

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

D04 – Testing Report



Grado en Ingeniería Informática – Ingeniería del Software

Diseño y Pruebas II

Curso 2023 – 2024

Fecha	Versión
08/07/2024	v1r2

Grupo de prácticas: C2.X02	
Autores	Correo Corporativo
José Ramón Baños Botón	josbanbot@alum.us.es

Link repositorio: <https://github.com/SoniaRM/Acme-SF-D04-24.5.0>



Índice

1. Tabla de versiones.....	3
2. Resumen ejecutivo	3
3. Introducción.....	5
4. Functional testing	5
5. Performance testing	10
6. Conclusiones.....	17
7. Bibliografía.....	18



1. Tabla de versiones

Fecha	Versión	Descripción
26/05/2024	v1r0	Creación del documento
27/05/2024	v1r1	Entrega 1
08/07/2024	v1r2	Modificación del contenido de acuerdo con los cambios realizados para subsanar los errores encontrados

2. Resumen ejecutivo

Para este documento de testing se han desarrollado y explicado cada una de las implementaciones presentes en los requisitos obligatorios de la entrega D04.



Se han evaluado tanto el rendimiento del sistema como el desempeño funcional de todas y cada una de las funciones solicitadas, en este caso, las relativas a “training module” y a “training session”. Para ello se siguió la metodología proporcionada en “S01 - Formal testing” y “S02 - Performance testing”.

El sistema muestra un comportamiento robusto en términos de funcionalidad pese a haber áreas que podrían precisar de una ligera toma de atención.



3. Introducción

Este documento se divide en dos secciones distintas:

1. Functional testing: se presentará un listado con los casos de prueba implementados, agrupados por funcionalidad. Por cada uno se dará una descripción y una indicación de cuan efectivo es detectando errores. Para la efectividad, se usará el coverage del código para comprobar que se han probado todas las decisiones posibles durante la ejecución del programa y así evitar la existencia de errores.

2. Performance testing: se proporcionarán los gráficos adecuados y un intervalo de confianza del 95% para el tiempo tomado para las solicitudes en las pruebas en dos ordenadores distintos. Además, tras las pruebas en los diferentes ordenadores, se indicará cual de estos es el más potente y ofrece mejor rendimiento.

4. Functional testing

Casos de pruebas relativos a training module:

- Create training module:
 - Descripción: Se prueban las restricciones de todos los campos del formulario de



creación de un training module con valores relativos a casos positivos, negativos y de hacking. Se valida que un usuario registrado con otro rol diferente a developer y no registrado no tenga acceso a esta funcionalidad.

- Coverage: 93.8%
- Efectividad: Alta. Entendemos que un coverage alrededor del 95% es un valor alto de efectividad. Vemos que lo que baja el coverage son las líneas de “assert object != null” que están en amarillo. Esto nos indica que, en ningún caso al realizar la función en la que se encuentra esta línea de código, se ha obtenido un objeto nulo. Por otra parte, dentro del método de validación, vemos que en la línea para comprobar que el updateMoment no sea nulo al crearlo, no completa todas las posibilidades. Esto es debido a que al crear un training module, se ha puesto que por defecto asigne a la variable updateMoment el momento actual, luego en ningún momento se podría crear un training module con updateMoment a null.
- Delete training module:
 - Descripción: Se prueba la eliminación de un training module. Se valida que un usuario registrado con otro rol diferente a developer y no registrado y un developer no propietario de un training module ajeno no tenga acceso a esta funcionalidad.
 - Coverage: 92.9%
 - Efectividad: Alta. Misma situación con “assert object != null”.
- List my training modules:
 - Descripción: Se prueba el listado de training modules pertenecientes al developer que ha iniciado sesión en el sistema. Se valida que un usuario registrado con otro rol diferente a developer y otro no registrado no tenga acceso a esta funcionalidad, y que otro developer solo pueda ver sus training modules.
 - Coverage: 95.6%
 - Efectividad: Alta. Misma situación con “assert object != null”.
- Publish training module:
 - Descripción: Se prueba la publicación de un training module teniendo en cuenta restricciones como que no se podrá publicar un training module si tiene al menos un training session sin publicar o si no tiene training sessions asociados. Se valida que un developer que no es propietario de un training module ajeno no tenga acceso a esta funcionalidad y que otro usuario registrado o no registrado que no es propietario de un training module ajeno no tenga acceso a esta funcionalidad tampoco.
 - Coverage: 94.3%
 - Efectividad: Alta. Misma situación con “assert object != null”.
- Show training module details
 - Descripción: Se prueba la muestra de detalles de un training module del que el developer es propietario. Se valida que un developer que no es propietario de un training module ajeno no tenga acceso a esta funcionalidad. Se valida, además, que otro usuario registrado o no registrado que no es propietario de un training module ajeno no tenga acceso a esta funcionalidad.



- Coverage: 96.8%
- Efectividad: Alta. Misma situación con “assert object != null”. Además, en la autorización, observamos en amarillo la línea “status = super.getRequest().getPrincipal().hasRole(developer) && object != null;”. Esto puede ser debido a un problema de Eclipse, ya que se han probado tanto los casos en los que se intenta acceder al show de un training module con otro rol que no sea el de developer, y con otro developer diferente y, con un training module con id nula en la URL.
- Update training module:
 - Descripción: Se prueban las restricciones de todos los campos del formulario de actualización de un training module con valores relativos a casos positivos, negativos y de hacking. Se valida que un developer que no es propietario de un training module ajeno no tenga acceso a esta funcionalidad y que otro usuario registrado o no registrado que no es propietario de un training module ajeno no tenga acceso a esta funcionalidad tampoco.
 - Coverage: 93.4%
 - Efectividad: Alta. Misma situación con “assert object != null”. También encontramos que la línea “if (!super.getBuffer().getErrors().hasErrors("updateMoment") && object.getUpdateMoment() != null)” está en amarillo. Esto se debe a que en las pruebas nunca se puede obtener la variable updateMoment como null, puesto que se ha definido que al actualizar un training module tome como updateMoment la fecha del sistema.

▼ acme.features.developer.trainingModule	94,4 %
> DeveloperTrainingModuleController.java	100,0 %
> DeveloperTrainingModuleCreateService.java	93,8 %
> DeveloperTrainingModuleDeleteService.java	92,9 %
> DeveloperTrainingModuleListMineService.java	95,6 %
> DeveloperTrainingModulePublishService.java	94,3 %
> DeveloperTrainingModuleShowService.java	96,8 %
> DeveloperTrainingModuleUpdateService.java	93,4 %

Casos de pruebas relativos a registros de training session:

- Create training session:
 - Descripción: Se prueban las restricciones todos los campos del formulario de creación de un training session con valores relativos a casos positivos, negativos y de hacking. Se valida que un developer que no es propietario de un training module ajeno no tenga acceso a esta funcionalidad y que otro usuario registrado o no registrado que no es propietario de un training module ajeno no tenga acceso a esta funcionalidad tampoco.
 - Coverage: 94.9%
 - Efectividad: Alta. Misma situación con “assert object != null”. Además de esto, la línea del código de error en la validación de la variable startPeriod está en amarillo: “if (!super.getBuffer().getErrors().hasErrors("startPeriod") && object.getStartPeriod()

!= null)”. Esto puede ser debido a que esta variable nunca puede obtener un valor null, puesto que se ha definido que al crear el training session esta variable tome la fecha del sistema. Ocurre igual con la variable updateMoment. También encontramos en amarillo la siguiente línea: “if (object.getTrainingModule() != null)” esto es debido a que al crear un training session siempre tiene que tener asociada un training module, luego el valor de getTrainingModule no podrá ser nulo.

- Delete training session:
 - Descripción: Se prueba la eliminación de un training session. Se valida que un developer que no es propietario de un training session ajeno no tenga acceso a esta funcionalidad y que otro usuario registrado o no registrado que no es propietario de un training session ajeno no tenga acceso a esta funcionalidad tampoco.
 - Coverage: 91.8%
 - Efectividad: Media. Misma situación con “assert object != null”. Además, la línea del código de error en la autorización: “status = trainingModule != null && trainingModule.isDraftMode() && object.isDraftMode() && super.getRequest().getPrincipal().hasRole(trainingModule.getDeveloper());” se encuentra en color amarillo. Esto es debido a que al existir el training session en modo borrador, existirá un training module en modo borrador también, luego no se podrá intentar borrar nunca una training session en modo no borrador que tenga asociado un training module en modo borrador.
- List my training sessions:
 - Descripción: Se prueba el listado de training sessions pertenecientes a un training module que pertenece a su vez al developer que ha iniciado sesión en el sistema. Se valida que un developer que no es propietario de un training session ajeno no tenga acceso a esta funcionalidad y que otro usuario registrado o no registrado que no es propietario de un training session ajeno no tenga acceso a esta funcionalidad tampoco.
 - Coverage: 95.8%
 - Efectividad: Alta. Misma situación con “assert object != null”.
- List all my training sessions:
 - Descripción: Se prueba el listado de training sessions pertenecientes al developer que ha iniciado sesión en el sistema. Se valida que un developer que no es propietario de un training session ajeno no tenga acceso a esta funcionalidad y que otro usuario registrado o no registrado que no es propietario de un training session ajeno no tenga acceso a esta funcionalidad tampoco.
 - Coverage: 95.8%
 - Efectividad: Alta. Misma situación con “assert object != null”.
- Publish training session:
 - Descripción: Se prueba la publicación de un training session. Se valida que un developer que no es propietario de un training session ajeno no tenga acceso a esta funcionalidad y que otro usuario registrado o no registrado que no es propietario de



un training session ajeno no tenga acceso a esta funcionalidad tampoco.

- Coverage: 95.0%
 - Efectividad: Alta. Misma situación con “assert object != null”. Misma situación con las variables getTrainingModule, creationMoment y updateMoment que para la creación de un training session.
- Show training session details:
 - Descripción: Se prueba la muestra de detalles de un training session del que el developer es propietario. Se valida que un developer que no es propietario de un training session ajeno no tenga acceso a esta funcionalidad y que otro usuario registrado o no registrado que no es propietario de un training session ajeno no tenga acceso a esta funcionalidad tampoco.
 - Coverage: 96.9%
 - Efectividad: Alta. Misma situación con “assert object != null” y “status = object != null && ...”. Además, en la autorización, encontramos la siguiente línea amarilla: “status = super.getRequest().getPrincipal().hasRole(trainingModule.getDeveloper()) && object != null;”. Esto es debido a que un training session tiene asociado un training module, luego éste no puede ser nulo.
 - Update training session:
 - Descripción: Se prueban las restricciones de todos los campos del formulario de actualización de un training session con valores relativos a casos positivos, negativos y de hacking. Se valida que un developer que no es propietario de un training session ajeno no tenga acceso a esta funcionalidad y que otro usuario registrado o no registrado que no es propietario de un training session ajeno no tenga acceso a esta funcionalidad tampoco.
 - Coverage: 94.9%
 - Efectividad: Alta. Misma situación con “assert object != null”. Misma situación con “status = super.getRequest().getPrincipal().hasRole(trainingModule.getDeveloper()) && object != null;” y la validación de las propiedades getTrainingModule, creationMoment y updateMoment.

▼	acme.features.developer.trainingSession	95,0 %
>	DeveloperTrainingSessionController.java	100,0 %
>	DeveloperTrainingSessionCreateService.java	94,9 %
>	DeveloperTrainingSessionDeleteService.java	91,8 %
>	DeveloperTrainingSessionListAllService.java	95,8 %
>	DeveloperTrainingSessionListMineService.java	95,8 %
>	DeveloperTrainingSessionPublishService.java	95,0 %
>	DeveloperTrainingSessionShowService.java	96,9 %
>	DeveloperTrainingSessionUpdateService.java	94,9 %

En resumen, mayormente las líneas amarillas que aparecen ya han sido comentadas anteriormente. Por todo esto y por tener un coverage del 94.4% de media en training module y de 95.0% de media en training session, se considera que la existencia de potenciales fallos o bugs es

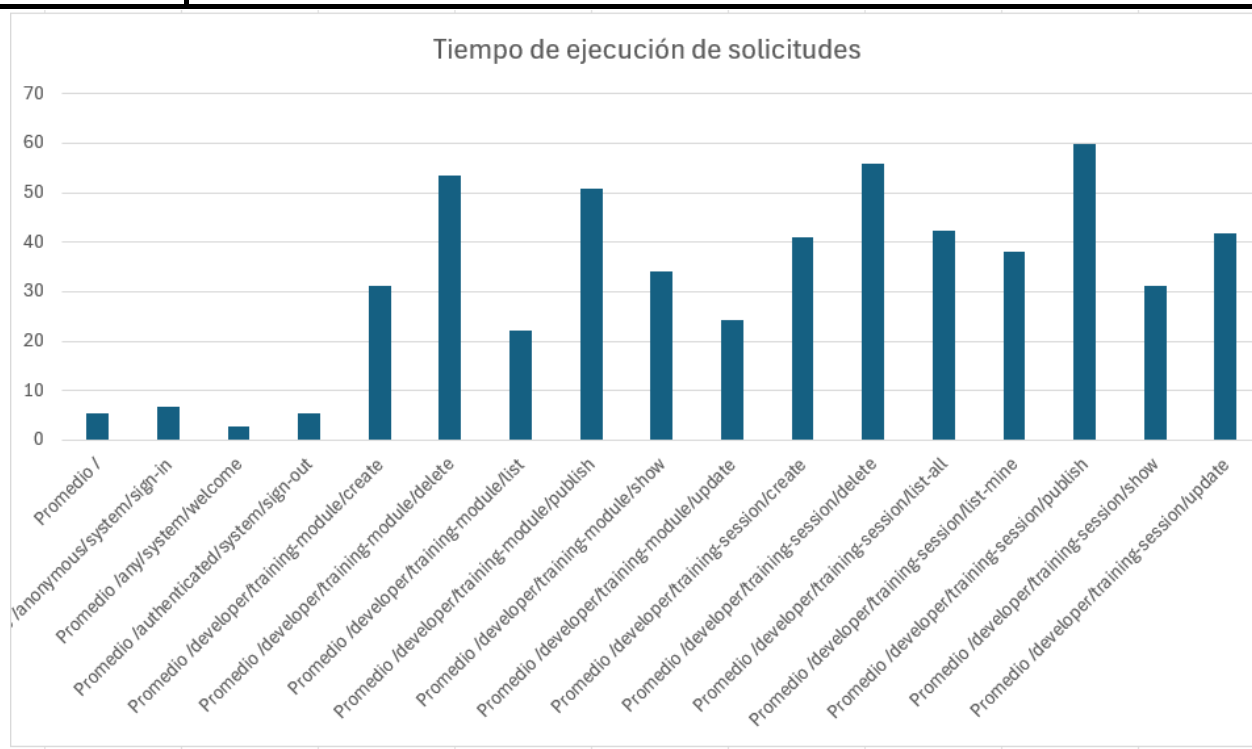


ínfima.

5. Performance testing

Tras realizar el conjunto de tests para las funcionalidades oportunas, se han realizado todos los pasos que se muestran en “S02 - Performance testing”, obteniendo los siguientes resultados:

Promedio /	5.37205919
Promedio /anonymous/system/sign-in	6.71094635
Promedio /any/system/welcome	2.69614141
Promedio /authenticated/system/sign-out	5.39560013
Promedio /developer/training-module/create	31.2845463
Promedio /developer/training-module/delete	53.4097201
Promedio /developer/training-module/list	22.1843999
Promedio /developer/training-module/publish	50.7088549
Promedio /developer/training-module/show	34.1456792
Promedio /developer/training-module/update	24.3788666
Promedio /developer/training-session/create	40.983
Promedio /developer/training-session/delete	55.9069626
Promedio /developer/training-session/list-all	42.2946528
Promedio /developer/training-session/list-mine	38.0071152
Promedio /developer/training-session/publish	59.7361363
Promedio /developer/training-session/show	31.2346927
Promedio /developer/training-session/update	41.6973943
Promedio general	25.7800525



Como se observa en las imágenes anteriores, el tiempo promedio que tarda el sistema en realizar una petición es de aproximadamente 25,78 ms, es decir, 0,0257 segundos, lo cual es bastante rápido.

Además, el gráfico de barras muestra claramente que las peticiones que requieren más tiempo son aquellas que manejan una mayor cantidad de datos y validaciones, específicamente las relacionadas con la creación, edición, eliminación y publicación de un training module. Esto se debe, en parte, a que se deben verificar varias validaciones tanto de la clase training module como de la clase asociada training session.

Posteriormente se obtuvieron los siguientes resultados con otra computadora, podemos apreciar la comparativa siguiente:

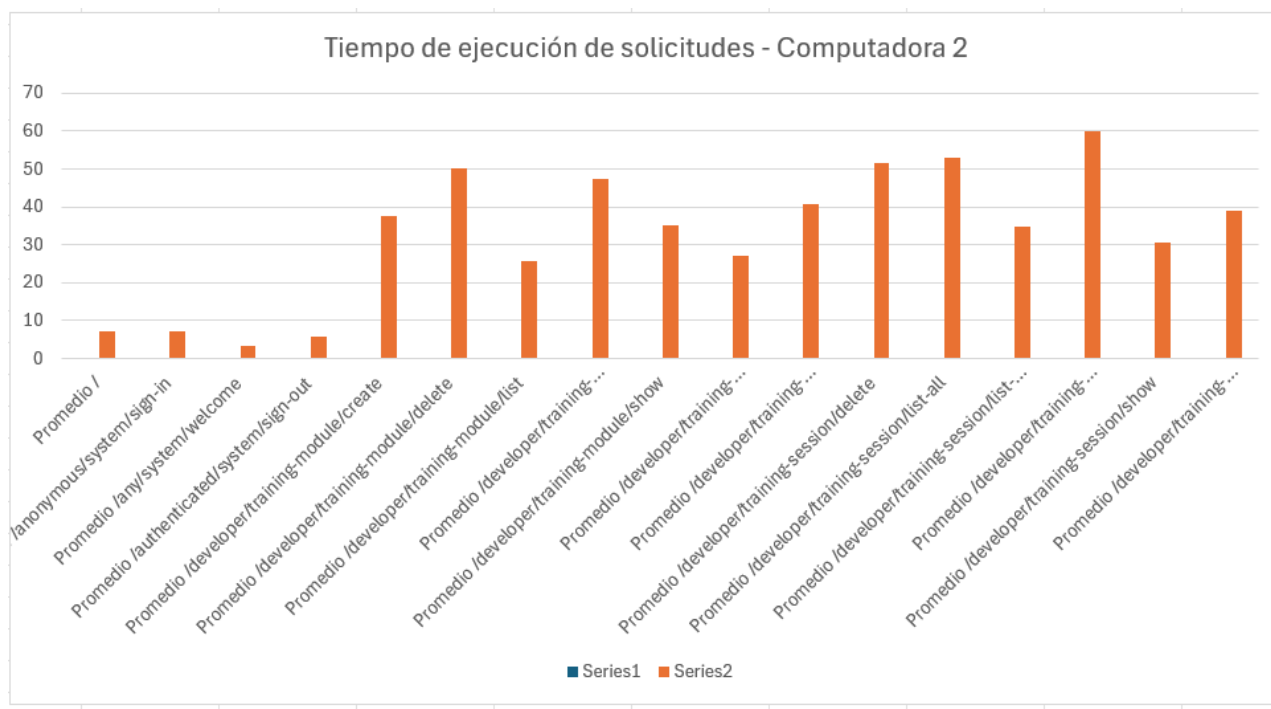
Computadora 1	Computadora 2
---------------	---------------



Testing report - José Ramón Baños Botón

Promedio /	5.37205919	Promedio /	7.0790625
Promedio /anonymous/system/sign-in	6.71094635	Promedio /anonymous/system/sign-in	7.15676
Promedio /any/system/welcome	2.69614141	Promedio /any/system/welcome	3.5094045
Promedio /authenticated/system/sign-out	5.39560013	Promedio /authenticated/system/sign-out	5.63598261
Promedio /developer/training-module/create	31.2845463	Promedio /developer/training-module/create	37.4451058
Promedio /developer/training-module/delete	53.4097201	Promedio /developer/training-module/delete	50.3102667
Promedio /developer/training-module/list	22.1843999	Promedio /developer/training-module/list	25.8066571
Promedio /developer/training-module/publish	50.7088549	Promedio /developer/training-module/publish	47.46363
Promedio /developer/training-module/show	34.1456792	Promedio /developer/training-module/show	35.0942931
Promedio /developer/training-module/update	24.3788666	Promedio /developer/training-module/update	27.1426397
Promedio /developer/training-session/create	40.983	Promedio /developer/training-session/create	40.9019321
Promedio /developer/training-session/delete	55.9069626	Promedio /developer/training-session/delete	51.651075
Promedio /developer/training-session/list-all	42.2946528	Promedio /developer/training-session/list-all	52.9745588
Promedio /developer/training-session/list-mine	38.0071152	Promedio /developer/training-session/list-mine	34.7946077
Promedio /developer/training-session/publish	59.7361363	Promedio /developer/training-session/publish	60.1495636
Promedio /developer/training-session/show	31.2346927	Promedio /developer/training-session/show	30.5727889
Promedio /developer/training-session/update	41.6973943	Promedio /developer/training-session/update	39.0656967
Promedio general	25.7800525	Promedio general	26.4657115

Para la computadora 2 obtenemos el siguiente grafo:



Se observa que el grafo queda muy similar con respecto al de la primera computadora, en el que destacan un mayor tiempo las peticiones de que manejan una mayor cantidad de datos y validaciones.

A continuación, se muestran estadísticas descriptivas de los tests elaborados para ambas computadoras:

Computadora 1	Computadora 2
---------------	---------------



Testing report - José Ramón Baños Botón

Computadora 1			Computadora 2		
Media	25.7800525		Media	26.4657115	
Error típico	1.02541134		Error típico	1.00906269	
Mediana	13.7098		Mediana	16.6687	
Moda	34.781401		Moda	15.9193	
Desviación estándar	30.8135682		Desviación estándar	30.3222918	
Varianza de la muestra	949.475985		Varianza de la muestra	919.44138	
Curtosis	4.87960829		Curtosis	11.8983946	
Coeficiente de asimetría	2.04198623		Coeficiente de asimetría	2.70137399	
Rango	204.1116		Rango	270.7914	
Mínimo	1.669499		Mínimo	2.2513	
Máximo	205.781099		Máximo	273.0427	
Suma	23279.3874		Suma	23898.5375	
Cuenta	903		Cuenta	903	
Nivel de confianza(95.0%)	2.01246971		Nivel de confianza(95.0%)	1.98038388	
Interval(ms)	23.7675828	27.7925222	Interval(ms)	24.4853276	28.4460954
Interval(s)	0.02376758	0.02779252	Interval(s)	0.02448533	0.0284461

Se puede comprobar que, ambos resultados son muy parecidos, por lo que se deduce que las características y rendimiento de ambas computadoras son parecidas. Además, con el z-test podemos comprobar que si el valor del two-tail p-value es mayor que alpha (0.05), los resultados se pueden considerar iguales.

Comparación del p-value de ambas computadoras:

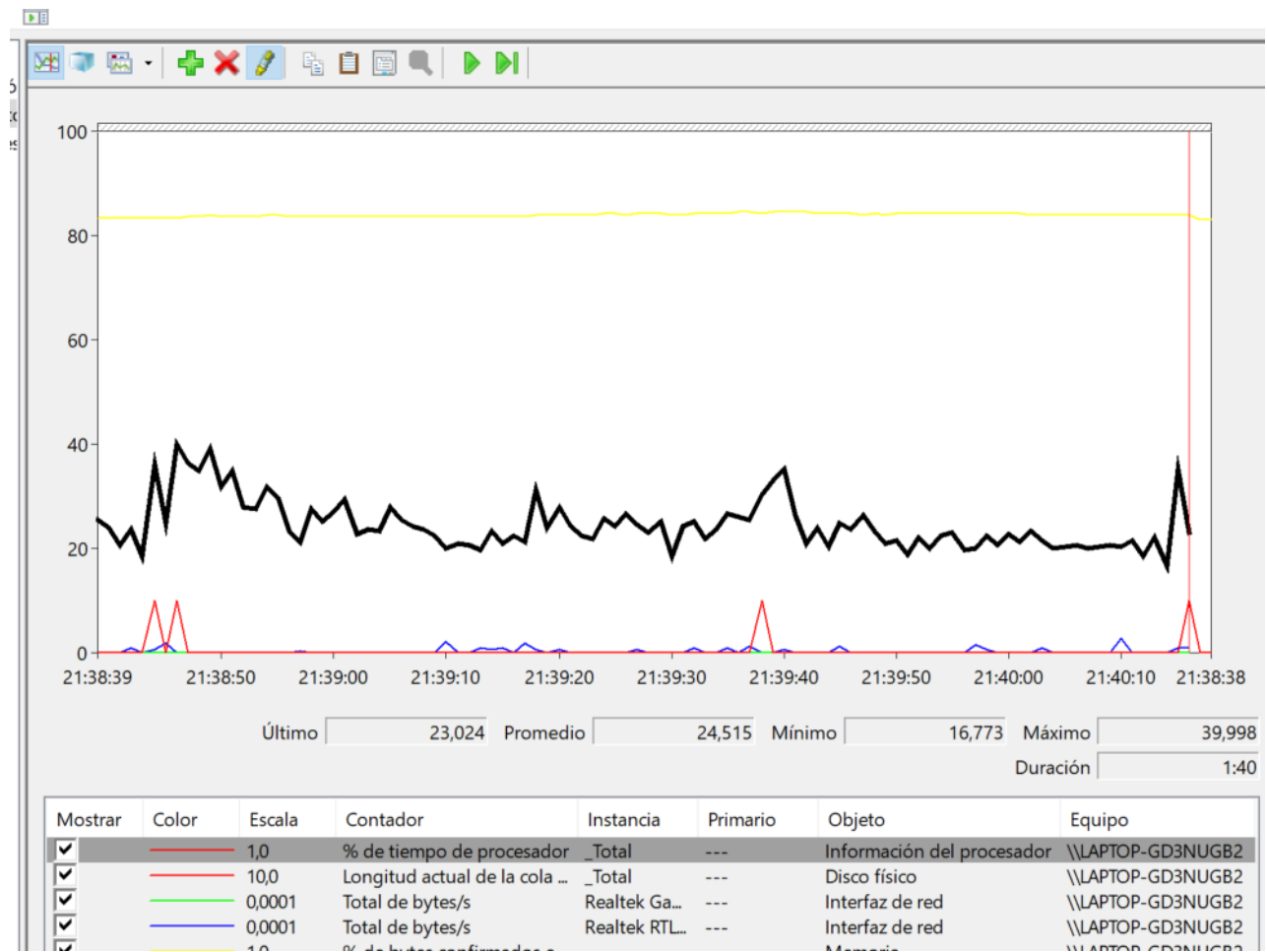
Prueba z para medias de dos muestras		
	Computadora1	Computadora2
Media	25.78005248	26.46571152
Varianza (conocida)	949.475985	919.44138
Observaciones	903	903
Diferencia hipotética de las medias	0	
z	-0.476603285	
P(Z<=z) una cola	0.316822322	
Valor crítico de z (una cola)	1.644853627	
Valor crítico de z (dos colas)	0.633644644	
Valor crítico de z (dos colas)	1.959963985	

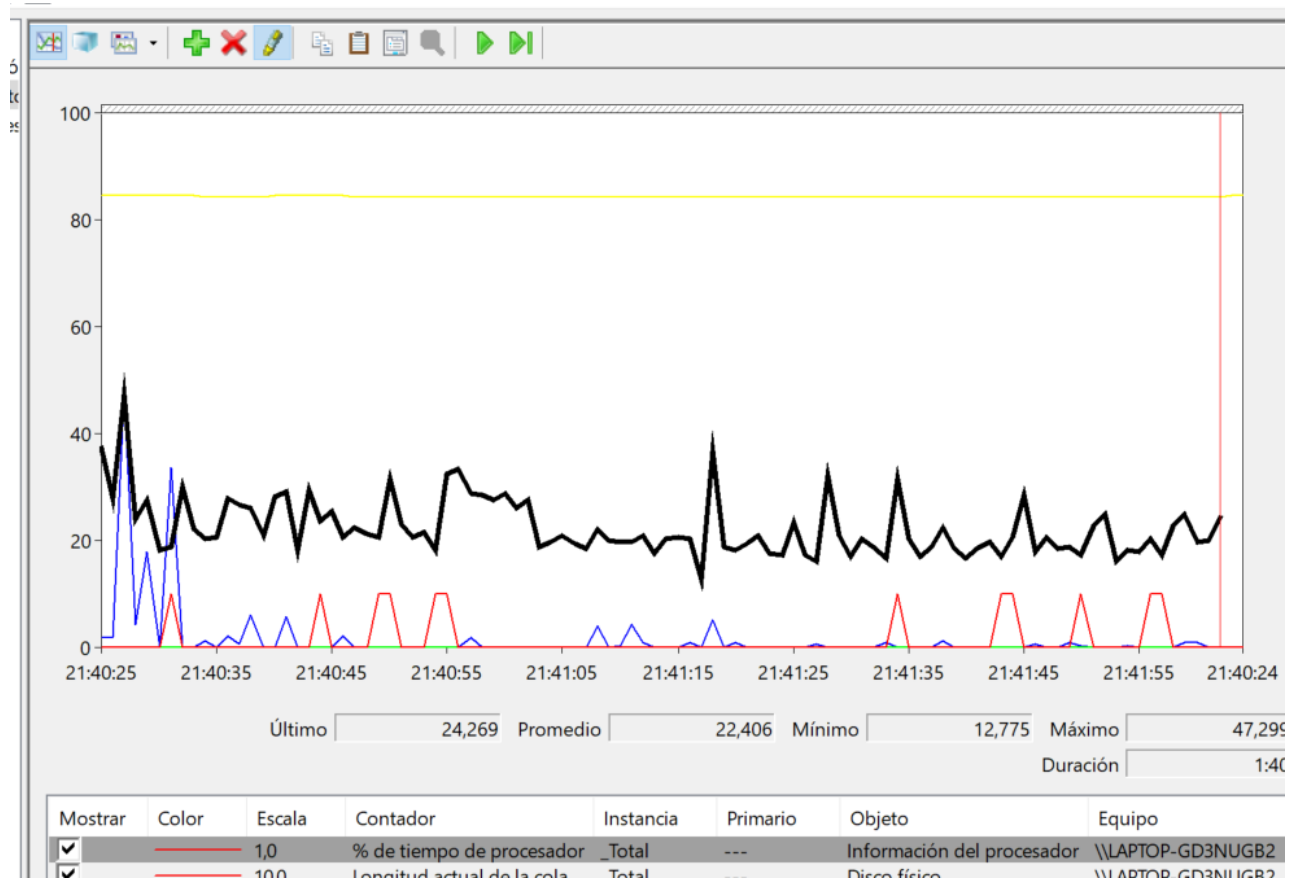
Estos son los resultados de la prueba Z para determinar si las dos medias son comparables. Como se ve, el valor del two-tail p-value es 0.633, un valor muy alejado de alpha (0.005), por lo que se puede afirmar que ambos computadores son muy similares. Sin embargo, pese a la leve diferencia, se puede afirmar que, en función del promedio de tiempo de ejecución de solicitudes que se puede ver en las tablas anteriores, el computador 1 es ligeramente superior al computador 2.

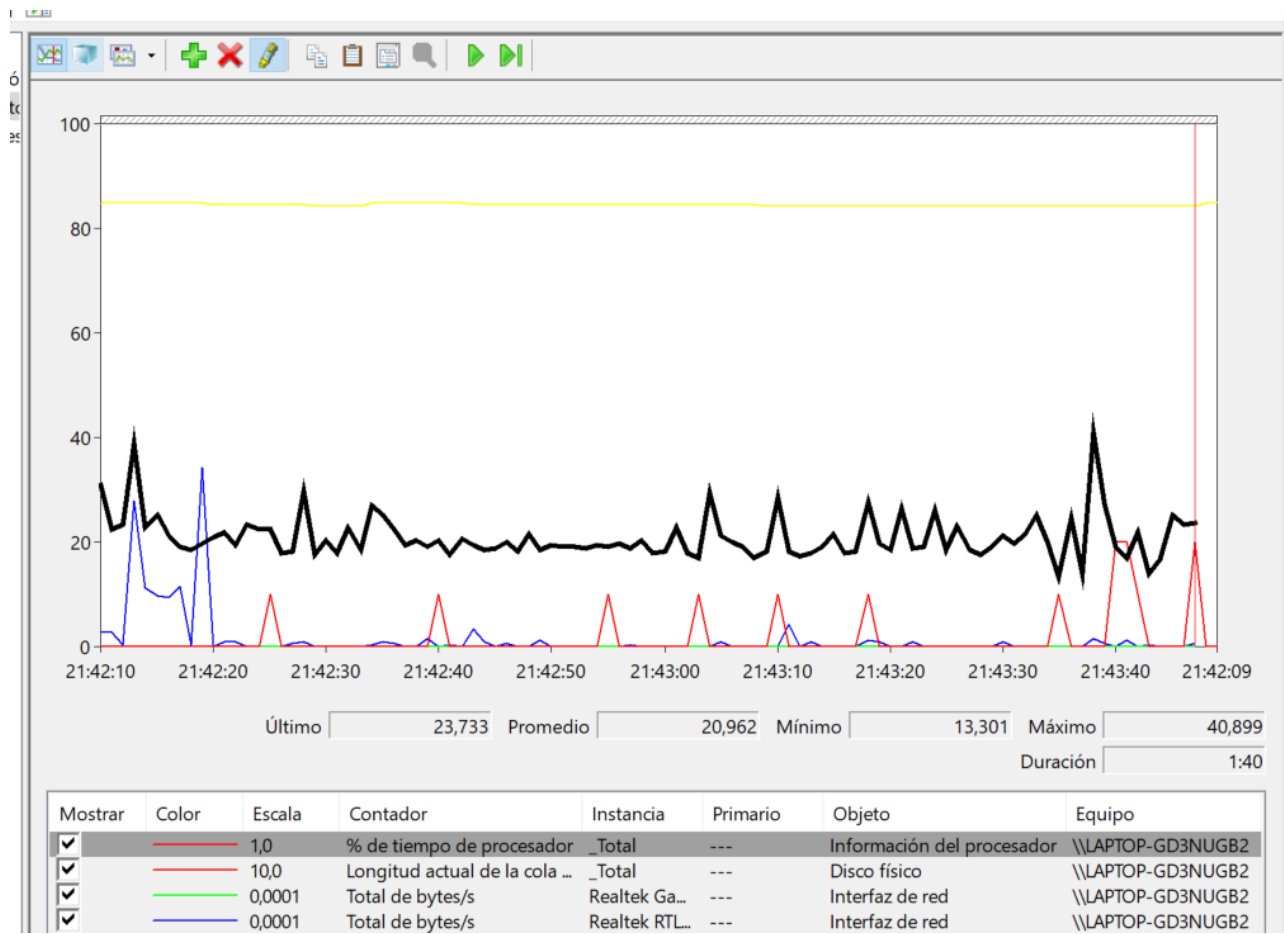


Análisis hardware:

Se presentan a continuación diferentes imágenes que muestran el rendimiento del hardware de la computadora 1. En ellas, se pueden observar un inicio más elevado, y algunos valores altos durante la ejecución de las pruebas. Al finalizar la ejecución se observa como desciende.









6. Conclusiones

Después de crear este documento sobre pruebas, se ha concluido que esta fase del ciclo de vida de un proyecto es vital. Verificar que todas las funciones desarrolladas funcionen correctamente y se revisen minuciosamente para minimizar errores o fallos, además de asegurar que el rendimiento esté optimizado al máximo, son aspectos fundamentales para el cliente. Un



sistema exhaustivamente probado permite que el usuario final lo utilice de manera rápida e intuitiva, evitando problemas que puedan perjudicar su experiencia. Además, un proceso de pruebas meticuloso contribuye a la satisfacción del cliente y a la buena reputación del producto, garantizando que las expectativas de calidad y eficiencia se cumplan consistentemente.

7. Bibliografía

- 08 Annexes – Material proporcionado en la asignatura *Diseño y Pruebas II* por la Universidad de Sevilla.
- L04 - S01 - Formal testing - Material proporcionado en la asignatura *Diseño y Pruebas II* por la Universidad de Sevilla.
- L04 - S02 - Performance testing - Material proporcionado en la asignatura *Diseño y Pruebas II* por la Universidad de Sevilla.



Diseño y Pruebas II

C2.X02

Testing report - José Ramón Baños Botón



Diseño y Pruebas II

C2.X02

Testing report - José Ramón Baños Botón