

  
Programación Go

## ▶ Testing

### // Práctica clase 2 - Go Testing

#### Objetivo

Implementar de forma correcta lo estudiado durante la clase. Tener en cuenta el diseño y ejecución de Tests de Integración sobre el proyecto Go Web.

#### Forma de trabajo

De forma análoga a la práctica anterior, los ejercicios serán ejecutados sobre el proyecto Go Web. Recordar que cada file de test, debe estar alojado en la misma carpeta del objeto de prueba. Y debe estar nombrado de la misma forma que el objeto de prueba, pero sucedido de la palabra “\_test”.

Ejemplo: service.go -> service\_test.go

¿Are you ready? 😎👍



## Ejercicio 1 - Service/Repo/Db Update()

Diseñar un test que pruebe en la capa service, el método o función Update(). Para lograrlo se deberá:

1. Crear un mock de Storage, dicho mock debe contener en su data un producto con las especificaciones que desee.
2. El método Read del Mock, debe contener una lógica que permita comprobar que dicho método fue invocado.
3. Para dar el test como OK debe validarse que al invocar el método del Service Update(), retorne el producto con mismo Id y los datos actualizados. Validar también que Read() del Store haya sido ejecutado durante el test.



## Ejercicio 2 - Service/Repo/Db Delete()

Diseñar un test que pruebe en la capa service, el método o función Delete(). Se debe probar la correcta eliminación de un producto, y el error cuando el producto no existe. Para lograrlo puede:

1. Crear un mock de Storage, dicho mock debe contener en su data un producto con las especificaciones que desee.
2. Ejecutar el test con dos id's de producto distintos, siendo uno de ellos un id inexistente en el Mock de Storage.
3. Para dar el test como OK debe validarse que efectivamente el producto borrado ya no exista en Storage luego del Delete(). También que cuando se intenta borrar un producto inexistente, se debe obtener el error correspondiente.