

# **CLASE 11**

# **TEST UNITARIOS**

Programación y Laboratorio II




**01.**

# **SOFTWARE TESTING**

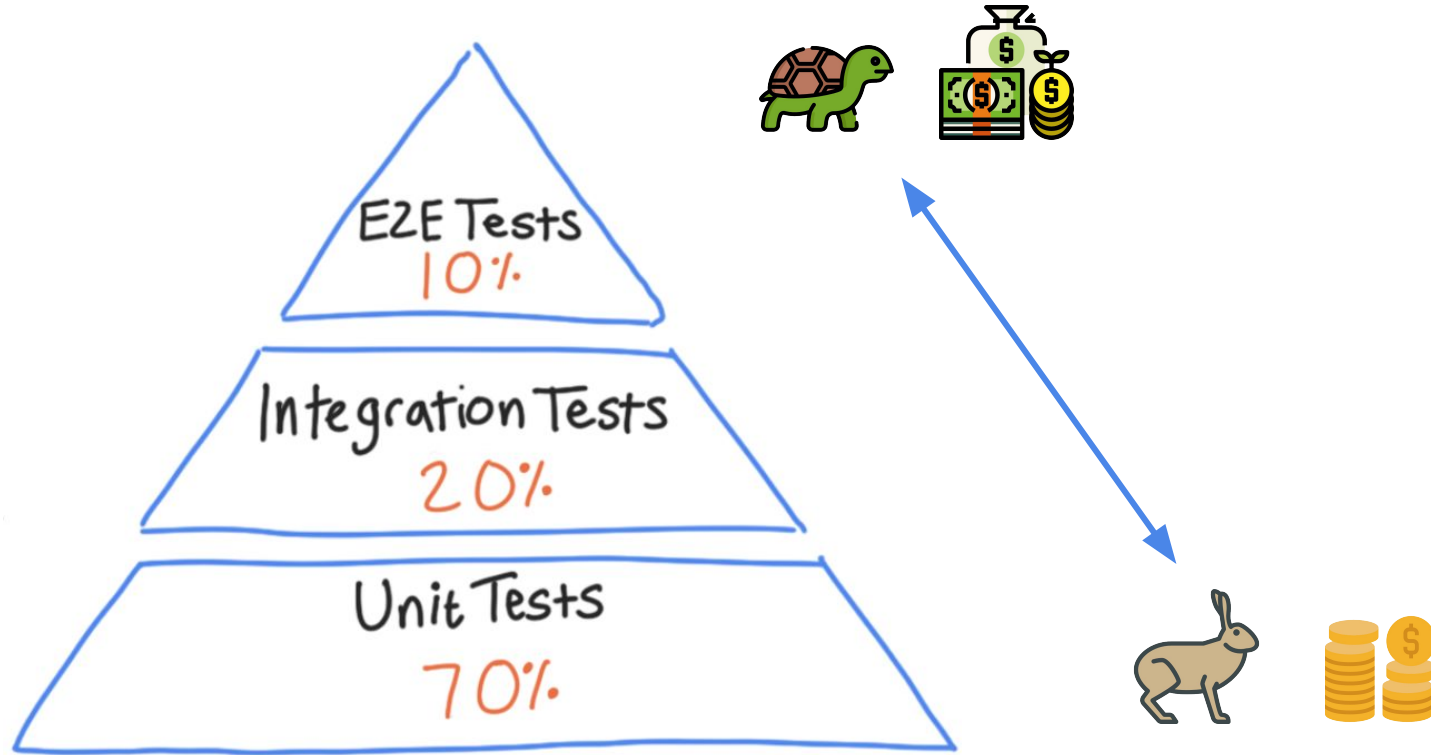


# SOFTWARE TESTING

Es una parte integral del ciclo de vida del software.  
Es el mecanismo, manual o automático, de verificar si el  
comportamiento del sistema es el deseado o no



# TIPOS DE TEST





# TIPOS DE TEST (TESTING PYRAMID)

## **FUNCIONALES**

Funcionalidad completa  
Verifican el flujo completo de la aplicación

## **INTEGRALES**

Verifican integración entre distintas partes y servicios

## **UNITARIOS**

Verifican funcionalidad específica, una pequeña parte






**02.**

**TEST UNITARIOS**



# ¿Qué son los TEST UNITARIOS?

Verifican la funcionalidad de una sección específica de código.



# NOMBRE DE UN TEST

**Método que se prueba**

Dividir

**Escenario**

CuandoDivisorEsCero

**Resultado**

DeberiaRetornarDoubleMinValue

Dividir\_CuandoDivisorEsCero\_DeberiaRetornarDoubleMinValue





# ESTRUCTURA DE UN TEST

## PATRÓN DE AAA

**ARRANGE**

Preparación

**ACT**

Invocación de los métodos a probar

**ASSERT**

Evaluación de los resultados

# F.I.R.S.T de Robert Martin

## **FAST**

Ejecutarse de forma rápida

## **ISOLATES**

No deben depender unas de otras

## **REPEATABLE**

Deben poder repetirse en cualquier entorno

## **SELF-VALIDATING**

No requieren validación manual

## **TIMELY**

Se escriben antes del código

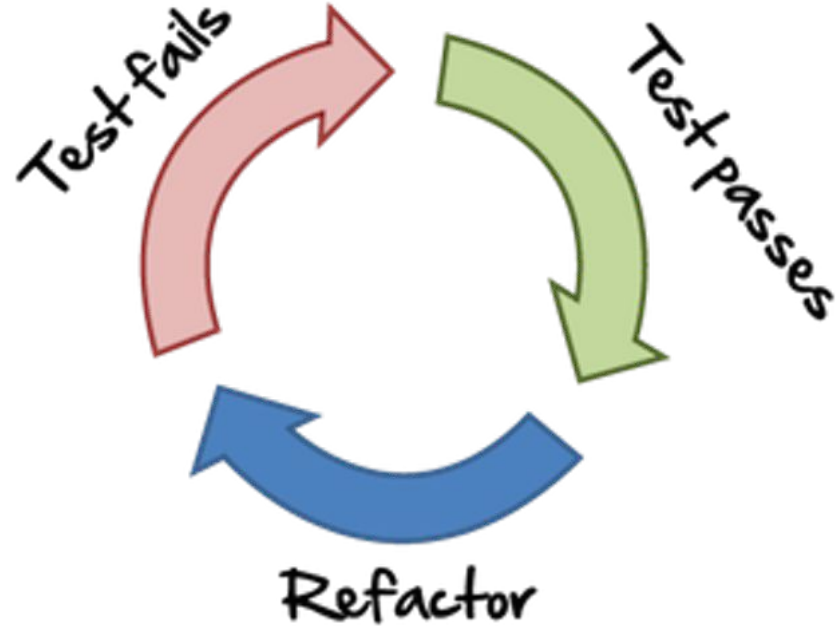




**03.**

**METODOLOGÍA DE TRABAJO**

# TEST DRIVEN DEVELOPMENT



**PUEDE  
FALLAR**



## TODO PUEDE FALLAR

Generar los test para el método FizzBuzz.

Probar los casos:

1. Cuando el número es divisible sólo por 3
2. Cuando el número es divisible sólo por 5
3. Cuando el número es divisible por 3 y 5
4. Cuando el número no es divisible por 5 o 3

