

Para saber mais - Sobre verificar código que não deve ser executado

Durante a execução do teste que simula uma exception, garantimos o comportamento esperado, que é apresentar a mensagem de falha esperada de acordo com a exception.

Porém, nesse tipo de teste, podem existir outros requisitos esperados, como por exemplo, código que não deve ser executado caso a situação excepcional ocorra. Entrando no nosso contexto (nos testes para o `EnviadorDeLance`), não faz sentido a execução da instrução `client.propoe()`, pois ela não deve acontecer caso ocorra exception.

Para verificarmos que esse código não é chamado quando, podemos utilizar o segundo parâmetro do método `verify()` do Mockito que é o `VerificationMode` (<https://static.javadoc.io/org.mockito/mockito-core/2.2.6/org/mockito/verification/VerificationMode.html>). Ele permite, por exemplo, verificar se o `propoe()` do `client` não foi chamado utilizando o modo `never()`:

```
@Test
public void deve_MostrarMensagemDeFalha_QuandoLanceForMenorQueUltimoLance() {
    EnviadorDeLance enviador = new EnviadorDeLance(
        client,
        listener,
        manager);
    doThrow(LanceMenorQueUltimoLanceException.class)
        .when(leilao).propoe(ArgumentMatchers.any(Lance.class));

    enviador.envia(leilao, new Lance(new Usuario("Fran"), 100));

    verify(manager).mostraAvisoLanceMenorQueUltimoLance();
    verify(client, never()).propoe(any(Lance.class), anyLong(), any(RespostaListener.class));
}
```

Existem outros modos que podem ser útil, como é o caso do `times()` que permite verificar quantas vezes um método foi executado, enviando um `int` via argumento, por exemplo, se você quer ter a certeza que foi apenas uma vez, basta chama `times(1)`.