

Estudiante: Rachit Díaz Vega

Carné: C12569

Practica #9

Preguntas:

1- Qué es un Mock (en testing)?

Un mock es un objeto que simula o imita el comportamiento de un objeto real en un entorno controlado. Se utilizan en pruebas unitarias para asegurarse de que ciertas funciones se comporten de manera esperada, como la llamada a métodos con parámetros específicos.

2- Qué es un Stub?

Un Stub es un objeto que proporciona respuestas predefinidas a llamadas realizadas durante una prueba. Son diferentes los Mocks porque los stubs no verifican interacciones, sino que simplemente devuelven datos específicos para permitir que el código bajo prueba funcione en un entorno controlado.

3- Busque o cree un ejemplo en el tipo de framework y explique cómo funciona.

Usando Jest:

```
export function getUsername(userId) {  
  Esto puede retomar el valor del user desde un API o un Base de Datos  
  return `Usuario_${userId}`;  
}  
  
export function showGreeting(userId) {  
  const name = getUsername(userId);  
  return `Hola, ${name}`;  
}
```

Ahora imitamos el comportamiento usando Mocks: `import * as userService from './userService';`

```
test('showGreeting llama a getUsername con el id correcto', () => {  
  const mockGetUsername = jest.spyOn(userService, 'getUsername');  
  mockGetUsername.mockReturnValue('MockUser');  
  
  const greeting = userService.showGreeting(5);  
  
  // Verificamos que fue llamada con el valor 5  
  expect(mockGetUsername).toHaveBeenCalledTimes(1);  
  expect(greeting).toBe('Hola, MockUser');  
})
```

Ahora con un Stub:

```
test('showGreeting retorna saludo usando stub', () => {  
  jest.spyOn(userService, 'getUserName').mockReturnValue('StubUser');  
  
  const greeting = userService.showGreeting(1);  
  
  expect(greeting).toBe('Hola, StubUser');  
});
```