



Escuela Técnica Superior de
Ingeniería Informática

Testing report Do4

Group C1.068 | Diseño y Pruebas II | 06/05/2025

Fecha	Versión	Autor
06-05-2025	1.0	Gabriel Vacaro Goytia

Miembros:

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

Repositorio de Github: <https://github.com/DP2-C1-068/Acme-ANS-Do4>

Contenido

Resumen ejecutivo 3

Introducción..... 4

Contenido 5

 Pruebas funcionales: 5

 Análisis de desempeño: 11

Conclusiones 16

Bibliografía 16

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-ANS, 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 y por último, la comparación de un mismo ordenador (antes y después de introducir índices en la base de datos).

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.

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 8 y 9 del estudiante #5 (Operaciones sobre registros de mantenimiento y tareas).

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 documento se estructura en dos capítulos principales dedicados al proceso de testing del proyecto. El primero está centrado en las pruebas funcionales e incluye un listado de los casos de prueba implementados, organizados por funcionalidad. Cada caso se acompaña de una breve descripción y una valoración de su eficacia a la hora de detectar errores. El segundo capítulo se enfoca en las pruebas de rendimiento e incorpora gráficos representativos, así como un intervalo de confianza del 95 % para el tiempo de respuesta del sistema al ejecutar las pruebas funcionales en dos escenarios distintos: antes y después de optimizar la búsqueda en la base de datos mediante índices. Además, se realiza un contraste de hipótesis con un nivel de confianza del 95 %, utilizando un Z-test, para determinar si los cambios introducidos producen una mejora estadísticamente significativa.

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”:

[/technician/maintenance-record/...](#)

X.safe	Descripción	Bugs detectados
list	Se listaron los registros de mantenimiento de los técnicos, se probó a mostrar listados con 0, 1 y varios registros. Tanto los propios como el catálogo de registros publicados por todos los técnicos.	No se detectaron bugs.
show	Se mostraron registros de mantenimiento publicados y no publicados.	No se detectaron bugs.
create	Se probó cada variación de cada campo del formulario de creación de registros de mantenimiento.	No se detectaron bugs.
update	Se probó cada variación de cada campo del formulario de actualización de registros de mantenimiento, además de probar actualizar registros con 0, 1 o varias tareas anexadas.	No se detectaron bugs.
publish	Se probó cada variación de cada campo del formulario de publicación de registros de mantenimiento, además de probar a publicar registros con el estado “pending” o “in progress” y registros sin tareas asignadas, 1 tarea asignada pero no publicada, 2 tareas asignadas (1 publicada y la otra no) y con 1 tarea publicada o más.	No se detectaron bugs.
delete	Se probó a eliminar registros de mantenimiento con tareas vinculadas a él (tareas tanto publicadas como sin publicar) como registros sin tareas vinculadas.	No se detectaron bugs.
Y.hack	Descripción	Bugs detectados
list	Se trató de listar registros de mantenimiento sin rol o con un rol incorrecto (tanto los personales como el catálogo público para todos los técnicos).	No se detectaron bugs.
show	Sin rol o con un rol incorrecto: <ul style="list-style-type: none"> - Mostrar registros con id 9999 - Mostrar registros existentes no publicados - Mostrar registros existentes publicados Con rol de técnico: <ul style="list-style-type: none"> - Mostrar registros con id 9999 - Mostrar registros existentes no publicados de otro técnico. 	No se detectaron bugs.
create	Sin rol o con un rol incorrecto: <ul style="list-style-type: none"> - Se trató de acceder a la pantalla de creación de registros de mantenimiento. Con rol de técnico:	No se detectaron bugs.

	- Creación de un registro con una aeronave con id 9999.	
update	<p>Sin rol o con un rol incorrecto:</p> <ul style="list-style-type: none"> - Se trató de acceder a la pantalla de actualización de registros de mantenimiento probando con ids de registros existentes tanto publicados como no publicados como con id 9999. <p>Con rol de técnico:</p> <ul style="list-style-type: none"> - Actualizar registros con id 9999 - Actualizar registros existentes tanto publicados como no publicados de otro técnico. - Actualizar registros existentes publicados del propio técnico. - Actualizar un registro no publicado del propio técnico con una aeronave con id 9999. 	No se detectaron bugs.
publish	<p>Sin rol o con un rol incorrecto:</p> <ul style="list-style-type: none"> - Se trató de acceder a la pantalla de publicación de registros de mantenimiento probando con ids de registros existentes tanto publicados como no publicados como con id 9999. <p>Con rol de técnico:</p> <ul style="list-style-type: none"> - Publicar registros con id 9999 - Publicar registros existentes tanto publicados como no publicados de otro técnico. - Publicar registros existentes publicados del propio técnico. - Publicar un registro no publicado del propio técnico con una aeronave con id 9999. 	Se detectó que era posible acceder a la vista de edición de un registro de mantenimiento publicado por el propio técnico.
delete	<p>Sin rol o con un rol incorrecto:</p> <ul style="list-style-type: none"> - Se trató de acceder a la pantalla de borrado de registros de mantenimiento probando con ids de registros existentes tanto publicados como no publicados como con id 9999. <p>Con rol de técnico:</p> <ul style="list-style-type: none"> - Eliminar registros con id 9999 - Eliminar registros existentes tanto publicados como no publicados de otro técnico. - Eliminar registros existentes publicados del propio técnico. 	No se detectaron bugs.

[/technician/task/...](#)

X.safe	Descripción	Bugs detectados
list	Se listaron las tareas de los técnicos, se probó a mostrar listados con 0, 1 y varias tareas. Tanto las propias como el catálogo de tareas publicadas por todos los técnicos. Por otra parte, se probó a listar las tareas vinculadas a registros de mantenimiento concretos (publicados y no publicados) con 0, 1 y varias tareas.	No se detectaron bugs.
show	Se mostraron tareas publicadas y no publicadas.	No se detectaron bugs.

create	Se probó cada variación de cada campo del formulario de creación de tareas tanto para tareas en solitario como para tareas que nacen asociadas a un registro de mantenimiento.	No se detectaron bugs.
update	Se probó cada variación de cada campo del formulario de actualización de tareas.	No se detectaron bugs.
publish	Se probó cada variación de cada campo del formulario de publicación de tareas.	No se detectaron bugs.
delete	Se probó a eliminar tareas vinculadas a registros de mantenimiento y tareas libres.	No se detectaron bugs
Y.hack	Descripcion	Bugs detectados
list	Se trató de listar tareas sin rol o con un rol incorrecto (tanto las personales como el catálogo público para técnicos).	No se detectaron bugs.
show	Sin rol o con un rol incorrecto: <ul style="list-style-type: none"> - Mostrar tareas con id 9999 - Mostrar tareas existentes no publicados - Mostrar tareas existentes publicados Con rol de técnico: <ul style="list-style-type: none"> - Mostrar tareas con id 9999 - Mostrar tareas existentes no publicados de otro técnico. 	No se detectaron bugs.
create	Sin rol o con un rol incorrecto: <ul style="list-style-type: none"> - Se trató de acceder a la pantalla de creación de tareas. Con rol de técnico: <ul style="list-style-type: none"> - Creación de una tarea dentro un registro de mantenimiento de otro técnico (tanto publicado como no publicado). - Creación de una tarea dentro de un registro de mantenimiento publicado del propio técnico. - Creación de una tarea para un registro de mantenimiento con id 9999. 	Se detectó que era posible crear tareas dentro de registros de mantenimiento creados por cualquier técnico independientemente de si estuvieran publicados o no. Por otra parte también era posible crear una tarea dentro de un registro de mantenimiento publicado propio.
update	Sin rol o con un rol incorrecto: <ul style="list-style-type: none"> - Se trató de acceder a la pantalla de actualización de tareas probando con ids de tareas existentes tanto publicados como no publicados como con id 9999. Con rol de técnico: <ul style="list-style-type: none"> - Actualizar tareas con id 9999 - Actualizar tareas existentes tanto publicados como no publicados de otro técnico. - Actualizar tareas existentes publicados del propio técnico. 	No se detectaron bugs.
publish	Sin rol o con un rol incorrecto: <ul style="list-style-type: none"> - Se trató de acceder a la pantalla de publicación de tareas probando con ids de tareas existentes tanto publicadas como no publicadas como con id 9999. Con rol de técnico: <ul style="list-style-type: none"> - Publicar tareas con id 9999 - Publicar tareas existentes tanto publicadas como no publicadas de otro técnico. - Publicar tareas existentes publicadas del propio técnico. 	No se detectaron bugs.
delete	Sin rol o con un rol incorrecto: <ul style="list-style-type: none"> - Se trató de acceder a la pantalla de borrado de tareas 	No se detectaron bugs.

	probando con ids de tareas existentes tanto publicadas como no publicadas como con id 9999. Con rol de técnico: <ul style="list-style-type: none"> - Eliminar tareas con id 9999 - Eliminar tareas existentes tanto publicadas como no publicadas de otro técnico. - Eliminar tareas existentes publicados del propio técnico. 	
--	--	--

/technician/involves/...
























X.safe	Descripcion	Bugs detectados
create	Se probó a vincular las tareas disponibles en el desplegable, además de probar sin tarea “---”.	No se encontraron bugs.
delete	Se probó a desvincular las tareas disponibles en el desplegable, además de probar sin tarea “---”.	No se encontraron bugs.
Y.hack	Descripcion	Bugs detectados
create	Sin rol o con un rol incorrecto: <ul style="list-style-type: none"> - Se trató de acceder a la pantalla de vinculación de tareas a registros de mantenimiento con id 9999 y a registros existentes (publicados y sin publicar). Con rol de técnico: <ul style="list-style-type: none"> - Acceder al formulario de vinculación de un registro de mantenimiento de otro técnico (tanto publicado como sin publicar) o con id 9999. - Acceder al formulario de vinculación de un registro propio publicado. - Vincular una tarea con id 9999 a un registro de mantenimiento. - Vincular una tarea existente, pero sin publicar de otro técnico a un registro de mantenimiento. 	Se descubrió que era posible vincular tareas a registros de mantenimiento de otros técnicos. Por otra parte, era posible vincular tareas a registros de mantenimiento propios publicados.
delete	Sin rol o con un rol incorrecto: <ul style="list-style-type: none"> - Se trató de acceder a la pantalla de desvinculación de tareas a registros de mantenimiento con id 9999 y a registros existentes (publicados y sin publicar). Con rol de técnico: <ul style="list-style-type: none"> - Acceder al formulario de desvinculación de un registro de mantenimiento de otro técnico (tanto publicado como sin publicar) o con id 9999. - Acceder al formulario de desvinculación de un registro propio publicado. - Desvincular una tarea con id 9999 a un registro de mantenimiento. - Desvincular una tarea existente a un registro de mantenimiento que no tiene vinculada esa tarea. 	Se descubrió que era posible desvincular tareas a registros de mantenimiento de otros técnicos. Por otra parte, era posible desvincular tareas a registros de mantenimiento propios publicados.

EXTRAS

Se creó de forma separada una prueba denominada miscellaneous.hack donde se pretendía cubrir la totalidad de la cobertura del código, pues había instrucciones que solo eran ejecutadas parcialmente, y era necesario utilizar urls con ids nulas para completar la cobertura total.

Recubrimiento del código:

Estas pruebas son completas, pues tienen una cobertura de código del 100%, independientemente de la feature.

>  acme.entities.aircraft	100,0 %	27	0	27
>  acme.entities.maintenance	100,0 %	87	0	87
▼  acme.features.technician.involves	100,0 %	426	0	426
>  TechnicianInvolvesController.java	100,0 %	14	0	14
>  TechnicianInvolvesCreateService.java	100,0 %	202	0	202
>  TechnicianInvolvesDeleteService.java	100,0 %	210	0	210
▼  acme.features.technician.maintenanceRecord	100,0 %	1.180	0	1.180
>  TechnicianMaintenanceRecordController.java	100,0 %	35	0	35
>  TechnicianMaintenanceRecordCreateService.java	100,0 %	179	0	179
>  TechnicianMaintenanceRecordDeleteService.java	100,0 %	205	0	205
>  TechnicianMaintenanceRecordListService.java	100,0 %	108	0	108
>  TechnicianMaintenanceRecordPublishService.java	100,0 %	279	0	279
>  TechnicianMaintenanceRecordShowService.java	100,0 %	145	0	145
>  TechnicianMaintenanceRecordUpdateService.java	100,0 %	229	0	229
▼  acme.features.technician.task	100,0 %	1.036	0	1.036
>  TechnicianTaskController.java	100,0 %	35	0	35
>  TechnicianTaskCreateService.java	100,0 %	212	0	212
>  TechnicianTaskDeleteService.java	100,0 %	150	0	150
>  TechnicianTaskListService.java	100,0 %	201	0	201
>  TechnicianTaskPublishService.java	100,0 %	164	0	164
>  TechnicianTaskShowService.java	100,0 %	113	0	113
>  TechnicianTaskUpdateService.java	100,0 %	161	0	161
>  acme.realms	100,0 %	3	0	3

Solo existe una instrucción la cual no ha podido ejecutarse en todas sus variantes, localizada en la feature /technician/task/ListService, en las líneas 94 y 95.

```

92         if (maintenanceRecordId != null) {
93             maintenanceRecord = this.repository.findMaintenanceRecordById(maintenanceRecordId);
94             showCreate = maintenanceRecord != null && maintenanceRecord.isDraftMode() //
95                 && super.getRequest().getPrincipal().hasRealm(maintenanceRecord.getTechnician());
96         } else if (mine)
97             showCreate = true;

```

Esto se debe a que la combinación de restricciones del showCreate que dicta tal que:

- maintenanceRecord != null -> TRUE
- maintenanceRecord.isDraftMode() -> TRUE
- super.getRequest().getPrincipal().hasRealm(maintenanceRecord.getTechnician()) -> FALSE

nunca va a poder ser comprobada, ya que, en este caso, estaríamos intentando acceder al listado de tareas de un registro de mantenimiento que, si bien existe, no está publicado, y por tanto si el principal no es el técnico adecuado va a saltar el método `authorise`, pues es un intento de hackeo. Como este código pertenece al método `unbind`, y no al método `authorise`, nunca llegará a ejecutarse.

Análisis de desempeño:

En primer lugar, comenzaremos con unas estadísticas básicas sobre el rendimiento de la aplicación antes y después de optimizar los índices, así como sus respectivas gráficas de tiempo/feature. Para poder comparar estos tiempos, se ha realizado un contraste de hipótesis mediante un Z-test que demuestra que los cambios si tienen una significancia estadística.

Prueba z para medias de dos muestras

	<i>Antes</i>	<i>Despues</i>
Media	4,33978098	3,67986744
Varianza (conocida)	30,5307382	19,4498044
Observaciones	1551	1551
Diferencia hipotética de las medias	0	
z	3,676143489	
P(Z<=z) una cola	0,000118393	
Valor crítico de z (una cola)	1,644853627	
Valor crítico de z (dos colas)	0,000236786	
Valor crítico de z (dos colas)	1,959963985	

Como podemos observar, el valor crítico de z (dos colas) es de 0,000236786, es decir, se encuentra entre 0 y 1-alpha (Alpha = 0.5, pues el intervalo de confianza es del 0.95). Esto quiere decir que los cambios realizados si tienen significancia estadística y podemos comprobar los tiempos antes y después, a lo cual procederemos a continuación:

<i>ANTES</i>		<i>DESPUES</i>	
Media	4,33978098	Media	3,67986744
Error típico	0,14030164	Error típico	0,111982903
Mediana	2,5647	Mediana	2,5078
Moda	0,0977	Moda	0,1003
Desviación estándar	5,52546271	Desviación estándar	4,410193239
Varianza de la muestra	30,5307382	Varianza de la muestra	19,4498044
Curtosis	10,1561432	Curtosis	15,29204845
Coeficiente de asimetría	2,65577382	Coeficiente de asimetría	2,83053199
Rango	44,4598	Rango	51,944
Mínimo	0,0692	Mínimo	0,0676
Máximo	44,529	Máximo	52,0116
Suma	6731,0003	Suma	5707,4744
Cuenta	1551	Cuenta	1551
Nivel de confianza(95,0%)	0,27520106	Nivel de confianza(95,0%)	0,219653977

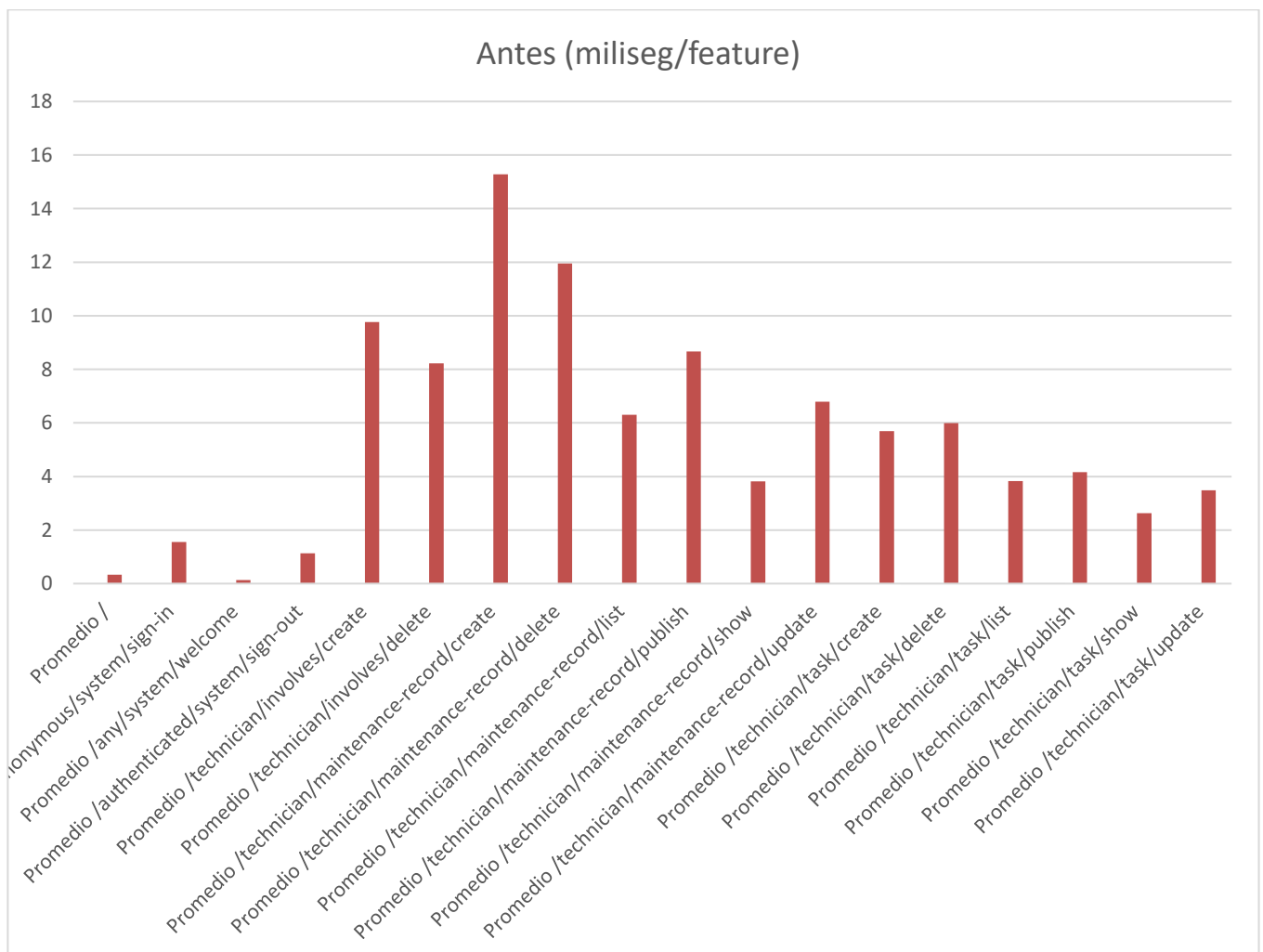
interval(ms) 4,06457992 4,61498204
interval(s) 0,00406458 0,00461498

interval(ms) 3,460213463 3,89952142
interval(s) 0,003460213 0,00389952

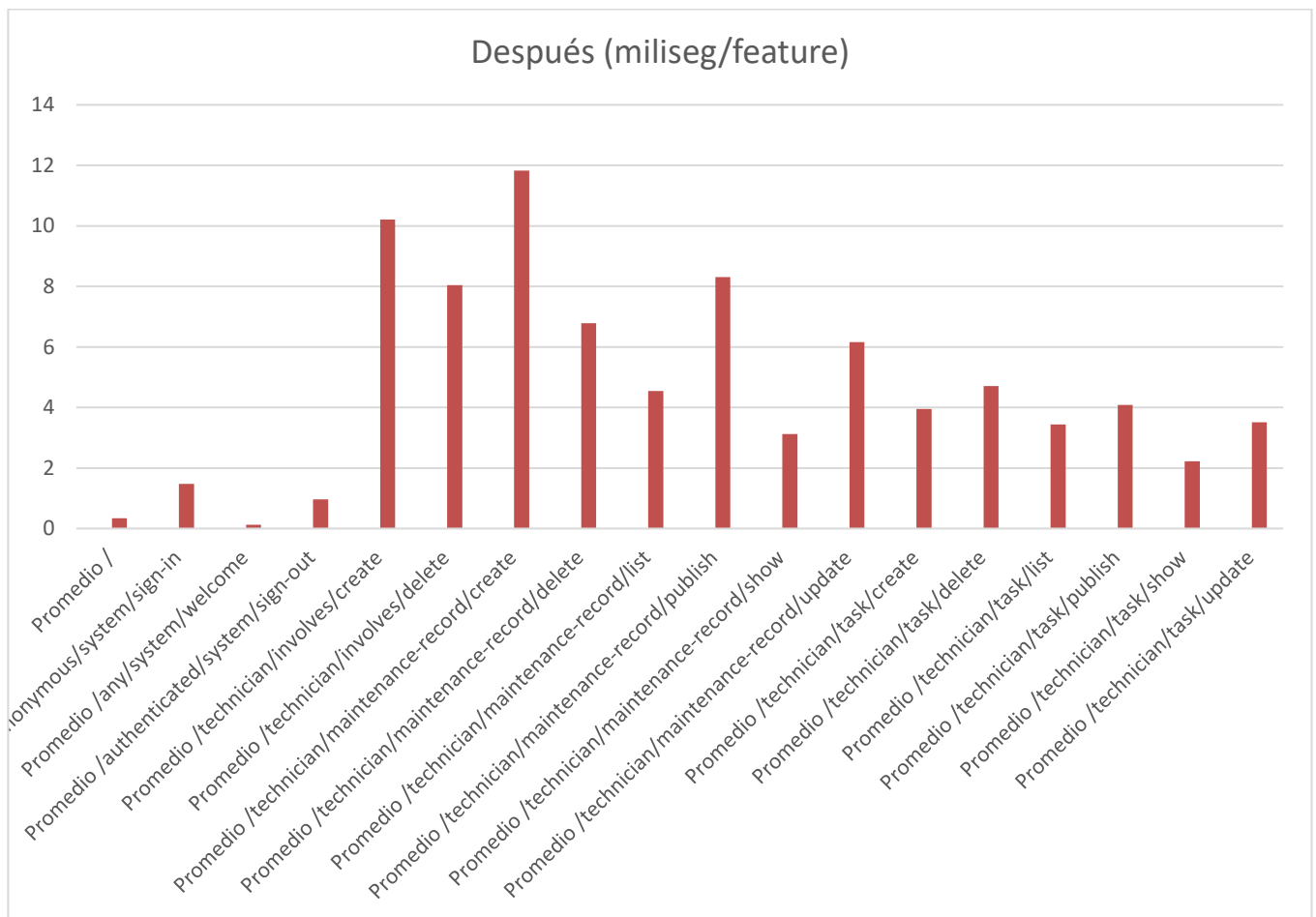
Si nos fijamos en el intervalo de confianza, antes de realizar los cambios sobre los índices, podíamos asegurar responder cualquier petición en un tiempo inferior o igual a 4,61498204 ms con un 95% de fiabilidad. Mientras que, tras la optimización en la búsqueda en base de datos, esta cota superior de tiempo ha disminuido hasta 3,89952142 ms, esto es, 0,71546062 ms menos. Esto se traduce en una mejora en el rendimiento de la aplicación del 15,50% aproximadamente. Independientemente a esta mejora, nos encontramos muy por debajo de la cota máxima del requisito no funcional que nos impone el cliente de poder servir cualquier petición en menos de 1 segundo con un 95% de confianza.

A continuación, procederemos a ver los gráficos de cada uno, para observar si se ha conseguido paliar el tiempo de respuesta del MIR (petición más ineficiente):

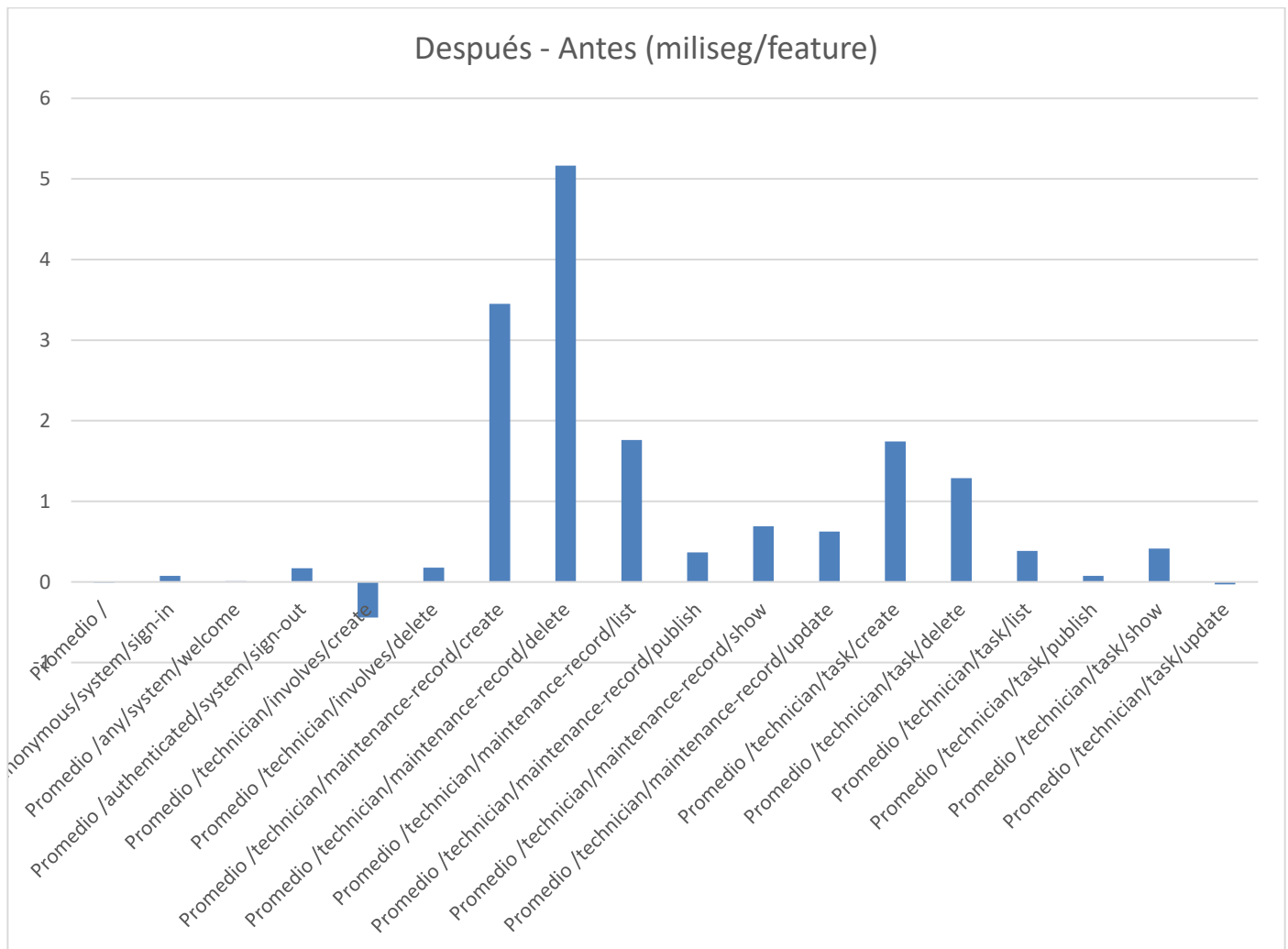
Grafica del antes:



Gráfica del después:



Como podemos comprobar, el MIR sigue perteneciendo a la misma feature (/technician/maintenance-record/create), pero esta se ha visto reducida en gran medida, en concreto de 15 ms a 12 ms aproximadamente. Una reducción similar se aplica para la mayoría de las features. En concreto, la mejor reducción se encuentra en /technician/maintenance-record/delete, con una mejoría de un 43.26% (de 11.95 ms a 6.78 ms). Para una mejor visualización de estos cambios, véase la siguiente gráfica, donde se muestra la diferencia de los tiempos (después - antes) en milisegundos.



Como podemos ver, a excepción de la feature technician/involves/create, la cual ha empeorado, el resto o siguen iguales o han mejorado en cierta medida. Lo que sí está claro es que este cambio ha resultado para bien, como se ha podido confirmar anteriormente.

Para finalizar este apartado, veamos una comparativa entre dos ordenadores distintos, para comprobar la relevancia del hardware a la hora de responder peticiones en un WIS (esta comparativa solo es tras la optimización de índices). Para el PC_1 se usará el mismo que el que se ha utilizado para este reporte (procesador Intel i7-14700), y para PC_2 se utilizará mi segundo entorno de trabajo, el cual es un portátil (procesador intel i7-1165G7):

PC_1	
Media	3,67986744
Error típico	0,1119829
Mediana	2,5078
Moda	0,1003
Desviación estándar	4,41019324
Varianza de la muestra	19,4498044
Curtosis	15,2920485
Coeficiente de asimetría	2,83053199
Rango	51,944
Mínimo	0,0676
Máximo	52,0116
Suma	5707,4744
Cuenta	1551
Nivel de confianza(95,0%)	0,21965398

interval(ms) 3,46021346 3,89952142
interval(s) 0,00346021 0,00389952

PC_2	
Media	11,8039432
Error típico	0,3662598
Mediana	7,05355
Moda	0,2491
Desviación estándar	14,4938957
Varianza de la muestra	210,073011
Curtosis	8,48382287
Coeficiente de asimetría	2,35445349
Rango	133,3698
Mínimo	0,1881
Máximo	133,5579
Suma	18484,9751
Cuenta	1566
Nivel de confianza(95,0%)	0,71841162

interval(ms) 11,0855316 12,5223549
interval(s) 0,01108553 0,01252235

Como podemos observar, en el caso del PC_2, nos movemos en un intervalo de confianza donde la cota superior es de 12,5223549 ms, lo cual es 3,21 veces mayor que en PC_1. En este caso, aunque sea complicado comparar por intervalos de confianza, queda claro cuál de los dos es más potente.

Conclusiones

A lo largo de este informe se ha demostrado que el proceso de testing aplicado al proyecto *Acme-ANS* ha sido riguroso, completo y orientado a garantizar tanto la corrección funcional como la eficiencia del sistema. Las pruebas funcionales, que abarcaron múltiples escenarios positivos, negativos y de hacking, han evidenciado una implementación sólida de los requisitos 8 y 9, permitiendo identificar y documentar solo un número muy reducido de vulnerabilidades. Estas incidencias fueron localizadas y paliadas en esta fase de testing formal.

Asimismo, el análisis de rendimiento ha confirmado, mediante un Z-test con una significancia estadística del 95 %, que las optimizaciones aplicadas en la base de datos, concretamente, la introducción de índices, han mejorado de forma notable el tiempo de respuesta del sistema, alcanzando una reducción de la cota superior del intervalo de confianza del 15,5 %. Esta mejora ha sido especialmente visible en ciertas operaciones como el borrado de registros de mantenimiento.

En conjunto, los resultados obtenidos avalan la calidad técnica del desarrollo y sientan una base robusta para futuras ampliaciones o despliegues en entornos productivos, asegurando una experiencia de usuario más fluida y un cumplimiento estricto de los estándares definidos en los requisitos con el cliente, así como el aseguramiento de la seguridad desde el diseño, con el objetivo de cumplir tanto con el Reglamento General de Protección de Datos (RGPD) como con la Ley Orgánica 3/2018, de Protección de Datos Personales y garantía de los derechos digitales.

Bibliografía

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