



Recibe una cálida:

¡Bienvenida!

Te estábamos esperando 😊 

➤ Pruebas Unitarias en Java

Plan formativo: Desarrollo de Aplicaciones Full Stack Java Trainee V2.0

HOJA DE RUTA

¿Cuáles **skill** conforman el programa?



REPASO CLASE ANTERIOR

En la clase anterior trabajamos :

- ✓ Implementación de *Fixtures* mediante anotaciones en *JUnit5*

LEARNING PATHWAY

4.7

Start! 🏁

Pruebas Unitarias en Java

Utilización de objetos
simulados (mocks)

Simuladores

Utilización de objetos
simulados (mocks)

OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?

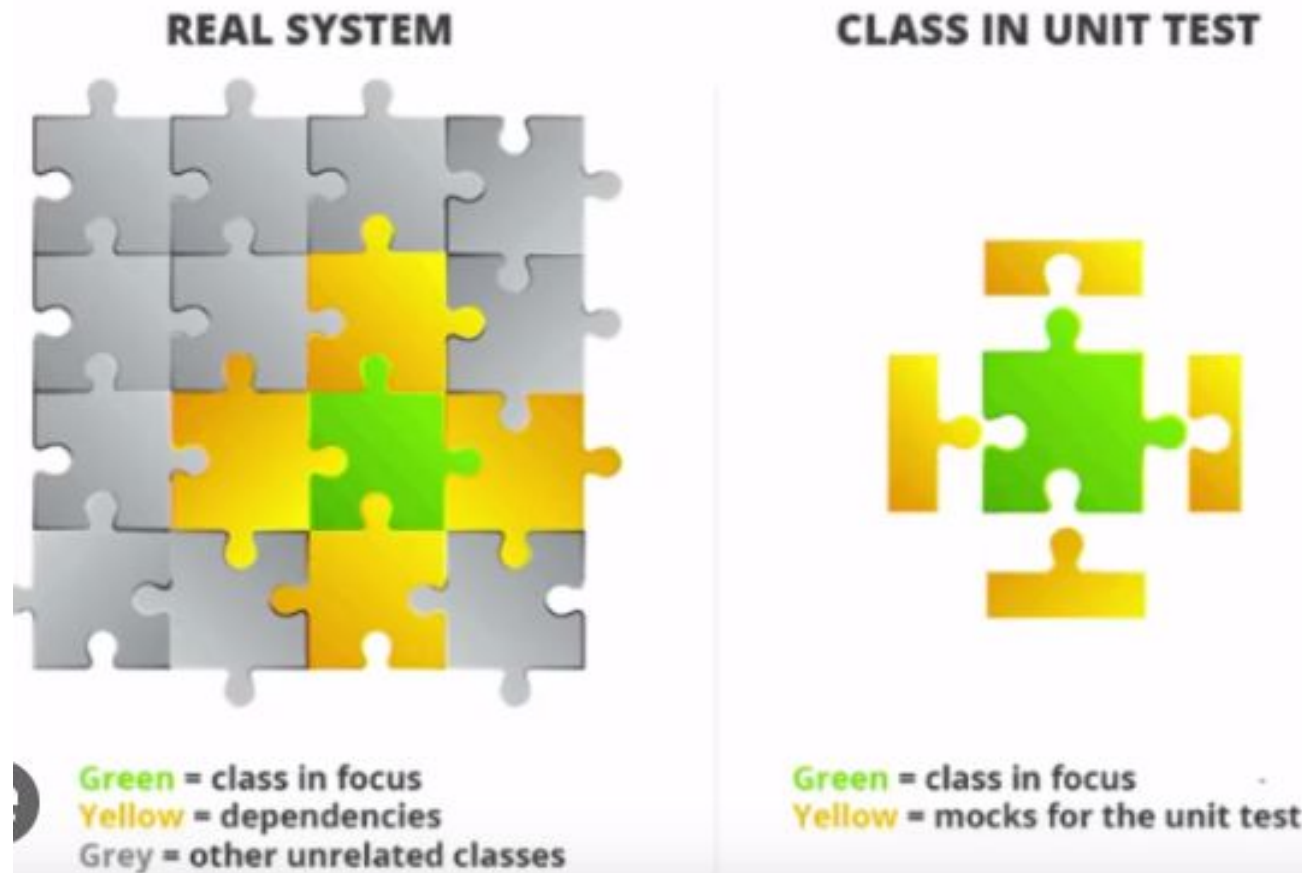


**Comprender el concepto y utilización de
objetos simulados en programación
orientada a objetos**





Rompehielo 🧊



Respondan levantando la mano: 🙋

Imaginen que deben probar un método que interactúa con un servicio externo como una API o base de datos. **¿Cómo podrían probarlo sin ejecutar el servicio real?**

› Utilización de objetos simulados (mocks)



Utilización de objetos simulados (mocks)



¿Qué es el Mocking?:

Es una técnica utilizada en las pruebas unitarias para **simular el comportamiento de objetos reales** cuando la unidad que se está probando tiene dependencias externas. **Las simulaciones, o mocks, se utilizan en lugar de los objetos reales.**



El **objetivo de los mocks es aislar y concentrarse en el código que se está probando** y no en el comportamiento o estado de las dependencias externas.

Dentro de Spring Boot existe **Mockito** que nos ayuda a simular pruebas teniendo en cuenta, por ejemplo, qué tipos de datos enviamos y qué tipos de datos esperamos como respuesta.



En síntesis, es un tipo concreto de doble de test.



Dobles de prueba

Estos dobles son utilizados para hacer que las pruebas sean más fáciles de escribir, más rápidas o se posea un mayor control al simular elementos que no son el propósito principal de la prueba específica o que son costosos, en tiempo o dinero, de duplicar como un servicio externo o una base de datos.



- **Dummy:** se considera como argumento pero nunca se usa realmente. Normalmente, los objetos dummy se usan sólo para rellenar listas de parámetros.
- **Fake:** tiene una implementación que realmente funciona pero, por lo general, toma algún atajo o cortocircuito que le hace inapropiado para producción (como una base de datos en memoria, por ejemplo).





Dobles de prueba

Stubs

Son la forma más simple de los tres dobles de prueba. **Se parecen al código real que están reemplazando, pero siempre devuelven la misma respuesta estática,** independientemente de la entrada. Generalmente, es la mejor opción de doble cuando no le importa cuál es el comportamiento de la entidad duplicada. Solo desea que se ejecute sin causar una excepción.

Como los dobles del Museo de Cera de Madame Tussauds, los stubs **sirven a un único y sencillo propósito.**





Dobles de prueba

Spies

Es casi lo opuesto a los stubs. Permiten que la entidad duplicada conserve su comportamiento original al tiempo que proporciona información sobre cómo interactuó con el código bajo prueba. El spy puede decirle a la prueba qué parámetros se le dieron, cuántas veces se llamó y cuál fue el valor de retorno, si lo hubo.

Los spies también se pueden usar para simular o hacer doblaje de una pequeña parte de la entidad duplicada mientras se deja el resto del comportamiento de esa entidad en su lugar.

En el cine se utiliza la animatrónica para simular los movimientos de un actor en determinadas escenas, que podrían ser complejas o peligrosas.






Dobles de prueba

Mocks

Son los dobles de prueba más poderosos. Permiten un control completo sobre la entidad duplicada y proporcionan la misma información que proporciona un spy sobre cómo se ha interactuado con la entidad. Los mocks **se configuran antes de que se ejecute el código** bajo prueba para que se comporte como nos gustaría.

Hay todo tipo de situaciones para las que puedes configurar tus mocks. Algunas de estas situaciones incluyen:

- Devolver un valor esperado si se dan los parámetros correctos.
-  Devolver un valor en la primera llamada y un valor diferente en la segunda llamada.
- Lanza una excepción.



Los Mocks son los verdaderos “dobles de acción”.

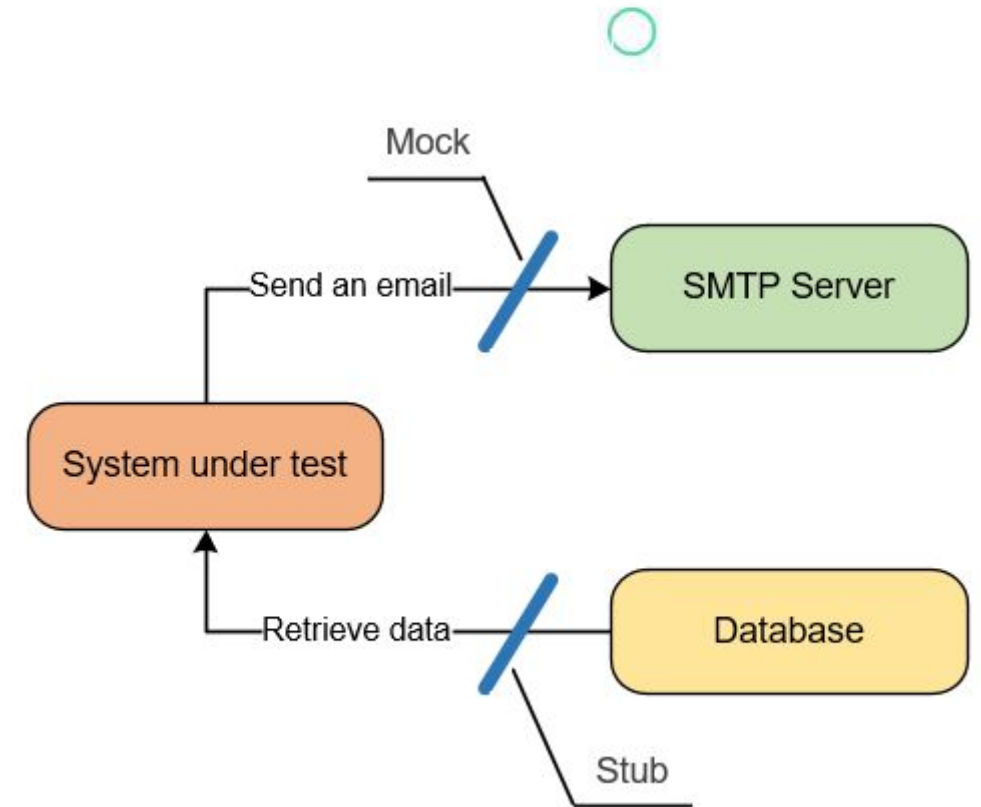


Utilización de objetos simulados (mocks)

Diferencia entre Stub y Mock

Aunque parezcan similares, podría decirse que el stub es como un mock con menor potencia, un subconjunto de su funcionalidad. Mientras que en el mock podemos definir expectativas con todo lujo de detalles, el stub tan solo devuelve respuestas preprogramadas a posibles llamadas.

Un **mock** valida comportamiento en la colaboración, mientras que el **stub** simplemente simula respuestas a consultas.



Evaluación Integradora ✨

¿Listos para un nuevo desafío? En esta clase comenzamos a construir nuestro...

Trabajo Integrador del Módulo 💪

Iremos completándolo progresivamente clase a clase.



LIVE DEMO

Ejemplo en vivo

Creando mocks:

En este caso vamos a evaluar algunos métodos creados en clases anteriores.

1. *Definir qué tipo de objetos simulados deberíamos crear (o elegimos crear) según los métodos/clases a testear*

 **Tiempo: 40 minutos**





Ejercicio N° 1

Simuladores



Simuladores



Consigna: 🖋️ **Vamos a evaluar qué objetos simulados utilizar**

Ahora vamos a definir que tipo de objetos simulados utilizarías para probar las clases Test generadas en el ejercicio anterior (de las clases de Polimorfismo)

En grupos reducidos, seleccionar el ejercicio de algún compañero y evaluar que tipo de pruebas realizarían.

Luego compartir las decisiones tomadas en la sala principal con el resto de la clase y analizar entre todos cuáles fueron las mejores alternativas.



Tiempo 🕒: 30 minutos

○

¿Alguna consulta?

+



RESUMEN

¿Qué logramos en esta clase?

- ✓ **Comprender el concepto e implementación de los objetos simulados (mocks)**



#WorkingTime

Continuemos ejercitando

¡Antes de cerrar la clase! Te invitamos a: 📌 📌 📌

1. Repasar nuevamente la grabación de esta clase
2. Revisar el material compartido en la plataforma de Moodle (lo que se vio en clase y algún ejercicio adicional)
 - a. *Lectura Módulo 4, Lección 7: páginas 13 - 14*
3. Traer al próximo encuentro, todas tus dudas y consultas para verlas antes de iniciar nuevo tema.

¡Muchas Gracias!

Nos vemos en la próxima clase 🙌



Momento: ✚

Time-out!

🕒 5 min.

