Introducción a los atributos y valores posibles de las entidades

Entidad: FlightAssignment

- leg: representa el tramo de vuelo asignado. Es obligatorio y en el frontend se representa con un componente <acme:input-select>.
- allocatedFlightCrewMember: es el miembro de la tripulación asignado. Es obligatorio y se muestra con un <acme:input-select readonly="true">.
- duty: indica la función del tripulante (por ejemplo, piloto, copiloto). Es obligatorio y se representa con <acme:input-select>.
- momentLastUpdate: almacena la fecha y hora de la última actualización. Es obligatorio y debe estar en el pasado. No se muestra como campo editable en el frontend.
- currentStatus: representa el estado de la asignación. Es obligatorio y se usa un <acme:input-select> para seleccionarlo.
- remarks: contiene observaciones adicionales. Es opcional y se introduce con un <acme:input-textarea>.
- draftMode: indica si la asignación está en modo borrador. Es obligatorio. No se muestra en el frontend.

Entidad: ActivityLog

- flightAssignment: hace referencia a la asignación de vuelo asociada. Es obligatorio y se muestra con un <acme:input-select readonly="true">.
- registrationMoment: indica la fecha y hora del registro del incidente. Es obligatorio, debe estar en el pasado, y se representa con un <acme:input-moment readonly="true">.
- incidentType: describe el tipo de incidente. Es obligatorio y se introduce mediante un <acme:input-textbox>.
- description: contiene una descripción detallada del incidente. Es obligatorio y se utiliza un <acme:input-textbox>.
- severityLevel: representa la severidad del incidente en una escala del 0 al 10. Es obligatorio y se introduce mediante un <acme:input-integer>.
- draftMode: indica si el registro está en modo borrador. Es obligatorio. No se muestra explícitamente en el fragmento de frontend proporcionado.

Funcionalidades a probar:

Entidad: FlightAssignment

Esta entidad permite gestionar asignaciones de vuelo y cuenta con las siguientes operaciones:

- **List**: muestra todas las asignaciones disponibles para el usuario.

- **List Planned**: muestra únicamente las asignaciones planificadas, es decir, aquellas que aún no se han ejecutado o cerrado.
- **Show**: permite ver los detalles de una asignación específica.
- **Create**: permite crear una nueva asignación de vuelo desde cero.
- **Update**: permite modificar una asignación existente, mientras esté en modo borrador (draftMode = true).
- **Publish**: cambia el estado de una asignación de borrador a definitiva, haciéndola visible o activa en el sistema. Después de publicarla, no se puede modificar.

Entidad: ActivityLog

Esta entidad se utiliza para registrar y gestionar eventos o incidentes ocurridos durante un vuelo asignado. Sus funcionalidades incluyen:

- **List**: muestra todos los registros de actividad creados por el usuario.
- **Show**: permite visualizar los detalles de un log específico.
- **Create**: permite registrar un nuevo evento asociado a una asignación de vuelo.
- **Update**: permite editar un registro existente mientras esté en modo borrador.
- **Delete**: permite eliminar un registro de actividad, solo si aún no ha sido publicado.

Estrategia de pruebas: .safe y .hack

Pruebas .safe – Acciones legales

Estas pruebas están diseñadas para verificar el comportamiento correcto de la aplicación bajo condiciones normales de uso. En ellas se comprueban:

- Los **límites de los atributos** de cada entidad, incluyendo longitudes mínimas y máximas, rangos numéricos y fechas válidas.
- El cumplimiento de las **restricciones de validación definidas en los requisitos**, como campos obligatorios, relaciones entre entidades y formatos válidos.
- El comportamiento esperado de las funcionalidades disponibles (crear, listar, actualizar, publicar, etc.) cuando se usan correctamente.

Pruebas .hack – Acciones ilegales

Estas pruebas están orientadas a comprobar la **resistencia del sistema frente a accesos indebidos** o manipulaciones no autorizadas. Se incluyen dos tipos de intentos:

- **GET hacking**: se intenta acceder directamente mediante la URL a recursos que no deberían ser accesibles por el usuario actual, como por ejemplo ver detalles de asignaciones ajenas.
- **POST hacking**: se realizan envíos de formularios modificando datos en el backend, con el objetivo de simular cambios o creaciones no autorizadas, como actualizar el valor de un objeto como por ejemplo Duty para que tome un valor no

especificado en el enumerado, para comprobar los atributos anotados como Read Only simplemente no se han añadido al método bind.

Resultados de las pruebas

Tanto las pruebas **.safe** como **.hack** se han ejecutado con éxito en las entidades FlightAssignment y ActivityLog. Todas las acciones legales han cumplido con los requisitos funcionales y de validación, y los intentos de acceso o modificación no autorizados han sido correctamente bloqueados por el sistema, demostrando un buen nivel de protección ante vulnerabilidades comunes.

En términos de cobertura de código, se han obtenido los siguientes resultados:

- **ActivityLog**: 90.9% de cobertura.
- **FlightAssignment**: 97.8% de cobertura.

Cabe destacar que servicios como ActivityLogDelete presentan una cobertura de código relativamente baja, debido a que no es posible acceder al método unbind de dicho servicio, ni siquiera como usuario autorizado ni como uno no autorizado. Esta limitación técnica impide alcanzar una cobertura completa en ese segmento, aunque no afecta al funcionamiento correcto del sistema ni a su seguridad.

acme.features.flightcrewmember.flightassignment	97,8 %	1.682	38	1.720
FlightAssignmentUpdateService.java	96,5 %	384	14	398
FlightAssignmentsCreateService.java	97,0 %	386	12	398
FlightAssignmentPublishService.java	98,4 %	562	9	571
FlightAssignmentShowService.java	98,1 %	154	3	157
FlightAssignmentsController.java	100,0 %	36	0	36
FlightAssignmentsLegLandedListService.java	100,0 %	79	0	79
FlightAssignmentsLegPlannedListService.java	100,0 %	81	0	81
acme.features.flightcrewmember.activitylog	90,9 %	849	85	934
ActivityLogDeleteService.java	58,3 %	91	65	156
ActivityLogListService.java	85,3 %	81	14	95
ActivityLogCreateService.java	97,3 %	218	6	224
ActivityLogController.java	100,0 %	35	0	35
ActivityLogPublishService.java	100,0 %	153	0	153
ActivityLogShowService.java	100,0 %	118	0	118
ActivityLogUpdateService.java	100,0 %	153	0	153

Pruebas de rendimiento

Posteriormente, se han llevado a cabo pruebas de rendimiento con el objetivo de optimizar el acceso a los datos y mejorar los tiempos de respuesta de los distintos servicios involucrados. Para ello, se han introducido índices en los campos clave de las entidades, acelerando de forma significativa las consultas a la base de datos, especialmente en las consultas de filtrado y búsqueda.

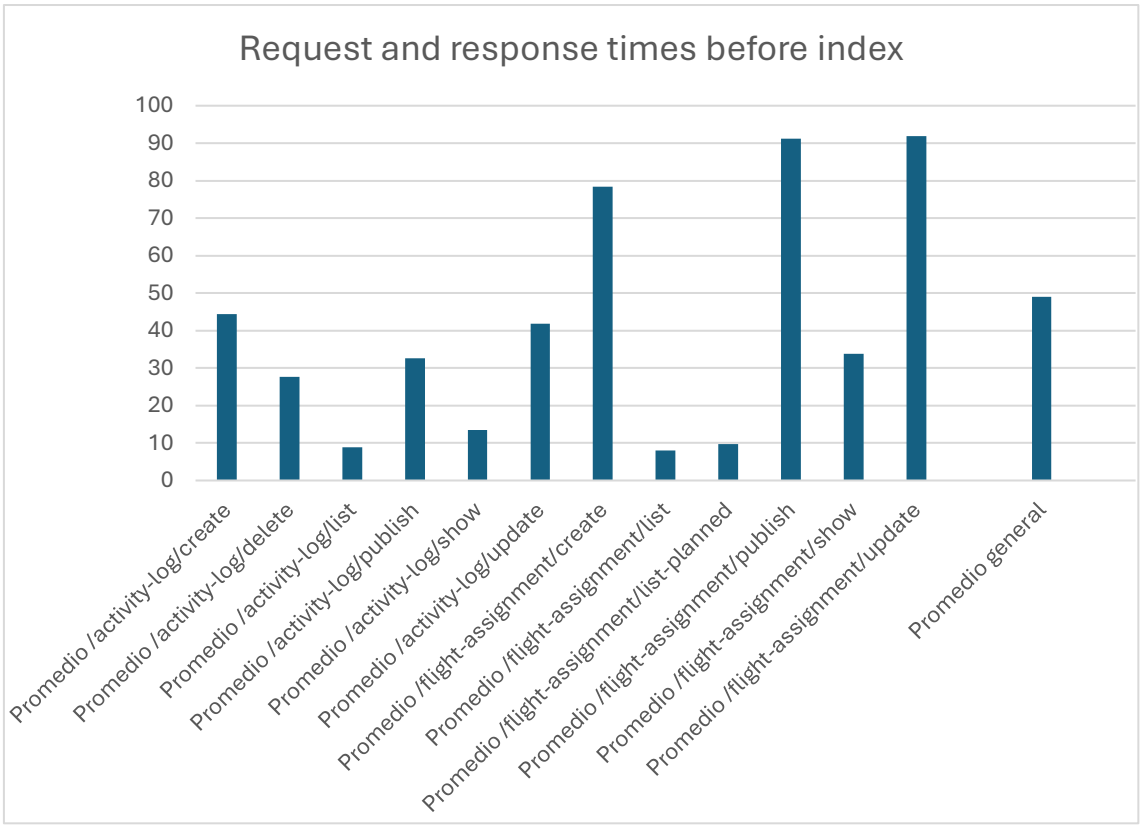
Como resultado, se ha conseguido una mejora notable en los tiempos de ejecución, lo que impacta positivamente en la experiencia del usuario y en la eficiencia general del sistema.

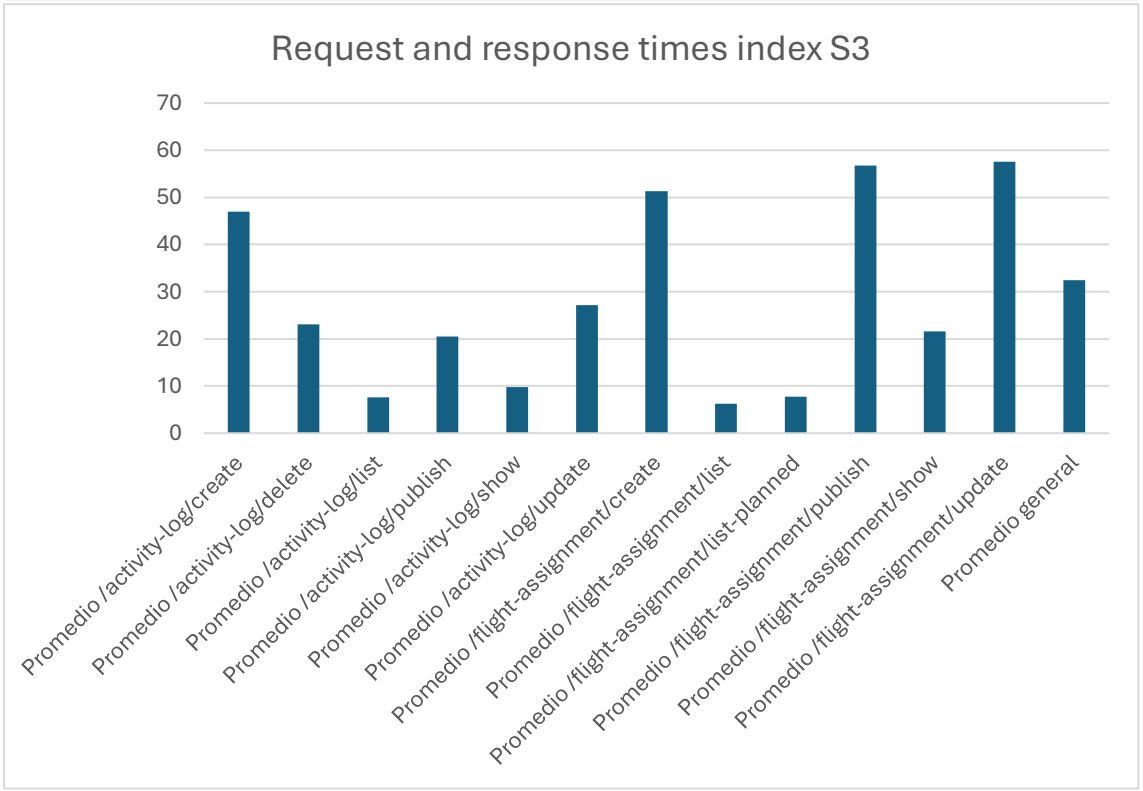
Análisis de la mejora

Para complementar las pruebas de rendimiento, se ha realizado un análisis estadístico utilizando Excel y la prueba Z (Z-test) con el objetivo de validar si la mejora en los tiempos de respuesta tras la introducción de los índices es estadísticamente significativa.

El resultado del análisis ha arrojado un valor $Pz < 0.00$, lo cual permite rechazar la hipótesis nula y concluir que los cambios aplicados han tenido un efecto significativo y positivo en el rendimiento del sistema.

Este resultado se refleja de forma clara en las gráficas de tiempos, donde se observa que servicios como FlightAssignment -> publish y update, que son los métodos que más consultas realizan a los repositorios y poseen queries más complejas, superaban los 90 milisegundos antes de la optimización. Tras la introducción de los índices, dichos tiempos se reducen de forma consistente, apenas alcanzando los 60 milisegundos.


Request and response times before index
100
90
80
70
60
50
40
30
20
10
0
Promedio /activity-log/create
Promedio /activity-log/delete
Promedio /activity-log/list
Promedio /activity-log/publish
Promedio /activity-log/show
Promedio /activity-log/update
Promedio /flight-assignment/create
Promedio /flight-assignment/list
Promedio /flight-assignment/list-planned
Promedio /flight-assignment/publish
Promedio /flight-assignment/show
Promedio /flight-assignment/update
Promedio general


Request and response times index S3
70
60
50
40
30
20
10
0
Promedio /activity-log/create
Promedio /activity-log/delete
Promedio /activity-log/list
Promedio /activity-log/publish
Promedio /activity-log/show
Promedio /activity-log/update
Promedio /flight-assignment/create
Promedio /flight-assignment/list
Promedio /flight-assignment/list-planned
Promedio /flight-assignment/publish
Promedio /flight-assignment/show
Promedio /flight-assignment/update
Promedio general

After	
Media	34,6336991
Error típico	2,06163028
Mediana	19,8632
Moda	#N/D
Desviación es	32,7922371
Varianza de la	1075,33081
Curtosis	1,89654044
Coeficiente de	1,32219718
Rango	187,9101
Mínimo	3,9914
Máximo	191,9015
Suma	8762,32587
Cuenta	253
Nivel de confia	4,06022079
Interval (ms)	30,5734783
Interval(s)	0,03057348

Before	
Media	48,6129188
Error típico	2,80996676
Mediana	27,6299
Moda	#N/D
Desviación es	44,6952574
Varianza de la	1997,66604
Curtosis	-0,55669657
Coeficiente de	0,84391935
Rango	199,3698
Mínimo	4,2034
Máximo	203,5732
Suma	12299,0685
Cuenta	253
Nivel de confia	5,53401138
Interval (ms)	43,0789074
Interval(s)	0,04307891

Prueba z para medias de dos muestras

	After	*before*
Media	34,63369908	48,6129188
Varianza (conocida)	1075,330814	1997,66604
Observaciones	253	253
Diferencia hipotética de las medias	0	
z	-4,011087476	
P(Z<=z) una cola	3,02199E-05	
Valor crítico de z (una cola)	1,644853627	
Valor crítico de z (dos colas)	6,04397E-05	
Valor crítico de z (dos colas)	1,959963985	

Software Profiling

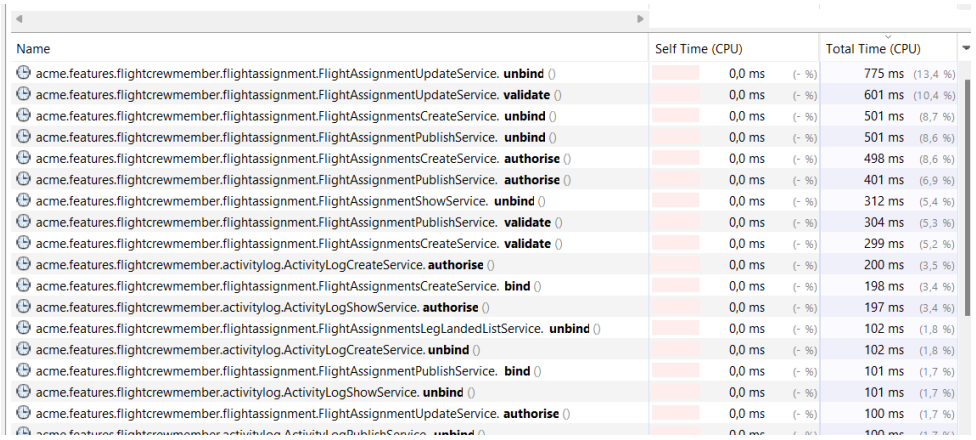
Además de las pruebas funcionales y de rendimiento, se ha realizado un test de Software Profiling con el objetivo de analizar el consumo de recursos y la eficiencia del sistema a nivel de ejecución. Esta técnica ha permitido identificar los métodos con mayor carga de procesamiento, el uso de memoria y la distribución del tiempo de CPU entre las distintas operaciones.

El *profiling* ha sido útil para confirmar que los puntos críticos del sistema (como los métodos publish y update de FlightAssignment) no solo hacen más llamadas a repositorio, sino que también concentran una mayor parte del uso de recursos.

Como puede observarse en las dos capturas adjuntas, la introducción de índices en la base de datos ha tenido un efecto directo y positivo en el rendimiento del sistema. Las gráficas y métricas muestran de forma clara que los tiempos de respuesta son significativamente menores cuando los índices están aplicados, en comparación con la versión sin optimización.

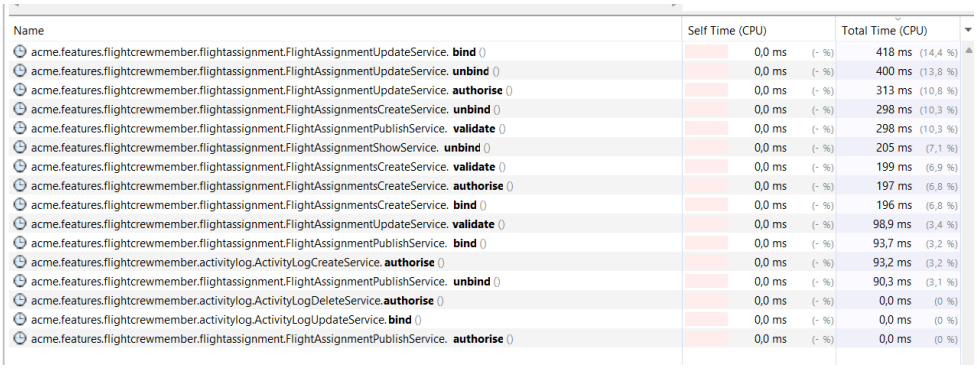
Los métodos más exigentes, como publish y update en la entidad FlightAssignment, reducen su tiempo de ejecución tras la aplicación de los índices, demostrando así que el sistema responde más rápido y con mayor eficiencia en los puntos donde más lo necesita.

Sin índices:



Name	Self Time (CPU)	Total Time (CPU)
acme.features.flightcrewmember.flightassignment.FlightAssignmentUpdateService. **unbind** ()	0,0 ms (- %)	775 ms (13,4 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentUpdateService. **validate** ()	0,0 ms (- %)	601 ms (10,4 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentsCreateService. **unbind** ()	0,0 ms (- %)	501 ms (8,7 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentPublishService. **unbind** ()	0,0 ms (- %)	501 ms (8,6 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentsCreateService. **authorise** ()	0,0 ms (- %)	498 ms (8,6 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentPublishService. **authorise** ()	0,0 ms (- %)	401 ms (6,9 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentShowService. **unbind** ()	0,0 ms (- %)	312 ms (5,4 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentPublishService. **validate** ()	0,0 ms (- %)	304 ms (5,3 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentsCreateService. **validate** ()	0,0 ms (- %)	299 ms (5,2 %)
acme.features.flightcrewmember.activitylog.ActivityLogCreateService. **authorise** ()	0,0 ms (- %)	200 ms (3,5 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentsCreateService. **bind** ()	0,0 ms (- %)	198 ms (3,4 %)
acme.features.flightcrewmember.activitylog.ActivityLogShowService. **authorise** ()	0,0 ms (- %)	197 ms (3,4 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentsLegLandedListService. **unbind** ()	0,0 ms (- %)	102 ms (1,8 %)
acme.features.flightcrewmember.activitylog.ActivityLogCreateService. **unbind** ()	0,0 ms (- %)	102 ms (1,8 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentPublishService. **bind** ()	0,0 ms (- %)	101 ms (1,7 %)
acme.features.flightcrewmember.activitylog.ActivityLogShowService. **unbind** ()	0,0 ms (- %)	101 ms (1,7 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentUpdateService. **authorise** ()	0,0 ms (- %)	100 ms (1,7 %)

Con índices:



Name	Self Time (CPU)	Total Time (CPU)
acme.features.flightcrewmember.flightassignment.FlightAssignmentUpdateService. **bind** ()	0,0 ms (- %)	418 ms (14,4 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentUpdateService. **unbind** ()	0,0 ms (- %)	400 ms (13,8 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentUpdateService. **authorise** ()	0,0 ms (- %)	313 ms (10,8 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentsCreateService. **unbind** ()	0,0 ms (- %)	298 ms (10,3 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentPublishService. **validate** ()	0,0 ms (- %)	298 ms (10,3 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentShowService. **unbind** ()	0,0 ms (- %)	205 ms (7,1 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentsCreateService. **validate** ()	0,0 ms (- %)	199 ms (6,9 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentsCreateService. **authorise** ()	0,0 ms (- %)	197 ms (6,8 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentsCreateService. **bind** ()	0,0 ms (- %)	196 ms (6,8 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentUpdateService. **validate** ()	0,0 ms (- %)	98,9 ms (3,4 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentPublishService. **bind** ()	0,0 ms (- %)	93,7 ms (3,2 %)
acme.features.flightcrewmember.activitylog.ActivityLogCreateService. **authorise** ()	0,0 ms (- %)	93,2 ms (3,2 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentPublishService. **unbind** ()	0,0 ms (- %)	90,3 ms (3,1 %)
acme.features.flightcrewmember.activitylog.ActivityLogDeleteService.**authorise** ()	0,0 ms (- %)	0,0 ms (0 %)
acme.features.flightcrewmember.activitylog.ActivityLogUpdateService.**bind** ()	0,0 ms (- %)	0,0 ms (0 %)
acme.features.flightcrewmember.flightassignment.FlightAssignmentPublishService. **authorise** ()	0,0 ms (- %)	0,0 ms (0 %)

Informe de Hardware Profiling

1. Current Disk Queue Length (verde)
Durante la ejecución, se observaron pequeños picos esporádicos, lo que indica que el acceso a disco no representa un cuello de botella crítico, aunque sí existen momentos puntuales de mayor carga.

2. Bytes Total/sec (azul)

Al igual que con la cola de disco, se detectaron picos ocasionales, generalmente coincidiendo con operaciones más complejas como la actualización o publicación de asignaciones. Estos picos son normales y no representan una degradación del sistema.

3. % Committed Bytes In Use (rosa)

La línea se mantiene estable alrededor del 90%, lo que sugiere que el sistema mantiene una carga de memoria constante, sin fugas ni consumo excesivo. Este comportamiento es positivo, ya que refleja estabilidad en el uso de memoria RAM.

4. % Processor Time (rojo)

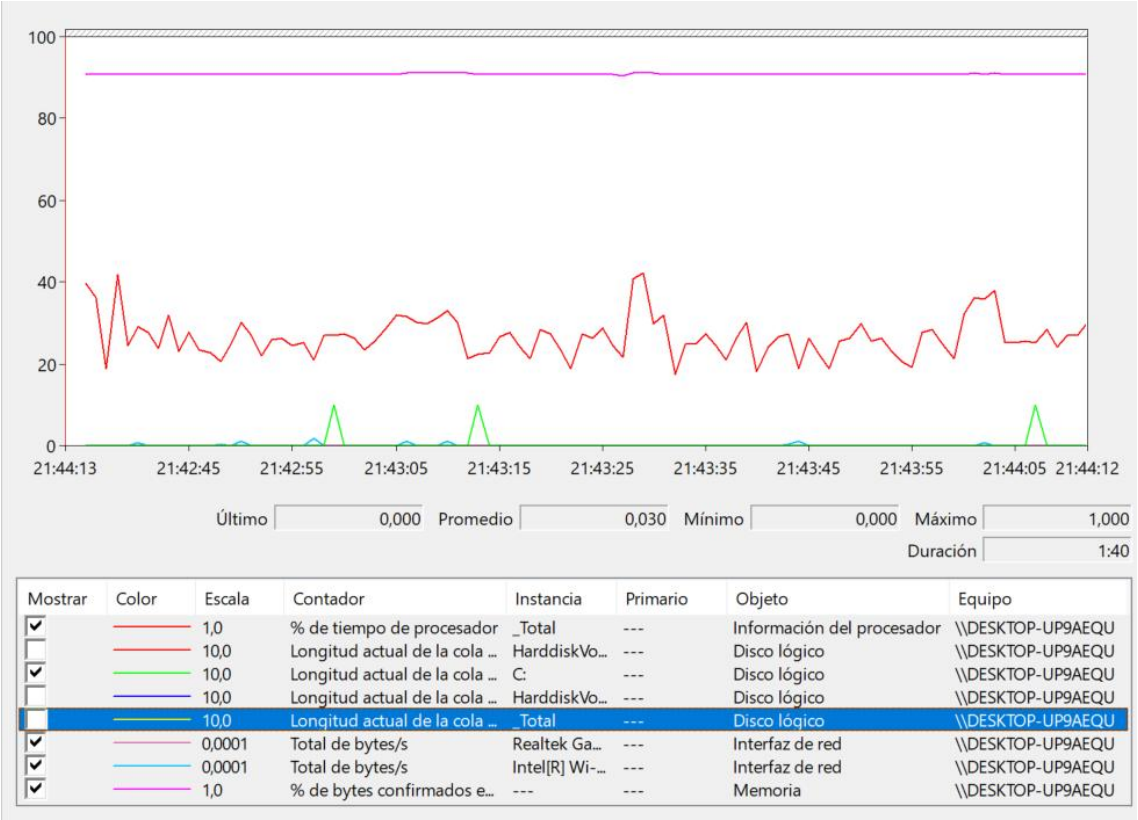
Representa el **porcentaje del tiempo que el procesador está ocupado ejecutando procesos activos**.
Este fue el contador más inestable, oscilando entre 20% y 40%, lo cual es esperable durante operaciones intensivas. Aunque se observan variaciones, los valores se mantienen dentro de un rango razonable y no indican saturación.

Conclusión.

La memoria RAM se mantiene constante en torno al 90% de uso, sin sobrecargas.

El procesador experimenta oscilaciones normales durante operaciones pesadas, pero sin alcanzar niveles críticos.

Los accesos a disco y el flujo de datos presentan picos puntuales, pero no sostenidos, lo que indica que el sistema responde adecuadamente a los momentos de mayor actividad.



Mostrar	Color	Escala	Contador	Instancia	Primario	Objeto	Equipo
☑		1,0	% de tiempo de procesador	_Total	---	Información del procesador	\\DESKTOP-UP9AEQU
☐		10,0	Longitud actual de la cola ...	HarddiskVo...	---	Disco lógico	\\DESKTOP-UP9AEQU
☑		10,0	Longitud actual de la cola ...	C:	---	Disco lógico	\\DESKTOP-UP9AEQU
☐		10,0	Longitud actual de la cola ...	HarddiskVo...	---	Disco lógico	\\DESKTOP-UP9AEQU
☐		10,0	Longitud actual de la cola ...	_Total	---	Disco lógico	\\DESKTOP-UP9AEQU
☑		0,0001	Total de bytes/s	Realtek Ga...	---	Interfaz de red	\\DESKTOP-UP9AEQU
☑		0,0001	Total de bytes/s	Intel[R] Wi-...	---	Interfaz de red	\\DESKTOP-UP9AEQU
☑		1,0	% de bytes confirmados e...	---	---	Memoria	\\DESKTOP-UP9AEQU