Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática

**Testing Report**



Grado en Ingeniería Informática – Ingeniería del Software  
Diseño y Pruebas 2.

Curso 2024 – 2025

|  |  |
| --- | --- |
| **Fecha** | **Versión** |
| 22/05/2025 | v1.0.0 |

|  |  |  |
| --- | --- | --- |
| **Grupo de prácticas: C1.005** | | |
| **Autores por orden alfabético** | **Rol** | **Descripción del rol** |
| Artero Bellido Manuel – manartbel@alum.us.es | Tester | Persona encargada de realizar pruebas sobre el código |
| Calderón Rodríguez, Manuel María -mancalrod@alum.us.es | Desarrollador | Persona encargada de desarrollar el código |
| González Benito, Claudio – clagonben@alum.us.es | Jefe de proyecto | Encargado de tomar decisiones de diseño y vigilar el correcto desarrollo. |
| Márquez Gutiérrez, José Manuel – josmargut@alum.us.es | Desarrollador | Persona encargada de desarrollar el código |
| Ramos Vargas, Alba – albramvar1@alum.us.es | Operador | Encargado de las tareas de campo, de las instalaciones y del mantenimiento de los sistemas de la empresa |

**Control de Versiones**

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** |
| 22/05/2025 | v1.0.0 | Desarrollo de la primera versión. |
|  |  |  |
|  |  |  |
|  |  |  |

Repositorio: <https://github.com/Manuelgithuv/Acme-ANS-D04>

Índice de contenido

[**1.** **Resumen ejecutivo** 2](#_Toc198917376)

[**2.** **Introducción** 3](#_Toc198917377)

[**3.** **Testeo funcional** 4](#_Toc198917378)

[**4.** **Reporte sobre las mutaciones** 13](#_Toc198917379)

[**5.** **Cobertura de las operaciones** 14](#_Toc198917380)

[**6.** **Testeo de rendimiento** 16](#_Toc198917381)

[**7.** **Conclusiones** 19](#_Toc198917382)

[**8.** **Bibliografía** 20](#_Toc198917383)

# **Resumen ejecutivo**

Este documento consiste en el informe necesario para cumplir con el requisito individual del estudiante 1 del reporte de pruebas funcionales y de rendimiento del estudiante 1 sobre sus entidades de mánager, de manera que consten los bugs encontrados en las pruebas, que tipo de pruebas se han realizado sobre las distintas operaciones y como de distinto es el rendimiento de la aplicación en distintos ordenadores.

El principal objetivo de esta entrega es probar de manera formal el funcionamiento correcto de nuestra aplicación, grabando para ello las pruebas necesarias sobre distintas operaciones legales e ilegales en las funcionalidades implementadas en la anterior entrega.

# **Introducción**

Este documento presenta dos capítulos esenciales: pruebas funcionales, con un listado de casos de prueba organizados por característica y su efectividad en detectar errores, y pruebas de rendimiento, con gráficos e intervalos de confianza del 95% para el tiempo de respuesta en dos computadoras, además de un análisis para determinar cuál es más potente. Estos capítulos son cruciales para evaluar la calidad y el rendimiento del software.

# **Testeo funcional**

En esta sección se detallarán todas las pruebas que hemos realizado para testear el funcionamiento de nuestro software. Se explicará qué es lo que se ha hecho, el resultado esperado, el resultado real (que hace referencia al resultado de la prueba corregido después de encontrar bugs que hicieran que el código estuviera mal) y, por último, la columna de bugs encontrados. Si ha habido bugs, se ha arreglado la prueba, por lo que el resultado real ya no los contiene.

Descripción de la prueba: Se detallará cada prueba realizada, incluyendo los pasos específicos llevados a cabo.

Resultado esperado: Se especificará cuál era el comportamiento o el resultado que se esperaba obtener con la prueba.

Resultado real: Se presentará el resultado obtenido tras la ejecución de la prueba, ya corregido después de encontrar y solucionar cualquier bug que afectará el código.

Bugs encontrados: Se listarán los bugs detectados durante la prueba y las correcciones realizadas. Una vez solucionados los bugs, se actualizará el resultado real para reflejar el funcionamiento correcto del software.

**List**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidad** | **Descripción** | **Resultado Esperado** | **Resultado Real** | **Bugs detectados** |
| **Flight safe** | Verificar que un mánager puede listar sus vuelos. | El sistema deberá mostrar los vuelos asociados al mánager | El sistema ha mostrado los vuelos asociados al mánager. | Ninguno |
| **Flight hack** | Verificar que un rol no mánager no puede listar los vuelos de los managers. | El sistema no debe mostrar los vuelos, usando un rol no mánager, dando un error de pánico | El sistema ha dado un error 500 de no autorizado cuando el rol no es mánager. | Ninguno |
| **Leg safe** | Comprobar que un mánager puede listar los tramos de sus vuelos. | El sistema deberá mostrar los tramos asociados a un vuelo del mánager. | El sistema ha mostrado todos los tramos asociados a los tramos de los vuelos de un mánager. | Ninguno |
| **Leg hack** | Comprobar que un rol no mánager no pueda acceder a los tramos de los vuelos y que un mánager no pueda acceder a los tramos de un vuelo de otro mánager y de un vuelo que no existe. | El sistema no debe mostrar los vuelos siendo un rol no mánager, intentando hacerlo sobre los tramos de un vuelo de otro mánager y sobre un vuelo inexistente, dando un 500 como resultado. | El sistema ha detectado cuando se intentaba acceder desde un rol no mánager y desde el vuelo de otro mánager, pero daba un 500 de null pointer exception cuando el vuelo era inexistente. | Saltaba **null pointer exception** cuando se **modificaba** la **url** usando un **id** de **vuelo** **no existente**, pues no se comprobaba si existía antes de intentar comprobar que no fuera un vuelo de otro mánager. Se solucionó y ahora el 500 es Not Authorised. |

**Show**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidad** | **Descripción** | **Resultado Esperado** | **Resultado Real** | **Bugs detectados** |
| **Flight safe** | Verificar que un mánager puede ver toda la información de sus vuelos. | El sistema deberá mostrar la información de los vuelos asociados al mánager. | El sistema ha mostrado la información de los vuelos asociados al mánager. | Ninguno |
| **Flight hack** | Verificar que un rol no mánager no puede ver la información de los vuelos, que otro mánager ver la información de un vuelo de otro mánager, ni un mánager intentar mostrar información de un vuelo no existente. | El sistema no debe mostrar la información de vuelos, usando un rol no mánager, ni mostrar información de vuelos de otro mánager, ni autorizar el intento de mostrar información de vuelos no existentes, dando un error de pánico. | El sistema ha dado un error 500 de no autorizado usando roles no mánager, usando vuelos de otro mánager y usando vuelos no existentes. | Ninguno |
| **Leg safe** | Verificar que un mánager puede ver toda la información de los tramos de sus vuelos. | El sistema deberá mostrar la información de los tramos asociados a los vuelos del mánager. | El sistema ha mostrado la información de los tramos asociados a los vuelos del mánager. | Ninguno |
| **Leg hack** | Verificar que un rol no mánager no puede ver la información de los tramos, que otro mánager ver la información de un tramo de otro mánager, ni un mánager intentar mostrar información de un tramo no existente. | El sistema no debe mostrar la información de tramos, usando un rol no mánager, ni mostrar información de tramos de otro mánager, ni autorizar el intento de mostrar información de tramos no existentes, dando un error de pánico. | El sistema ha dado un error 500 de no autorizado en cada rol El sistema ha dado un error 500 de no autorizado usando roles no mánager, usando tramos de otro mánager y usando tramos no existentes. | Ninguno |

**Create**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidad** | **Descripción** | **Resultado Esperado** | **Resultado Real** | **Bugs detectados** |
| **Flight safe** | Verificar que un mánager puede crear vuelos de forma correcta. | El sistema deberá permitir crear vuelos válidos. Y rechazar aquellos que no cumplan con las restricciones. | El sistema ha dejado crear vuelos con datos válidos y ha rechazado la creación de vuelos con campos nulos cuando debía haber un mínimo o cuando el tipo de campo era incorrecto. | Ninguno |
| **Flight hack** | Verificar que un rol no mánager no pueda acceder a la creación de vuelos, que un mánager no pueda crear vuelos a otros managers y que pudiera crearlos con datos de solo lectura mediante su modificación en la herramienta de inspección de manera que estos se ignoren. | El sistema deberá dar un 500 al intentar acceder a la creación desde un rol no mánager, ha de permitir la creación siempre que se sea mánager ya que se hace sobre el mánager autenticado, es decir que no se puede hacer Get Hacking y por último debe ignorar los campos de solo lectura escritos en la herramienta de inspección. | El sistema ha dado un error 500 de no autorizado en los roles no mánager. Ha permitido la creación desde un rol mánager, ya que la url no es posible hacerle Get Hacking y a ignorado los campos de solo lectura escritos. | Ninguno |
| **Leg safe** | Verificar que un mánager puede crear tramos de vuelos de forma correcta. | El sistema deberá permitir crear tramos válidos y rechazar aquellos datos que no cumplan con las restricciones de negocio, como fechas de tramo anterior a la actual, tramos solapados o aircrafts en uso entre otras. | El sistema ha permitido crear tramos válidos. Y ha rechazado la creación de tramos que incumplían reglas del del negocio | Ninguno |
| **Leg hack** | Verificar que un rol no mánager no pueda acceder a la creación de tramos, que un mánager no pueda crear tramos a otros managers y que no pudiera crearlos con datos navegables que no aparecieran en los desplegables. | El sistema debe de lanzar error de pánico si un rol no mánager intenta acceder a la URL, si se intenta crear un tramo cuyo vuelo pertenece a otro mánager, si se intenta crear para un vuelo publicado. También deberá dar errores 500 al intentar cambiar los ids de los atributos navegables por unos no existentes. | El sistema ha dado un 500 al intentar crear tramo en un rol no mánager, al intentar hacerlo sobre un tramo de un vuelo de otro mánager y sobre un vuelo publicado. También ha dado un 500 por cada post hacking hecho cambiando los datos navegables por unos no existentes excepto para cuando un aircraft si existe, pero está en mantenimiento, que ha dado un error recuperable. | Ninguno |

**Update**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidad** | **Descripción** | **Resultado Esperado** | **Resultado Real** | **Bugs detectados** |
| **Flight safe** | Verificar que un mánager puede editar sus vuelos de vuelos de forma correcta. | El sistema deberá permitir actualizar vuelos válidos. Y deberá rechazar la edición de vuelos con datos que incumplan las reglas de negocio. | El sistema ha dejado actualizar vuelos con datos válidos. Y ha rechazado los de los datos inválidos. | Ninguno |
| **Flight hack** | Verificar que un rol no mánager no pueda acceder a la edición de vuelos, que un mánager no pueda editar vuelos a que no estén en modo borrador por un show previo y que pudiera editarlos con datos de solo lectura mediante su modificación en la herramienta de inspección de manera que estos se ignoren. | El sistema deberá dar un 500 al intentar acceder a la edición desde un rol no mánager, o cuando se pone la url de update sin haber hecho un show antes y por último debe ignorar los campos de solo lectura escritos en la herramienta de inspección. | El sistema ha generado un 500 al intentar acceder a la edición desde un rol no mánager, o cuando se pone la url de update sin haber hecho un show antes y por último ha ignorado los campos de solo lectura escritos en la herramienta de inspección. | Ninguno |
| **Leg safe** | Verificar que un mánager puede editar tramos de vuelos de forma correcta. | El sistema deberá actualizar tramos válidos. Y rechazar la actualización de tramos que incumplan las reglas de negocio. | El sistema ha dejado actualizar tramos con datos válidos. Y ha rechazado los de los datos inválidos. | Ninguno |
| **Leg hack** | Verificar que un rol no mánager no pueda acceder a la edición de tramos, que un mánager no pueda editar tramos que no estén en modo borrador por un show previo y que no pudiera editarlos con datos navegables que no aparecieran en los desplegables. | El sistema debe de lanzar error de pánico si un rol no mánager intenta acceder a la URL y si se intenta editar un tramo sin haberle hecho un show. También deberá dar errores 500 al intentar cambiar los ids de los atributos navegables por unos no existentes. | El sistema ha dado un 500 al intentar editar tramos en un rol no mánager y al intentar hacerlo sobre un tramo que no estuviera en un modo borrador por un show previo. También ha dado un 500 por cada post hacking hecho cambiando los datos navegables por unos no existentes excepto para cuando un aircraft si existe, pero está en mantenimiento, que ha dado un error recuperable. | Ninguno |

**Publish**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidad** | **Descripción** | **Resultado Esperado** | **Resultado Real** | **Bugs detectados** |
| **Flight safe** | Verificar que un mánager puede publicar sus vuelos de vuelos de forma correcta. | El sistema deberá permitir publicar vuelos válidos. Y deberá rechazar la publicación de vuelos con datos que incumplan las reglas de negocio. | El sistema ha dejado publicar vuelos con datos válidos. Y ha rechazado los de los datos inválidos. | Ninguno |
| **Flight hack** | Verificar que un rol no mánager no pueda acceder a la publicación de vuelos, que un mánager no pueda publicar vuelos a que no estén en modo borrador por un show previo y que pudiera publicarlos con datos de solo lectura mediante su modificación en la herramienta de inspección de manera que estos se ignoren. | El sistema deberá dar un 500 al intentar acceder a la publicación desde un rol no mánager, o cuando se pone la url de publish sin haber hecho un show antes y por último debe ignorar los campos de solo lectura escritos en la herramienta de inspección. | El sistema ha generado un 500 al intentar acceder a la publicación desde un rol no mánager, o cuando se pone la url de publish sin haber hecho un show antes y por último ha ignorado los campos de solo lectura escritos en la herramienta de inspección. | Ninguno |
| **Leg safe** | Verificar que un mánager puede publicar tramos de vuelos de forma correcta. | El sistema deberá publicar tramos válidos. Y rechazar la publicación de tramos que incumplan las reglas de negocio. | El sistema ha dejado publicar tramos con datos válidos. Y ha rechazado los de los datos inválidos. | Ninguno |
| **Leg hack** | Verificar que un rol no mánager no pueda acceder a la publicación de tramos, que un mánager no pueda publicar tramos que no estén en modo borrador por un show previo y que no pudiera publicarlos con datos navegables que no aparecieran en los desplegables. | El sistema debe de lanzar error de pánico si un rol no mánager intenta acceder a la URL y si se intenta publicar un tramo sin haberle hecho un show. También deberá dar errores 500 al intentar cambiar los ids de los atributos navegables por unos no existentes. | El sistema ha dado un 500 al intentar publicar tramos en un rol no mánager y al intentar hacerlo sobre un tramo que no estuviera en un modo borrador por un show previo. También ha dado un 500 por cada post hacking hecho cambiando los datos navegables por unos no existentes excepto para cuando un aircraft si existe, pero está en mantenimiento, que ha dado un error recuperable. | Ninguno |

**Delete**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidad** | **Descripción** | **Resultado Esperado** | **Resultado Real** | **Bugs detectados** |
| **Flight safe** | Verificar que un mánager puede eliminar sus vuelos de vuelos de forma correcta. | El sistema deberá permitir eliminar vuelos válidos. Y deberá rechazar la eliminación de vuelos con datos que incumplan las reglas de negocio, como que tenga tramos publicados. | El sistema ha dejado eliminar vuelos con datos válidos. Y ha rechazado los de los datos inválidos. | Ninguno |
| **Flight hack** | Verificar que un rol no mánager no pueda acceder a la eliminación de vuelos, que un mánager no pueda eliminar vuelos a que no estén en modo borrador por un show previo y que pudiera eliminarlos con datos de solo lectura mediante su modificación en la herramienta de inspección de manera que estos se ignoren. | El sistema deberá dar un 500 al intentar acceder a la eliminación desde un rol no mánager, o cuando se pone la url de delete sin haber hecho un show antes y por último debe ignorar los campos de solo lectura escritos en la herramienta de inspección. | El sistema ha generado un 500 al intentar acceder a la eliminación desde un rol no mánager, o cuando se pone la url de delete sin haber hecho un show antes y por último ha ignorado los campos de solo lectura escritos en la herramienta de inspección. | Ninguno |
| **Leg safe** | Verificar que un mánager puede eliminar tramos de vuelos de forma correcta. | El sistema deberá eliminar tramos válidos. Y rechazar la eliminación de tramos que incumplan las reglas de negocio. | El sistema ha dejado eliminar tramos con datos válidos. Y ha rechazado los de los datos inválidos. | Ninguno |
| **Leg hack** | Verificar que un rol no mánager no pueda acceder a la eliminación de tramos, que un mánager no pueda eliminar tramos que no estén en modo borrador por un show previo y que no pudiera eliminarlos con datos navegables que no aparecieran en los desplegables. | El sistema debe de lanzar error de pánico si un rol no mánager intenta acceder a la URL y si se intenta eliminar un tramo sin haberle hecho un show. También deberá dar errores 500 al intentar cambiar los ids de los atributos navegables por unos no existentes. | El sistema ha dado un 500 al intentar eliminar tramos en un rol no mánager y al intentar hacerlo sobre un tramo que no estuviera en un modo borrador por un show previo. También ha dado un 500 por cada post hacking hecho cambiando los datos navegables por unos no existentes. | Ninguno |

# **Reporte sobre las mutaciones**

1. Mutación en create leg service

La línea afectada es la 77, el cambio es: status = flight != null && flight.getManager().getId() **!=** super.getRequest().getPrincipal().getActiveRealm().getId() && !flight.isPublished() && manager.getId() **!=** flight.getManager().getId() && entitiesExist && flightId != 0;

Imagen que contiene agua, sostener, grande, tabla

El contenido generado por IA puede ser incorrecto.

Como se puede ver, al haber cambiado la lógica de autorización, las operaciones de POST legales sobre la creación de un leg ya no devuelven un 200, sino un 500 porque no puede autorizar ninguna petición.

1. Mutación en create flight service

La línea afectada es la 38, el cambio es: flight.setManager(null);

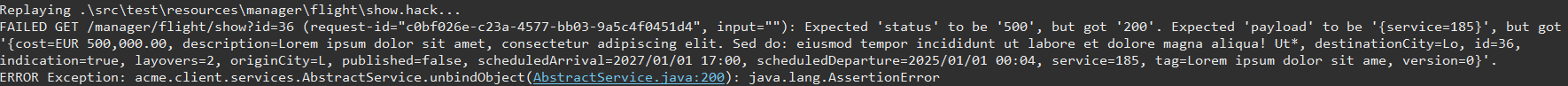
Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

Como se puede ver, al haber cambiado la lógica de asociar un mánager al vuelo, las operaciones de POST sobre la creación de un flight ya no devuelven con el estado que esperaba, pues detecta que el mánager es nulo cuando esperaba que fuera un mánager existente.

1. Mutación en flight show service:

La línea afectada es la 32, el cambio es: super.getResponse().setAuthorised(true);



Como se puede ver, al haber cambiado la lógica de autorización, las operaciones de GET sobre la visualización de un flight ya no devuelven un 500 en los intentos de hacking, sino un 200 porque siempre autoriza la respuesta pese a que esperaba desautorizar los intentos de hacking.

1. Mutación en leg show service

La línea afectada es la 45, el cambio es: super.getResponse().setAuthorised(true);

Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Como se puede ver, al haber cambiado la lógica de autorización, las operaciones de GET sobre la visualización de un leg ya no devuelven un 500 en los intentos de hacking, sino un 200 porque siempre autoriza la respuesta pese a que esperaba desautorizar los intentos de hacking.

1. Mutación en update flight service

La línea afectada es la 36 el cambio es: Boolean isFlightFromAuthenticatedManager = flight == null && flight.getManager().getId() == managerId;

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Como se puede ver, al haber cambiado la lógica de autorización, las operaciones de POST legales sobre la edición de un flight ya no devuelven un 200 ni el payload esperado, sino un 500 porque no puede autorizar ninguna petición.

# **Cobertura de las operaciones**

En este apartado se muestra el porcentaje de cobertura obtenido con las pruebas formales, confirmando que supera el umbral mínimo del 90 % antes de iniciar las pruebas de rendimiento. De este modo, queda documentado que todas las operaciones críticas han sido validadas y cumple el requisito de cobertura establecido.

1. Cobertura de Flight

Pantalla de computadora con letras

El contenido generado por IA puede ser incorrecto.

1. Cobertura de Leg

Pantalla de computadora con letras

El contenido generado por IA puede ser incorrecto. En las capturas se ve cómo el replay de pruebas cubre prácticamente todo el módulo “estudiante 1”, reduciendo al mínimo la posibilidad de errores ocultos. Además, las pruebas de mutación introducen cambios deliberados (mutantes) y exigen que la suite original falle al detectar esas alteraciones; el hecho de que los tests fallen ante cada mutante demuestra su solidez y valida que la cobertura alcanzada es real y efectiva.

# **Testeo de rendimiento**

En este capítulo, se evaluará el rendimiento de nuestro proyecto mediante pruebas detalladas que analizan el tiempo de respuesta al atender las solicitudes en nuestras pruebas funcionales. Utilizaremos dos archivos de testeo, uno como replay de las pruebas hechas un portátil y el segundo como replay de las pruebas en un pc con unas características algo mejores que el primero.

Para realizar el análisis de estas pruebas, presentaremos gráficos que ilustran los resultados obtenidos. Además, calcularemos intervalos de confianza del 95% para los tiempos de respuesta medidos y realizaremos un contraste de hipótesis mediante un test z para pruebas de dos medias, también con un intervalo de confianza del 95%, para determinar la diferencia de hacer replay sobre pruebas en un ordenador con respecto a otro.

**Gráficas de pruebas en primer ordenador:**

**Gráficas de pruebas en segundo ordenador:**

**Comparación de resultados:**

|  |  |  |
| --- | --- | --- |
| Prueba z para medias de dos muestras | | |
|  |  |  |
|  | *Before* | *After* |
| Media | 6.39667044 | 6.28795056 |
| Varianza (conocida) | 42.0941666 | 42.4282378 |
| Observaciones | 900 | 900 |
| Diferencia hipotética de las medias | 0 |  |
| z | 0.35476791 |  |
| P(Z<=z) una cola | 0.36138173 |  |
| Valor crítico de z (una cola) | 1.64485363 |  |
| Valor crítico de z (dos colas) | 0.72276347 |  |
| Valor crítico de z (dos colas) | 1.95996398 |  |

En nuestras pruebas de rendimiento, hemos obtenido un valor crítico de z de **0.** **72276347**para un nivel de significancia (α) de **0.95**. Este valor se utiliza para determinar si las diferencias en los tiempos de respuesta entre las dos pruebas son estadísticamente significativas.

De acuerdo con nuestra metodología, si el p-valor se encuentra en el intervalo (α, 1.00], esto indica que los cambios no resultaron en mejoras relevantes; es decir, aunque los tiempos de muestra sean diferentes, globalmente son equivalentes. Dado que nuestro valor crítico de z está en este intervalo, concluimos que no hay una mejora significativa en el rendimiento al comparar los tiempos de respuesta de las pruebas. En resumen, las diferencias observadas no son lo suficientemente grandes como para considerarse estadísticamente relevantes con el nivel de confianza del 95%. Por lo que tampoco tiene sentido comparar las medias de tiempo de ejecución de las baterías de pruebas ejecutadas en un ordenador con respecto al otro.

# **Conclusiones**

En este proceso de testing hemos aprendido que definir con precisión las pruebas para cada funcionalidad solicitada por el cliente es fundamental para alinear resultados esperados y reales. Un entorno de pruebas aislado, por su parte, facilita la reproducción de errores y acelera su resolución en caso de haber encontrado bugs haciendo el testing. Asimismo, el análisis de métricas de rendimiento nos permite evaluar requisitos no funcionales, detectar ineficiencias en el código y determinar si su origen es de hardware o software para proceder, en caso de ser razón de software, con la mejora en la eficiencia del código.

# **Bibliografía**

Intencionalmente en blanco.