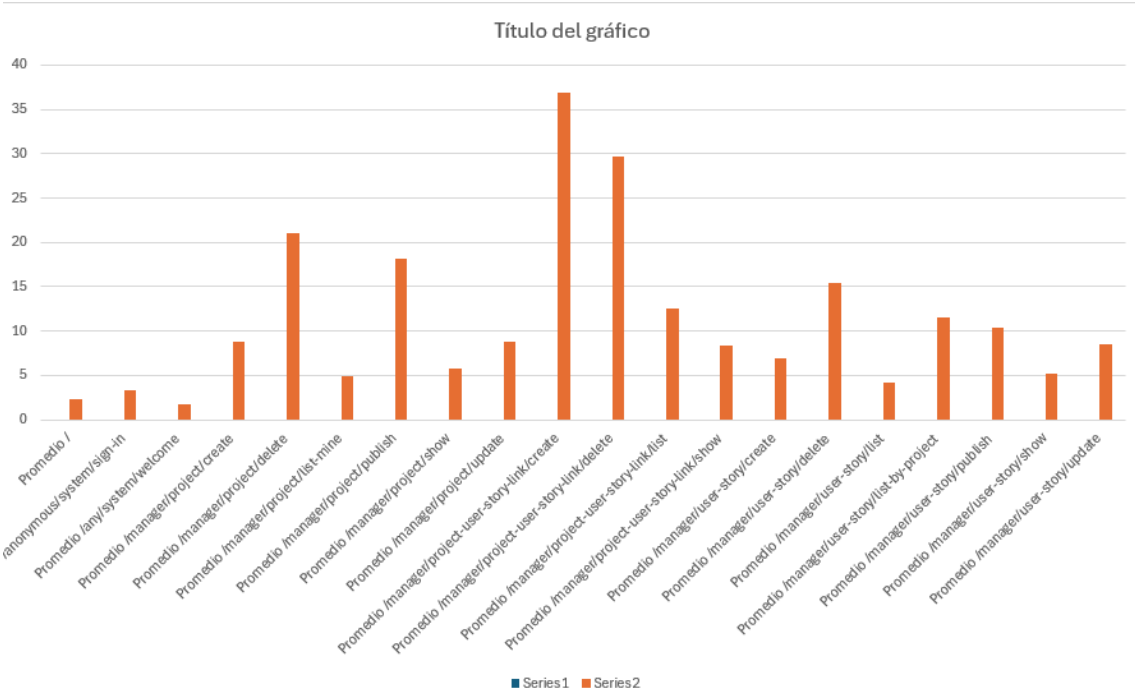
Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede a los detalles de una asignación, para conocer su id. En la otra se inicia sesión como manager2 y se intenta acceder a la url en la que se encuentra el manager1, obteniendo error 500 al no estar autorizado. No se encontraron errores.

- **show-wrong-role.hack:**

Se abren dos pestañas, una pública y otra en modo incógnito, en la primera se inicia sesión como manager1 y se accede a los detalles de una asignación, para conocer su id. En la otra se inicia sesión como otro rol y se intenta acceder a la url en la que se encuentra el manager1, obteniendo error 500 al no estar autorizado. No se encontraron errores.

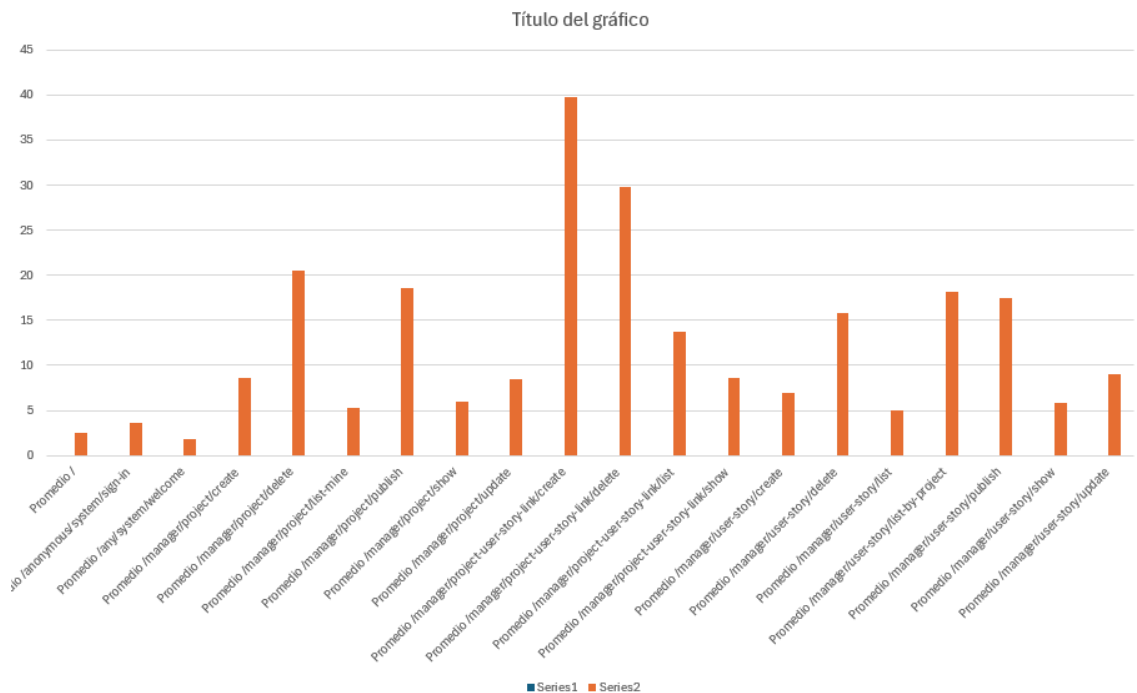
5.2 Pruebas de rendimiento

Pruebas previas a la refactorización con índices



Columna1				
Media	5,55315739	Interval (ms)	4,9492149	6,15709988
Error típico	0,307708429	Interval (s)	0,00494921	0,0061571
Mediana	3,31805			
Moda	5,1304			
Desviación estándar	9,055206076			
Varianza de la muestra	81,99675709			
Curtosis	44,11867897			
Coeficiente de asimetría	5,837692017			
Rango	113,2895			
Mínimo	0,8755			
Máximo	114,165			
Suma	4809,0343			
Cuenta	866			
Nivel de confianza(95,0%)	0,603942493			

Pruebas posteriores a la refactorización con índices



Columna1				
Media	1,369429477	Interval (ms)	1,30727048	1,43158847
Error típico	0,031710486	Interval (s)	0,00130727	0,00143159
Mediana	0,8681			
Moda	0,8107			
Desviación estándar	3,158816428			
Varianza de la muestra	9,978121228			
Curtosis	362,0543907			
Coficiente de asimetría	16,68017431			
Rango	98,4214			
Mínimo	0,6193			
Máximo	99,0407			
Suma	13588,8487			
Cuenta	9923			
Nivel de confianza(95,0%)	0,062158992			

Se puede comprobar visualizando ambas gráficas que las peticiones que más tiempo requieren son la creación de la asignación de proyecto a historia de usuario y su borrado.

También que, de manera global antes de refactorizar, las peticiones se tardan más en servir en comparación a la posterior refactorización, en la cual como media se tarda bastante menos, pero aquellas que más se tardaba en el caso anterior se tarda algo más.

Los intervalos de confianza con un nivel de confianza del 95% son de (4.95, 6.15) previos a la refactorización, y de (1.3,1.4) posteriores, con lo cual con esa bajada de valor del parámetro podemos ver una mejora significativa en el rendimiento, y al haber una variabilidad tan grande de uno frente a otro, que el sistema es más consistente en su rendimiento.

Hipótesis de contraste

Prueba z para medias de dos muestras		
	<i>Before</i>	<i>After</i>
Media	5,55315739	5,942765704
Varianza (conocida)	81,99675709	9,978121228
Observaciones	866	866
Diferencia hipotética de las medias	0	
z	-1,195508177	
P(Z<=z) una cola	0,115944272	
Valor crítico de z (una cola)	1,644853627	
Valor crítico de z (dos colas)	0,231888543	
Valor crítico de z (dos colas)	1,959963985	

Con estos valores, tomados con un 95% de nivel de confianza, observamos que el valor de P se encuentra por encima de 0,05.

Por lo cual podemos concluir que a la hora de comparar, tras la refactorización de los índices hace el sistema más eficiente en cuanto al tiempo que provee las peticiones de manera individual, pero de manera global viene a surgir el mismo efecto.

6. Conclusiones

En el informe se recogen todas las pruebas realizadas por el estudiante 1, las cuales han servido para detectar algún error en el código y calcular el adecuado funcionamiento; y poder calcular el rendimiento, pudiendo sacar así conclusiones sobre la eficiencia de nuestro sistema sirviendo las peticiones.

7. Bibliografía

Intencionalmente en blanco.