**Teste com Dependências**

**Introdução**

* Algumas classes dependem de outras para fazer seu trabalho
* É importante testar essas classes trabalhando em conjunto com sua despendências

**Dubles**

* Usada para testar classes que dependem de outras sem testar a integração entre elas
* “Imita” o comportamento de uma classe real de uma maneira controlada
* Maior controle, podendo definir o comportamento esperado para o duble
* Mais velocidade, visto que não precisa esperar um serviço ou banco de dados
* Melhoria no projeto das classes, refletindo sobre contratos e limites conceituais
* **Dummys:** objetos passados para a classe alvo, mas nunca usados (ex.: parâmetros não utilizados)
* **Fakes:** objetos que possuem uma implementação mais simples que o real (ex.: lista com objetos fixos)
* **Stubs:** objetos que provém resposta prontas para as chamadas executadas durante o teste
* **Mocks:** equivalentes aos “stubs”, mas podem registrar as chamadas de métodos que foram feitas para a conferência posterior (se ou quantas vezes foi chamado)
* **Spies:** observa sua interação com os demais, evelope ao redor do objeto registrando as interações
* **Stubs x Mocks:** verificação de estado x verificação de comportamento
* **Mockito:** permite a criação de spies sobre objetos reais

**Usar dubles quando**

* Classes que a classe alvo depende são lentas
* Classe alvo depender de conexão com infraestrutura externa
* Quiser testar situações difíceis de simular
* Classes alvo dependem de muitas outras classes

**Não usar dubles quando**

* Classe alvo depende de entidade que não são complexas de instanciar
* Bibliotecas externas ou métodos utilitários
* Objetos muito simples