

## Usando Annotations:

Entendemos como o mockito funciona, hora de **deixar as coisas mais simples** com as **Annotations**.

Primeiro vamos dizer as nossas **váriaves de instância** (aquelas que inicializamos como mock no setup) que elas serão mockadas, com a **annotation @Mock**:

```
@Mock
private SPCService spcService;

@Mock
private LocacaoDAO dao;

@Mock
private EmailService emailService;
```

Agora vamos **dizer onde nós usaremos esse mock**, que vai ser **na LocacaoService**. Isso é bem simples, basta anotarmos com **@InjectMocks** a **variável de instância que faz referência a LocacaoService**:

```
@InjectMocks
private LocacaoService servicoDeLocacao;
```

Agora **vamos transformar nosso @Before método de setUp( )**, invés de utilizarmos todas essas instâncias e seta-las:

```
@Before
public void setUp() {

    servicoDeLocacao = new LocacaoService();

    dao = Mockito.mock(LocacaoDAO.class);
    servicoDeLocacao.setLocacaoDAO(dao);

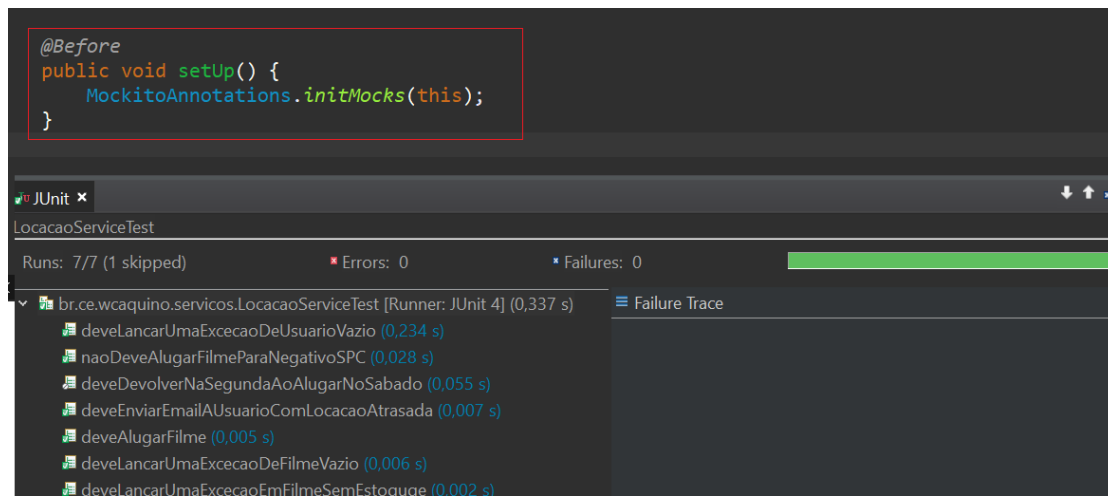
    spcService = Mockito.mock(SPCService.class);
    servicoDeLocacao.setSpcService(spcService);

    emailService = Mockito.mock(EmailService.class);
    servicoDeLocacao.setEmailService(emailService);
}
```

Vamos usar apenas a classe **MockitoAnnotations** com o método **initMocks( )** usando a **classe como parâmetro**. Com isso **vamos inicializar todos os Mocks** que vão ser **utilizados na**

`servicoDeLocacao` e estão na classe passada por parâmetro:

```
@Before
public void setUp() {
    MockitoAnnotations.initMocks(this);
}
```



JUnit x  
LocacaoServiceTest

Runs: 7/7 (1 skipped)    Errors: 0    Failures: 0

br.ce.wcaquino.servicos.LocacaoServiceTest [Runner: JUnit 4] (0,337 s)

- deveLancarUmaExeccaoDeUsuarioVazio (0,234 s)
- naoDeveAlugarFilmeParaNegativoSPC (0,028 s)
- deveDevolverNaSegundaAoAlugarNoSabado (0,055 s)
- deveEnviarEmailAUusuarioComLocacaoAtrasada (0,007 s)
- deveAlugarFilme (0,005 s)
- deveLancarUmaExeccaoDeFilmeVazio (0,006 s)
- deveLancarUmaExeccaoEmFilmeSemEstoque (0,002 s)

Failure Trace

Com isso não precisamos nem mais dos **métodos de set** que criamos na classe de serviço apenas **para satisfazer os testes** que **injetavam os mocks** nela:

```
public void setLocacaoDAO(LocacaoDAO dao) {
    this.dao = dao;
}

public void setSpcService(SPCService spcService) {
    this.spcService = spcService;
}

public void setEmailService(EmailService emailService) {
    this.emailService = emailService;
}
```

Podemos apenas excluí-los.



