
TESTING REPORT

Entidades *Flight* y *Leg*

Nombre: Francisco Ayala Díaz

Grupo: C1.058

Repositorio: <https://github.com/C1-058/Acme-ANS-D04>

Correo de contacto: fraayadia@alum.us.es

Tabla de contenido

- Testing* funcional..... 3
- Testing* de rendimiento 4
 - Antes 4
 - Después 5
 - Comparación 6
- Software profiling*..... 7
- Hardware profiling* 7

Testing funcional

Para el *testing* de las funcionalidades relacionadas con las entidades Flight y Leg se han grabado casos de prueba, tanto positivos y negativos como de *hacking*. Dichas pruebas son:

Flight

Funcionalidad	+ / -	Hacking
<i>List</i>	Se ha probado a listar los vuelos asociados a dos <i>managers</i> : uno que devuelve una lista vacía y otro que devuelve una lista con elementos.	Se ha probado a listar vuelos tanto habiendo iniciado sesión con un rol distinto a <i>manager</i> , como sin haber iniciado sesión.
<i>Show</i>	Se ha probado a mostrar los detalles de un vuelo concreto.	Se ha probado a mostrar los detalles de un vuelo (no publicado) sin haber iniciado sesión y habiendo iniciado sesión con una cuenta distinta a la que pertenece dicho vuelo.
<i>Create</i>	Se ha probado a rellenar un formulario de creación introduciendo datos incorrectos en cada atributo. Finalmente, se ha creado un vuelo con datos correctos.	Se ha probado a crear un vuelo introduciendo datos inválidos desde consola. También se ha intentado crear un vuelo con una cuenta distinta a <i>manager</i> y sin haber iniciado sesión.
<i>Update</i>	Se ha probado a rellenar un formulario de actualización introduciendo datos incorrectos en cada atributo. Finalmente, se ha actualizado un vuelo con datos correctos.	Se ha probado a actualizar un vuelo introduciendo datos inválidos desde consola. También se ha intentado actualizar un vuelo con una cuenta distinta al <i>manager</i> de dicho vuelo, actualizar un vuelo publicado y actualizar sin haber iniciado sesión.
<i>Delete</i>	Se ha probado a eliminar un vuelo tanto con <i>legs</i> asociadas como sin ellas.	Se ha probado a borrar un vuelo publicado, un vuelo de otro <i>manager</i> y un vuelo sin haber iniciado sesión.
<i>Publish</i>	Se ha probado a publicar un vuelo sin <i>legs</i> asociadas, con <i>legs</i> sin publicar y con todas las <i>legs</i> publicadas.	Se ha probado a publicar un vuelo sin haber iniciado sesión, con una cuenta distinta a la asociada a dicho vuelo y también se ha probado a publicar un vuelo ya publicado.

Leg

Las pruebas para las funcionalidades relacionadas con la entidad *Leg* son análogas a las de *Flight*.

Testing de rendimiento

El análisis de rendimiento se ha realizado dos veces, antes y después de añadir los índices a las entidades *Flight* y *Leg*.

Antes

Promedio de tiempo de respuesta por funcionalidad:

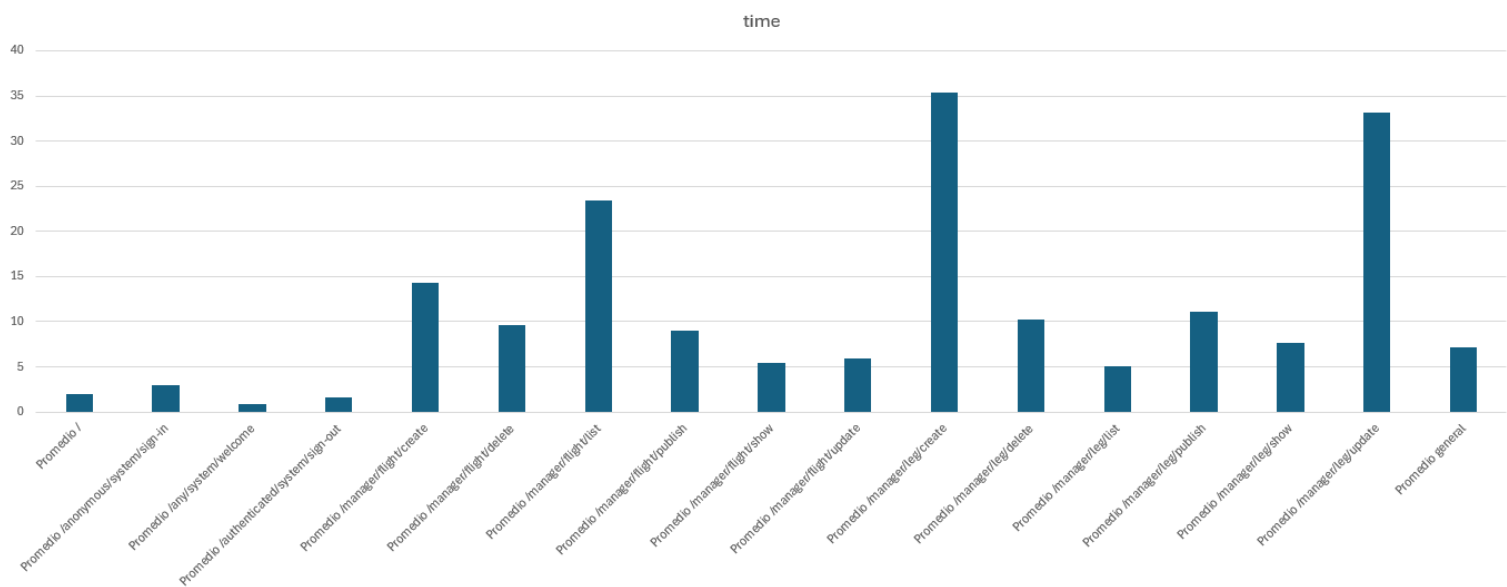


Tabla de análisis:

Columna1	Columna2
Media	7,226397985
Error típico	0,588912128
Mediana	1,74235
Moda	0,9535
Desviación estándar	13,86134525
Varianza de la muestra	192,1368922
Curtosis	11,34973945
Coefficiente de asimetría	3,352828254
Rango	82,0515
Mínimo	0,5223
Máximo	82,5738
Suma	4003,424484
Cuenta	554
Nivel de confianza(95,0%)	1,156778334

Después

Promedio de tiempo de respuesta por funcionalidad:

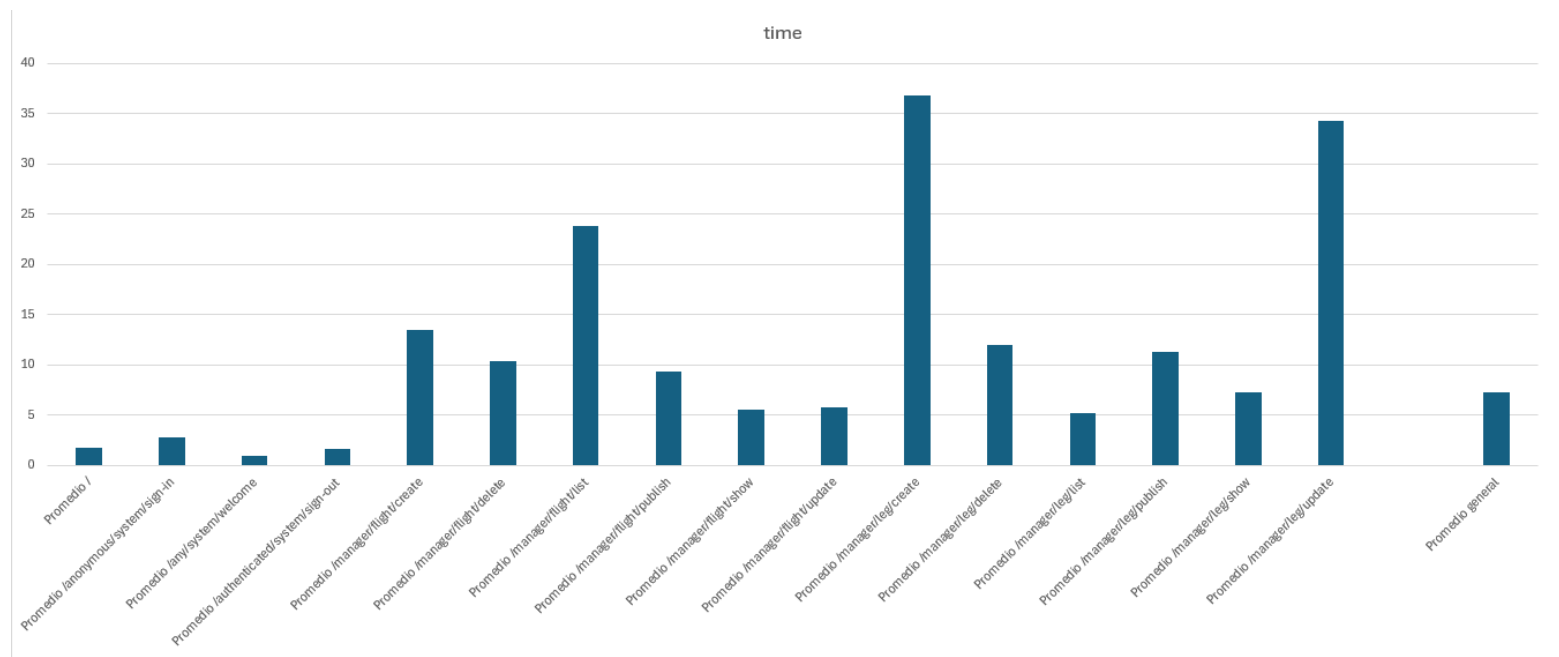


Tabla de análisis:

Columna1	Columna2
Columna1	
Media	7,321429105
Error típico	0,604971648
Mediana	1,74655
Moda	0,7587
Desviación estándar	14,23934145
Varianza de la muestra	202,758845
Curtosis	12,59399922
Coefficiente de asimetría	3,470935147
Rango	93,2498
Mínimo	0,527
Máximo	93,7768
Suma	4056,071724
Cuenta	554
Nivel de confianza(95,0%)	1,188323455

Comparación

Prueba z para medias de dos muestras		
	78,7801	56,3354
Media	7,14398744	7,28160044
Varianza (conocida)	192649413	2033374015
Observaciones	555	555
Diferencia hipotética de las medias	0	
z	-6,871E-05	
P(Z<=z) una cola	0,49997259	
Valor crítico de z (una cola)	1,64485363	
Valor crítico de z (dos colas)	0,99994517	
Valor crítico de z (dos colas)	1,95996398	

Como se puede observar, el valor crítico de z (dos colas) no está en el intervalo [0,0.05]. Esto puede deberse a que las peticiones que más tiempo tardan en completarse, con bastante diferencia del resto, utilizan *queries* que no son optimizadas por los índices.

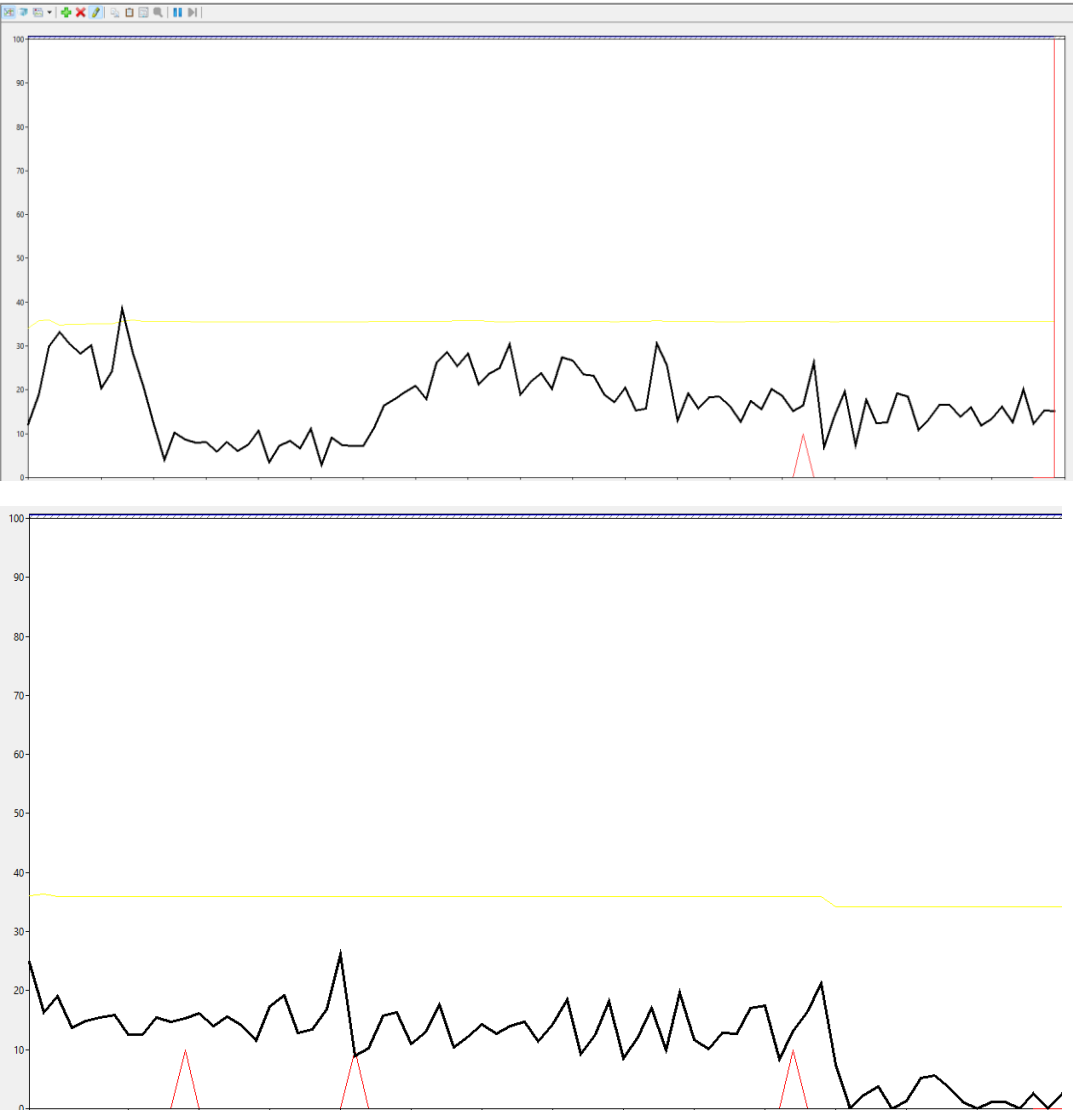
Software profiling

Los resultados del *software profiling* son:

Name	Self Time (CPU)			Total Time (CPU)	
acme.features.manager.leg.ManagerLegCreateService.bind ()	0,0 ms	(- %)		647 ms	(27 %)
acme.features.manager.flight.ManagerFlightListService.unbind ()	0,0 ms	(- %)		627 ms	(26,1 %)
acme.features.manager.leg.ManagerLegCreateService.validate ()	0,0 ms	(- %)		295 ms	(12,3 %)
acme.features.manager.leg.ManagerLegUpdateService.authorise ()	0,0 ms	(- %)		200 ms	(8,4 %)
acme.features.manager.leg.ManagerLegUpdateService.validate ()	0,0 ms	(- %)		198 ms	(8,3 %)
acme.features.manager.leg.ManagerLegShowService.authorise ()	0,0 ms	(- %)		101 ms	(4,2 %)
acme.features.manager.leg.ManagerLegUpdateService.bind ()	0,0 ms	(- %)		100 ms	(4,2 %)
acme.features.manager.flight.ManagerFlightShowService.unbind ()	0,0 ms	(- %)		93,5 ms	(3,9 %)
acme.features.manager.flight.ManagerFlightCreateService.load ()	0,0 ms	(- %)		73,4 ms	(3,1 %)
acme.features.manager.flight.ManagerFlightUpdateService.authorise ()	0,0 ms	(- %)		61,2 ms	(2,6 %)
acme.features.manager.leg.ManagerLegCreateService.unbind ()	0,0 ms	(- %)		0,0 ms	(0 %)

Hardware profiling

Los resultados de *hardware profiling* son:



Leyenda:

- Negro: % de tiempo del procesador
- Rojo: Longitud actual de la cola de disco
- Amarillo: % de bytes confirmados en uso