

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
Sonia María Rus Morales	sonrusmor@alum.us.es

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



## ***Índice***

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



## 1. Tabla de versiones

Fecha	Versión	Descripción
26/05/2024	v1r0	Creación del documento
27/05/2024	v1r1	Entrega
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 “project” y a “user story”. 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 project:**

- Create project:



- Descripción: Se prueban las restricciones de todos los campos del formulario de creación de un project con valores relativos a casos positivos, negativos y de hacking. Se valida que un usuario registrado con otro rol diferente a manager y no registrado no tenga acceso a esta funcionalidad.
- Coverage: 90.6%
- 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.
- Delete project
  - Descripción: Se prueba la eliminación de un project. Se valida que un usuario registrado con otro rol diferente a manager y no registrado y un manager no propietario de un project ajeno no tenga acceso a esta funcionalidad.
  - Coverage: 91.2%
  - Efectividad: Alta. Misma situación con “assert object != null”.
- List my projects:
  - Descripción: Se prueba el listado de projects pertenecientes al manager que ha iniciado sesión en el sistema. Se valida que un usuario registrado con otro rol diferente a manager y otro no registrado no tenga acceso a esta funcionalidad, y que otro manager solo pueda ver sus projects.
  - Coverage: 95.5%
  - Efectividad: Alta. Misma situación con “assert object != null”.
- Publish project
  - Descripción: Se prueba la publicación de un project teniendo en cuenta restricciones como que no se podrá publicar un project si tiene al menos un user story sin publicar o si no tiene user stories asociadas. Se valida que un manager que no es propietario de un project ajeno no tenga acceso a esta funcionalidad y que otro usuario registrado o no registrado que no es propietario de un project ajeno no tenga acceso a esta funcionalidad tampoco.
  - Coverage: 94.0%
  - Efectividad: Alta. Misma situación con “assert object != null”.
- Show project details
  - Descripción: Se prueba la muestra de detalles de un project del que el manager es propietario. Se valida que un manager que no es propietario de un project ajeno no tenga acceso a esta funcionalidad. Se valida, además, que otro usuario registrado o no registrado que no es propietario de un project ajeno no tenga acceso a esta funcionalidad.
  - Coverage: 96.2%

- Efectividad: Alta. Misma situación con “assert object != null” y “status = object != null && ...”. Se encuentra la siguiente línea amarilla: “status = super.getRequest().getPrincipal().hasRole(manager) && object != null;” en la autorización. 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 project con otro rol que no sea el de manager, y con otro manager diferente y, con un project con id nula en la URL.
- Update project
  - Descripción: Se prueban las restricciones de todos los campos del formulario de actualización de un project con valores relativos a casos positivos, negativos y de hacking. Se valida que un manager que no es propietario de un project ajeno no tenga acceso a esta funcionalidad y que otro usuario registrado o no registrado que no es propietario de un project ajeno no tenga acceso a esta funcionalidad tampoco.
  - Coverage: 92.0%
  - Efectividad: Alta. Misma situación con “assert object != null”.

▼  acme.features.manager.project	 93,3 %
>  ManagerProjectController.java	 100,0 %
>  ManagerProjectCreateService.java	 90,6 %
>  ManagerProjectDeleteService.java	 91,2 %
>  ManagerProjectListMineService.java	 95,5 %
>  ManagerProjectPublishService.java	 94,0 %
>  ManagerProjectShowService.java	 96,2 %
>  ManagerProjectUpdateService.java	 92,0 %

### Casos de pruebas relativos a user story:

- Create user story:
  - Descripción: Se prueban las restricciones todos los campos del formulario de creación de un user story con valores relativos a casos positivos, negativos y de hacking. Se valida que un usuario registrado con otro rol diferente a manager y no registrado no tenga acceso a esta funcionalidad.
  - Coverage: 90.3%
  - Efectividad: Alta. Misma situación con “assert object != null”.
- Delete user story:



- Descripción: Se prueba la eliminación de un user story. Se valida que un usuario registrado con otro rol diferente a manager y no registrado y un manager no propietario de un user story ajeno no tenga acceso a esta funcionalidad.
- Coverage: 91.7%
- Efectividad: Alta. Misma situación con “assert object != null” y “status = object != null && ...”. Además, en la autorización, observamos en amarillo la línea “status = super.getRequest().getPrincipal().hasRole(object.getManager()) && object != null && object.isDraftMode();”. Esto puede ser debido a un problema de Eclipse, ya que se han probado tanto los casos en los que se intenta acceder al delete de un user story con otro rol que no sea el de manager, y con otro manager diferente y, con un user story con id nula en la URL.
- List my user stories:
  - Descripción: Se prueba el listado de user stories pertenecientes al manager que ha iniciado sesión en el sistema. Se valida que un usuario registrado con otro rol diferente a manager y otro no registrado no tenga acceso a esta funcionalidad, y que otro manager solo pueda ver sus user stories.
  - Coverage: 97.4%
  - Efectividad: Alta. Misma situación con “assert object != null”.
- List all my user stories:
  - Descripción: Se prueba el listado de user stories pertenecientes al manager que ha iniciado sesión en el sistema. Se valida que un manager que no es propietario de un user story ajeno no tenga acceso a esta funcionalidad y que otro usuario registrado o no registrado que no es propietario de un user story ajeno no tenga acceso a esta funcionalidad tampoco.
  - Coverage: 95.5%
  - Efectividad: Alta. Misma situación con “assert object != null”.
- Publish user story:
  - Descripción: Se prueba la publicación de un user story. Se valida que un manager que no es propietario de un user story ajeno no tenga acceso a esta funcionalidad y que otro usuario registrado o no registrado que no es propietario de un user story ajeno no tenga acceso a esta funcionalidad tampoco.
  - Coverage: 91.4%
  - Efectividad: Alta. Misma situación con “assert object != null”. Además, en la autorización, observamos en amarillo la línea “status = super.getRequest().getPrincipal().hasRole(object.getManager()) && object != null && object.isDraftMode();”. Esto puede ser debido a un problema de Eclipse, ya que se han probado tanto los casos en los que se intenta acceder al delete de un user story con otro rol que no sea el de manager, y con otro manager diferente y, con un user story con id nula en la URL.
- Show user story details:





## Testing report – Sonia María Rus Morales

- Descripción: Se prueba la muestra de detalles de un user story del que el manager es propietario. Se valida que un manager que no es propietario de un user story ajeno no tenga acceso a esta funcionalidad y que otro usuario registrado o no registrado que no es propietario de un user story ajeno no tenga acceso a esta funcionalidad tampoco.
  - Coverage: 95.3%
  - Efectividad: Alta. Misma situación con “assert object != null” . Además, en la autorización, observamos en amarillo la línea “status = super.getRequest().getPrincipal().hasRole(manager) && 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 delete de un user story con otro rol que no sea el de manager, y con otro manager diferente y, con un user story con id nula en la URL.
- Update user story:
    - Descripción: Se prueban las restricciones de todos los campos del formulario de actualización de un user story con valores relativos a casos positivos, negativos y de hacking. Se valida que un manager que no es propietario de un user story ajeno no tenga acceso a esta funcionalidad y que otro usuario registrado o no registrado que no es propietario de un user story ajeno no tenga acceso a esta funcionalidad tampoco.
    - Coverage: 91.3 %
    - Efectividad: Alta. Misma situación con “assert object != null” y “ status = super.getRequest().getPrincipal().hasRole(object.getManager()) && object != null && object.isDraftMode();”.

▼  acme.features.manager.userStory	93,1 %
>  ManagerUserStoryController.java	100,0 %
>  ManagerUserStoryCreateService.java	90,3 %
>  ManagerUserStoryDeleteService.java	91,7 %
>  ManagerUserStoryListAllService.java	95,5 %
>  ManagerUserStoryListMineService.java	97,4 %
>  ManagerUserStoryPublishService.java	91,4 %
>  ManagerUserStoryShowService.java	95,3 %
>  ManagerUserStoryUpdateService.java	91,3 %

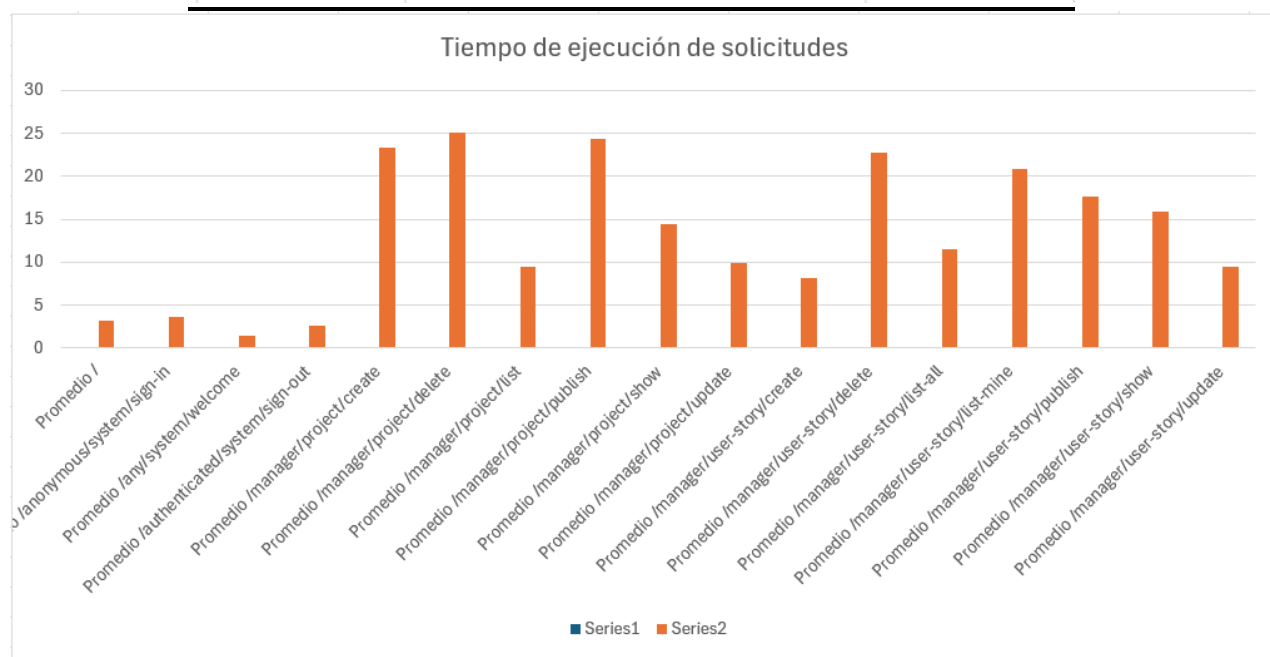
En resumen, mayormente las líneas amarillas que aparecen ya han sido comentadas anteriormente. Por todo esto y por tener un coverage del 93.3% de media en project y de 93.1% de media en user story, se considera que la existencia de potenciales fallos o bugs es ínfima.



## ***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 /	3.18123735
Promedio /anonymous/system/sign-in	3.63706827
Promedio /any/system/welcome	1.49193333
Promedio /authenticated/system/sign-out	2.58778182
Promedio /manager/project/create	23.3952121
Promedio /manager/project/delete	25.1213813
Promedio /manager/project/list	9.45280357
Promedio /manager/project/publish	24.3480045
Promedio /manager/project/show	14.4166586
Promedio /manager/project/update	9.92540875
Promedio /manager/user-story/create	8.23466912
Promedio /manager/user-story/delete	22.75264
Promedio /manager/user-story/list-all	11.56874
Promedio /manager/user-story/list-mine	20.7966111
Promedio /manager/user-story/publish	17.5876727
Promedio /manager/user-story/show	15.8642409
Promedio /manager/user-story/update	9.52922911
Promedio general	9.76350995



Como se puede observar en las imágenes colocadas anteriormente, el tiempo promedio que tarda el sistema en realizar una petición es de unos 9.76 ms, es decir, 0,0097 segundos, un tiempo muy rápido.

También se puede visualizar en el gráfico de barras de forma menos compleja, que las peticiones que más tiempo tardan son aquellas que manejan mayor cantidad de datos y validaciones; las relativas a la creación, edición, eliminación y publicación de un proyecto e historia de usuario. Esto



se debe a la comprobación de todas las validaciones que se han realizado, tanto de la propia clase project, como de la clase userStory.

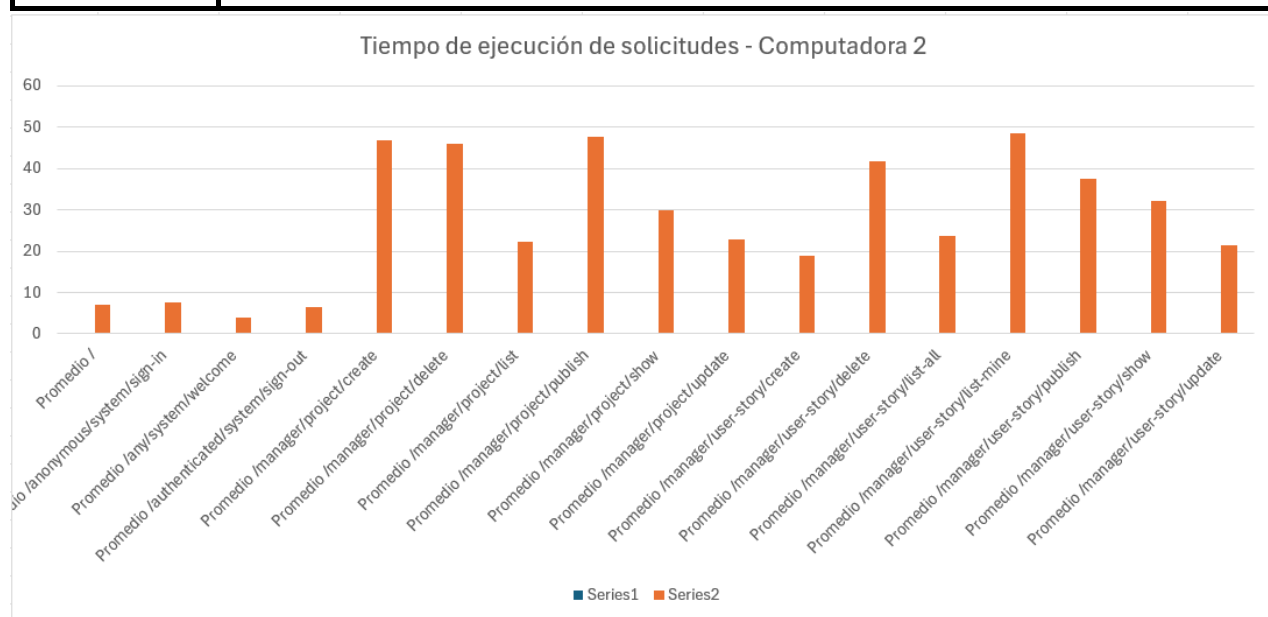
Posteriormente se obtuvieron los siguientes resultados con otra computadora, podemos apreciar la comparativa siguiente:

Computadora 1			Computadora 2		
Promedio /		3.18123735	Promedio /		7.0032443
Promedio /anonymous/system/sign-in		3.63706827	Promedio /anonymous/system/sign-in		7.645924
Promedio /any/system/welcome		1.49193333	Promedio /any/system/welcome		3.82848416
Promedio /authenticated/system/sign-out		2.58778182	Promedio /authenticated/system/sign-out		6.37232727
Promedio /manager/project/create		23.3952121	Promedio /manager/project/create		46.9168879
Promedio /manager/project/delete		25.1213813	Promedio /manager/project/delete		46.0055063
Promedio /manager/project/list		9.45280357	Promedio /manager/project/list		22.2323864
Promedio /manager/project/publish		24.3480045	Promedio /manager/project/publish		47.5362833
Promedio /manager/project/show		14.4166586	Promedio /manager/project/show		30.00352
Promedio /manager/project/update		9.92540875	Promedio /manager/project/update		22.9360052
Promedio /manager/user-story/create		8.23466912	Promedio /manager/user-story/create		18.7807588
Promedio /manager/user-story/delete		22.75264	Promedio /manager/user-story/delete		41.71904
Promedio /manager/user-story/list-all		11.56874	Promedio /manager/user-story/list-all		23.6593643
Promedio /manager/user-story/list-mine		20.7966111	Promedio /manager/user-story/list-mine		48.4403467
Promedio /manager/user-story/publish		17.5876727	Promedio /manager/user-story/publish		37.567455
Promedio /manager/user-story/show		15.8642409	Promedio /manager/user-story/show		32.0607643
Promedio /manager/user-story/update		9.52922911	Promedio /manager/user-story/update		21.3884506
Promedio general		9.76350995	Promedio general		20.4911403

Para la computadora 2 obtenemos el siguiente grafo: \_



Testing report – Sonia María Rus Morales



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, aunque para la segunda computadora, los tiempos son más elevados.

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

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



Testing report – Sonia María Rus Morales

Computadora 1			Computadora 2		
Media	10.5923717		Media	22.2609129	
Error típico	0.41198027		Error típico	0.83998069	
Mediana	6.2309		Mediana	15.1087	
Moda	1.1518		Moda	3.0646	
Desviación estándar	11.1234672		Desviación estándar	22.0004396	
Varianza de la muestra	123.731523		Varianza de la muestra	484.019344	
Curtosis	7.1655276		Curtosis	12.4350861	
Coeficiente de asimetría	2.08050527		Coeficiente de asimetría	2.44021942	
Rango	95.8956		Rango	224.747	
Mínimo	0.8986		Mínimo	2.6308	
Máximo	96.7942		Máximo	227.3778	
Suma	7721.839		Suma	15270.9862	
Cuenta	729		Cuenta	686	
Nivel de confianza(95.0%)	0.80881117		Nivel de confianza(95.0%)	1.64924595	
Interval(ms)	9.78356058	11.4011829	Interval(ms)	20.6116669	23.9101588
Interval(s)	0.00978356	0.01140118	Interval(s)	0.02061167	0.02391016

Se puede comprobar que, ambos resultados son un poco diferentes, por lo que se deduce que las características y rendimiento de ambas computadoras no son del todo 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:

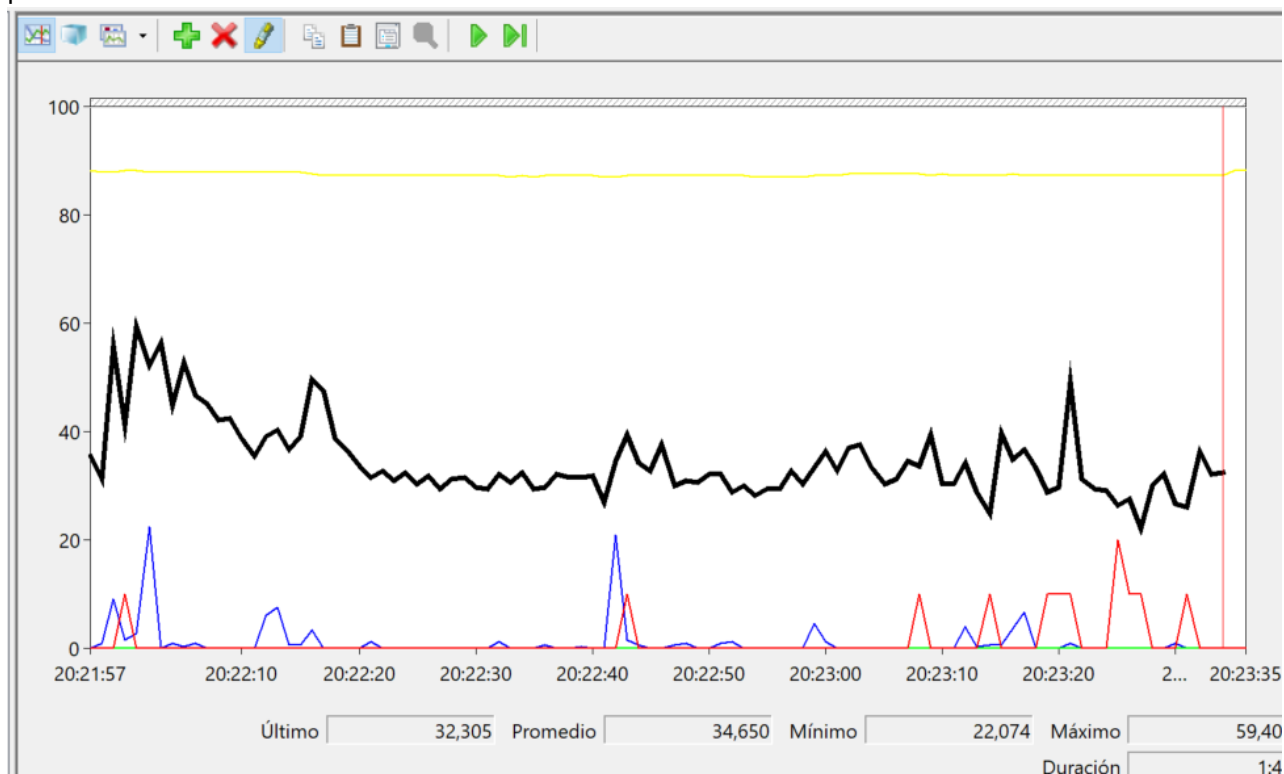
Prueba z para medias de dos muestras		
	Computadora 1	Computadora 2
Media	9.834829683	20.64165707
Varianza (conocida)	123.731523	484.019344
Observaciones	812	769
Diferencia hipotética de las medias	0	
z	-12.22228513	
P(Z<=z) una cola	0	
Valor crítico de z (una cola)	1.644853627	
Valor crítico de z (dos colas)	0	
Valor crítico de z (dos colas)	1.959963985	

Estos son los resultados de la prueba Z para determinar si las dos medias son comparables.

Análisis hardware:

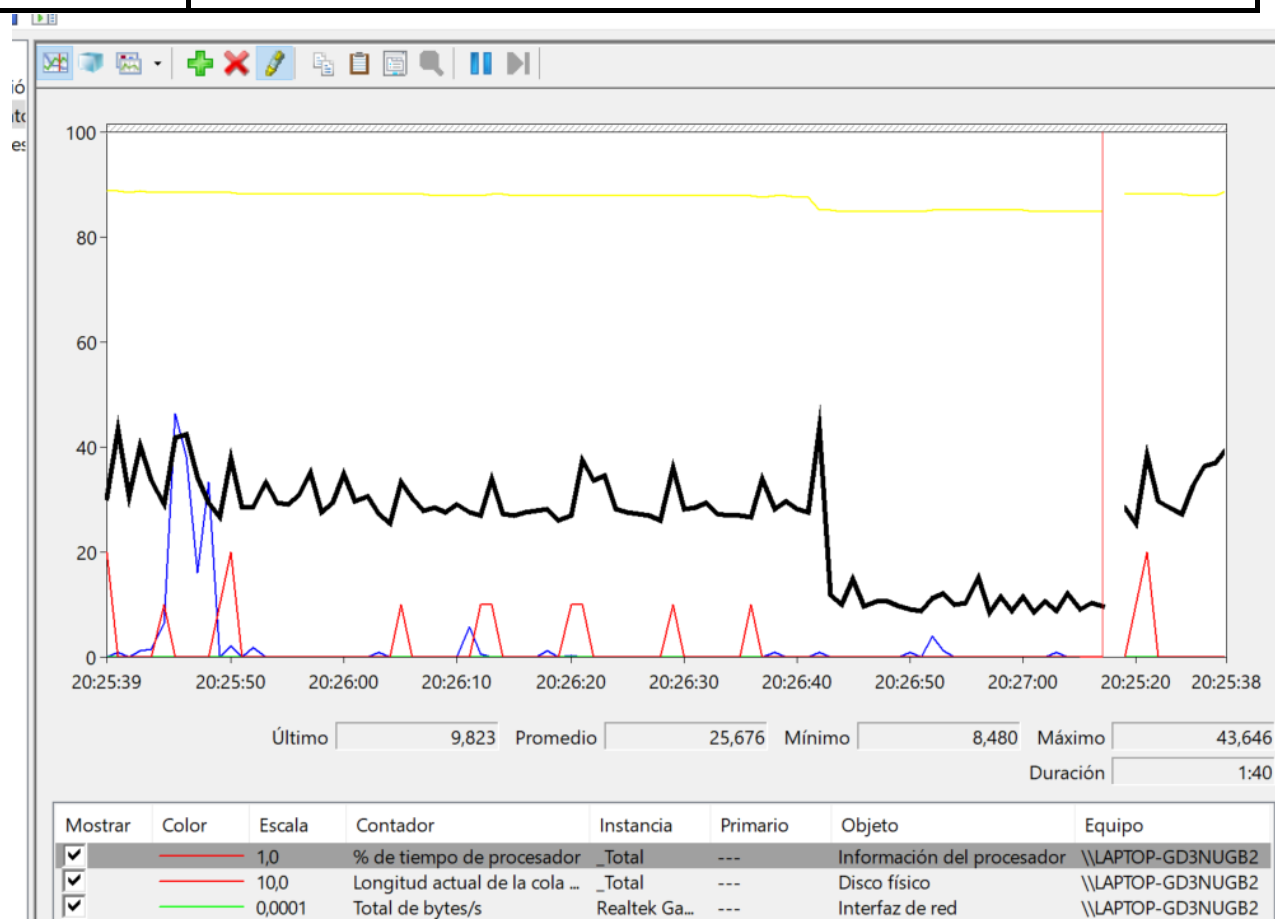
Se presentan a continuación diferentes imágenes que muestran el rendimiento del hardware de la

computadora 1. En ellas, se pueden observar una pequeña elevación al inicio, algún pico en la zona intermedia y algunos valores bajos en la parte final de la ejecución de las pruebas. El resto del proceso se ha mantenido considerablemente estable.









## 6. Conclusiones

Después de preparar este documento sobre pruebas, se ha determinado que esta etapa del ciclo de vida de un proyecto es esencial. Es fundamental verificar que todas las funciones



desarrolladas operen correctamente y sean revisadas detalladamente para minimizar errores o fallos, además de asegurar que el rendimiento esté optimizado al máximo. Estos factores son cruciales para el cliente. Un sistema bien probado permite que el usuario final lo utilice de manera rápida e intuitiva, evitando problemas que puedan afectar negativamente su experiencia. Además, un proceso de pruebas riguroso contribuye a la satisfacción del cliente y a la reputación del producto, asegurando que las expectativas de calidad y eficiencia se cumplan de manera consistente.

## **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 – Sonia María Rus Morales



Diseño y Pruebas II

C2.X02

Testing report – Sonia María Rus Morales