



# Pruebas unitarias y TDD

Sesión Experimental 02





## Activación de conceptos

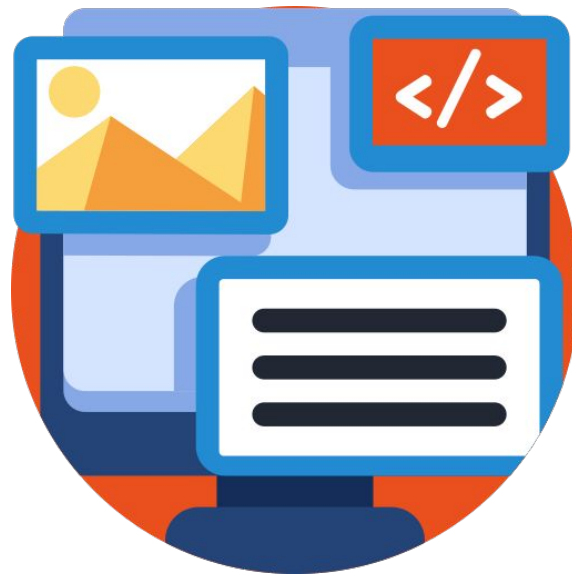


# Mockito

*Librería para trabajar con Mocks, muy popular en el mundo Java*

## Características

- Permite escribir pruebas expresivas.
- Ofrece una gama de servicios simples.
- Rodeado de una gran comunidad.



# TDD

*TDD es un procedimiento en donde se escriben las pruebas antes de la implementación real.*

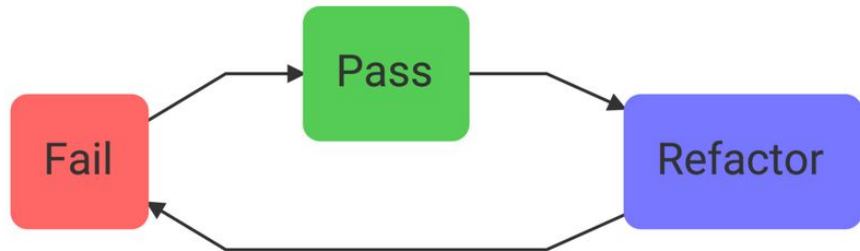
Reglas:

- Escribe sólo lo suficiente de una prueba unitaria para fallar.
- Escriba solo el código de producción suficiente para hacer que la prueba unitaria que falló pase.



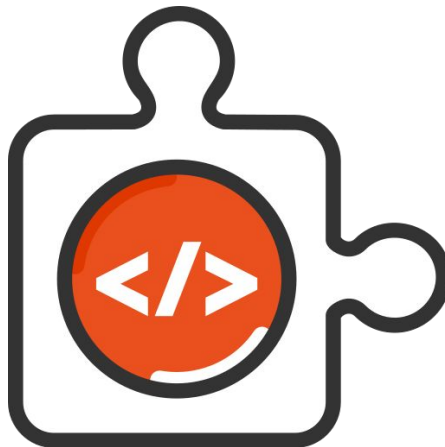
# Ciclo de TDD

*Se define como un ciclo red-green-refactor*



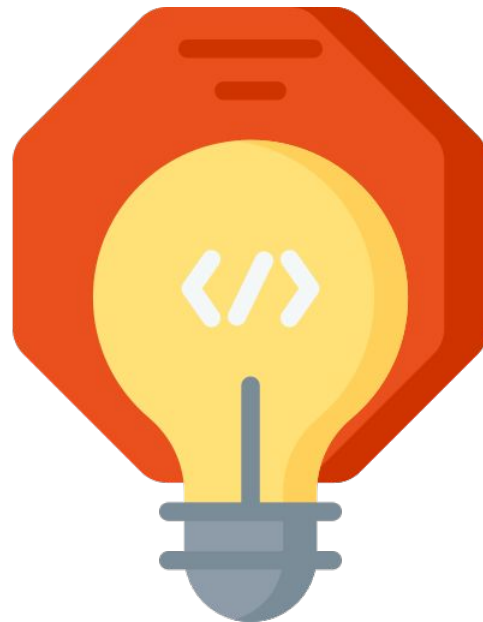
## Fase Roja

- Se debe escribir una prueba sobre un comportamiento que está a punto de implementar.
- No se debe pensar en cómo escribir el código de producción.



## Fase Verde

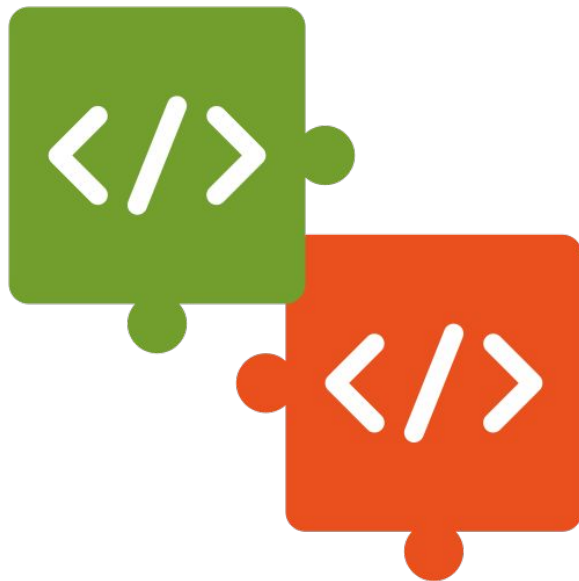
- Se escribe el código de producción.
- Escribir una solución sencilla que haga pasar la prueba.





## Fase de Refactorización

- Se permite cambiar el código manteniendo las pruebas en verde.
- Eliminar la duplicación de código.





## Desafío



140 minutos


{desafío}  
latam\_



## Panel de discusión



# Panel de discusión



¿Qué fue lo que más me costó?

¿Qué fue lo que menos me costó?

¿Qué se necesita reforzar?

¿Cómo podría hacerlo mejor la próxima vez?



Cierre



**¿Existe algún concepto que no  
hayas comprendido?**

**Volvamos a revisar los conceptos que más te  
hayan costado antes de seguir adelante**

**Reflexionemos**





*Academia de  
talentos digitales*

[www.desafiolatam.com](http://www.desafiolatam.com)



/DesafioLatam



/DesafioLatam



/DesafioLatam



/DesafioLatam