Escuela Técnica Superior de Ingeniería Informática

Testing report D04

Group C2.020 | Diseño y Pruebas II | 06/07/2024

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, asi 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

Miembros:

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

Repositorio de Github: https://github.com/DP2-C1-020/Acme-SF-D04

Contenido

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 seria 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 del analista 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.

Pruebas de Hacking:

- Intentar acceder a la URL con un rol incorrecto (GET hacking).

OPERACIONES "SHOW"

Pruebas Seguras:

- Seleccionar varios proyectos o historias de usuario de una lista para ver sus detalles.

Pruebas de Hacking:

- Intentar acceder a los detalles de un proyecto o historia de usuario con un rol incorrecto o con un gerente diferente (GET hacking).
- Intentar acceder a un proyecto o historia de usuario inexistente con un identificador inválido (GET hacking).

OPERACIONES "CREATE"

Pruebas Seguras:

- Intentar crear un nuevo proyecto o historia de usuario.
- Verificar que el sistema rechace datos inválidos para cada atributo.
- Verificar que el sistema acepte datos válidos para cada atributo.

Pruebas de Hacking:

- No se encontraron formas de hacking.

OPERACIONES "UPDATE"

Pruebas Seguras:

- Actualizar un proyecto o historia de usuario existente.
- Verificar que el sistema rechace datos inválidos para cada atributo.
- Verificar que el sistema acepte datos válidos para cada atributo.

Pruebas de Hacking:

- Probar URLs de forma maliciosa para ver si se encuentran vulnerabilidades (POST hacking).
- Cambiar datos mediante la herramienta inspeccionar del navegador, y posteriormente intentar actualizarlos mediante POST hacking.

OPERACIONES "DELETE"

Pruebas Seguras:

- Intentar eliminar proyectos o historias de usuario, considerando restricciones como elementos dependientes o estados de publicación.
- Eliminar correctamente un proyecto o historia de usuario cuando es posible.

Pruebas de Hacking:

- Intentar acceder al borrado de un proyecto sin tener los permisos de este mediante URLs (POST hacking)

OPERACIONES "PUBLISH"

Pruebas Seguras:

- Intentar publicar proyectos o historias de usuario, considerando condiciones previas necesarias para la publicación.
- Verificar el proceso de actualización de atributos durante la publicación.
- Intentar publicar con errores fatales.

Pruebas de Hacking:

- Intentar publicar un proyecto o historias de usuario sin tener los permisos de este mediante URLs (POST hacking)

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.

Pruebas de Hacking:

- Intentar acceder a las páginas de vinculación y desvinculación con un rol incorrecto o con un gerente diferente (GET hacking).

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

Ruta de la feature	X.safe	Y.hack	Efectividad	Coverage
/any/...	List-project Show-project	Show-project	1.0 Baja, No se encontraron fallos ni vulnerabilidades.	94.4%
/manager/Project/...	Créate Delete List-mine Show Update Publish	Delete List-mine Show Update Publish	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 condiciones provocaba que al borrar un proyecto no se borraran algunas historias de usuario. 2.0 Alta, Solución de errores al meter una cantidad muy grande de cualquier divisa.	93.6%
/manager/Project-user-story/...	Link Unlink	Link Unlink	1.0 Alta, encontrado bug que bajo ciertos datos de prueba las relaciones en las que el proyecto era nulo ocasionaban un panic. 2.0 Alta, se refactorizó el código de validaciones debido a restricciones innecesarias	91.2%
/manager/User-stories/...	Créate Delete List-mine List-for-project Show Update Publish	Delete List-mine Show Update Publish	1.0 Media, bug menor encontrado que hacía comprobaciones redundantes al borrar una historia de usuario bajo ciertas condiciones.	86.7%

			2.0 Baja, no se detectaron nuevos errores.	

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	93,6 %	1.194	81	1.275
ManagerProjectController.java	100,0 %	36	0	36
ManagerProjectCreateService.java	92,9 %	210	16	226
ManagerProjectDeleteService.java	92,2 %	226	19	245
ManagerProjectListMineService.java	94,3 %	66	4	70
ManagerProjectPublishService.java	93,9 %	291	19	310
ManagerProjectShowService.java	96,2 %	101	4	105
ManagerProjectUpdateService.java	93,3 %	264	19	283

Como se puede observar en la imagen la cobertura es de un 93.6%, 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), 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 creación de proyectos:

```
48        project = this.repository.findOneProjectById(masterId);
49        manager = project == null ? null : project.getManager();
50        status = project != null && project.isDraftMode() && super.getRequest().getPrincipal().hasRole(manager);
51
```

Las líneas 49 y 50 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 creación 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	86,7 %	846	130	976
ManagerUserStoryController.java	100,0 %	42	0	42
ManagerUserStoryCreateService.java	89,0 %	130	16	146
ManagerUserStoryDeleteService.java	65,1 %	114	61	175
ManagerUserStoryListForProjectServic	92,1 %	82	7	89
ManagerUserStoryListMineService.jav	92,6 %	50	4	54
ManagerUserStoryPublishService.java	89,4 %	160	19	179
ManagerUserStoryShowService.java	96,5 %	111	4	115
ManagerUserStoryUpdateService.java	89,2 %	157	19	176

Como se puede observar en la imagen la cobertura es de un 86.7%, 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), 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	91,2 %	372	36	408
ManagerProjectUserStoryController.java	100,0 %	14	0	14
ManagerProjectUserStoryCreateService.java	90,8 %	178	18	196
ManagerProjectUserStoryDeleteService.java	90,9 %	180	18	198

Como se puede observar en la imagen la cobertura es de un 91.2%, 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) , 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: 100% coverage (¡en este caso como no hay ningún assert!=null, ninguna internalización ni condición que sabemos no va a comprobarse nunca en caso de que la aplicación funcione bien ni un condicional if, el coverage es completo)

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
acme.features.manager.managerDashboard	100,0 %	497	0	497
ManagerDashboardController.java	100,0 %	9	0	9
ManagerDashboardShowService.java	100,0 %	488	0	488

Recubrimiento en los servicios relacionados con los proyectos para cualquier usuario: 94.4% coverage, no llegando al 100% por los motivos anteriormente mencionados.

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

Análisis de desempeño:

En este apartado veremos las estadísticas propias de la aplicación en la 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 2.0 del entregable, y por último comprobaremos que los dos contrastes de hipótesis coinciden:

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

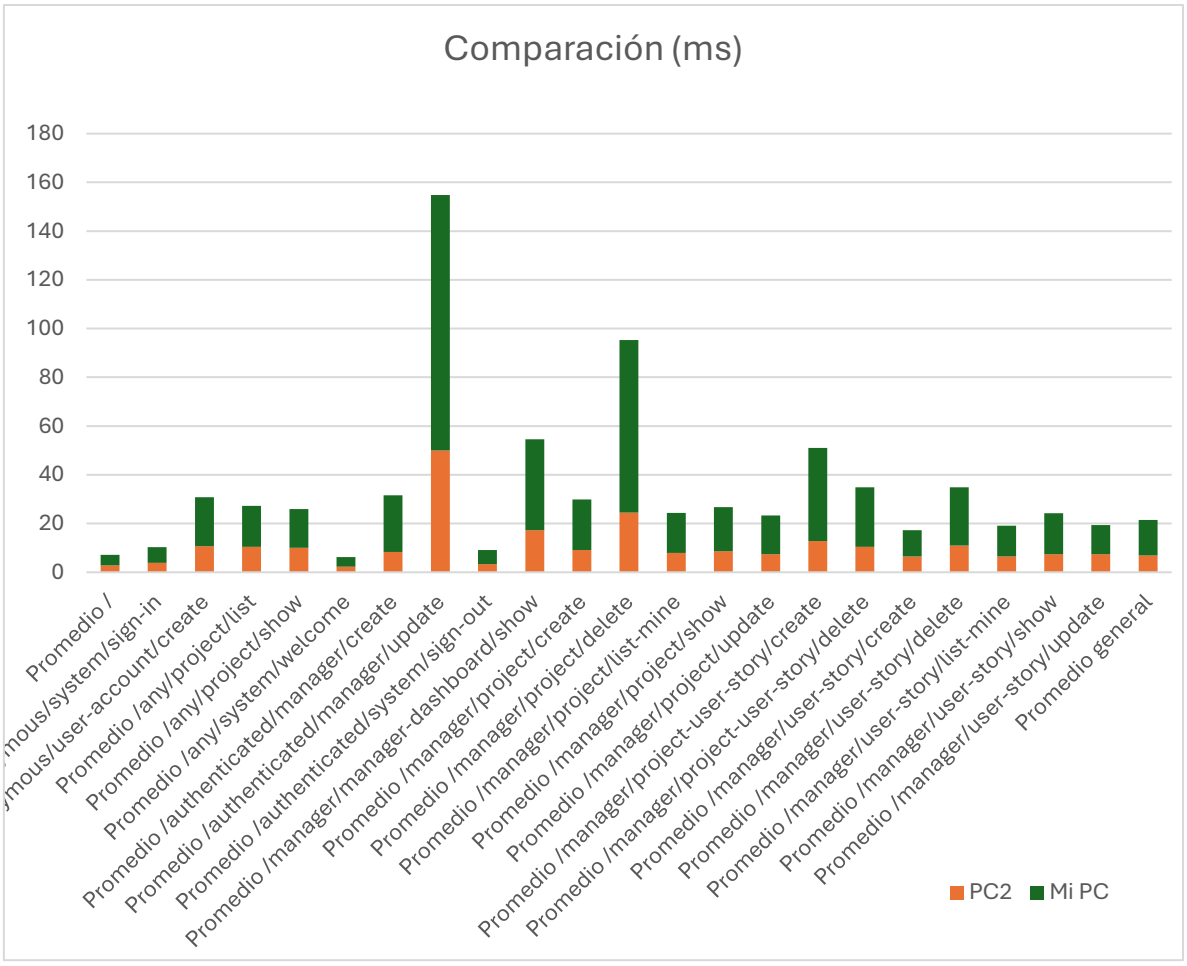
STATISTICS PC2				
Media	7,032846065	Interval(ms)	6,205195166	7,860496964
Error típico	0,421209404	Interval(s)	0,006205195	0,007860497
Mediana	5,677401			
Moda	5,6025			
Desviación estándar	9,218617915			
Varianza de la muestra	84,98291626			
Curtosis	150,9159598			
Coeficiente de asimetría	10,23075059			
Rango	155,122699			
Mínimo	1,6883			
Máximo	156,810999			
Suma	3368,733265			
Cuenta	479			
Nivel de confianza(95,0%)	0,827650899			

time-pc2	time-myPc
4,3535	7,5932
3,052101	5,1372
3,8435	4,8268
3,232201	5,7233
2,650201	4,5814
4,6674	5,0114
2,9709	4,2208
3,2419	5,1148
3,323801	6,1338
18,823899	6,5744
3,345701	4,2615
2,3352	3,9968
3,1075	4,1644
3,4948	7,0456
3,068899	3,9241

STATISTICS-MYPC				
Media	15,05672832	Interval(ms)	12,81553318	17,29792346
Error típico	1,140622545	Interval(s)	0,012815533	0,017297923
Mediana	9,55035			
Moda	3,7192			
Desviación estándar	25,09369599			
Varianza de la muestra	629,6935786			
Curtosis	60,79906946			
Coeficiente de asimetría	6,815479595			
Rango	291,9664			
Mínimo	2,6699			
Máximo	294,6363			
Suma	7287,456506			
Cuenta	484			
Nivel de confianza(95,0%)	2,241195142			

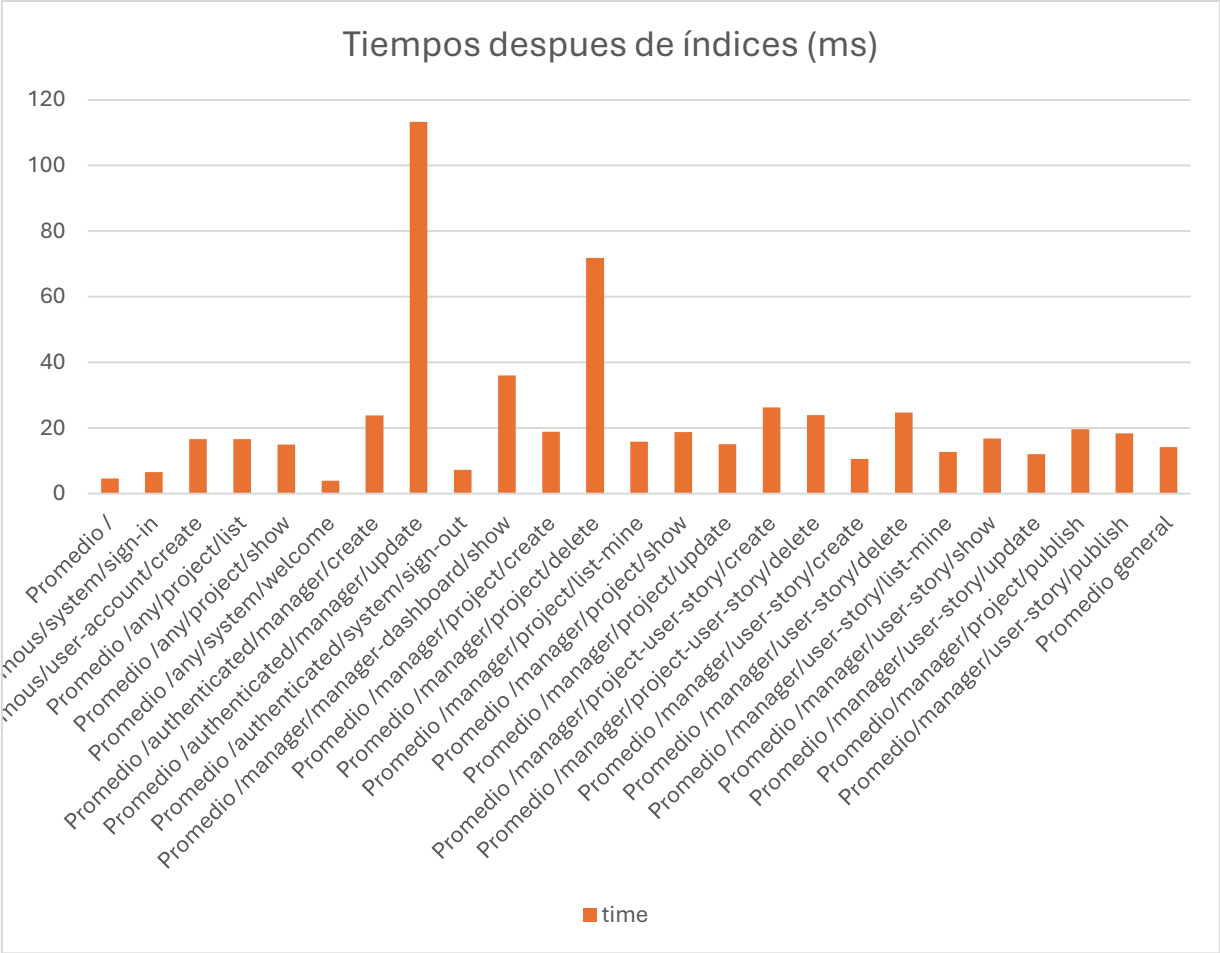
Ilustración 1 Tabla comparativa entre el pc2 y mi pc

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 1.0los 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:



ESTADISTICAS 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			
Coeficiente 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 2.0 del entregable, cabe destacar que la comparación entre estas dos versiones tiene una diferencia, en esta segunda versión los test de autenticación de mánager no han sido realizados para priorizar los requisitos obligatorios que debemos satisfacer completamente:

1) Version 2.0 Sin optimizar índices:

Before				
Media	15,80094655	Interval (ms)	14,169783	17,43211
Error típico	0,830800407	Interval(s)	0,0141698	0,0174321
Mediana	9,12675			
Moda	2,6792			
Desviación estándar	21,98091266			
Varianza de la muestra	483,1605215			
Curtosis	33,30957012			
Coeficiente de asimetría	4,405609444			
Rango	269,8214			
Mínimo	2,4329			
Máximo	272,2543			
Suma	11060,66258			
Cuenta	700			
Nivel de confianza(95,0%)	1,631163258			

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 17 milisegundos), y la media se mantiene en 15 milisegundos.

2) Version 2.0 Tras realizar la modificación en la aplicación para agilizar búsquedas mediante los índices en la base de datos, obtenemos este nuevo resultado:

After				
Media	14,59618181	Interval (ms)	13,037397	16,154966
Error típico	0,793935641	Interval(s)	0,0130374	0,016155
Mediana	8,15255			
Moda	3,8717			
Desviación estándar	21,00556264			
Varianza de la muestra	441,2336617			
Curtosis	25,86918084			
Coeficiente de asimetría	4,152238438			
Rango	220,8023			
Mínimo	2,4128			
Máximo	223,2151			
Suma	10217,32727			
Cuenta	700			
Nivel de confianza(95,0%)	1,55878432			

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, que son prácticamente los mismos que en al version 1.0.

Por último, en el contraste de hipótesis, nos damos cuenta de que el p-value (0.276891783), 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		
	154,65	*184,7924*
Media	15,60231966	14,352808
Varianza (conocida)	483,160521	441,23366
Observaciones	700	700
Diferencia hipotética de las medias	0	
z	1,087328217	
P(Z<=z) una cola	0,138445891	
Valor crítico de z (una cola)	1,644853627	
Valor crítico de z (dos colas)	0,276891783	
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 entre los usuarios y clientes, 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