**Anotações sobre JUnit**

O JUnit é um framework simples e de código aberto, feito para fazer testes unitários(ou seja testar um a função ou método especifico). Ajuda na organização do código, sendo que cada classe tem que ter uma classe teste(o Maven ajuda bastante nisso) e independente da alteração do código ele deve ser sempre testado.

**Asserts**

Para validar os testes usamos o Asserts para validar os testes, podemos ter asserts para os tipos primitivos, objetos e arrays.O asserts usam o import static o que acaba tornando o código mais simples e a ordem dos parâmetros é o valor esperado e depois o valor atual.

São métodos utilitários usados para determinar se um caso de teste deu certo ou não.O métodos podem ser usados diretamente mas o uso de import deixa o código de melhor compreensão.

**Rules**

Componente que intercepta uma chamada dos métodos do teste, o que permite fazer alguma coisa antes ou depois do teste rodar.

Podemos criara arquivos,criar diretórios temporários,deletar um teste se um timeout for excedido,configurar um recurso externo.

**Exceção**

Importante garantir que o código irá agir normalmente mesmo que aconteça algum imprevisto.

**Mocks**

Mocks é usado para denominar objetos que simulam objetos reais em ambientes controlados,podemos usar quando precisamos validas testes sem validar se a comunicação com o banco de dados ocorreu.

Uma das limitações do Mock é o seu comportamento ser atrelado ao objeto, se as alterações não forem feitas corretamente os testes podem falhar ,mas o resultado vai sair como se o teste estivesse certo, o que torna o teste unitário inútil.

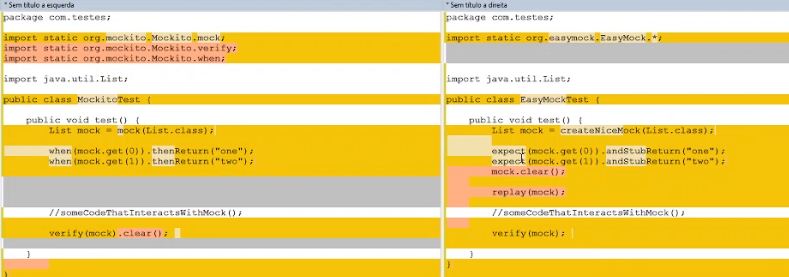
**Frameworks**

* Mockito

Framework feito pra Java que permite construir apis simples

* EasyMock

Foi o primeiro gerador de objetos mock, muito semelhante ao mockito



(mockito e easymock)

* PowerMockito

É uma extensão que permite o uso de certos recursos pelo mockito, de maneira simples, e ajuda a lidar com algumas limitações como a dificuldade de lidar com métodos finais,estáticos e privados.

**Hamcrest e Matcher**

* Hamcrest

É um framework usado para teste de unidade.

Macher

Bean Matcher

Permite inspecionar uma propriedade de um objeto

**AssertJ**

Permite escrever asserts de entendimento mais fácil, ele prove um conjunto de classes e métodos para escrever facilmente.

**TDD**

Uma técnica de construção de software que orienta o desenvolvimento por meio dos testes primeiro.

