

**PRÁCTICA INTEGRACIÓN**

IWM41

**─**

Grupo 8:

Omar Piñeiro Parada

Violeta Macías de Miguel

Javier Gómez de Lucio

Javier Herrero Sánchez

Alejandro Del río Álvarez

# Índice

[**Índice**](#_3vfxlc84w664) **2**

[**Test addRemoteSystem(String userId, Object remote)**](#_f6cka3k6ckqt) **3**

[Resultados](#_bp8nhmyao2tf) 4

[**TestStartRemoteSystem()**](#_b4ph6xm85bn6) **12**

[Resultados:](#_wm1o7atjrro7) 14

[**TestStopRemoteSystem()**](#_dubvlvkbwk8n) **14**

[Resultados:](#_bmbhln7pphtz) 16

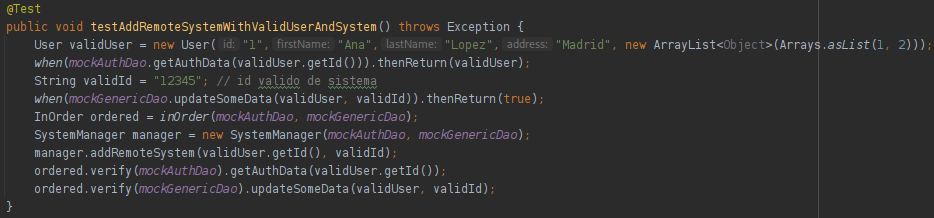
# 

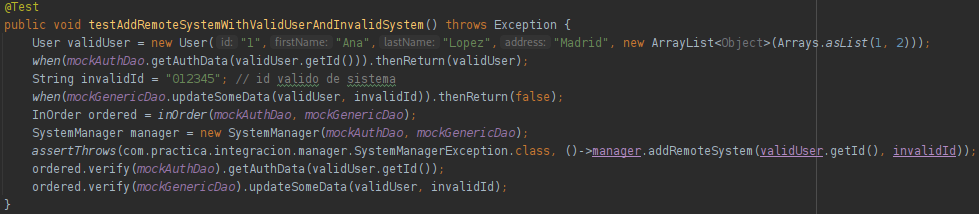
# Test addRemoteSystem(String userId, Object remote)

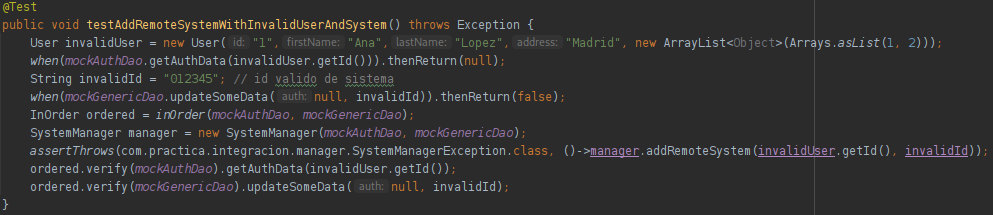
En addRemoteSystem se llama a los métodos getAuthData y updateSomeData.

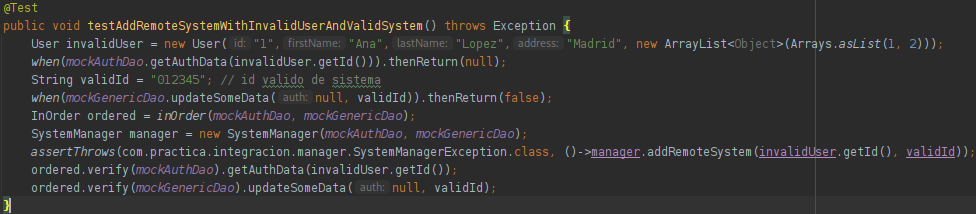
Se tiene que probar lo siguiente para los casos de usuario y sistema válidos, usuario y sistema inválidos, usuario válido y sistema inválido, usuario inválido y sistema válido:

* Llamada al DAO de autenticación y al DAO acceso a datos del sistema así como sus validaciones.
* Instanciación del SystemManager con mocks creados.
* Llamada a addRemoteSystem y comprobación llamadas y orden de getAuthData y updateSomeData.

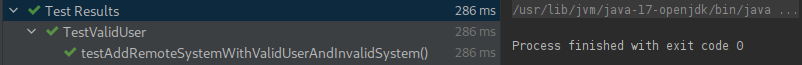
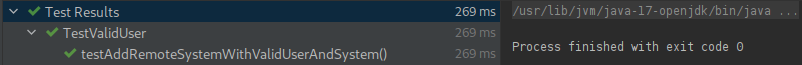
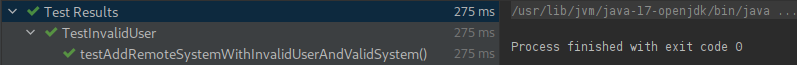








## Resultados



Todos los tests se pasan satisfactoriamente

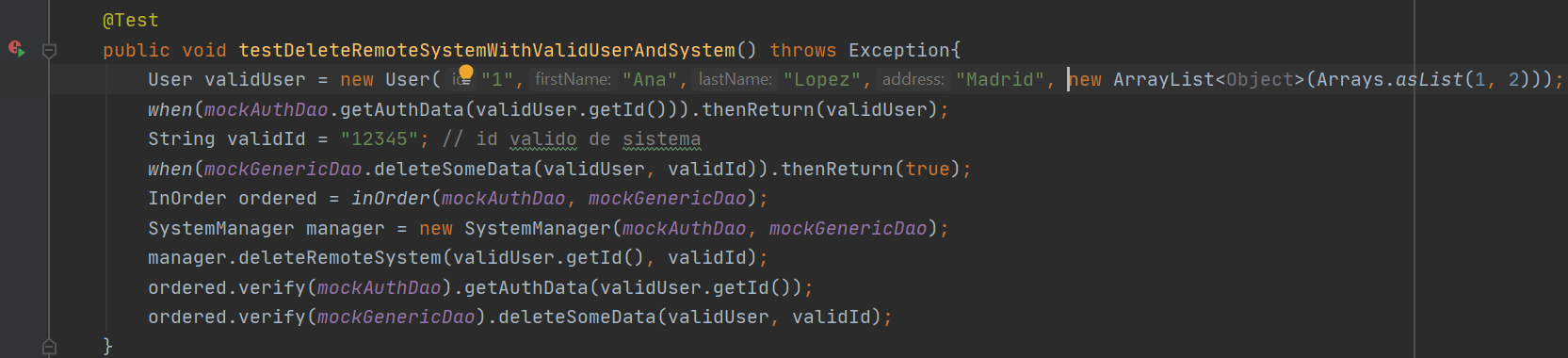
Test deleteRemoteSystem(String userId, String remoteId)

En deleteRemoteSystem se llama a los métodos getAuthData y deleteSomeData.

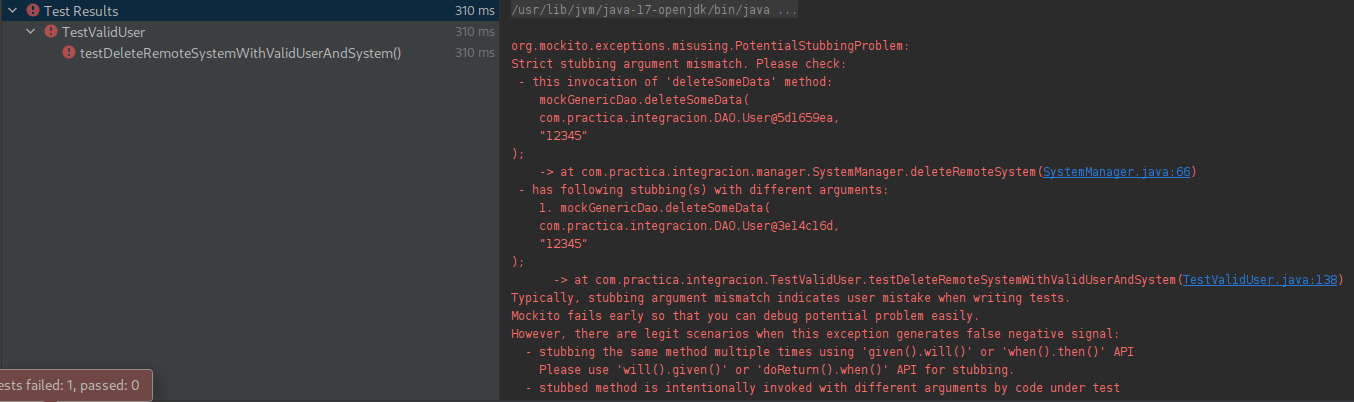
Se tiene que probar lo siguiente para los casos de usuario y sistema válidos, usuario y sistema inválidos, usuario válido y sistema inválido, usuario inválido y sistema válido:

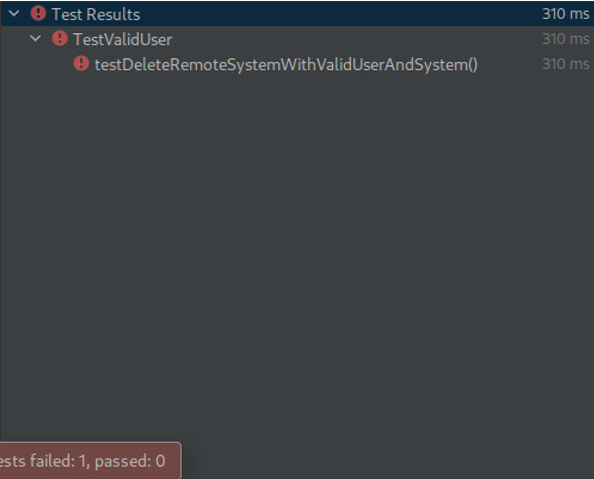
* Llamada al DAO de autenticación y al DAO acceso a datos del sistema así como sus validaciones a través de la creación de mocks de la clase AuthDao y de GenericDao.
* Instanciación del SystemManager con los mocks creados.
* Llamada a deleteRemoteSystem y comprobación llamadas y orden de getAuthData y deleteSomeData.

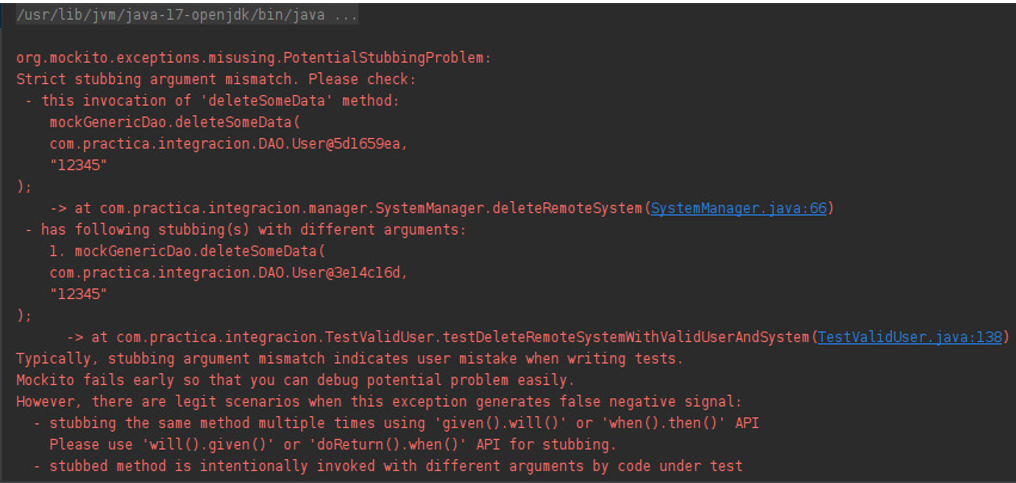
Usuario y Sistema válidos:



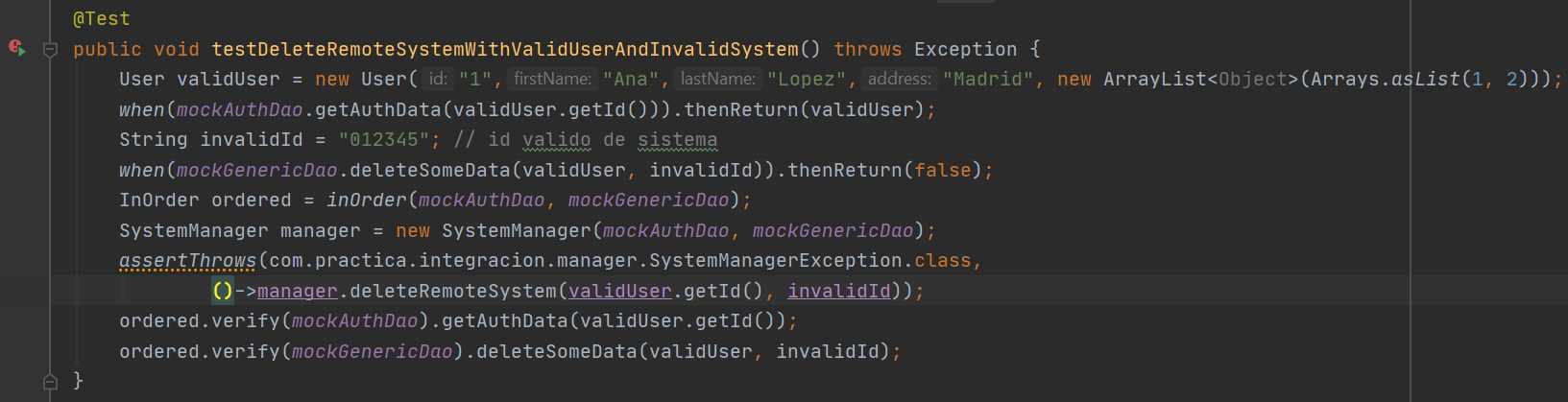
Resultado:



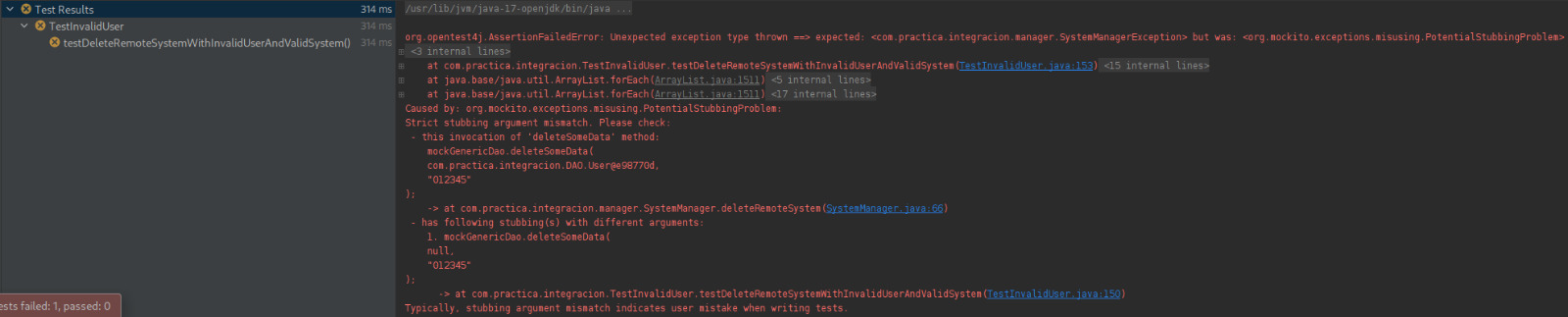


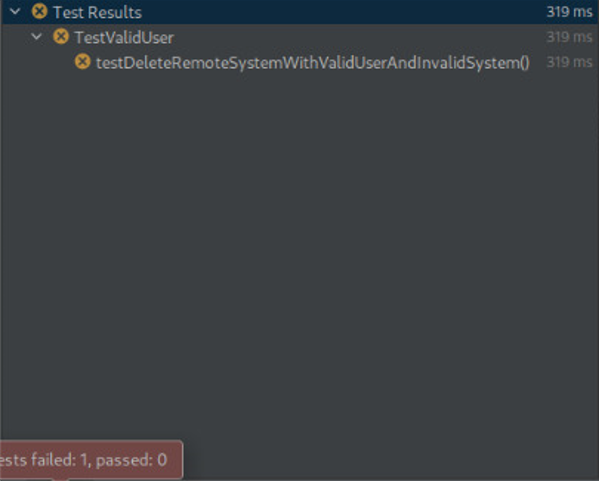


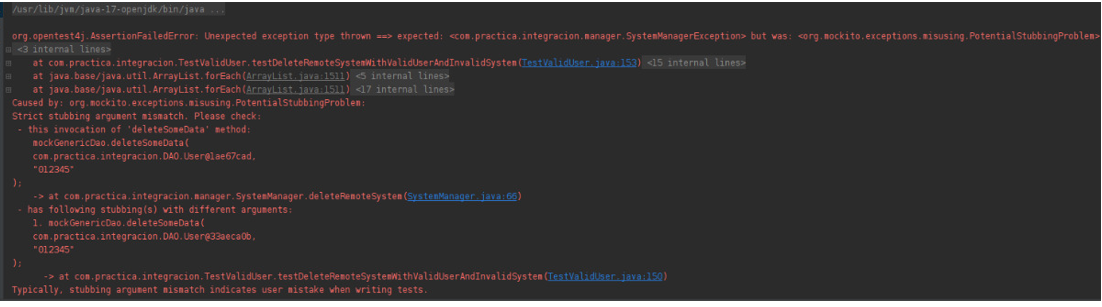
Usuario Válido y Sistema Inválido:



Resultado:

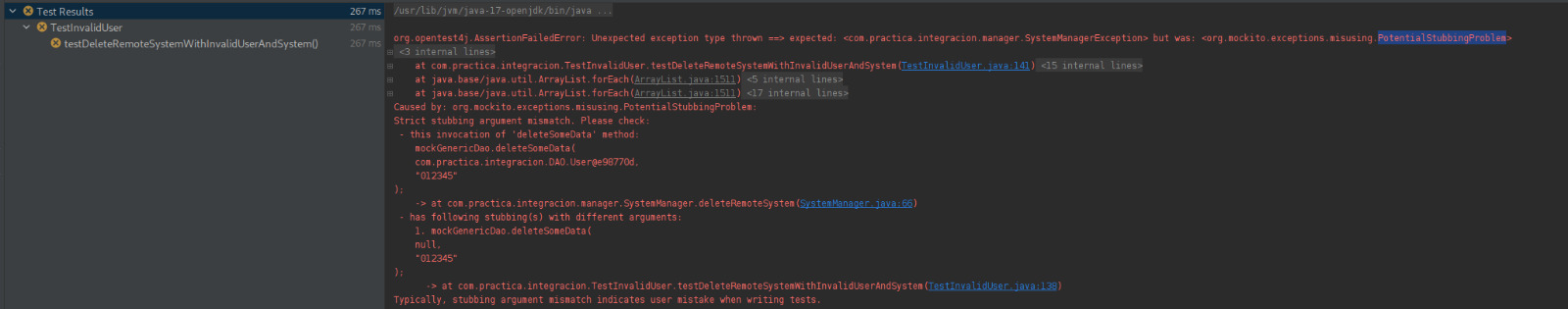


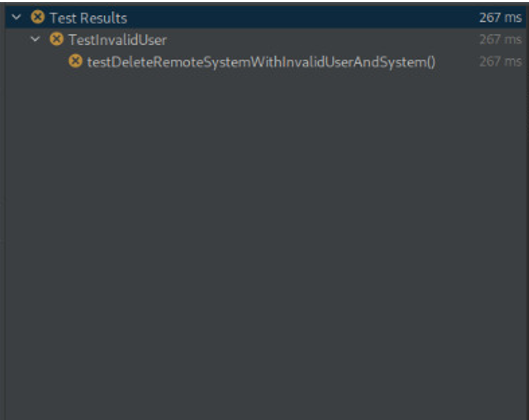


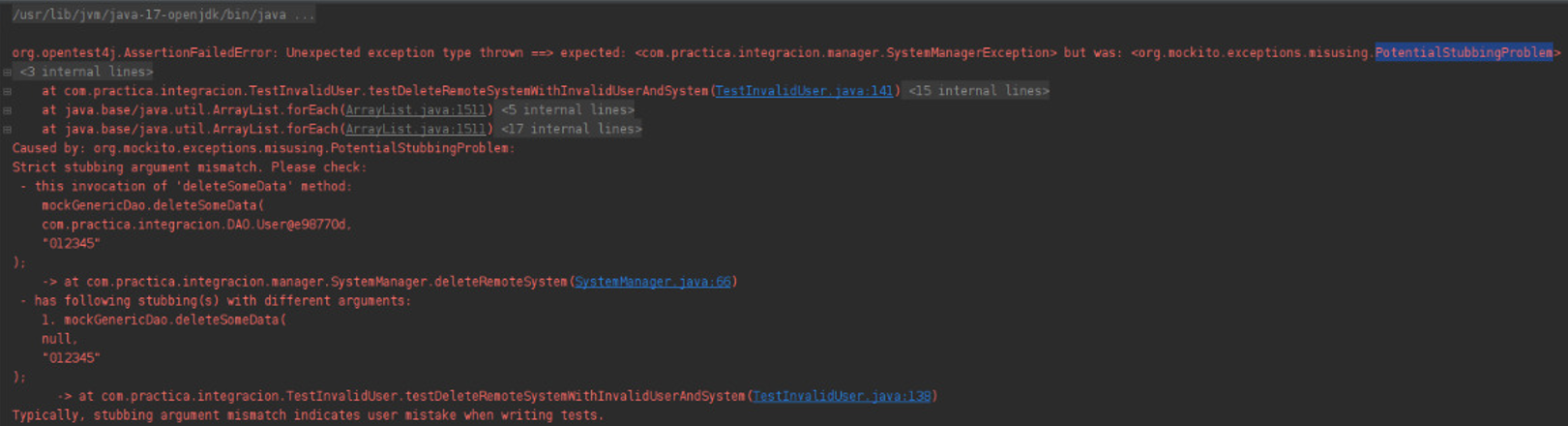


Usuario y Sistema inválidos:

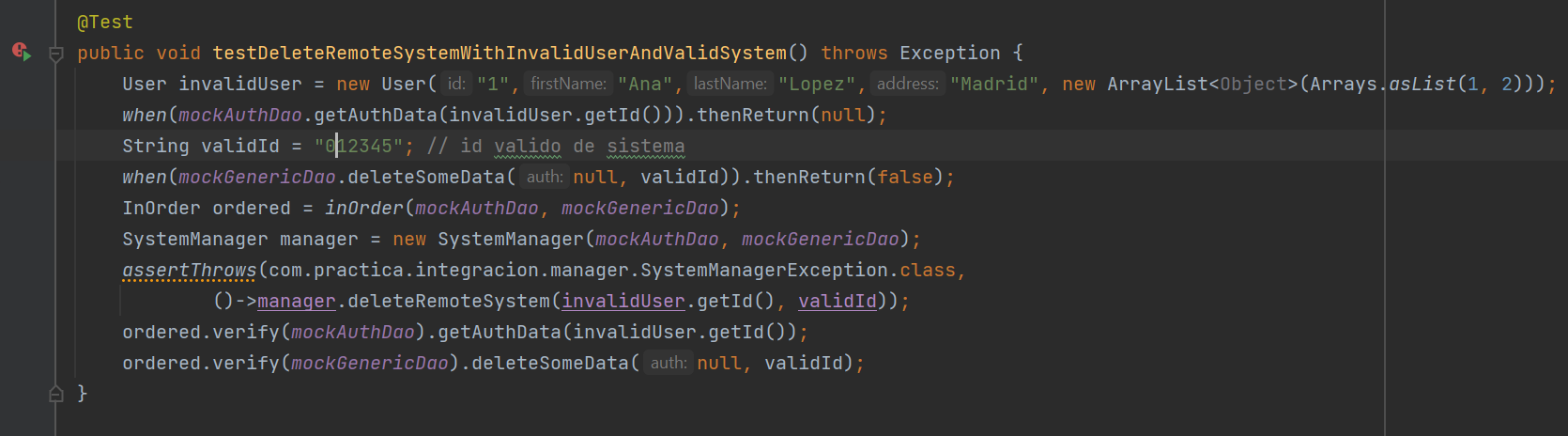


Resultado:

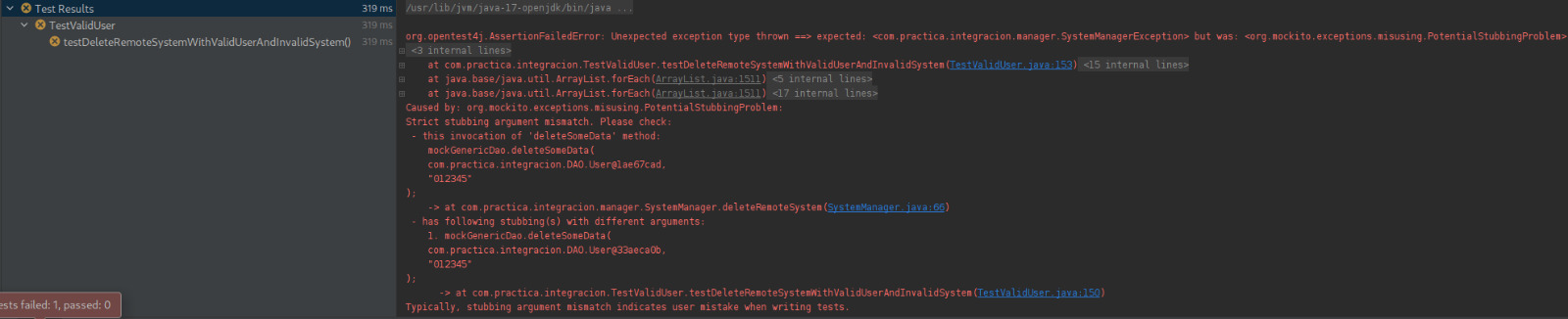


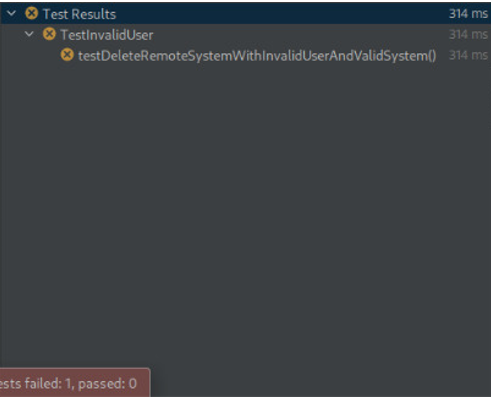


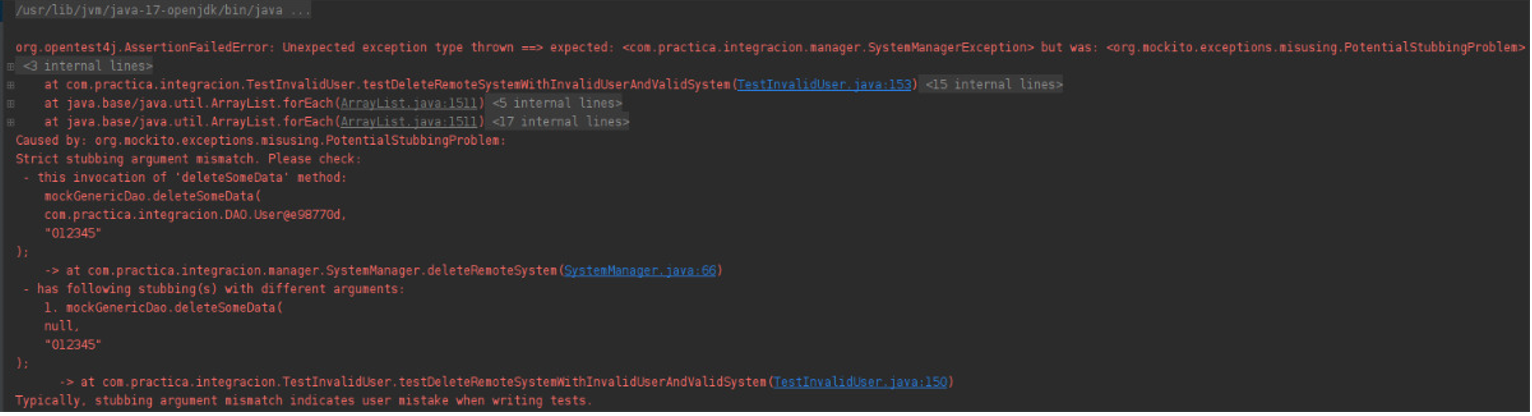
Usuario inálido y Sistema válido:



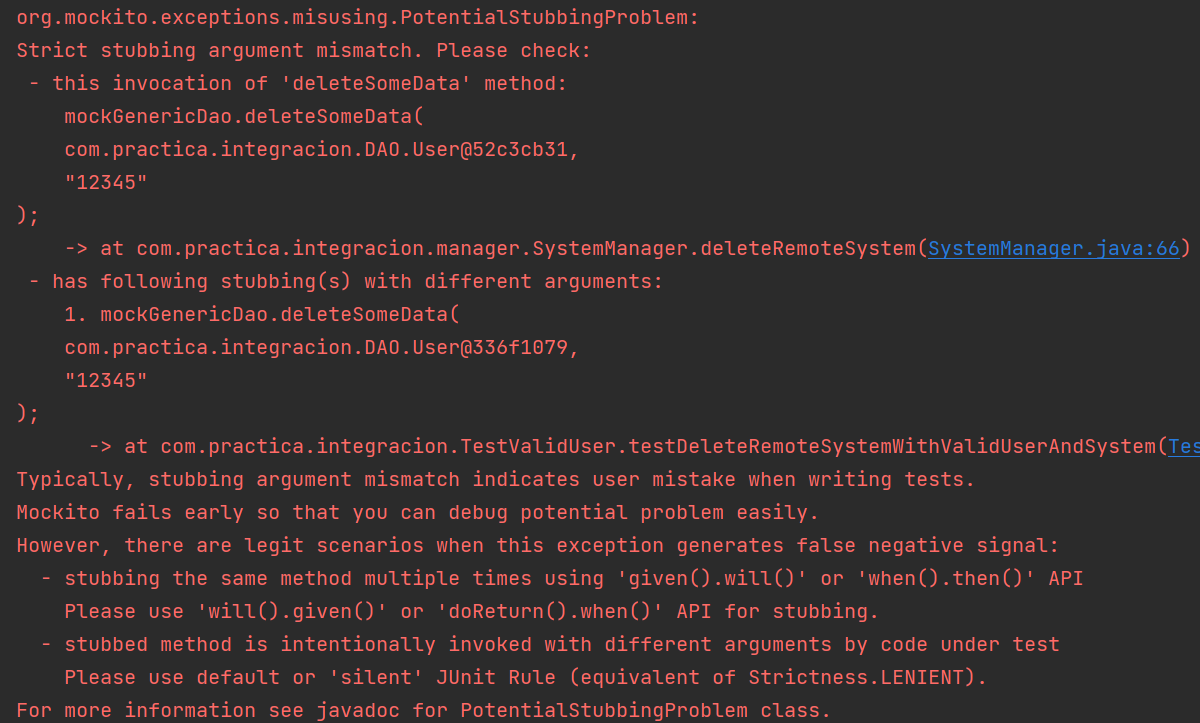
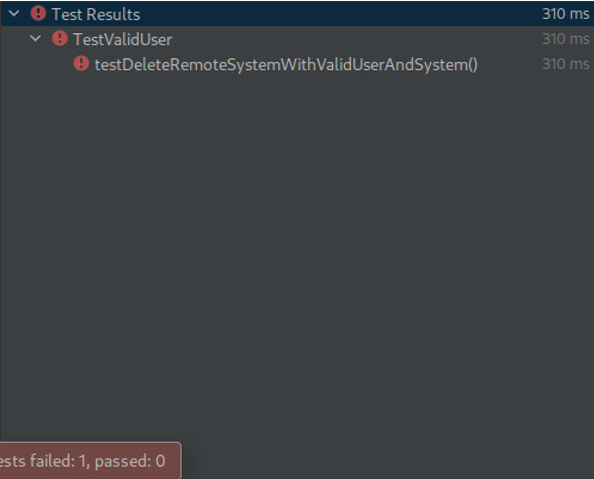
Resultado:

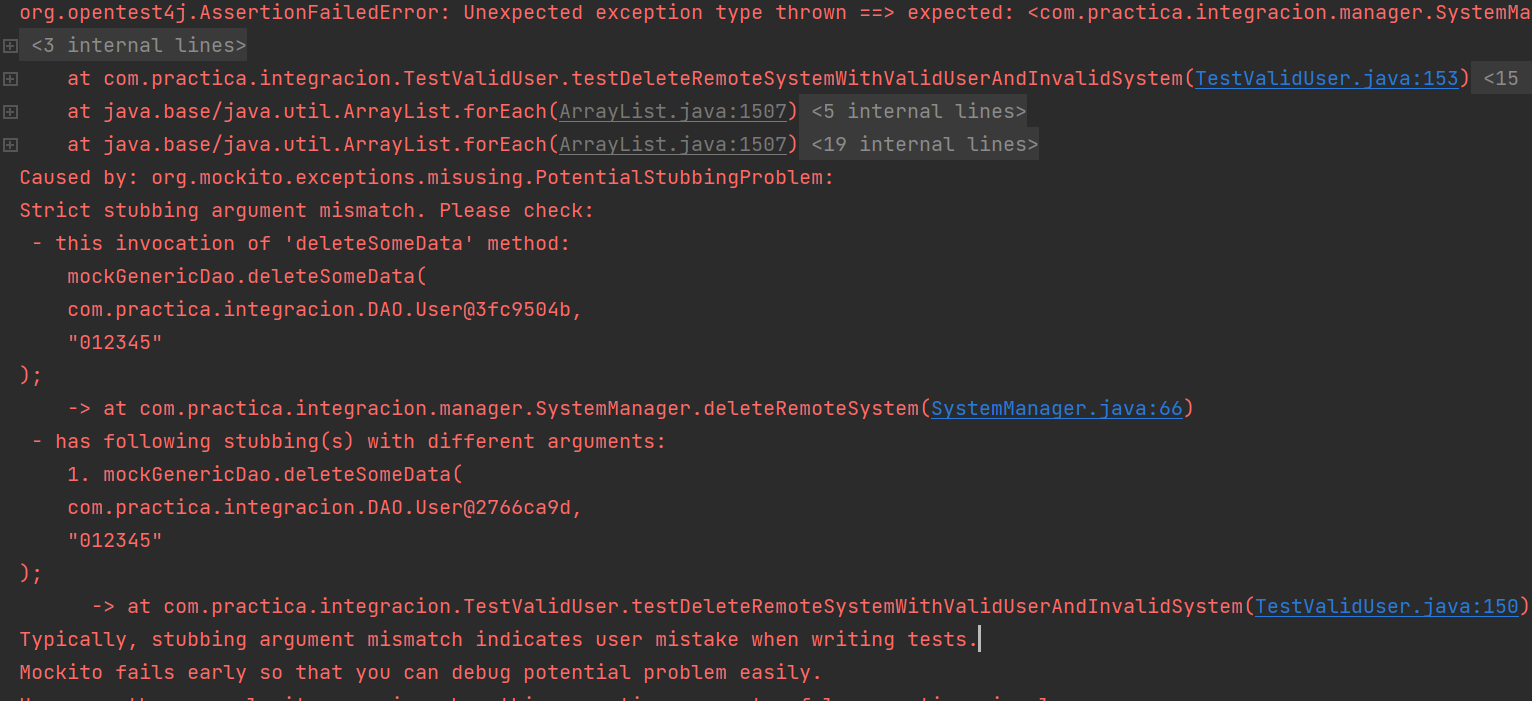
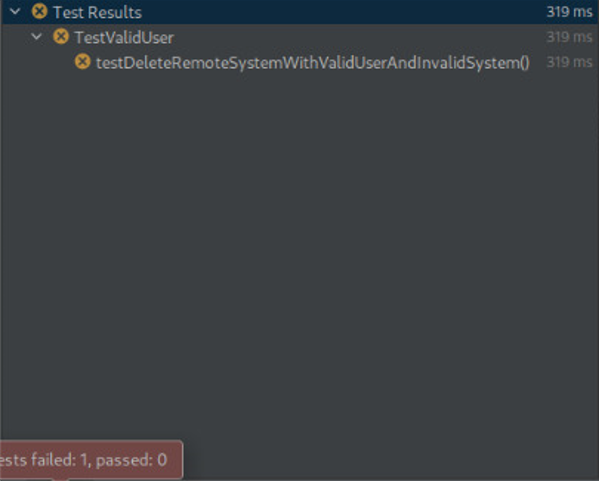


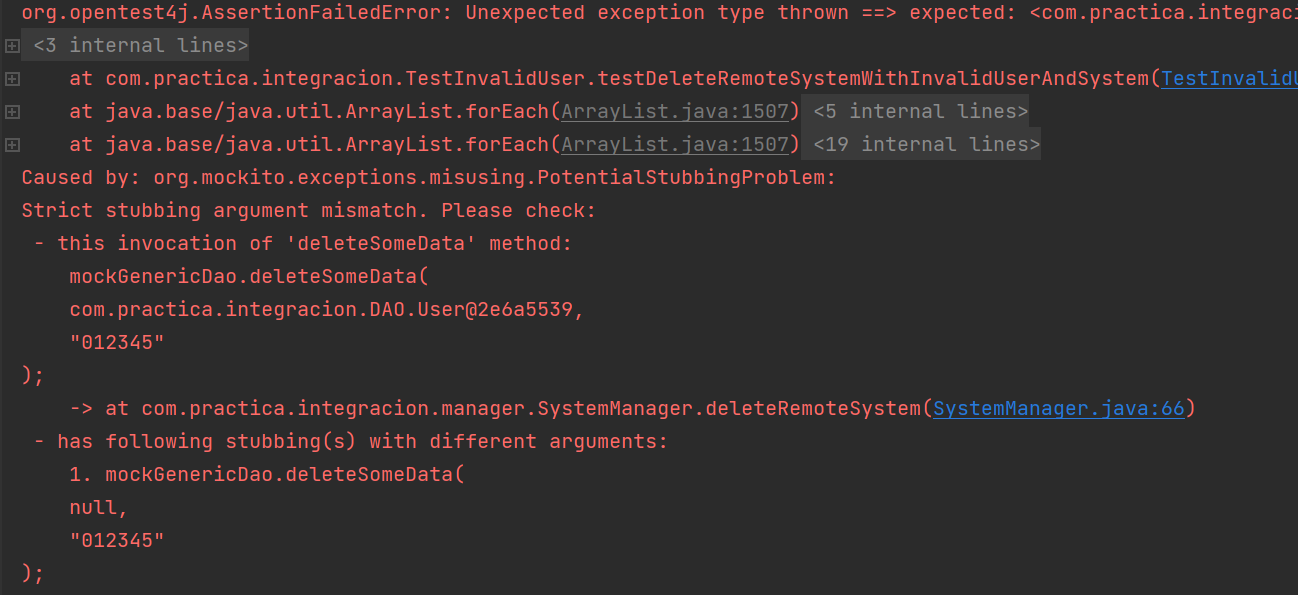
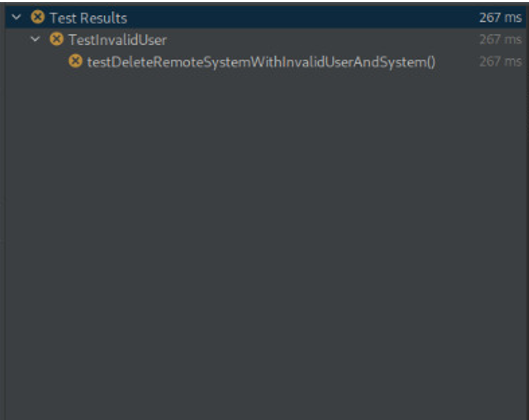


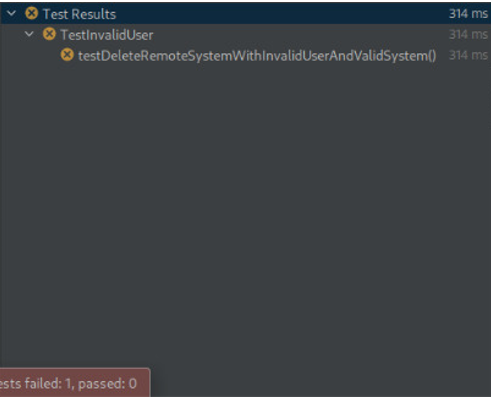


Se han producido errores de argumentos en todos los tests:





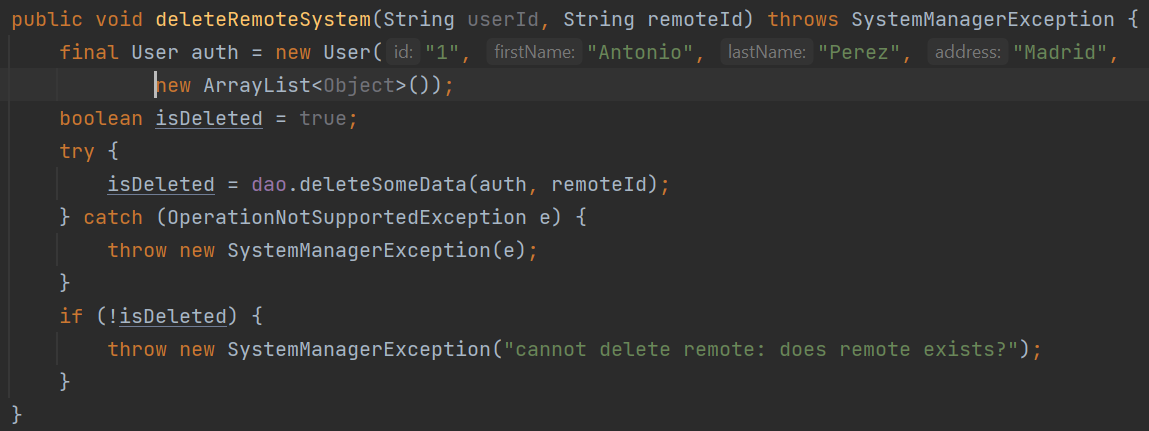




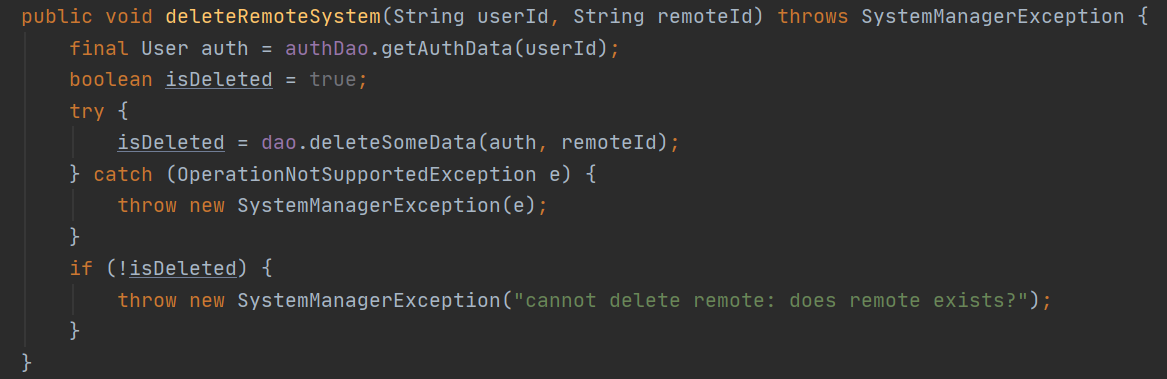


En el método deleteRemoteSystem, a la llamada deleteSomeData se le está pasando un objeto User (auth) creado en el propio método (deleteRemoteSystem) en vez de el objeto User que se ha utilizado en la creación del mock.

Podemos concluir que el método deleteRemoteSystem está mal implementado, ya que no sigue el comportamiento esperado.



Para evitar esto habría que cambiar la creación de un nuevo usuario para el objeto auth de tipo usuario por uno con el id del objeto usuario creado en el mock, quedaría de la siguiente manera:



De esta forma los test se pasarían satisfactoriamente.

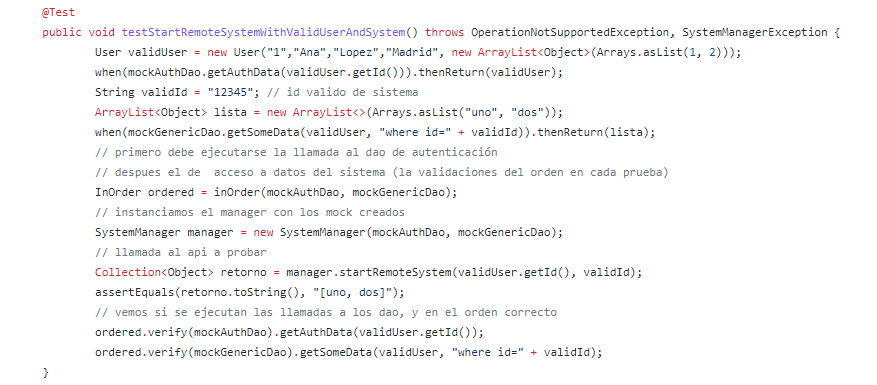
# TestStartRemoteSystem()

Se probarán las 4 combinaciones de usuario valido/no valido con sistema valido/no valido.

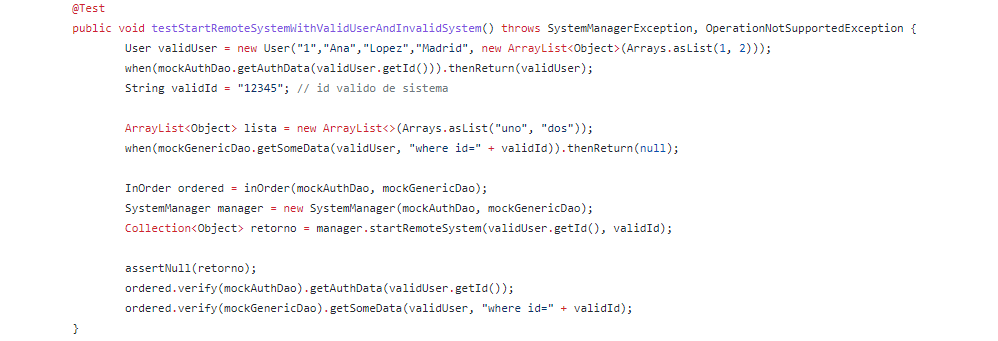
Para lo cual se crearán mocks de la clase AuthDao y de GenericDao.

Para las pruebas se comprobará lo que retorna el método así como que los metodos getAuthData y el metodo getSomeData se hayan ejecutado en dicho orden.

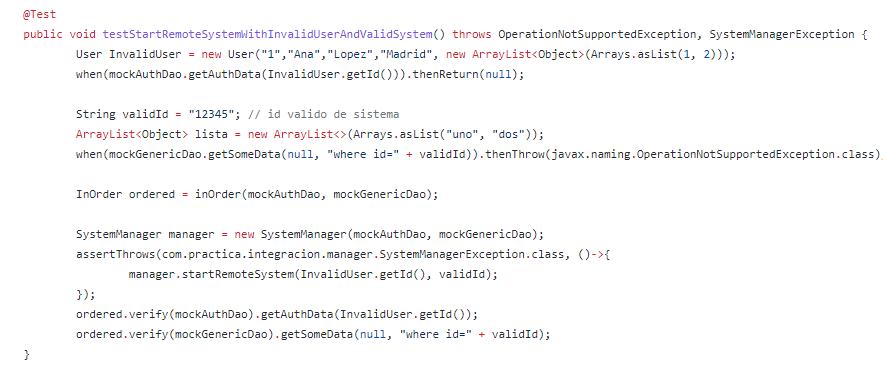
Usuario y Sistema Válidos



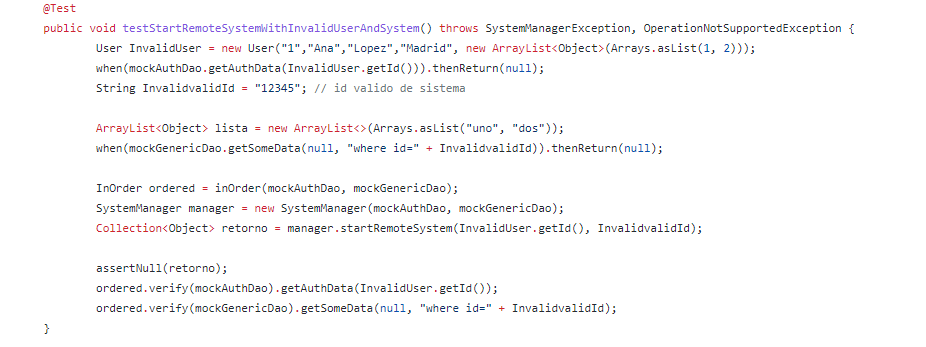
Usuario Válido y Sistema Inválido



Usuario Inválido y Sistema Válido

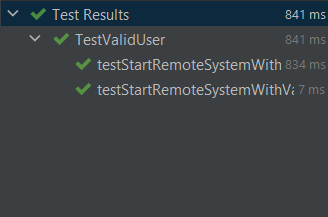


Usuario y Sistema Invalido

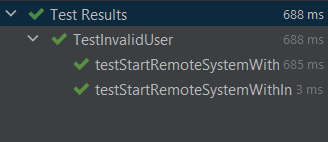


## Resultados:

Valid User.



Invalid User.



Todos los test se pasan con éxito.

# TestStopRemoteSystem()

Se probarán las 4 combinaciones de usuario valido/no valido con sistema valido/no valido.

Para lo cual se crearán mocks de la clase AuthDao y de GenericDao.

Para las pruebas se comprobará lo que retorna el método así como que los metodos getAuthData y el metodo getSomeData se hayan ejecutado en dicho orden.

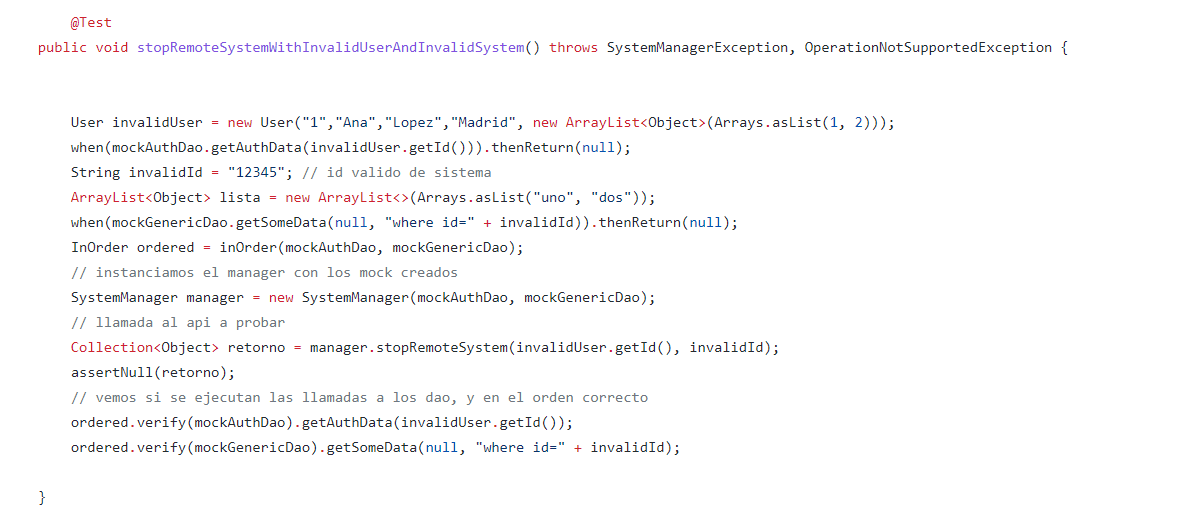
Usuario y Sistema Válidos

Usuario Válido y Sistema Inválido

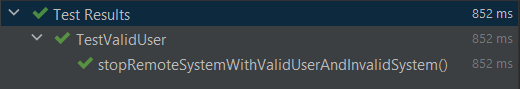
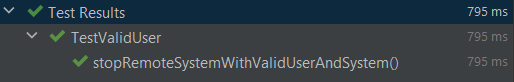
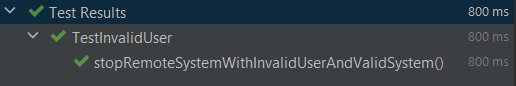


Usuario Inválido y Sistema Válido

Usuario y Sistema Inválidos



## Resultados:



Todos los test son pasados con éxito