26/02/2025
Repositorio: https://github.com/paugonpae67/Acme-ANS
Trabajo individual realizado por: Paula Rosa González Páez (paugonpae@alum.us.es)
Grupo: C1.064

TESTING REPORT

ACME ANS – C1.064

TABLA DE CONTENIDOS

RESUMEN EJECUTIVO

En este documento se va a recoger la documentación acerca de las pruebas funcionales y de rendimiento realizadas sobre los requisitos 8 y 9 del Student 5, en concreto, sobre las entidades y funcionalidades de MaintenanceRecord, Task e InvolvedIn. De igual forma, se realizará un análisis de dichas pruebas exponiendo los resultados y conclusiones.

El objetivo principal de este análisis es ver la eficiencia e impacto que tienen los cambios y refactorizaciones en el rendimiento del sistema.

Para la realización de las pruebas, se implementaron casos de pruebas positivos, casos de prueba negativos y casos de hacking aprendidos en la asignatura. Estas pruebas funcionales se desarrollaron a través de Eclipse mediante herramientas de grabación (tester-record). Las distintas pruebas se clasifican en ficheros .safe y .hack según se tratara de pruebas de hacking o no.

Para la realización del análisis, se observó el rendimiento de forma individual tanto antes como después de realizar las refactorizaciones en el código. Se utilizó el parámetro p (two-tail p-value) que nos ayuda a determinar si el promedio de tiempo del sistema antes y después de los cambios puede considerarse igual o no. Todo ello gracias al análisis estadístico.

TABLA DE REVISIONES

Versión	Fecha	Descripción
1.0	25/05/2025	Primera versión del documento
1.1	26/05/2025	Última versión, revisión del documento y preparación para la entrega

INTRODUCCIÓN

Este documento, contiene en primer lugar, una explicación de los casos de pruebas generados, así como las respuestas esperadas y su efectividad para ayudar a encontrar bugs en el código. Estas pruebas en todo momento han tenido 2 objetivos, verificar el cumplimiento de los requisitos del Student #5 y el correcto funcionamiento del proyecto y sus validaciones, y ver el rendimiento que tiene Acme-ANS.

De igual forma, se incluirán capturas de la cobertura que se ha alcanzado con las pruebas.

A continuación, de forma ordenada se exponen tablas y gráficos que reflejan el tiempo de respuesta del sistema antes de implementar los cambios (Before) y después de ellos (After). Esta información ha sido recopilada a través de un Excel, ficheros .trace (generados con la replicación de las pruebas) y herramientas de análisis.

Por último, se expone una conclusión en base a si los cambios realizados han sido lo suficientemente buenos como para mejorar el rendimiento del sistema. Todo ello, partiendo del parámetro p obtenido por el análisis estadístico con los datos del Excel mencionado anteriormente.

Todo el análisis parte de un nivel de confianza del 95%.

CONTENIDO

TESTING FUNCIONAL

Para empezar, en este apartado se muestran los casos de pruebas que se han llevado a cabo, así como la respuesta esperada de los mismo y su efectividad para localización de bugs. De esta forma, queda reflejado el cumplimiento de los requisitos del Student #5 y la realización de pruebas para verificar el correcto funcionamiento del mismo. Estructuralmente, se mostrarán las listas por entidad y a su vez separadas por funcionalidades y métodos. En primer lugar, la información acerca de MaintenanceRecord, posteriormente la información sobre Task y para finalizar la información sobre InvolvedIn.

Para abreviar un poco la tabla no se ha puesto cada caso en específico si como respuesta daban lo mismo. El procedimiento general para los casos de prueba ha sido:

1. Probar que las listas se vean y cuya url si no es un *realm* del mismo tipo debe saltar Error Not Authorised.
2. Probar los métodos *show* de forma que se vean todos los atributos de las clases y si se hace un *show* con otro *realm* distinto debe saltar Error Not Authorised. De igual forma, las entidades publicadas sí pueden verlas otros usuarios con el mismo *realm* sin saltar Not Authorised, pero las no publicadas solo pueden verlas los técnicos responsables de ellas.
3. Probar el método *create* enviando primero un formulario vacío donde saltan las validaciones de los campos nulos y progresivamente ir probando en cada campo todos los valores incorrectos que no cumplen su formato específico para comprobar que aparecen las validaciones correspondientes. Una vez hecho, se ponen datos correctos y debe dejar crear la entidad. Como hacking, en caso de atributos navegables, si introducimos un id del atributo nulo o inexistente debe lanzar error Not Authorised. De igual forma, hacer una petición GET de *create* directamente desde un *show* debe lanzar error Not Authorised. Por último, se comprueba que otra persona con el mismo *realm* no pueda crear entidades en otros técnicos y además que no se pueda crear una entidad con id distinto de 0.
4. Probar el método update y publish ambos de igual forma. A continuación, mandar el formulario con los valores en null para que aparezcan las validaciones Not Null, realizar las peticiones comprobando en cada campo los valores incorrectos de acuerdo con sus restricciones para que aparezcan los errores de validación correspondientes y por último, realizar las peticiones con los campos rellenados de forma correcta. Para los hacking, se comprueba que el id de los atributos navegables no pueda ser null o inexistente, se comprueba que no puedan hacerse peticiones GET de estos métodos y que si es un usuario de otro realm u otro técnico intenta actualizar o publicar una entidad de otro técnico, lance error Not Authorised.
5. Probar el método *delete*, eliminando correctamente una entidad propia (deberán aparecer validaciones si corresponden), eliminando una entidad de otro usuario saltanto error Not Authorised y probar una petición GET del delete que debe lanzar error Not Authorised.

Para los atributos navegables de la clase Involved, en caso de estar añadiendo tareas debe saltar Not Authorised en caso de añadir dos veces la misma tarea a un registro y en cuanto a la eliminación de la asociación entre una tarea y un registro, si el id de la tarea no corresponde a ninguna relación con ese registro también debe saltar error Not Authorised (siendo el técnico responsable del registro).

Para todos los casos se comprueba también que el masterId o id si se modifica la URL introduciendo uno inválido o inexistente salte error Not Authorised.

TESTING FUNCIONAL MAINTENANCE-RECORD

Case Id	Descripción	Repuesta Esperada	Bugs Detectados	Efectividad
LIST and SHOW				
TC-01	Listar todos los registros de mantenimiento de un técnico	Mostrar lista de los registros de ese técnico	0	Baja
TC-02	Mostrar los detalles de un registro de mantenimiento de un técnico	Mostrar formulario con los datos del registro de mantenimiento de ese técnico	0	Baja
TC-03	Listar registros de mantenimiento como anónimo	Error Not Authorised	0	Media
TC-04	Mostrar detalles de un registro de mantenimiento con otro realm o como anónimo	Error Not Authorised	0	Media
TC-05	Mostrar detalles de un registro de mantenimiento publicado con otro técnico	Mostrar los detalles del registro de mantenimiento	0	Baja
TC-06	Mostrar detalles de un registro de mantenimiento no publicado con otro técnico	Error Not Authorised	0	Media
TC-07	Mostrar detalles de un registro de mantenimiento con una id que no existe	Error Not Authorised	0	Media
CREATE				
TC-08	Post de un registro de mantenimiento con valores inválidos en sus campos	Saltan las validaciones y no permite crearlo	0	Media
TC-09	Post de un registro de mantenimiento con un id de aircraft nulo o que no existe	Error Not Authorised	0	Media
TC-10	Post de un registro de mantenimiento con id distinta de 0	Error Not Authorised	1	Alta

TC-11	Post de un registro de mantenimiento con los valores nulos	Saltan validaciones Not Null en los campos correspondientes.	0	Baja
TC-12	Post de un registro de mantenimiento con los valores correctos	Se crea correctamente el registro de mantenimiento	0	Baja
UPDATE				
TC-13	Actualizar un registro de mantenimiento con valores nulos en sus campos	Saltan las validaciones Not Null y no permite actualizar	0	Baja
TC-14	Actualizar un registro de mantenimiento con valores incorrectos en sus campos	Saltan las validaciones correspondientes y no permite actualizar	0	Baja
TC-15	Actualizar un registro de mantenimiento con valores válidos en sus campos siendo el técnico del registro	Se actualiza correctamente el registro de mantenimiento	0	Baja
TC-16	Realizar una petición GET del método Update con cualquier realm o como anónimo	Error Not Authorised	1	Alta
TC-17	Actualizar un registro de mantenimiento con un id de aircraft nulo o inexistente	Error Not Authorised	0	Media
TC-18	Actualizar con método Post el registro de mantenimiento de otro técnico	Error Not Authorised	0	Media
PUBLISH				
TC-19	Publicar un registro de mantenimiento con valores nulos en sus campos	Saltan las validaciones Not Null y no permite publicarlo	0	Baja
TC-20	Publicar un registro de mantenimiento con valores incorrectos en sus campos	Saltan las validaciones correspondientes y no permite publicarlo	0	Baja
TC-21	Publicar un registro de mantenimiento con valores válidos en sus campos siendo el técnico del registro	Se publica correctamente el registro de mantenimiento	0	Baja
TC-22	Realizar una petición GET del método Publish con cualquier realm o como anónimo	Error Not Authorised	1	Media
TC-23	Publicar un registro de mantenimiento con un id de aircraft nulo o inexistente	Error Not Authorised	0	Media
TC-24	Publicar con método Post el registro de mantenimiento de otro técnico	Error Not Authorised	1	Alta
TC-25	Publicar un registro sin tener al menos una tarea publicada asociada o si alguna de las tareas asociadas no está publicada	Error de validación mostrando que debe tener al menos una tarea publicada asociadas y que	0	Media

		todas las que tenga asociada deben estar publicadas		
DELETE				
TC-26	Eliminar un registro de mantenimiento siendo el técnico del registro	Se elimina correctamente	0	Baja
TC-27	Eliminar un registro de mantenimiento de otro técnico	Error Not Authorised	0	Media
TC-28	Realizar una petición GET del método eliminar (Delete) con cualquier realm o como anónimo	Error Not Authorised	1	Alta
TC-29	Eliminar un registro teniendo tareas asociadas	Error de validación para eliminar primero las tareas	0	Media

TESTING FUNCIONAL TASK

Case Id	Descripción	Repuesta Esperada	Bugs Detectados	Efectividad
LIST and SHOW				
TC-01	Listar todas las tareas de un técnico	Mostrar lista de las tareas de ese técnico	0	Baja
TC-02	Mostrar los detalles de una tarea de un técnico	Mostrar formulario con los datos de la tarea de ese técnico	0	Baja
TC-03	Listar tareas como anónimo	Error Not Authorised	0	Media
TC-04	Mostrar detalles de una tarea con otro realm o como anónimo	Error Not Authorised	0	Media
TC-05	Mostrar detalles de una tarea publicada con otro técnico	Mostrar los detalles de la tarea	0	Baja
TC-06	Mostrar detalles de una tarea no publicada con otro técnico	Error Not Authorised	0	Media
TC-07	Mostrar detalles de una tarea con una id que no existe o no es mío	Error Not Authorised	0	Media
CREATE				
TC-08	Post de una tarea con valores inválidos en sus campos	Saltan las validaciones y no permite crearla	0	Media
TC-9	Post de una tarea con id distinta de 0	Error Not Authorised	1	Alta
TC-10	Post de una tarea con los valores nulos	Saltan validaciones Not Null en los campos correspondientes.	0	Baja

TC-11	Post de una tarea con los valores correctos	Se crea correctamente la tarea	0	Baja
UPDATE				
TC-12	Actualizar una tarea con valores nulos en sus campos	Saltan las validaciones Not Null y no permite actualizar	0	Baja
TC-13	Actualizar una tarea con valores incorrectos en sus campos	Saltan las validaciones correspondientes y no permite actualizar	0	Baja
TC-14	Actualizar una tarea con valores válidos en sus campos siendo el técnico del registro	Se actualiza correctamente la tarea	0	Baja
TC-15	Realizar una petición GET del método Update con cualquier realm o como anónimo	Error Not Authorised	1	Alta
TC-16	Actualizar con método Post el registro de mantenimiento de otro técnico	Error Not Authorised	0	Media
PUBLISH				
TC-17	Publicar una tarea con valores nulos en sus campos	Saltan las validaciones Not Null y no permite publicarla	0	Media
TC-18	Publicar una tarea con valores incorrectos en sus campos	Saltan las validaciones correspondientes y no permite publicarla	0	Media
TC-19	Publicar una tarea con valores válidos en sus campos siendo el técnico del registro	Se publica correctamente la tarea	0	Media
TC-20	Realizar una petición GET del método Publish con cualquier realm o como anónimo	Error Not Authorised	0	Media
TC-21	Publicar con método Post la tarea de otro técnico	Error Not Authorised	1	Alta
DELETE				
TC-22	Eliminar una tarea siendo el técnico del registro	Se elimina correctamente	0	Baja
TC-23	Eliminar una tarea de otro técnico	Error Not Authorised	0	Media
TC-24	Realizar una petición GET del método eliminar (Delete) con cualquier realm o como anónimo	Error Not Authorised	0	Media
TC-25	Eliminar una tarea estando asociada a uno o más registros	Error de validación que indica que existen relaciones con esa tarea y algunos registros	1	Alta

TESTING FUNCIONAL INVOLVED-IN

Case Id	Descripción	Repuesta Esperada	Bugs Detectados	Efectividad
LIST and SHOW				
TC-01	Listar las tareas de un registro de mantenimiento de un técnico	Mostrar lista de tareas de un registro de ese técnico	0	Baja
TC-02	Mostrar los detalles de la relación entre la tarea y el registro de mantenimiento de un técnico	Mostrar formulario con los datos de la relación de ese técnico	0	Baja
TC-03	Listar las tareas de un registro de mantenimiento como anónimo	Error Not Authorised	0	Media
TC-04	Mostrar detalles de la relación entre la tarea y el registro de mantenimiento con otro realm o como anónimo	Error Not Authorised	0	Media
TC-05	Mostrar detalles de la relación entre la tarea y el registro de mantenimiento publicado con otro técnico	Mostrar los detalles de la relación entre la tarea y el registro de mantenimiento	0	Media
TC-06	Mostrar detalles de la relación entre la tarea y el registro de mantenimiento no publicado con otro técnico	Error Not Authorised	0	Media
TC-07	Mostrar detalles de la relación entre la tarea y el registro de mantenimiento con una id que no existe	Error Not Authorised	0	Media
TC-08	Listar las tareas de un registro de mantenimiento con una id que no existe	Error Not Authorised	0	Media
CREATE				
TC-09	Post de una relación entre una tarea y un registro de mantenimiento con el campo nulo	Saltan la validación Not Null y no permite crearla	0	Media
TC-10	Post de una relación entre una tarea y un registro de mantenimiento con un id de task nulo o que no existe	Error Not Authorised	0	Media
TC-11	Post de una relación entre una tarea y un registro de mantenimiento con id distinta de 0	Error Not Authorised	1	Alta
TC-12	Post de una relación entre una tarea y un registro de mantenimiento con los valores correctos	Se crea correctamente la relación	0	Baja
TC-13	Post de una relación con el id de la tarea que ya se ha añadido a ese registro	Error Not Authorised	0	Media
TC-14	Realizar una petición GET del método Create de un registro de un técnico	Error Not Authorised	1	Alta

	con otro técnico, otro realm o como anónimo			
DELETE				
TC-15	Eliminar una relación entre una tarea correcta y un registro de mantenimiento siendo el técnico del registro	Se elimina correctamente	0	Baja
TC-16	Eliminar una relación entre una tarea y un registro de mantenimiento con un id de la tarea nulo o inexistente o que no tenía relación con ese registro.	Error Not Authorised	0	Media
TC-17	Eliminar una tarea de un registro de otro técnico	Error Not Authorised	0	Media
TC-18	Realizar una petición GET del método Delete de un registro de un técnico con otro técnico, otro realm o como anónimo	Error Not Authorised	0	Media

ANÁLISIS TESTING

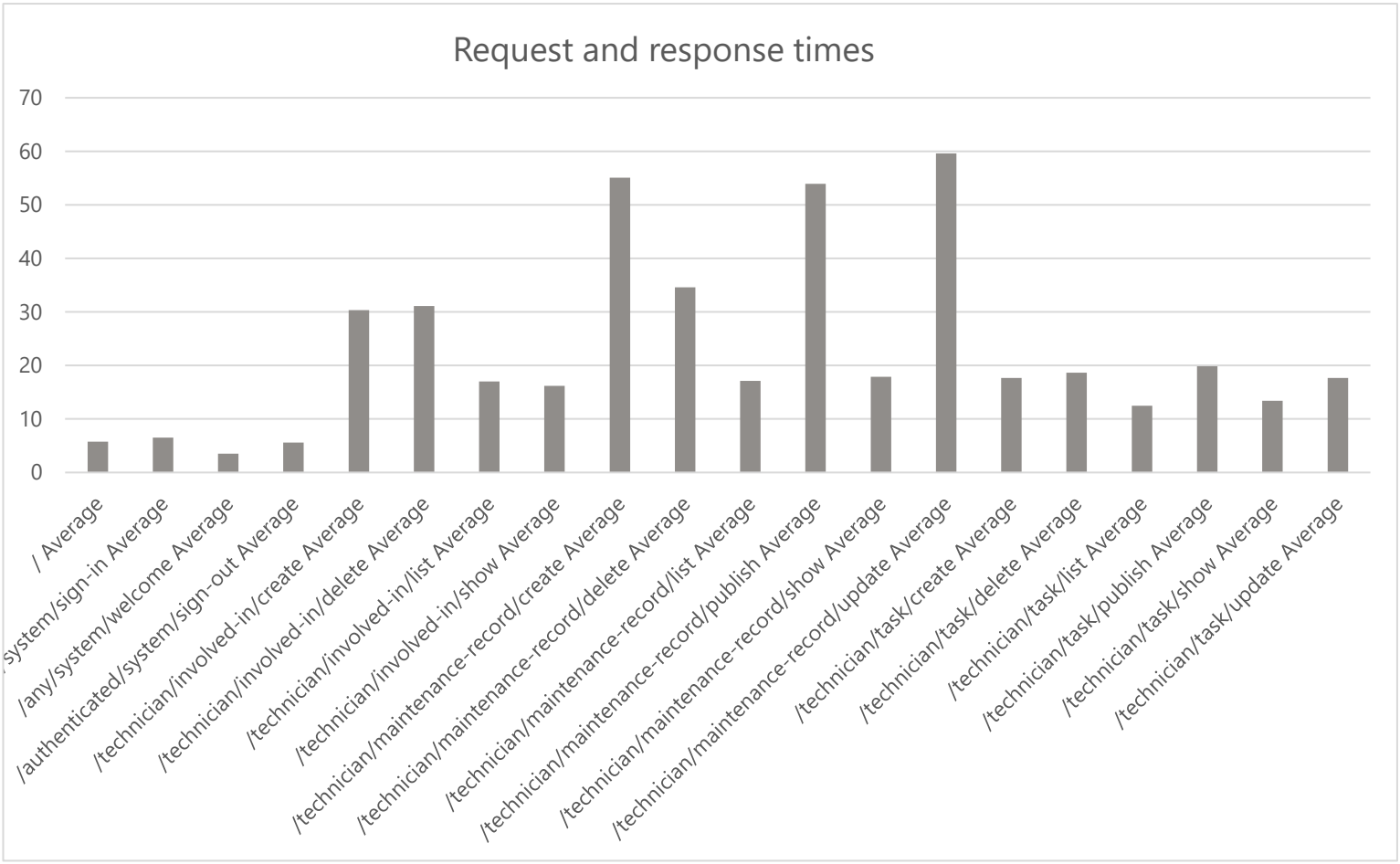
Los casos de pruebas detallados anteriormente se han probado tanto antes como después de realizar la refactorización y añadir los índices a las entidades con el objetivo de ver la calidad del código en cuanto a rendimiento.

Análisis testing before refactoring

Previo a la incorporación de las refactorizaciones, se determinó los siguientes promedios de tiempo de las peticiones agrupadas por funcionalidad.

/ Average	5,755178205
/anonymous/system/sign-in Average	6,519851852
/any/system/welcome Average	3,533789091
/authenticated/system/sign-out Average	5,588193103
/technician/involved-in/create Average	30,31880645
/technician/involved-in/delete Average	31,11727222
/technician/involved-in/list Average	17,00893333
/technician/involved-in/show Average	16,1918625
/technician/maintenance-record/create Average	55,052468
/technician/maintenance-record/delete Average	34,62091667
/technician/maintenance-record/list Average	17,13007576
/technician/maintenance-record/publish Average	53,93988148
/technician/maintenance-record/show Average	17,88360682
/technician/maintenance-record/update Average	59,62474
/technician/task/create Average	17,65615455
/technician/task/delete Average	18,62527143
/technician/task/list Average	12,497225
/technician/task/publish Average	19,84129048
/technician/task/show Average	13,40005294
/technician/task/update Average	17,64935652
Grand Average	16,81820829

Gráficamente se ve de la siguiente forma:



Por tanto, antes de aplicar la refactorización tenemos un intervalo de 95% de nivel de confianza de:

Before	
Mean	16,81820829
Standard Error	0,764318115
Median	10,69005
Mode	12,0822
Standard Deviation	20,22195654
Sample Variance	408,9275265
Kurtosis	6,658118462
Skewness	2,495461626
Range	136,8976
Minimum	2,2417
Maximum	139,1393
Sum	11772,7458
Count	700
Confidence Level(95,0%)	1,500634347

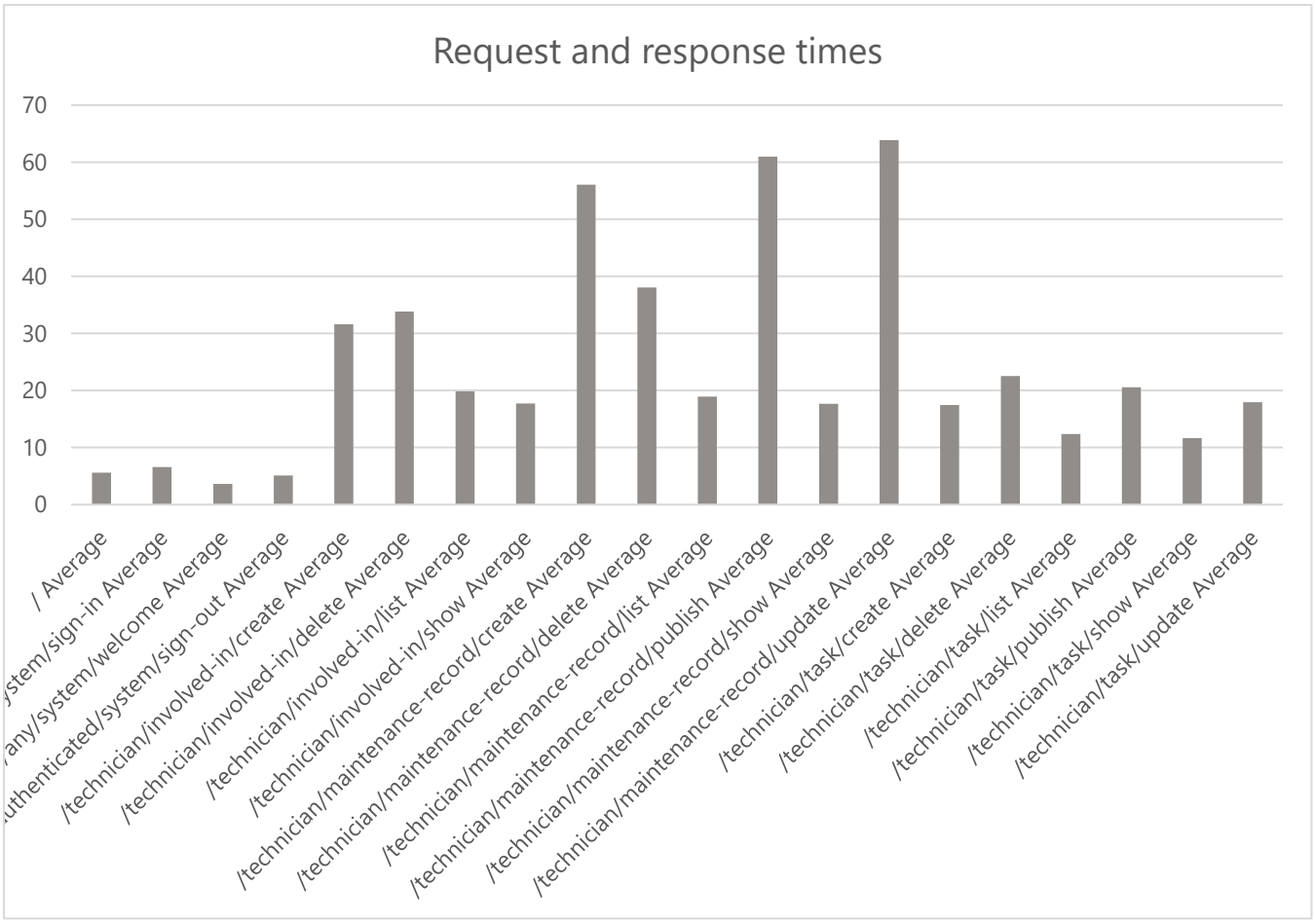
Interval (ms)	15,31757	18,31884
Interval (s)	0,015318	0,018319

Análisis testing before refactoring

Posteriormente a la incorporación de las refactorizaciones, se determinó los siguientes promedios de tiempo de las peticiones agrupadas por funcionalidad:

/ Average	5,610764103
/anonymous/system/sign-in Average	6,567900926
/any/system/welcome Average	3,629423636
/authenticated/system/sign-out Average	5,11657931
/technician/involved-in/create Average	31,62349032
/technician/involved-in/delete Average	33,84617778
/technician/involved-in/list Average	19,835
/technician/involved-in/show Average	17,700275
/technician/maintenance-record/create Average	56,080188
/technician/maintenance-record/delete Average	38,02565
/technician/maintenance-record/list Average	18,91934545
/technician/maintenance-record/publish Average	60,97467407
/technician/maintenance-record/show Average	17,64885909
/technician/maintenance-record/update Average	63,880388
/technician/task/create Average	17,42734545
/technician/task/delete Average	22,55154286
/technician/task/list Average	12,36709167
/technician/task/publish Average	20,5675
/technician/task/show Average	11,68027647
/technician/task/update Average	17,96957826
Grand Average	17,59441343

Gráficamente se ve de la siguiente forma:



Por tanto, tras de aplicar la refactorización tenemos un intervalo de 95% de nivel de confianza de:

After	
Mean	17,59441343
Standard Error	0,822517253
Median	11,12245
Mode	3,1322
Standard Deviation	21,76176102
Sample Variance	473,5742425
Kurtosis	9,403026955
Skewness	2,733384499
Range	184,5031
Minimum	2,2462
Maximum	186,7493
Sum	12316,0894
Count	700
Confidence Level(95,0%)	1,614900416

Interval (ms)	15,97951	19,20931
Interval (s)	0,01598	0,019209

Contraste de hipótesis con un 95% de nivel de confianza

Una vez obtenido los análisis, desde dos ordenadores diferentes, antes y después de implementar los cambios y refactorizaciones, mediante la prueba Z, hemos podido obtener cuál de los intervalos es mejor. La tabla obtenida es la siguiente:

	Before	*After*
Mean	16,81820829	17,59441343
Known Variance	408,9275265	473,5742425
Observations	700	700
Hypothesized Mean Difference	0	
z	-0,691301889	
P(Z<=z) one-tail	0,244687922	
z Critical one-tail	1,644853627	
P(Z<=z) two-tail	0,489375845	
z Critical two-tail	1,959963985	

Hemos tomado nuestro valor Alpha como 0.05. Para poder realizar la comparación necesitamos fijarnos en el valor P(Z<= z) two-tail que es el que nos determinará si los cambios han producido un impacto o si por el contrario ambos promedios de tiempo pueden considerarse iguales.

Tenemos un valor P(Z<= z) two-tail igual a 0,489375845, esto significa que se encuentra en el intervalo [Alpha, 1.00], lo que se traduce en que los cambios realizados no han supuesto un cambio significativo, es decir, la diferencia entre los tiempos antes y después de las refactorizaciones no difieren mucho.

Principalmente los cambios añadidos han sido la implementación de los índices en las entidades implicadas, Technician, MaintenanceRecord, Task e InvolvedIn. La base de datos, a la hora de la realización del análisis no contenía la suficiente información ni cantidad de datos como para que dicho cambio reflejara una mejora del rendimiento del sistema y, por consiguiente, los tiempos se consideran prácticamente los mismos.

Para solucionar esto tenemos varias opciones. Una de ellas sería añadir más datos a la base de datos, aunque es algo ineficiente. Otra opción es monitorizar los casos de pruebas identificando los MIR para después ver cuáles son aquellos métodos más ineficientes e intentar refactorizarlos haciendo el código más simple, por ejemplo. De igual forma, también se puede monitorizar el ordenador mientras se ejecutan los casos de prueba para así determinar los componentes que se utilizan durante todo el tiempo al 100%, de esta forma averiguamos los cuellos de botella.

Una vez apliquemos algunas de las medidas lo ideal sería repetir el análisis y comprobar que efectivamente el valor P(Z<= z) two-tail va disminuyendo hasta que consigamos un buen rendimiento.

CONCLUSIÓN

La realización de esta parte del trabajo ha consistido en añadir cambios, en concreto índices en las entidades Technician, MaintenanceRecord, Task e InvolvedIn, para evaluar como de eficiente son dichos cambios ante el rendimiento del sistema. En primer lugar, se realizó un análisis del rendimiento antes de la implementación de los cambios. Posteriormente, se introdujeron dichos cambios y se realizó un segundo análisis. Y, por último, se llevó a cabo una comparación entre ambas situaciones tomando de base los resultados obtenidos en los pasos previos.

De esta forma, se ha concluido que dichos cambios no han supuesto una mejora significativa principalmente debido a la escasa cantidad de datos en la base de datos (que eran los que se tomaban de base para las pruebas).

Como se especificó anteriormente, posibles soluciones para mejorar el rendimiento serían monitorizar de forma más exhaustiva los casos de pruebas para identificar así los métodos más ineficientes y también aquellas partes del código que generan cuellos de botella. Tras estas mejoras habría que realizar un tercer análisis y de vuelta una comparativa que nos permita apreciar la mejora o no de rendimiento.

BIBLIOGRAFÍA

Intentionally blank