

## Mockito:

Nosso problema na versão anterior do projeto(2.9) era que precisavamos do objeto que implementava o DAO para continuarmos no teste. Uma solução seria usar objetos falsos para simular o comportamento do DAO, mas isso seria uma solução muito trabalhosa dependendo do tamanho do projeto.

Existem frameworks que ajudam a dinamicamente criarmos esses objetos falsos e é aí que ele entra, o **MOCKITO**.

O mockito **não gera objetos fake**, ele **gera objetos Mock**. A diferença entre Mock e fake é que nos mocks somos capazes de **definir comportamentos dinâmicos** e inclusive se ele **forá executado conforme o esperado**.

Primeiro de tudo devemos adicionar a dependência do mockito ao projeto, nesse caso como é um projeto gerenciado pelo Maven, devemos adicionar ao **pom.xml**:

```
</dependency>  
  
<dependency>  
  <groupId>org.mockito</groupId>  
  <artifactId>mockito-all</artifactId>  
  <version>1.10.19</version>  
</dependency>
```

Antes de fazer um **mock** que **implementará a interface de DAO**, fizemos um método **setLocacaoDAO** que **fará a injeção de dependência**, assim poderemos **mockar um objeto que implemente a interface de dao** e passar por **parâmetro através do método**:

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

Agora podemos passar a instanciar nossa interface de DAO com um objeto mockado que a implemente através do mockito:

```
@Before  
public void setUp() {  
    servicoDeLocacao = new LocacaoService();  
    LocacaoDAO dao = Mockito.mock(LocacaoDAO.class);  
    servicoDeLocacao.setLocacaoDAO(dao);  
}
```

A instância de um **objeto mock** responde como se **tivesse implementando a classe que ele**

**usou como origem**, porém ele é genérico e não tem como saber como deve se comportar, então ele faz um **comportamento padrão** de **acordo com a assinatura do método**. Se o método **for void ele não faz nada**, se **retornar um número ele retorna zero**, se **retornar uma string ele retorna uma string vazia** e se **o retorno for um objeto ele retorna null**.

Isso não significa que esses retornos não possam ser alterados, é o que veremos na próxima versão do projeto.