

INFORME DE PRUEBAS

GRUPO C1.02.03



Autores:

- **Alejandro García Sánchez-Hermosilla** – alegarsan11@alum.us.es

Repositorio: <https://github.com/Alejandrocg024/Acme-L3-D04>

Fecha del documento: 20/05/2023

Tabla de contenidos.

Resumen general..... 3

Tabla de revisión. 4

Introducción. 5

Contenido..... 6

Conclusiones. 18

Bibliografía. 19

Resumen general.

En este informe hablaremos sobre las pruebas que tiene el sistema sobre los requisitos desarrollados del estudiante número 3 dado que este es el informe individual de este estudiante, hemos agrupado el tipo de pruebas dependiendo a la clase que se esta probando y el tipo de funcionalidad, después de esta clasificación diferenciamos entre 3 diferentes tipos de casos de prueba como son: los casos de pruebas positivos, los casos de prueba negativos y por ultimo los casos de pruebas de hacking.

Después de ver detalladamente de que trata cada prueba, al final de este documento también realizaremos una tarea de análisis para poder indicar a nuestro cliente que estadísticamente nuestras peticiones se sirven dentro de un intervalo de confianza

Tabla de revisión.

Fecha	Versión	Descripción
12/05/2023	0.1	Creación del documento
20/05/2023	0.2	Contenido y conclusión

Introducción.

Tras haber finalizado de implementar la funcionalidad que nos ha pedido el cliente y realizar una comprobación informal para asegurar parcialmente que lo implementado hasta la fecha funciona correctamente, nos dedicaremos ahora a realizar un testeo formal con pruebas automatizadas centradas en probar la funcionalidad totalmente con todos los casos de pruebas correspondientes que son los casos de prueba positivos, los casos de prueba negativos y por ultimo los casos de prueba de hacking, cada tipo trata de comprobar comportamientos diferentes del sistema de acuerdo con las diferentes situaciones que se le presentan al sistema, dependiendo si se están realizando operaciones legales o si se encuentran errores que el sistema puede manejar.

Por otra parte, también tenemos que informar al cliente de una forma objetiva mediante un estudio estadístico que las peticiones que realiza el sistema se encuentran dentro de un intervalo de confianza que el haya establecido por lo que después de comentar de que trata cada prueba realizaremos un análisis del tiempo de ejecución de las pruebas realizando pruebas de rendimiento para el sistema.

Contenido.

Conjunto de pruebas realizadas:

Para las tutorías:

List:

- Test100Positive():
 - Descripción: Iniciamos sesión como “assistant1” y accedemos al listado de sus tutorías, una vez que accedemos al listado ordenamos ascendentemente la primera columna y comprobamos que los valores que aparecen son los esperados.
 - Efectividad: No se encontraron fallos gracias a esta prueba.
- Test200Negative():
 - No existen casos negativos para esta prueba.
- Test300Hacking():
 - Descripción: Iniciamos sesión con diferentes roles en el sistema, pero ninguno de ellos es el rol asistente por lo que no tienen acceso a ver el listado de tutorías.
 - Efectividad: No se encontraron fallos gracias a esta prueba.

Show:

- Test100Positive():
 - Descripción: Iniciamos sesión como “assistant1” accedemos a la lista de tutorías y ordenamos la lista, vamos accediendo a las diferentes tutorías del listado y comprobamos los datos del formulario se corresponden con los esperados.
 - Efectividad: No se encontraron fallos gracias a esta prueba.
- Test200Negative():
 - No existen casos negativos para esta prueba.
- Test300Hacking():
 - Descripción: Iniciamos sesión con diferentes roles en el sistema, pero ninguno de ellos es el rol asistente e intentamos visualizar tutorías que no estén publicadas, por lo que el sistema no permite visualizarlas dado que no se corresponden con nuestras tutorías ni tenemos el rol necesario para visualizar las tutorías.
 - Efectividad: No se encontraron fallos gracias a esta prueba.

- Test301Hacking():
 - Descripción: Iniciamos sesión como “assistant2” e intentamos visualizar una tutoría que no esta publicada de otro asistente y como no somos el creador de esta tutoría el sistema no nos permite visualizarla.
 - Efectividad: No se encontraron fallos gracias a esta prueba.

Create:

- Test100Positive():
 - Descripción: Iniciamos sesión como “assistant1”, accedemos al listado de nuestras tutorías y clicamos en el botón de crear una nueva tutoría y rellenamos los campos correspondientes de una tutoría siguiendo la metodología vista en clase, 1º creamos un caso normal y después vamos modificando los diferentes campos con los limites correspondientes, además de añadir casos con letras de otro idioma y probar a insertar líneas de código en los diferentes campos. (Todos los casos probados en esta prueba han de ser positivos por lo que los limites se comprueban dentro del rango que permite crear una tutoría)
 - Efectividad: No se encontraron fallos gracias a esta prueba.
- Test200Negative():
 - Descripción: Iniciamos sesión como “assistant1”, accedemos al listado de nuestras tutorías y clicamos en el botón de crear una nueva tutoría y rellenamos los campos correspondientes de una tutoría siguiendo la metodología vista en clase, 1º enviamos el formulario vacío y luego vamos modificando cada atributo con valores no permitidos y comprobamos que el sistema encuentra fallos y sabe manejarlos. (Para este caso de pruebas no se permite que se cree ninguna tutoría por loque algún campo debe de tener algún error)
 - Efectividad: No se encontraron fallos en esta prueba.
- Test300Hacking():
 - Descripción: Iniciamos sesión con diferentes roles en el sistema, pero ninguno de ellos es el rol asistente e intentamos crear tutorías, y el sistema no permite crearlas dado que no tenemos el rol necesario para realizar operaciones en las tutorías.
 - Efectividad: No se encontraron fallos gracias a esta prueba.

Update:

- Test100Positive():
 - Descripción: Iniciamos sesión como “assistant1”, accedemos al listado de tutorías y accedemos a la primera tutoría y la vamos modificando con la metodología vista en clase, primero en caso de prueba positivos general, y después vamos modificando cada atributo para comprobar sus límites, además de ello también probamos con líneas de código HTML.
 - Efectividad: No se encontraron fallos gracias a esta prueba.

- Test200Negative():
 - Descripción: Iniciamos sesión como “assistant1” y accedemos a una tutoría que no esté publicada de este usuario, y rellenamos los campos siguiendo la metodología vista en clase, primero insertamos el formulario vacío y luego vamos probando los límites de cada atributo de forma que se detecten fallos en el formulario.
 - Efectividad: No se encontraron fallos gracias a esta prueba.
- Test300Hacking():
 - Descripción: Buscamos las tutorías no publicadas de “assistant1” y vamos accediendo a cada una de ellas de la siguiente manera: Primero sin iniciar sesión, luego con los diferentes roles disponibles del sistema, pero sin que ninguno de ellos sea assistant1. Por lo que el sistema no nos permitirá visualizar si quiere la tutoría.
 - Efectividad: No se encontraron fallos gracias a esta prueba.
- Test301Hacking():
 - Descripción: Buscamos las tutorías no publicadas de “assistant1” y tras eso iniciamos sesión como “assistant2” y como no somos los creadores de esta tutoría el sistema nos deniega el acceso a cada una de las tutorías.
 - Efectividad: No se encontraron fallos gracias a esta prueba.
- Test302Hacking():
 - Descripción: Iniciamos sesión como “assistant1” y buscamos las tutorías que ya están publicadas de este usuario, intentamos acceder a la dirección de actualizar la tutoría, pero dado que ya esta publicada el sistema no nos permite actualizar la tutoría.
 - Efectividad: No se encontraron fallos gracias a esta prueba.

Publish:

- Test100Positive():
 - Descripción: Iniciamos sesión como “assistant1”, accedemos al listado de tutorías, ordenamos, cliqueamos en la primera tutoría y pulsamos el botón de sesiones de tutoría, clicamos en el botón de crear y rellenamos los datos correspondientes para crear una sesión para esa tutoría, después volvemos a la tutoría anterior y tocamos en publicar y comprobamos que no podemos realizar ninguna operación en esa tutoría que no sea visualizarla.
 - Efectividad: No se encontraron fallos gracias a esta prueba.
- Test200Negative():
 - Descripción: Iniciamos sesión como “assistant1” y accedemos al listado de tutorías e intentamos publicar una tutoría que no tiene sesiones.
 - Efectividad: No se encontraron fallos gracias a esta prueba.

- Test201Negative():
 - Descripción: Iniciamos sesión como “assistant1” y accedemos a una tutoría que esta publicada e intentamos acceder a la dirección de publicar una tutoría haciendo referencia a esta tutoría y comprobamos que encontramos un error de pánico.
 - Efectividad: No se encontraron fallos gracias a esta prueba.
- Test300Hacking():
 - Descripción: Buscamos todas las tutorías que no estén publicadas de “assistant1” y probamos a intentar publicarlas con la dirección primero sin iniciar sesión y luego con todos los roles posibles menos el rol de asistente.
 - Efectividad: No se encontraron fallos gracias a esta prueba.
- Test301Hacking():
 - Descripción: Buscamos todas las tutorías no publicadas de “assistant1” e iniciamos sesión como “assistant2” e intentamos publicar cada tutoría de “assistant1” y dado que no somos los creadores de esta tutoría pues el sistema nos deniega el acceso.
 - Efectividad: No se encontraron fallos gracias a esta prueba.
- Test302Hacking():
 - Descripción: Iniciamos sesión como “assistant1” y buscamos las tutorías publicadas de este usuario, y vamos accediendo a la dirección de publicar estas tutorías y como ya se encuentran publicadas, no nos permite volver a publicarlas
 - Efectividad: No se encontraron fallos gracias a esta prueba

Delete:

- Test100Positive():
 - Descripción: Iniciamos sesión como “assistant1” y accedemos al listado de tutorías, ordenamos y clicamos en tutorías que no estén publicadas y las eliminamos.
 - Efectividad: No se encontraron fallos en esta prueba.
- Test200Negative():
 - No existen casos negativos para esta prueba.
- Test300Hacking():
 - Descripción: Buscamos tutorías que no estén publicadas de “assistant2” e intentamos eliminar las tutorías de este usuario, pero primero sin iniciar sesión y después con los diferentes roles del sistema, pero sin ser el rol de asistente, por lo que el sistema nos devolverá un error de pánico.
 - Efectividad: No se encontraron fallos gracias a esta prueba.
- Test301Hacking():
 - Descripción: Buscamos tutorías que no estén publicadas de “assistant2” e intentamos eliminarlas como “assistant1” y como no somos el creador de esas tutorías el sistema nos devolverá un error de pánico.
 - Efectividad: No se encontraron fallos gracias a esta prueba.

- Test302Hacking():
 - Descripción: Iniciamos sesión como “assistant1” y después buscamos las tutorías que están publicadas de este usuario e intentamos eliminarlas con la dirección y dado que están publicadas, pues el sistema nos devolverá un error de pánico.
 - Efectividad: No se encontraron fallos gracias a esta prueba.

Para las sesiones de tutorías:

List:

- Test100Positive():
 - Descripción: Iniciamos sesión como “assistant1” buscamos una la tutoría que contiene más sesiones y accedemos al listado de sus sesiones, una vez que accedemos al listado ordenamos ascendentemente la primera columna y comprobamos que los valores que aparecen son los esperados.
 - Efectividad: No se encontraron fallos gracias a esta prueba.
- Test200Negative():
 - No existen casos negativos para esta prueba.
- Test300Hacking():
 - Descripción: buscamos la tutoría con más sesiones e iniciamos sesión con diferentes roles en el sistema, pero ninguno de ellos es el rol asistente por lo que no tienen acceso a ver el listado de sesiones de la tutoría.
 - Efectividad: No se encontraron fallos gracias a esta prueba.

Show:

- Test100Positive():
 - Descripción: Iniciamos sesión como “assistant1” accedemos a la lista de tutorías y ordenamos la lista, accedemos a la tutoría número 5 y clicamos en el listado de sus sesiones y vamos accediendo a las diferentes sesiones del listado y comprobamos los datos del formulario se corresponden con los esperados.
 - Efectividad: No se encontraron fallos gracias a esta prueba.
- Test200Negative():
 - No existen casos negativos para esta prueba.
- Test300Hacking():
 - Descripción: Iniciamos sesión con diferentes roles en el sistema, pero ninguno de ellos es el rol asistente e intentamos visualizar sesiones de tutorías del "assistant1", por lo que el sistema no permite visualizarlas dado que no se corresponden con nuestras tutorías ni tenemos el rol necesario para visualizar las tutorías.
 - Efectividad: No se encontraron fallos gracias a esta prueba.

- Test301Hacking():
 - Descripción: Iniciamos sesión como “assistant1” y buscamos sesiones de tutoría de “assistant2” después intentamos acceder a cada una de sus sesiones y como no somos el creador de las sesiones pues el sistema nos devuelve un pánico.
 - Efectividad: No se encontraron fallos gracias a esta prueba.

Create:

- Test100Positive():
 - Descripción: Iniciamos sesión como “assistant1”, accedemos al listado de nuestras tutorías y clicamos en la primera tutoría de la lista y accedemos a sus sesiones, pulsamos en crear nuevas sesiones y rellenamos los campos correspondientes de una tutoría siguiendo la metodología vista en clase, 1º creamos un caso normal y después vamos modificando los diferentes campos con los limites correspondientes, además de añadir casos con letras de otro idioma y probar a insertar líneas de código en los diferentes campos. (Todos los casos probados en esta prueba han de ser positivos por lo que los limites se comprueban dentro del rango que permite crear una tutoría)
 - Efectividad: Hemos observado fallos en esta prueba dado que el sistema no comprobaba los limites superiores e inferiores de las fechas.
- Test200Negative():
 - Descripción: Iniciamos sesión como “assistant1”, accedemos al listado de nuestras tutorías y clicamos en la primera tutoría de la lista, clicamos en sesiones de tutoría y le damos a crear nueva sesión rellenamos los campos correspondientes de una tutoría siguiendo la metodología vista en clase, 1º enviamos el formulario vacío y luego vamos modificando cada atributo con valores no permitidos y comprobamos que el sistema encuentra fallos y sabe manejarlos. (Para este caso de pruebas no se permite que se cree ninguna tutoría por loque algún campo debe de tener algún error)
 - Efectividad: No se encontraron fallos en esta prueba.
- Test300Hacking():
 - Descripción: Iniciamos sesión con diferentes roles en el sistema, pero ninguno de ellos es el rol asistente e intentamos crear sesiones de tutorías, y el sistema no permite crearlas dado que no tenemos el rol necesario para realizar operaciones en las tutorías.
 - Efectividad: No se encontraron fallos gracias a esta prueba.
- Test300Hacking():
 - Descripción: Iniciamos sesión como “assistant1” y buscamos tutorías del “assistant2” y mediante el uso de una dirección intentamos crear una sesión de tutoría en un tutorial que no sea el nuestro y el sistema nos devuelve un error de pánico dado que no somos el creador de esa tutoría.
 - Efectividad: No se encontraron fallos gracias a esta prueba.

Update:

- Test100Positive():
 - Descripción: Iniciamos sesión como “assistant1”, accedemos al listado de tutorías y accedemos a la tercera tutoría y pulsamos en los diferentes índices que corresponda y vamos modificando los datos con la metodología vista en clase, primero en caso de prueba positivos general, y después vamos modificando cada atributo para comprobar sus límites, además de ello también probamos con líneas de código HTML.
 - Efectividad: Hemos observado fallos en esta prueba debido a que cuando se actualizaba una fecha de inicio a nula y una fecha de fin específica no se comprobaba un objeto nulo en el servicio lo que provocaba un NullPointerException.
- Test200Negative():
 - Descripción: Iniciamos sesión como “assistant1” y accedemos a la 3ª tutoría de este usuario, clicamos en el listado de sus sesiones y seleccionamos la sesión que corresponda, y vamos rellenamos los campos siguiendo la metodología vista en clase, primero insertamos el formulario vacío y luego vamos probando los límites de cada atributo de forma que se detecten fallos en el formulario.
 - Efectividad: No se encontraron fallos gracias a esta prueba.
- Test300Hacking():
 - Descripción: Buscamos las sesiones de tutorías de “assistant1” y vamos accediendo a cada una de ellas de la siguiente manera: Primero sin iniciar sesión, luego con los diferentes roles disponibles del sistema, pero sin que ninguno de ellos sea assistant1. Por lo que el sistema no nos permitirá visualizar si quiere la tutoría, ni actualizarla.
 - Efectividad: No se encontraron fallos gracias a esta prueba.
- Test301Hacking():
 - Descripción: Buscamos las sesiones de tutorías no publicadas de “assistant2” y tras eso iniciamos sesión como “assistant1” y como no somos los creadores de esas sesiones de tutoría el sistema nos deniega el acceso a la dirección de actualizar.
 - Efectividad: No se encontraron fallos gracias a esta prueba.

Publish:

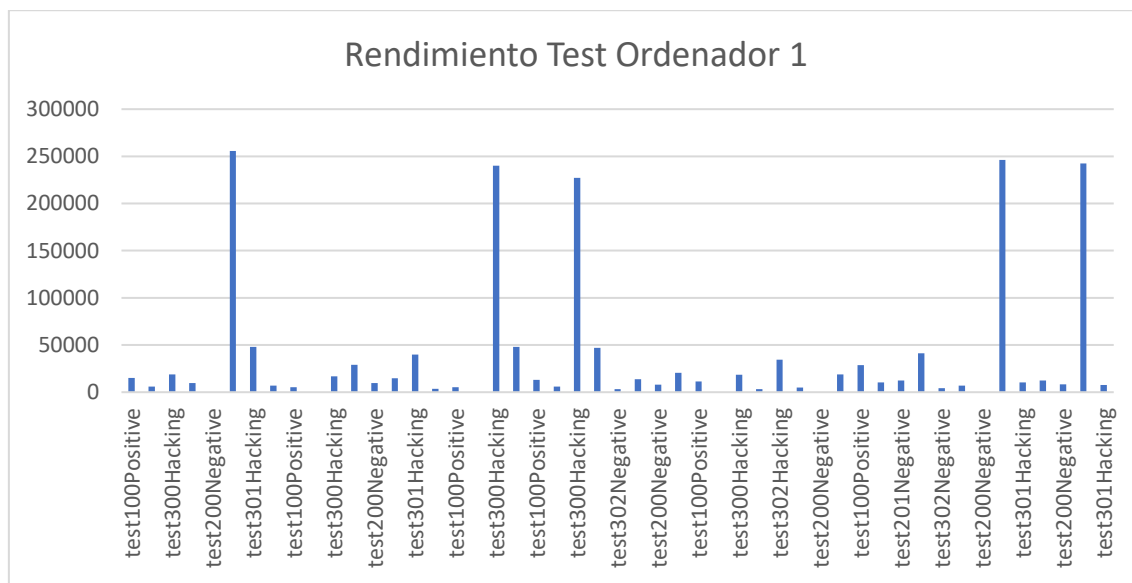
- Al publicarse las sesiones de una tutoría de forma automática al publicar la propia tutoría a la que pertenecen, la prueba es exactamente el mismo que la prueba de publicar una tutoría.

Delete:

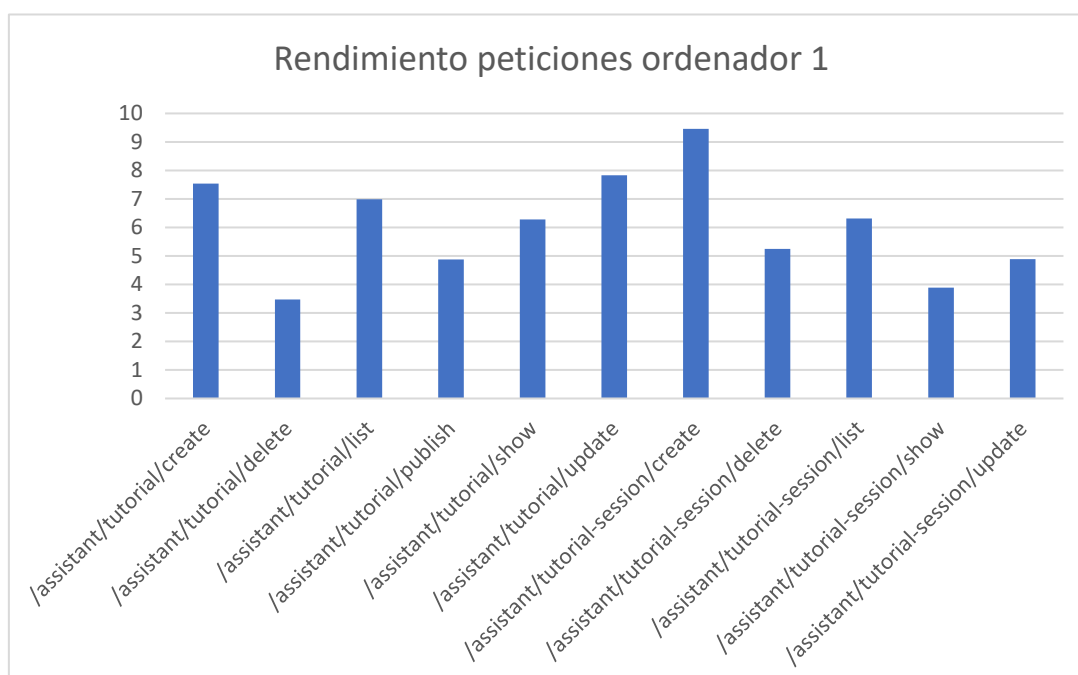
- Test100Positive():
 - Descripción: Iniciamos sesión como “assistant1” y accedemos al listado de tutorías, ordenamos y clicamos en la 3ª tutoría que no está publicada, accedemos a sus sesiones y las eliminamos.
 - Efectividad: No se encontraron fallos en esta prueba.
- Test200Negative():
 - No existen casos negativos para esta prueba.
- Test300Hacking():
 - Descripción: Buscamos sesiones de tutorías que no estén publicadas de “assistant2” e intentamos eliminar las sesiones de tutorías de este usuario pero primero sin iniciar sesión y después con los diferentes roles del sistema, pero sin ser el rol de asistente, por lo que el sistema nos devolverá un error de pánico.
 - Efectividad: No se encontraron fallos gracias a esta prueba.
- Test301Hacking():
 - Descripción: Buscamos sesiones de tutorías que no estén publicadas de “assistant1” e intentamos eliminarlas como “assistant2” y como no somos el creador de esas sesiones de tutorías el sistema nos devolverá un error de pánico.
 - Efectividad: No se encontraron fallos gracias a esta prueba.
- Test302Hacking():
 - Descripción: Iniciamos sesión como “assistant1” y después buscamos las sesiones tutorías que están publicadas de este usuario e intentamos eliminarlas con la dirección y dado que están publicadas, pues el sistema nos devolverá un error de pánico.
 - Efectividad: No se encontraron fallos gracias a esta prueba.

Después de comentar las distintas pruebas del sistema que tienen que ver con la funcionalidad implementada por el estudiante número 3 vamos a realizar un análisis del rendimiento de la funcionalidad mediante la prueba, para ello vamos a hacer una serie de cálculos estadísticos que nos revelaran el tiempo medio y el intervalo de confianza obtenido de realizar las pruebas.

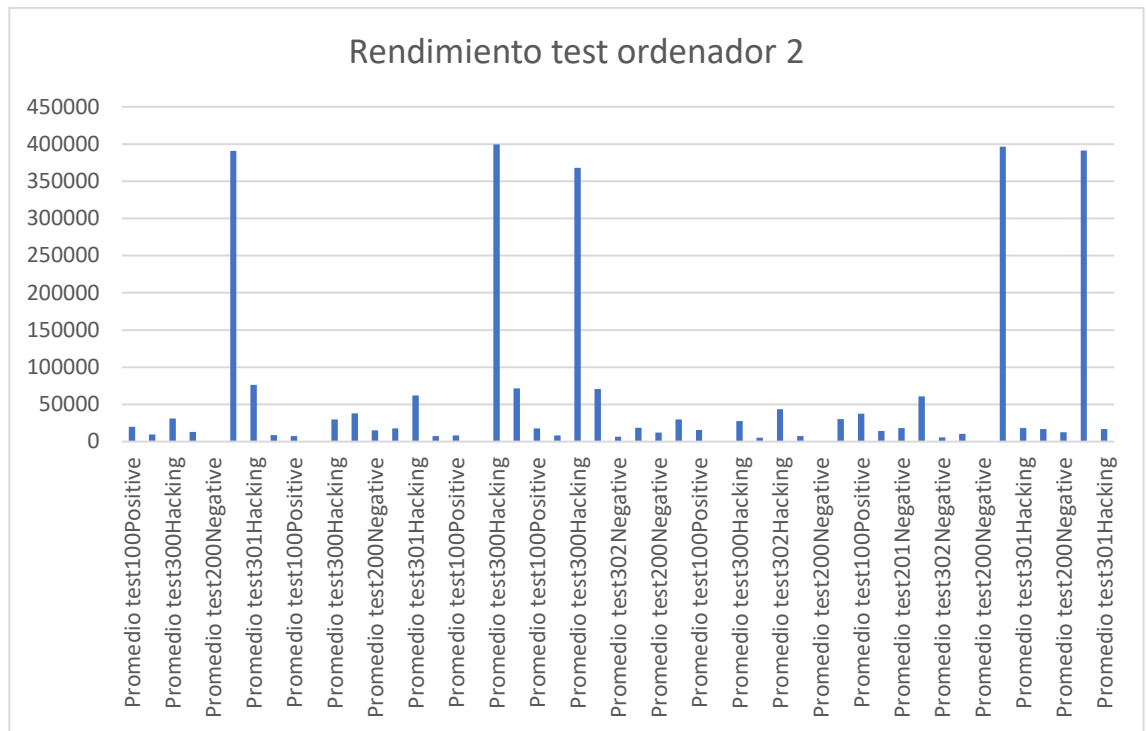
Para ello ejecutaremos las pruebas en 2 ordenadores diferentes el primero de ellos dispone de los siguientes componentes, un procesador Intel Core i5-9300H 2.5 GHz y 16 Gigas de RAM, el tiempo que ha tardado en ejecutar cada prueba puede verse en esta gráfica:



En cuanto a la velocidad con la que ha servido las peticiones en este ordenador, se muestra en la siguiente gráfica:



El segundo ordenador tiene como componentes principales, un procesador AMD Ryzen 7 3750H 2.3 GHz y 16 GB de RAM y el tiempo que ha tardado en ejecutar cada prueba puede verse en esta gráfica:



Y en cuanto a su tiempo con respecto cuanto tarda en servir las peticiones se muestra en la siguiente gráfica:



Tras realizar el estudio previo lo que nos queda es hacer los cálculos para definir nuestro rango de confianza y demostrar al cliente que estadísticamente las peticiones se sirven dentro de un rango controlado y lo hemos calculado mediante la siguiente tabla de Excel:

Columna1			Columna1		
Ordenador 1			Ordenador 2		
Media	6,16581957		Media	6,11054638	
Error típico	0,13198716		Error típico	0,14207205	
Mediana	5		Mediana	5	
Moda	5		Moda	5	
Desviación es	5,23641469		Desviación es	5,63651956	
Varianza de la	27,4200388		Varianza de la	31,7703528	
Curtosis	113,910677		Curtosis	189,156914	
Coeficiente de	7,32158469		Coeficiente de	10,7109553	
Rango	109		Rango	123	
Mínimo	0		Mínimo	1	
Máximo	109		Máximo	124	
Suma	9705		Suma	9618	
Cuenta	1574		Cuenta	1574	
Nivel de confi	0,25888927		Nivel de confi	0,27867053	
Intervalo (ms)	5,90693029	6,42470884	Intervalo (ms)	5,83187585	6,38921691
Intervalo (s)	0,00590693	0,00642471	Intervalo (s)	0,00583188	0,00638922

Después de esto vamos a realizar un análisis con la prueba z para averiguar cuál es el ordenador con mejor rendimiento y tras realizar la prueba podemos observar en la siguiente tabla que el resultado es más o menos similar, pero sí que es cierto que se observan algunas mejoras en el ordenador 2 por lo que los datos que tomaremos de referencia son de este.

	Ordenador 1	Ordenador 2
Media	6,165819568	6,110546379
Varianza (conocida)	27,4200388	31,7703528
Observaciones	1574	1574
Diferencia hipotética de las medias	0	
z	0,285030465	
P(Z<=z) una cola	0,387810401	
Valor crítico de z (una cola)	1,644853627	
Valor crítico de z (dos colas)	0,775620802	
Valor crítico de z (dos colas)	1,959963985	

Se observan que las pruebas son parecidas debido a que el valor de z es cercano a 1 pero fijándonos en la media el ordenador 2 obtiene una mejor media en cuanto a la duración de la ejecución de las pruebas por lo que tomaremos de referencia los cálculos obtenidos en la prueba de rendimiento del segundo ordenador.

Columna1		
Ordenador 2		
Media	6,11054638	
Error típico	0,14207205	
Mediana	5	
Moda	5	
Desviación es	5,63651956	
Varianza de la	31,7703528	
Curtosis	189,156914	
Coefficiente de	10,7109553	
Rango	123	
Mínimo	1	
Máximo	124	
Suma	9618	
Cuenta	1574	
Nivel de confi	0,27867053	
Intervalo (ms)	5,83187585	6,38921691
Intervalo (s)	0,00583188	0,00638922

Obteniendo como resultado el intervalo de confianza de entre 0,00583188 y 0,00638922 segundos.

Conclusiones.

Como conclusión de este informe podemos decir que las pruebas han sido muy útiles dado que hemos encontrado fallos en ellas, por lo que hemos podido refinar la implementación pedida por el cliente y nos hemos asegurado de que todo funciona como debería y como se espera que se comporte.

Además, hemos analizado cuanto se demora el sistema en realizar las peticiones y hemos hecho un análisis estadístico de los datos proporcionados pudiendo afirmar a nuestro cliente objetivamente que las peticiones del sistema se sirven dentro del intervalo establecido, mediante el cálculo de un intervalo de confianza y dicho intervalo de confianza se encuentra entre 0,00583188 y 0,00638922 segundos.

Bibliografía.

Intencionadamente vacío.