



Programación Go



## Testing

### // Práctica clase 1 - Go Testing

#### Objetivo

El objetivo de esta guía práctica es identificar y diferenciar los distintos tipos de test, entender el objetivo de cada uno y su impacto en la calidad de código y del software productivo. Por lo que se harán distintos planteamientos a los que se debe responder según lo visto en la clase1

#### Forma de trabajo

En base al material visto en la clase 1, se deberá responder las siguientes preguntas. Agregando la respuesta abajo de cada pregunta, ejemplo:

¿Are you ready?



#### Ejercicio 1

¿Cuáles son las diferencias entre White Box y Black Box?

En **White Box**, quien realiza la prueba conoce y tiene la posibilidad de visualizar el código. Contrario al **Black Box**, no tiene detalles del software; así como su estructura, diseño o funcionamiento **interno** es desconocido por quien ejecuta la prueba.

Además, en **White Box** se aplica para probar el **flujo, seguridad y estructura** del código. Y pueden aplicarse por medio de *Test Unitarios y de Integración*. Mientras en **Black Box**, su objetivo principal es probar la funcionalidad del código, evaluando las respuestas y reacciones (**comportamiento**) del componente testeado ante distintos escenarios. Y se aplican mediante: *Tests funcionales, no funcionales y de regresión*.



## Ejercicio 2

¿Qué es un test funcional?

Son pruebas que se basan en la entrada y salida del software, tienen como objetivo comprobar la respuesta del software ante cada escenario, y este mismo coincida con el resultado esperado. Validan el comportamiento funcional del software.



## Ejercicio 3

¿Qué es un Test de Integración?

Son aquellos test que prueban la comunicación entre distintos componentes o capas de la aplicación. Su objetivo es comprobar que todos aquellos bloques de código que fueron probados de forma unitaria, interactúen y se comuniquen entre sí generando los resultados esperados.

Los test de integración exponen el funcionamiento general de la aplicación, permitiendo evaluar su diseño, desempeño y comportamiento.



## Ejercicio 4

Indicar las dimensiones de calidad prioritarias en MELI.

Existen diversos estándares que enumeran las dimensiones a considerar para la calidad de software. En MELI, toma como prioritarias las siguientes:

- Funcionalidad (Functionality)
- Rendimiento (Performance)
- Fiabilidad (Reliability)
- Seguridad (Security)
- Mantenibilidad (Maintainability)

