

ACME AirNav Solutions



Testing report - Student 3

Grupo: C1.050

Miembros: Cristina Fernández Chica (criferchi@alum.us.es), Ángel Amo Sánchez (angamosan@alum.us.es), Candela Jazmín Gutiérrez González (canqutgon@alum.us.es), Marta Aguilar Morcillo (maragumor@alum.us.es) y Luis Emmanuel Chávez Malavé (luichamal@alum.us.es)

Repositorio: <https://github.com/Cristinafernandezchica/Acme-ANS>

Planning dashboard: <https://github.com/users/Cristinafernandezchica/projects/1/views/1>

Sevilla 26 Mayo, 2025

Tabla de contenidos

Resumen Ejecutivo.....	3
Tabla de Revisiones	3
Introducción.....	4
Pruebas funcionales	4
Clase Flight Assignment	4
Clase Activity Log	6
Recubrimiento de código	7
Análisis del desempeño.....	8
Conclusión	12

Resumen Ejecutivo

El propósito de este informe es presentar una explicación completa de los distintos pasos llevados a cabo en el marco del testing formal del proyecto Acme-ANS. Esto incluye desde la creación de una cantidad adecuada de datos de prueba, hasta la revisión detallada del cumplimiento de los requisitos funcionales. También se aplicaron diversas herramientas estadísticas para analizar y comparar los tiempos de ejecución de diferentes peticiones, evaluar el rendimiento entre dos equipos distintos y observar los cambios en el desempeño de un mismo equipo antes y después de añadir índices en la base de datos. En resumen, se ha seguido un enfoque riguroso y cuidadoso para detectar y corregir errores a lo largo del proceso de pruebas, con el fin de garantizar un producto final de alta calidad que cumpla con las expectativas del cliente.

Tabla de Revisiones

Número de revisión	Fecha	Descripción de revisión	Autor
1.0	26/06/2025		Candela Jazmín Gutiérrez González

Introducción

En este informe se detallará cómo se llevó a cabo el testing funcional y el análisis del rendimiento, señalando los errores detectados durante el proceso. Esta evaluación se centrará en los requisitos funcionales 8 y 9 del Student #1.

La redacción del informe seguirá la estructura establecida en los anexos de la asignatura. Se dividirá en dos secciones principales: la primera abordará las pruebas funcionales, donde se describirán las evaluaciones realizadas por cada funcionalidad y su grado de efectividad, es decir, cómo ayudaron a identificar fallos en el código. La segunda sección se enfocará en las pruebas de rendimiento, aplicando un nivel de confianza del 95% para comparar los resultados antes y después de optimizar las búsquedas en la base de datos mediante índices. Además, en esta parte se incluirán gráficos con los tiempos de ejecución según cada funcionalidad. Finalmente, se presentará una conclusión basada en el análisis de ambas secciones.

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”:

Clase Flight Assignment

X.safe	Descripción	Bugs detectados
List	Se listaron las asignaciones de vuelo tanto planeadas como completadas. Se emplearon distintos flightCrewMembers para mostrar una lista vacía, otra con una asignación y otra con varias.	No se detectaron bugs.
Show	Se mostró una asignación de vuelo de un flightCrewMember con los atributos de la entidad.	No se detectaron bugs.
Create	Se probó cada variación de cada campo del formulario de creación de asignaciones de vuelo: datos vacíos, límites de los atributos y validaciones.	No se detectaron bugs.
Update	Se probó cada variación de cada campo del formulario de actualización de asignaciones de vuelo: datos vacíos, límites de los atributos y validaciones. Además, se probó la actualización de una asignación de vuelo.	No se detectaron bugs.
Publish	Se probó cada variación de cada campo del formulario de publicación de asignaciones de vuelo: datos vacíos, límites de los atributos y validaciones. Además, se probó la publicación de una asignación de vuelo.	No se detectaron bugs.
Delete	Se probó a eliminar una asignación de vuelo tanto con confirmación como sin él.	No se detectaron bugs.
Y.hack	Descripción	Bugs detectados
List	Se trató de listar las asignaciones de vuelo de otro flightCrewMember, distinto al que inició sesión.	No se detectaron bugs.

Show	<ul style="list-style-type: none"> - Mostrar asignación de vuelo con id nula. - Mostrar asignación de vuelo con id inexistente en base de datos. - Mostrar asignación de vuelo perteneciente a otro flightCrewMember. 	No se detectaron bugs.
Create	<ul style="list-style-type: none"> - Creación de una asignación de vuelo con id de leg nula. - Creación de una asignación de vuelo con id de leg inexistente en base de datos. - Creación de asignación de vuelo cambiado su id por uno existente para sobrescribirlo. 	Se detectó que era posible sobrescribir una asignación de vuelo existente.
Update	<ul style="list-style-type: none"> - Actualización de asignación de vuelo con id vacío. - Actualización de asignación de vuelo con id inexistente en base de datos. - Actualización de asignación de vuelo con id perteneciente a otro flightCrewMember. - Actualización de asignación de vuelo publicada. - Actualización de una asignación de vuelo con id de leg nula. - Actualización de una asignación de vuelo con id de leg inexistente en base de datos. - Actualización de una asignación de vuelo con id de flightCrewsDuty nula. - Actualización de una asignación de vuelo con id de flightCrewsDuty inexistente en base de datos. 	No se detectaron bugs.
Publish	<ul style="list-style-type: none"> - Publicación de asignación de vuelo con id vacío. - Publicación de asignación de vuelo con id inexistente en base de datos. - Publicación de asignación de vuelo con id perteneciente a otro flightCrewMember. - Publicación de asignación de vuelo publicada. - Publicación de una asignación de vuelo con id de leg nula. - Publicación de una asignación de vuelo con id de leg inexistente en base de datos. - Publicación de una asignación de vuelo con id de flightCrewsDuty nula. - Publicación de una asignación de vuelo con id de flightCrewsDuty inexistente en base de datos. - Publicación de una asignación de vuelo con id de CurrentStatus inexistente en base de datos. - Publicación de una asignación de vuelo con id de CurrentStatus nula. 	No se detectaron bugs.

Delete	<ul style="list-style-type: none"> - Borrado de asignación de vuelo con id vacío. - Borrado de asignación de vuelo con id inexistente en base de datos. - Borrado de asignación de vuelo con id perteneciente a otro flightCrewMember. - Borrado de asignación de vuelo publicada. 	Se detectó que no era posible borrar una asignación de vuelo no publicada con activityLogs (caso imposible a menos que sea una asignación introducida por csv).
---------------	--	---

Clase Activity Log

X.safe	Descripción	Bugs detectados
List	Se listaron los reportes de una asignación de vuelo. Se emplearon distintas asignaciones de vuelo para mostrar una lista vacía, otra con una asignación y otra con varias.	No se detectaron bugs.
Show	Se mostró un reporte de una asignación de vuelo con los atributos de la entidad.	No se detectaron bugs.
Create	Se probó cada variación de cada campo del formulario de creación de reportes: datos vacíos, límites de los atributos y validaciones.	No se detectaron bugs.
Update	Se probó cada variación de cada campo del formulario de actualización de reportes: datos vacíos, límites de los atributos y validaciones. Además, se probó la actualización de un reporte.	No se detectaron bugs.
Publish	Se probó cada variación de cada campo del formulario de publicación de asignaciones de vuelo: datos vacíos, límites de los atributos y validaciones. Además, se probó la publicación de un reporte.	No se detectaron bugs.
Delete	Se probó a eliminar un reporte tanto con confirmación como sin él.	No se detectaron bugs.
Y.hack	Descripción	Bugs detectados
List	<ul style="list-style-type: none"> - Mostrar listado de reportes de una asignación de vuelo cuya id es nula. - Mostrar listado de reportes de una asignación de vuelo cuya id es inexistente en base de datos. - Mostrar listado de reportes de una asignación de vuelo perteneciente a otro flightCrewMember. 	No se detectaron bugs.
Show	<ul style="list-style-type: none"> - Mostrar reporte con id nula. - Mostrar reporte con id inexistente en base de datos. - Mostrar reporte perteneciente a otro flightCrewMember. 	No se detectaron bugs.
Create	<ul style="list-style-type: none"> - Creación de asignación de vuelo cambiado su id por uno existente para sobrescribirlo. - Creación de un reporte con id de asignación de vuelo inexistente en base de datos. 	Se detectó que era posible crear un reporte para una asignación de vuelo en borrador.

	<ul style="list-style-type: none"> - Creación de un reporte con id de asignación de vuelo perteneciente a otro flightCrewMember. - Creación de un reporte con id de asignación de vuelo nulo. 	
Update	<ul style="list-style-type: none"> - Actualización de reporte con id vacío. - Actualización de reporte con id inexistente en base de datos. - Actualización de reporte con id perteneciente a otro flightCrewMember. - Actualización de reporte publicada. 	No se detectaron bugs.
Publish	<ul style="list-style-type: none"> - Publicación de reporte con id vacío. - Publicación de reporte con id inexistente en base de datos. - Publicación de reporte con id perteneciente a otro flightCrewMember. - Publicación de reporte publicada. 	No se detectaron bugs.
Delete	<ul style="list-style-type: none"> - Borrado de reporte con id vacío. - Borrado de reporte con id inexistente en base de datos. - Borrado de reporte con id perteneciente a otro flightCrewMember. - Borrado de reporte publicada. 	No se detectaron bugs.

Recubrimiento de código

Las pruebas realizadas son consideradas completas, pues aquellas realizadas a la clase FlightAssignment poseen una cobertura del código del 100%, y las realizadas a la clase ActivityLog un 99'1%. A partir de estos resultados, podemos concluir que el código se ha probado en profundidad debido a una búsqueda exhaustiva de todos los posibles nodos de cada funcionalidad de las clases

acme.features.flightCrewMembe	99,1 %	1.151	10	1.161
> ActivityLogPublishService.java	98,5 %	203	3	206
> ActivityLogUpdateService.java	98,5 %	200	3	203
> ActivityLogCreateService.java	99,3 %	275	2	277
> ActivityLogDeleteService.java	99,0 %	204	2	206
> ActivityLogController.java	100,0 %	35	0	35
> ActivityLogListService.java	100,0 %	113	0	113
> ActivityLogShowService.java	100,0 %	121	0	121
acme.features.flightCrewMembe	100,0 %	2.341	0	2.341
> FlightAssignmentCompletedL	100,0 %	51	0	51
> FlightAssignmentController.ja	100,0 %	42	0	42
> FlightAssignmentCreateServic	100,0 %	477	0	477
> FlightAssignmentDeleteServic	100,0 %	290	0	290
> FlightAssignmentPlannedListS	100,0 %	47	0	47
> FlightAssignmentPublishServi	100,0 %	666	0	666
> FlightAssignmentShowService	100,0 %	192	0	192
> FlightAssignmentUpdateServi	100,0 %	576	0	576

Análisis del desempeño

Para evaluar el rendimiento de la aplicación antes y después de optimizar los índices se evaluará un contraste de hipótesis mediante un Z-test que indicarán si los cambios realizados tienen una mejora significativa.

	<i>pre-índices</i>	<i>post-índices</i>
Media	16,07172439	16,54312348
Varianza (conocida)	153,9420483	171,0672384
Observaciones	328	328
Diferencia hipotética de las m	0	
z	-0,47356301	
P(Z<=z) una cola	0,317905779	
Valor crítico de z (una cola)	1,644853627	
Valor crítico de z (dos colas)	0,635811559	
Valor crítico de z (dos colas)	1,959963985	

El p-valor obtenido en la prueba Z es de 0,317905779259069. Al considerarse un nivel de confianza del 95%, el nivel de significancia es de un 5%. Comparando dichos datos, no se llegó a una conclusión, pues el p-valor no es menor que la significancia ni tampoco supera el 100%, que indicaría el empeoramiento del rendimiento. No existe tal diferencia significativa como para asegurar que ha habido una mejora.

A continuación, se presentan las estadísticas de los tiempos antes y después de la implementación de índices:

<i>pre-índices</i>		<i>post-índices</i>	
Media	16,07172439	Media	16,54312348
Error típico	0,685080662	Error típico	0,722181729
Mediana	13,439	Mediana	13,4235
Moda	13,8394	Moda	13,4473
Desviación estándar	12,40733848	Desviación estándar	13,07926739
Varianza de la muestra	153,9420483	Varianza de la muestra	171,0672354
Curtosis	6,081005187	Curtosis	3,500321278
Coefficiente de asimetría	2,039166225	Coefficiente de asimetría	1,779646623
Rango	91,7678	Rango	79,4046
Mínimo	2,2675	Mínimo	2,3443
Máximo	94,0353	Máximo	81,7489
Suma	5271,5256	Suma	5426,1445
Cuenta	328	Cuenta	328
Nivel de confianza(95,0%)	1,347721579	Nivel de confianza(95,0%)	1,420708471

interval (ms)	14,72400281	17,41944597
interval (s)	0,014724003	0,017419446

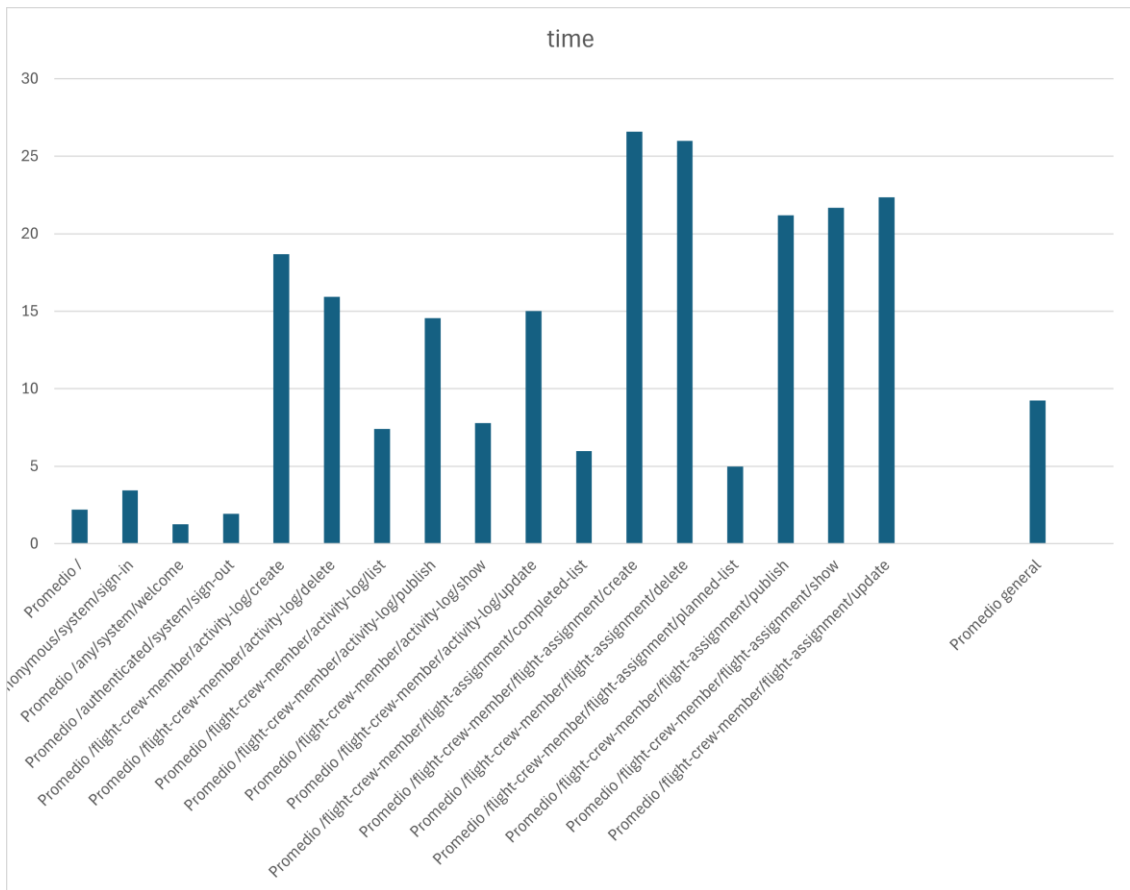
interval (ms)	15,122415	17,9638319
interval (s)	0,015122415	0,01796383

Analizando el intervalo de confianza, podemos asegurar que cada petición será respondida con un tiempo de respuesta igual o menor a 17'41944597 con un 95% de fiabilidad. Sin

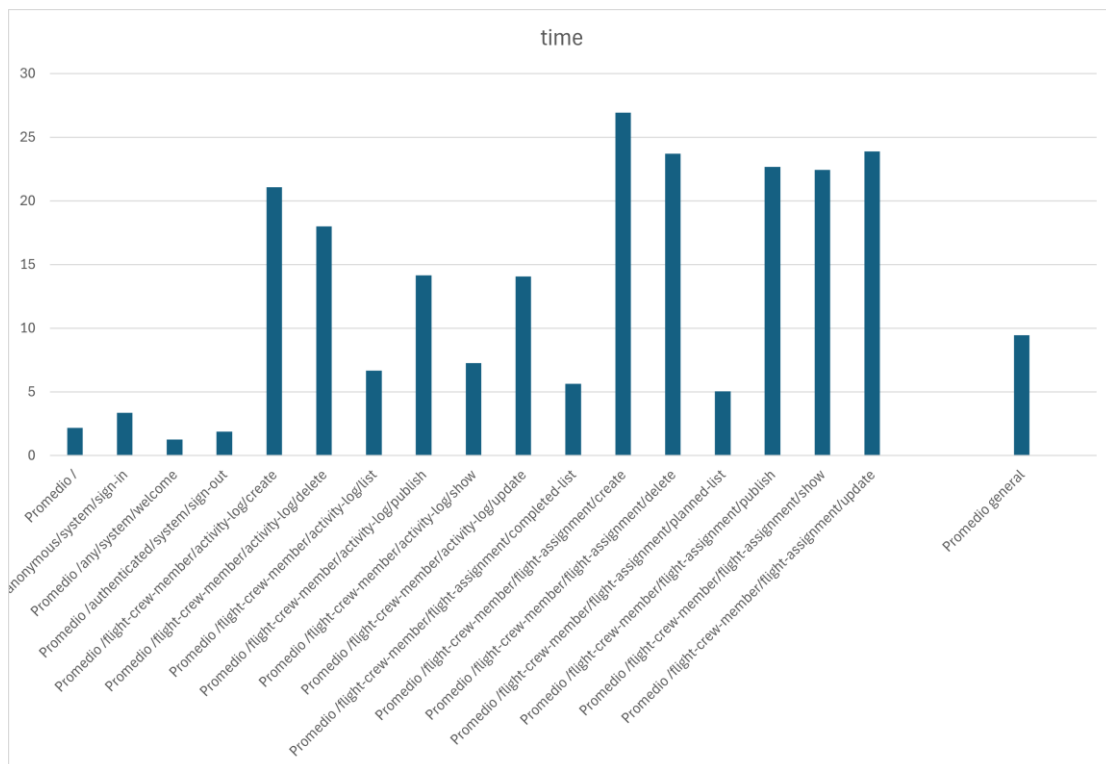
embargo, tras optimizar las búsquedas a base de datos mediante la implementación de índices, no se obtiene una mejora sino un empeoramiento leve, siendo un 0,54438593 ms más lento. Este resultado se puede contrastar con p-valor, el cambio es tan leve que no ha sido posible analizarlo a partir de dicho valor.

Posteriormente, se presentarán los gráficos antes y después de la optimización que indicarán aquellas llamadas MIR que no han sido reducidas:

Pre-índices

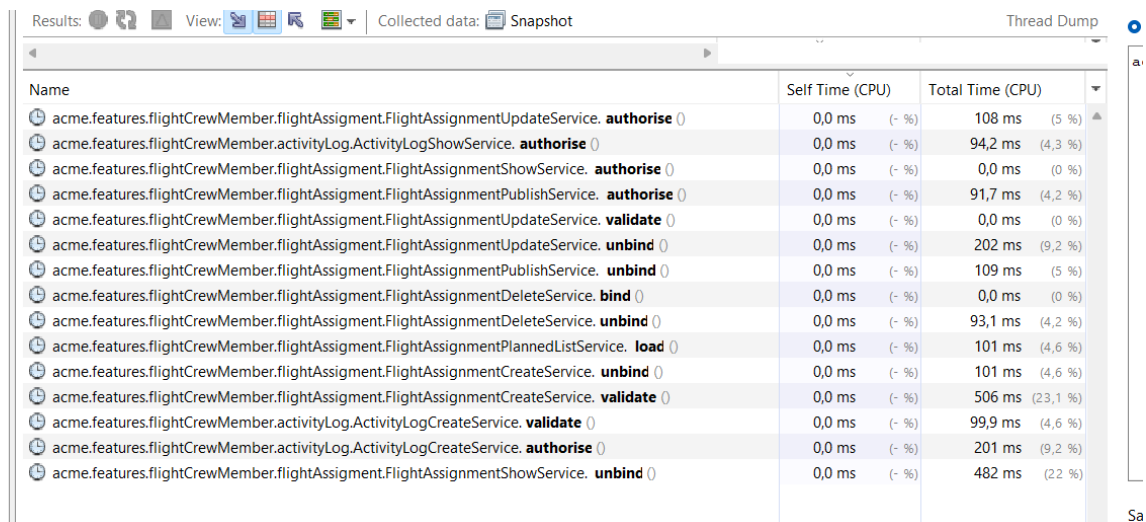


Post-índices



Como se ha concluido anteriormente, en el promedio general se puede observar el leve empeoramiento que se presenta tras realizar la optimización. Por un lado, se mejoran ciertos MIR como `/flight-crew-member/flight-assignment/delete` y `/flight-crew-member/activity-log/update`, sin embargo, también se presentan MIR cuyo tiempo ha aumentado, como `/flight-crew-member/flight-assignment/delete` y `/flight-crew-member/activity-log/delete`.

Al conseguirse una mejora notable con la aplicación de índices, se decidió realizar un análisis más exhaustivo con VisualVM.



The screenshot shows the 'Thread Dump' window in VisualVM. It displays a list of threads with their names, self times, and total times. The threads are sorted by total time in descending order. The methods being executed are mostly `authorize()` and `validate()` methods of various services.

Name	Self Time (CPU)	Total Time (CPU)
acme.features.flightCrewMember.flightAssignment.FlightAssignmentUpdateService. authorize ()	0,0 ms (- %)	108 ms (5 %)
acme.features.flightCrewMember.activityLog.ActivityLogShowService. authorize ()	0,0 ms (- %)	94,2 ms (4,3 %)
acme.features.flightCrewMember.flightAssignment.FlightAssignmentShowService. authorize ()	0,0 ms (- %)	0,0 ms (0 %)
acme.features.flightCrewMember.flightAssignment.FlightAssignmentPublishService. authorize ()	0,0 ms (- %)	91,7 ms (4,2 %)
acme.features.flightCrewMember.flightAssignment.FlightAssignmentUpdateService. validate ()	0,0 ms (- %)	0,0 ms (0 %)
acme.features.flightCrewMember.flightAssignment.FlightAssignmentUpdateService. unbind ()	0,0 ms (- %)	202 ms (9,2 %)
acme.features.flightCrewMember.flightAssignment.FlightAssignmentPublishService. unbind ()	0,0 ms (- %)	109 ms (5 %)
acme.features.flightCrewMember.flightAssignment.FlightAssignmentDeleteService. bind ()	0,0 ms (- %)	0,0 ms (0 %)
acme.features.flightCrewMember.flightAssignment.FlightAssignmentDeleteService. unbind ()	0,0 ms (- %)	93,1 ms (4,2 %)
acme.features.flightCrewMember.flightAssignment.FlightAssignmentPlannedListService. load ()	0,0 ms (- %)	101 ms (4,6 %)
acme.features.flightCrewMember.flightAssignment.FlightAssignmentCreateService. unbind ()	0,0 ms (- %)	101 ms (4,6 %)
acme.features.flightCrewMember.flightAssignment.FlightAssignmentCreateService. validate ()	0,0 ms (- %)	506 ms (23,1 %)
acme.features.flightCrewMember.activityLog.ActivityLogCreateService. validate ()	0,0 ms (- %)	99,9 ms (4,6 %)
acme.features.flightCrewMember.activityLog.ActivityLogCreateService. authorize ()	0,0 ms (- %)	201 ms (9,2 %)
acme.features.flightCrewMember.flightAssignment.FlightAssignmentShowService. unbind ()	0,0 ms (- %)	482 ms (22 %)

A partir de estos datos podemos concluir que en una gran mayoría, el `authorize()` y el `validate` son los métodos con mayor tiempo de ejecución. Esto se podría mejorar si se pudiera mejorar la complejidad de dicho código, pues, el self time es cero en todos los casos, lo que indica que el tiempo invertido solo en el código del método no es el problema, sino aquellos a los que llaman internamente. Por ello, se procedió con la implementación de índices, para disminuir el tiempo de consulta a la base de datos.

Finalmente, se realizó una comparativa con otro ordenador, para analizar el impacto del hardware al responder llamadas. Para ello, se ha utilizado las estadísticas del dispositivo de trabajo para el proyecto del student 3 (ASUS TUF Dash F15 con procesador I7-12650H) con la versión tras la implementación de índices y en el ordenador de mi compañera (HP Victus 16-d1001ns con procesador i7 12700H).

En los datos proporcionados, podemos observar que los intervalos de confianza presentan una diferencia de 9'2977519 de ms, siendo el segundo dispositivo aquel que presenta una cota superior. Debido a ello, concluimos que el desempeño del primer ordenador (el dispositivo de trabajo para el proyecto del student 3) es mejor, al ofrecer tiempos de respuesta significativamente más cortos que los del PC 2.

post-índices		PC 2	
Media	16,54312348	Media	27,26158354
Error típico	0,722181729	Error típico	1,226085287
Mediana	13,4235	Mediana	21,72185
Moda	13,4473	Moda	10,9964
Desviación estándar	13,07926739	Desviación estándar	22,20534897
Varianza de la muestra	171,0672354	Varianza de la muestra	493,0775229
Curtosis	3,500321278	Curtosis	3,651737065
Coefficiente de asimetría	1,779646623	Coefficiente de asimetría	1,700709758
Rango	79,4046	Rango	139,7936
Mínimo	2,3443	Mínimo	2,6906
Máximo	81,7489	Máximo	142,4842
Suma	5426,1445	Suma	8941,7994
Cuenta	328	Cuenta	328
Nivel de confianza(95,0%)	1,420708471	Nivel de confianza(95,0%)	2,41201028
interval (ms)		15,122415	17,9638319
interval (s)		0,015122415	0,01796383
inter (ms)		24,84957326	27,2615835
inter (s)		0,024849573	0,02726158

Conclusión

Este informe demuestra la diversidad de pruebas realizadas para detectar el máximo de errores posible y corregirlos. Las pruebas evidencian la variedad de escenarios probados para los requisitos 8 y 9 del Student 3. Gracias a ello, se pudieron minimizar el riesgo de fallos en el sistema.

En cuanto al rendimiento, se empleó como principal sistema de evaluación la prueba Z con un nivel de confianza del 95%, sin embargo, dio una respuesta no conclusiva, y se debió recurrir a otros tipos de prueba, como las gráficas de llamadas MIR que demostraron la mejora en ciertas peticiones.

En general, podemos concluir que el sistema cuenta con una funcionalidad bastante sólida según las pruebas realizadas. Sin embargo, el rendimiento podría verse afectado en algunos dispositivos con tecnología más antigua, donde podrían presentarse problemas.

Como conclusión, es importante evaluar desde el inicio la eficiencia del sistema cuando se desarrolla un proyecto, ya que al momento de hacer refactorizaciones se podrían introducir varios inconvenientes y terminar con un sistema demasiado lento.

Bibliografía

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