

# INFORME DE PRUEBAS

GRUPO C1.02.03



## Autores:

- David Cortabarra Romero – [davcorrom@alum.us.es](mailto:davcorrom@alum.us.es)

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

Fecha del documento: 22/05/2023

Tabla de contenidos

Resumen general ..... 3

Tabla de revisión..... 4

Introducción. .... 5

Contenido ..... 6

    Descripción de las pruebas realizadas ..... 6

    Rendimiento..... 11

Bibliografía..... 16

## Resumen general

Este informe de pruebas se centra fundamentalmente en describir el conjunto de pruebas que se han elaborado y ejecutado con la finalidad de probar la correcta implementación de las funcionalidades de nuestra aplicación. Concretamente, en este documento se va a hablar de las pruebas relacionadas con las prácticas (*Practicum*) y las sesiones de práctica (*PracticumSession*). Así, podemos decir que se han creado pruebas para las operaciones de creación, eliminación, muestra, listado, publicación y actualización de las prácticas; y pruebas para las operaciones de creación, eliminación, muestra, listado y actualización de las sesiones de práctica.

Por otra parte, se ha analizado el rendimiento de la aplicación a partir de las pruebas ejecutadas en dos ordenadores con distintas especificaciones. Así, se ha obtenido una serie de gráficas que representan el tiempo medio en realizar cada una de las peticiones que soporta el sistema, y el tiempo necesario en ejecutar cada caso de prueba.

Por último, se han calculado métricas asociadas al rendimiento, entre ellas, el intervalo de confianza, consiguiendo así, un intervalo de tiempo de respuesta a las peticiones que se le garantiza al cliente.

## Tabla de revisión

Fecha	Versión	Descripción
20/05/2023	0.1.0	Elaboración del documento.
22/05/2023	1.0.0	Correcciones y mejoras del documento.

## Introducción.

El presente informe tiene como objetivo documentar el conjunto de pruebas realizadas en nuestro proyecto. Nos vamos a centrar concretamente en las pruebas a las funcionalidades implementadas para el *Student #4*. Estamos hablando, de esta forma, de las pruebas sobre las operaciones para las prácticas y para las sesiones de práctica. Además de esto, se llevará a cabo un análisis del rendimiento evaluando los tiempos empleados para resolver las peticiones y generando una serie de gráficas que ayuden a comprender mejor los datos obtenidos.

Es importante explicar la estructura de este informe de pruebas, el cual está organizado de la siguiente forma: en primer lugar nos encontramos con una portada donde se reflejan los datos del proyecto, el autor del documento y la fecha de la última versión del informe; a continuación, tenemos el índice donde se pueden ver los diferentes apartados que se van a desarrollar; luego, vamos a tener un resumen general que mostrará todo el contenido de una forma sintetizada para poder conocer las nociones más importantes que se tratan sin necesidad de entrar en profundidad, continuando con una tabla de versiones del documento, y, posteriormente, el desarrollo del mismo con la introducción, el contenido y las conclusiones; y, por último, la bibliografía donde se hace referencia a los lugares que han servido de ayuda para la redacción del informe.

## Contenido

### Descripción de las pruebas realizadas

En este apartado se van a describir las diferentes pruebas que se han realizado viendo en detalle los pasos que se han seguido para realizar las comprobaciones pertinentes sobre una determinada funcionalidad.

Comencemos hablando de las pruebas realizadas a las prácticas (*Practicum*):

Create (*CompanyPracticumCreateTest*):

- Test100Positive(): Mediante esta prueba, iniciamos sesión como una empresa y listamos todas sus prácticas, para luego crear nuevas prácticas y corroborar que se han creado correctamente. Concretamente, iniciamos sesión como “company5”, la cual no tiene ninguna práctica asociada. Accedemos al listado de las prácticas, el cual estará vacío, y crearemos una práctica pulsando en el correspondiente botón. Una vez rellenados todos los campos y creada la práctica, se comprobará si esta se ha creado correctamente verificando si aparece en el listado de prácticas y viendo si sus detalles son correctos al mostrarla. Este procedimiento se repite para todas las prácticas que se vayan a crear.
- Test200Negative(): Esta prueba intenta crear una práctica con datos incorrectos. Seguimos el mismo procedimiento que en la prueba anterior, sin embargo, en este caso, al rellenar los campos y pulsar en el botón para crear la práctica, debe aparecer algún mensaje de error en uno o varios de los campos a rellenar que indique que los datos introducidos no son correctos.
- Test300Hacking(): Esta prueba intenta crear una nueva práctica usando un rol distinto de “Company”. Para ser exactos, se intenta acceder a la URL “/company/practicum/create” sin haber iniciado sesión, o bien, con la sesión iniciada como “lecturer1”, “student1”, “assistant1” o “auditor1”. Por último, se verifica si se produce un *panic* como resultado de la petición.

Delete (*CompanyPracticumDeleteTest*):

- Test100Positive(): Mediante esta prueba, iniciamos sesión como una empresa y listamos todas sus prácticas, a continuación, seleccionamos una de ellas y la eliminamos. Concretamente, iniciamos sesión como “company1” y accedemos a su listado de prácticas. Luego pulsamos en una de las prácticas sin publicar y la borramos, comprobando que no se han producido errores.
- Test200Negative(): Para esta funcionalidad no existe ninguna prueba negativa ya que no es posible obtener un mensaje de error pulsando el botón de eliminar.
- Test300Hacking(): Esta prueba intenta eliminar una práctica concreta con un rol distinto a “Company” lo cual resulta en un *panic*.
- Test301Hacking(): Esta prueba intenta eliminar una práctica publicada por la empresa que previamente había creado la práctica, lo cual resulta en un *panic*.

- Test302Hacking(): Esta prueba intenta eliminar una práctica creada por una empresa determinada siendo otra empresa completamente distinta.

List (*CompanyPracticumListTest*):

- Test100Positive(): Mediante esta prueba, iniciamos sesión como una empresa, en este caso “*company1*”, listamos todas sus prácticas y comprobamos que los datos del listado son los esperados.
- Test200Negative(): Para esta funcionalidad no existe ninguna prueba negativa ya que no involucra introducir datos en un formulario, por lo tanto, no es posible obtener un mensaje de error.
- Test300Hacking(): Esta prueba intenta listar las prácticas de una empresa con un rol distinto a “*Company*”, lo cual resulta en un *panic*.

Publish (*CompanyPracticumPublishTest*):

- Test100Positive(): Mediante esta prueba, iniciamos sesión como una empresa y listamos todas sus prácticas, luego, seleccionamos una de ellas y la publicamos. Concretamente, iniciamos sesión como “*company1*”, accedemos a su listado de prácticas y entramos en una práctica sin publicar. Pulsamos el botón correspondiente para publicar y comprobamos que no se produce ningún tipo de error.
- Test200Negative(): Esta prueba intenta publicar una práctica que no puede ser publicada todavía debido a que no tiene ninguna sesión de práctica asociada. Por lo tanto, al intentar publicar la práctica se devuelve un mensaje de error impidiendo la realización de la operación.
- Test300Hacking(): Esta prueba intenta publicar una práctica con un rol distinto a “*Company*”, lo cual resulta en un *panic*.
- Test301Hacking(): Esta prueba intenta publicar una práctica de la empresa que ya la había publicado anteriormente, lo cual resulta en *panic*.
- Test302Hacking(): Esta prueba intenta publicar una práctica, ya publicada o no, por una empresa distinta a la que creó la práctica. Esto resultará en un *panic*.

Show (*CompanyPracticumShowTest*):

- Test100Positive(): Mediante esta prueba, iniciamos sesión como una empresa, en este caso, “*company1*” y, luego, listamos todas sus prácticas y vamos entrando en ellas para comprobar que el formulario tiene los datos correctos.
- Test200Negative(): Para esta funcionalidad no existe ninguna prueba negativa ya que no involucra introducir datos en un formulario, por lo tanto, no es posible obtener un mensaje de error.
- Test300Hacking(): Esta prueba intenta mostrar una práctica que no está publicada por alguien distinto al que ha creado la práctica. Concretamente se prueba a mostrar una práctica creada por “*company2*” sin iniciar sesión, o bien, con la sesión iniciada como: “*administrator*”, “*lecturer1*”, “*student1*”, “*assistant1*”, “*company1*” o “*auditor1*”. La prueba resultará en *panic* para todos estos casos.

Update (*CompanyPracticumUpdateTest*):

- Test100Positive(): Mediante esta prueba, iniciamos sesión como una empresa y listamos sus prácticas. Luego, seleccionamos una de ellas y la actualizamos, para luego comprobar que la actualización se ha llevado a cabo correctamente. Concretamente, iniciamos sesión como “*company1*”, accedemos a su listado de prácticas y entramos en una práctica sin publicar. A continuación, modificamos los campos del formulario de la práctica y la actualizamos, para luego, comprobar que la práctica aparece actualizada en el listado y una vez se muestra.
- Test200Negative(): Esta prueba intenta actualizar una práctica con datos incorrectos. Se sigue el mismo procedimiento que en la prueba anterior, a diferencia que esta vez, al modificar los datos del formulario e intentar actualizar la práctica, se produce un mensaje de error.
- Test300Hacking(): Esta prueba intenta actualizar una práctica con un rol distinto a “*Company*” o mediante una empresa que no es la creadora de la práctica. Esto resultará en un *panic*.
- Test301Hacking(): Esta prueba intenta actualizar una práctica que ya ha sido publicada, lo cual resulta en un *panic*.

Pasemos ahora a hablar de las pruebas realizadas a las sesiones de práctica (*PracticumSession*):

Create (*CompanyPracticumSessionCreateTest*):

- Test100Positive(): Mediante esta prueba, iniciamos sesión como una empresa, en este caso “*company1*”, y listamos sus prácticas. Luego, seleccionamos una de ellas que no haya sido publicada y listamos sus sesiones de práctica para, a continuación, crear una nueva rellenando el formulario pertinente y comprobar que esta se ha creado correctamente tanto en el listado como al mostrar la práctica con los datos previamente introducidos.
- Test101Positive(): Mediante esta prueba, iniciamos sesión como una empresa, en este caso “*company1*”, y listamos sus prácticas. Luego, seleccionamos una de las prácticas que se encuentren publicadas y listamos sus sesiones de práctica para, a continuación, crear una sesión de práctica excepcional rellenando el formulario pertinente y marcando la casilla de confirmación. Para finalizar, se comprueba que la práctica excepcional se ha creado correctamente tanto en el listado como al mostrar la práctica con los datos previamente introducidos.
- Test200Negative(): Esta prueba intenta crear una nueva sesión de práctica usando datos incorrectos. Se sigue exactamente el mismo procedimiento que en el Test100Positive aunque, en este caso, al introducir los datos en el formulario e intentar crear la sesión de práctica, se debería devolver algún mensaje de error en uno o varios campos del formulario.



- Test201Negative(): Esta prueba intenta crear una sesión de práctica adicional sin marcar la casilla de confirmación en el formulario. Esto resultará en un mensaje de error.
- Test300Hacking(): Esta prueba intenta crear una sesión de práctica con un rol distinto a *"Company"*. Para ser exactos, se intenta acceder a la URL *"/company/practicum-session/create"* sin haber iniciado sesión, o bien, con la sesión iniciada como *"lecturer1"*, *"student1"*, *"assistant1"* o *"auditor1"*. Por último, se verifica si se produce un *panic* como resultado de la petición para todos estos casos.
- Test301Hacking(): Esta prueba intenta crear una sesión de práctica para una práctica por una empresa que no es la que creó la práctica. Esto debería resultar en un *panic*.

Delete (*CompanyPracticumSessionDeleteTest*):

- Test100Positive(): Mediante esta prueba, iniciamos sesión como una empresa, en este caso *"company1"*, y listamos sus prácticas. Luego, seleccionamos una de ellas que no esté publicada, listamos sus sesiones de práctica, entramos en una y la eliminamos.
- Test200Negative(): Para esta funcionalidad no existe ninguna prueba negativa ya que no es posible obtener un mensaje de error pulsando el botón de eliminar.
- Test300Hacking(): Esta prueba intenta eliminar una sesión de práctica con un rol distinto a *"Company"*. Concretamente, se probará esta acción sin iniciar sesión o iniciando sesión como: *"lecturer1"*, *"student1"*, *"assistant1"* o *"auditor1"*. El resultado debería ser un *panic* para todos estos casos.
- Test301Hacking(): Esta prueba intenta eliminar una sesión de práctica de una práctica ya publicada. La empresa que creó la sesión de práctica y la que intenta eliminarla son la misma. El resultado debería ser un *panic*.
- Test302Hacking(): Esta prueba intenta eliminar una sesión de práctica de una práctica que no ha sido publicada aún. La empresa que creó la práctica es distinta a la que intenta eliminarla. El resultado debería ser un *panic*.

List (*CompanyPracticumSessionListTest*):

- Test100Positive(): Mediante esta prueba, iniciamos sesión como una empresa, en este caso *"Company1"*, y listamos sus prácticas. Luego, seleccionamos una de ellas y listamos sus sesiones de práctica comprobando que los resultados obtenidos son los esperados.
- Test200Negative(): Para esta funcionalidad no existe ninguna prueba negativa ya que no involucra introducir datos en un formulario, por lo tanto, no es posible obtener un mensaje de error.
- Test300Hacking(): Esta prueba intenta listar las sesiones de práctica de una práctica no publicada habiendo iniciado sesión como una empresa distinta a la que ha creado la práctica, o bien, como un rol completamente distinto a *"Company"*. Concretamente se prueban sin iniciar sesión, como *"lecturer1"*,

*“student1”, “assistant1”, “company1” o “auditor1”*. Esto resultará en un *panic* para todos estos casos.

Show (*CompanyPracticumSessionShowTest*):

- Test100Positive(): Mediante esta prueba, iniciamos sesión como una empresa, en este caso *“Company1”*, y listamos sus prácticas. Luego, seleccionamos una de las prácticas y listamos sus sesiones de práctica. A continuación, comprobamos si se muestran correctamente las sesiones de práctica con los datos pertinentes.
- Test200Negative(): Para esta funcionalidad no existe ninguna prueba negativa ya que no involucra introducir datos en un formulario, por lo tanto, no es posible obtener un mensaje de error.
- Test300Hacking(): Esta prueba intenta mostrar las sesiones de práctica de una práctica no publicada habiendo iniciado sesión como una empresa distinta a aquella que creó la práctica, o bien, como un rol completamente distinto a *“Company”*. Concretamente se prueban sin iniciar sesión, como *“lecturer1”, “student1”, “assistant1”, “company1” o “auditor1”*. Esto resultará en un *panic* para todos estos casos.

Update (*CompanyPracticumSessionUpdateTest*):

- Test100Positive(): Mediante esta prueba, iniciamos sesión como una empresa y listamos sus prácticas. Luego, seleccionamos una de ellas que no haya sido publicada, listamos sus sesiones de práctica, mostramos una de ellas y la actualizamos, para luego comprobar que la actualización se ha llevado a cabo correctamente. Concretamente, iniciamos sesión como *“company1”*, accedemos a su listado de prácticas y entramos en una práctica sin publicar. A continuación, listamos sus sesiones de práctica y mostramos una de ellas. Por último, modificamos los campos del formulario de la sesión de práctica y la actualizamos, para luego, comprobar que la sesión de práctica aparece actualizada en el listado y una vez se muestra.
- Test200Negative(): Esta prueba intenta actualizar una sesión de práctica con datos incorrectos. Se sigue el mismo procedimiento que en la prueba anterior, a diferencia que esta vez, al modificar los datos del formulario e intentar actualizar la sesión de práctica, se produce un mensaje de error.
- Test300Hacking(): Esta prueba intenta actualizar una práctica con un rol distinto a *“Company”* o mediante una empresa que no es la creadora de la práctica. Concretamente se prueban sin iniciar sesión, como *“lecturer1”, “student1”, “assistant1”, “company1” o “auditor1”*. Esto resultará en un *panic* para todos estos casos.
- Test301Hacking(): Esta prueba intenta actualizar una sesión de práctica perteneciente a una práctica que ya ha sido publicada, lo cual resulta en un *panic*.

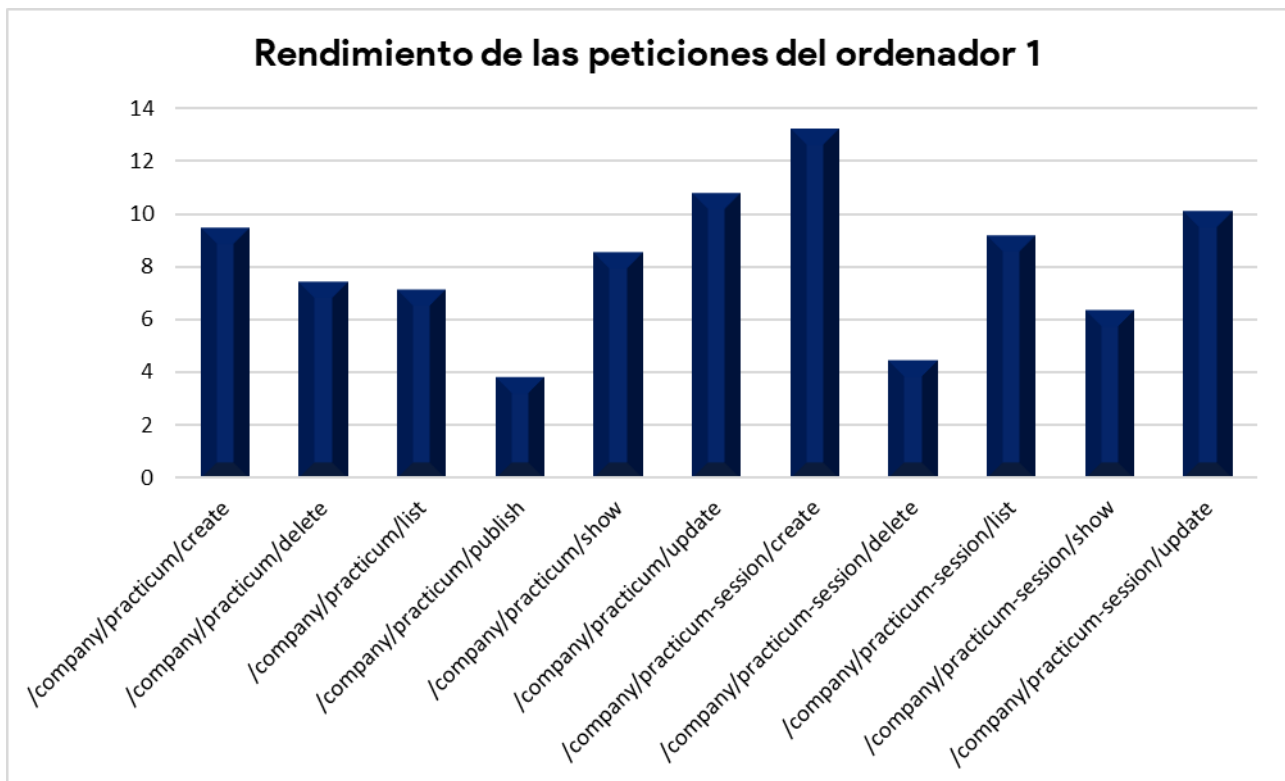
## Rendimiento

En este apartado vamos a hablar del rendimiento de las funcionalidades implementadas analizando los tiempos de ejecución de las pruebas que se han realizado. En primer lugar, veremos una serie de gráficas que recogen el rendimiento de las peticiones realizadas por dos ordenadores diferentes, así como el rendimiento de los casos de prueba.

Analicemos en primer lugar el tiempo medio en realizar las diferentes peticiones en dos ordenadores con distintas especificaciones.

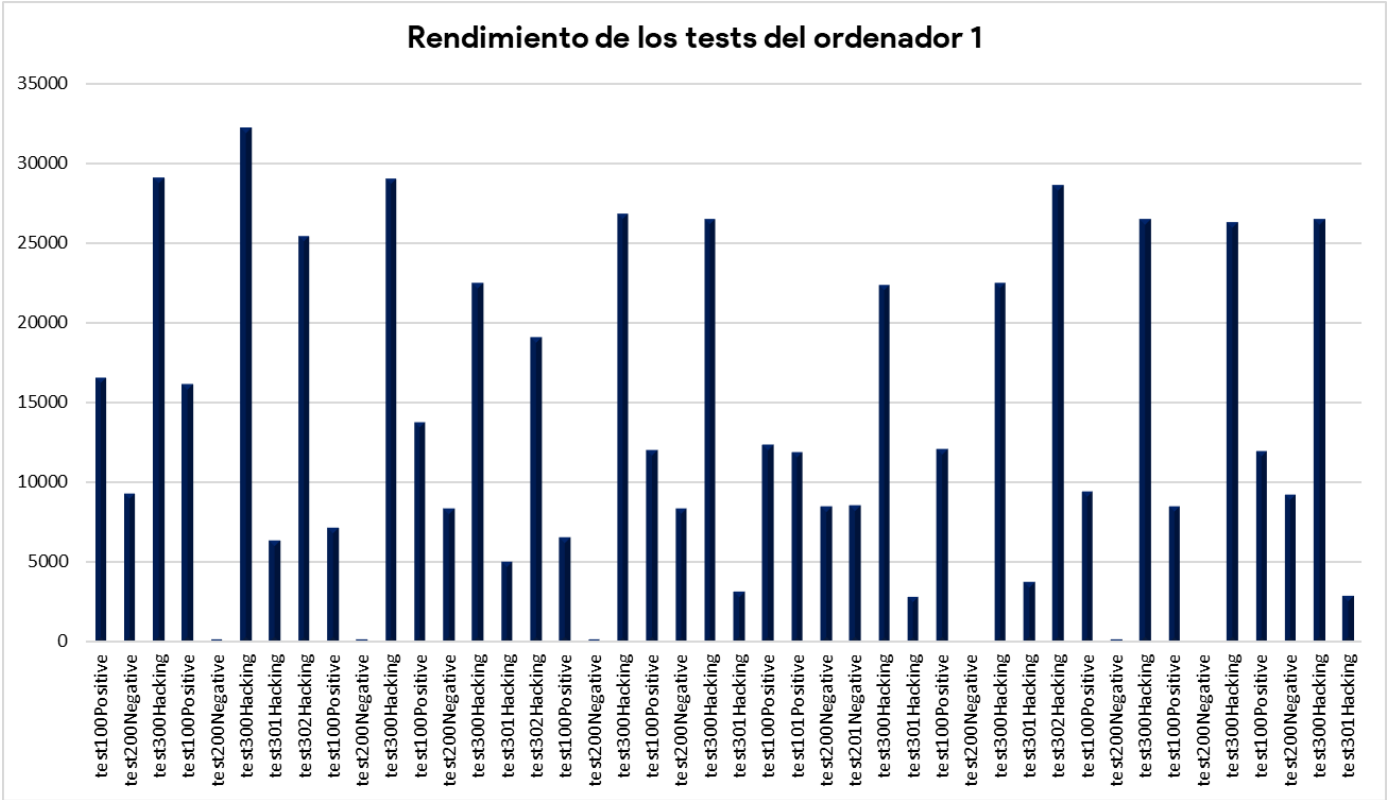
El ordenador 1 cuenta con las siguientes especificaciones: procesador AMD Ryzen 7 3750H a 2.3 GHz y 16 GB de memoria RAM.

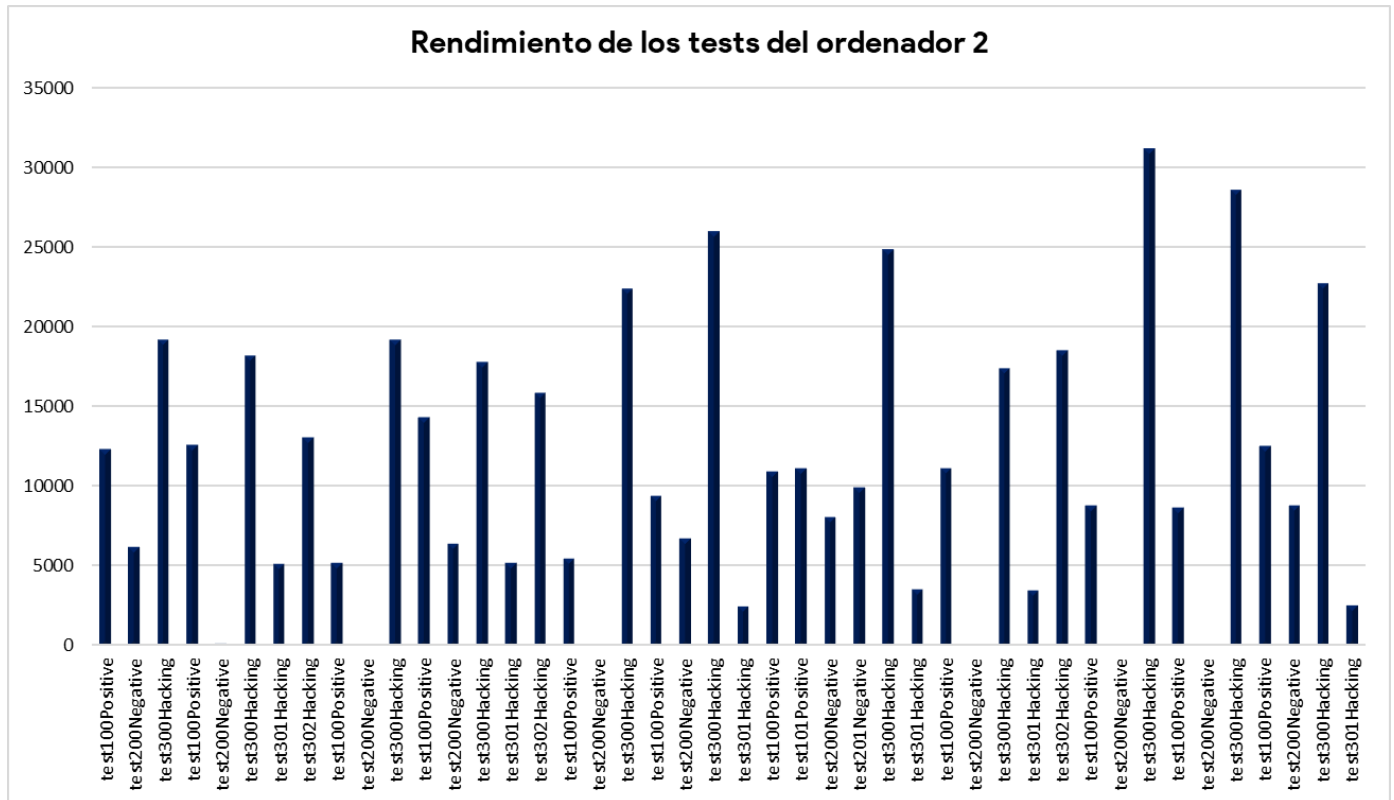
El ordenador 2 cuenta con las siguientes especificaciones: procesador Intel Core i5-9300H a 2.5 GHz y 16 GB de memoria RAM.





Ahora, veamos en las siguientes dos gráficas el tiempo que ha tardado en ejecutarse cada caso de prueba.





A continuación, vamos a analizar cuál de los dos ordenadores ha obtenido mejor rendimiento, para luego definir el intervalo de confianza. Para esto, se va a comprobar cuánto varían las medias de tiempo con respecto a ambos ordenadores. Concretamente, vamos a realizar un Z-Test para analizar los tiempos en servir las peticiones en ambos ordenadores. Con las métricas obtenidas, vamos a evaluar el rendimiento para determinar qué ordenador es mejor para el cálculo del intervalo de confianza. En la siguiente tabla vemos las métricas obtenidas tras la realización del Z-Test.

Prueba z para medias de dos muestras		
	Ordenador1	Ordenador2
Media	8,44272605	8,59223307
Varianza (conocida)	49,1653581	77,8692686
Observaciones	1193	1193
Diferencia hipotética de las medias	0	
z	-0,45816376	
P(Z<=z) una cola	0,3234174	
Valor crítico de z (una cola)	1,64485363	
Valor crítico de z (dos colas)	0,64683479	
Valor crítico de z (dos colas)	1,95996398	

El valor crítico de z (dos colas) es 0,64683479. Obtengamos el valor de *alpha*. En este caso,  $\alpha = 1 - \text{nivel de confianza}$ . Así,  $\alpha = 1 - 0,95 = 0,05$ .

Como el valor crítico de  $z$  (dos colas) se encuentra dentro del intervalo  $[0.05, 1.00)$ , los rendimientos de ambos ordenadores son muy similares, por lo que no existen diferencias notables entre ambos equipos para calcular el intervalo de confianza.

Para finalizar, se van a mostrar los cálculos de diferentes métricas estadísticas gracias a los resultados obtenidos al realizar las diferentes pruebas. Así, se va a definir el intervalo de confianza que se puede garantizar a los clientes para servir las peticiones. Se han calculado las métricas para el ordenador 1 pues, tal y como hemos visto antes, las diferencias de rendimiento entre ambos equipos en los que se ha ejecutado las pruebas no son notables.

Columna1					
Media	8,407190635		Interval (ms)	8,00936122	8,80502006
Error típico	0,20277234		Interval (s)	0,00800936	0,00880502
Mediana	7				
Moda	6				
Desviación estándar	7,012523058				
Varianza de la muestra	49,17547963				
Curtosis	162,468332				
Coeficiente de asimetría	10,33818708				
Rango	146				
Mínimo	0				
Máximo	146				
Suma	10055				
Cuenta	1196				
Nivel de confianza(95,0%)	0,39782942				

De media, las peticiones realizadas al sistema se servirán en un intervalo de entre 0.00800936 segundos y 0.00880502 segundos.

## Conclusiones

La creación de las pruebas ha servido para probar toda la funcionalidad creada en relación a las prácticas y las sesiones de práctica. Esto es muy útil para detectar diferentes errores que hayan podido surgir durante el proceso de desarrollo con el objetivo de solventarlos, mejorando así la aplicación creada. Cada una de las pruebas cubre una parte de la funcionalidad del sistema, facilitando la identificación de los problemas.

En cuanto al rendimiento, se puede afirmar que la ejecución de las pruebas ha sido un éxito logrando unos buenos tiempos de respuesta a las peticiones. Además, podemos garantizar al cliente que la aplicación funcionará sin problemas dentro del intervalo de tiempo calculado, tal y como se ha podido observar en el análisis del rendimiento.

## Bibliografía

Intencionalmente en blanco.